Conversation Analysis for Computational Modelling of Task-oriented Dialogue

Nathan Duran

A thesis submitted in partial fulfilment of the requirements of the University of the West of England, Bristol for the degree of Doctor of Philosophy

Faculty of Environment and Technology
University of the West of England
Bristol, United Kingdom
January 2023

Conversation Analysis for Computational Modelling of Task-oriented Dialogue

Nathan Duran

Abstract

Current methods of dialogue modelling for Conversational AI (CAI) bear little resemblance to the manner in which humans organise conversational interactions. The way utterances are represented, interpreted, and generated are determined by the necessities of the chosen technique and do not resemble those used during natural conversation. In this research we propose a new method of representing task-oriented dialogue, for the purpose of computational modelling, which draws inspiration from the study of human conversational structures, Conversation Analysis (CA). Our approach unifies two well established, yet disparate, methods of dialogue representation: Dialogue Acts (DA), which provide valuable semantic and intentional information, and the Adjacency Pair (AP), which are the predominant method by which structure is defined within CA. This computationally compatible approach subsequently benefits from the strengths, whilst overcoming the weaknesses, of its components.

To evaluate this thesis we first develop and evaluate a novel CA Modelling Schema (CAMS), which combines concepts of DA's and AP's to form *AP-type* labels. Thus creating a single annotation scheme that is able to capture the semantic and syntactic structure of dialogue. We additionally annotate a task-oriented corpus with our schema to create CAMS-KVRET, a first-of-its-kind DA and AP labelled dataset. Next, we conduct detailed investigations of input representation and architectural considerations in order to develop and refine several ML models capable of automatically labelling dialogue with CAMS labels. Finally, we evaluate our proposed method of dialogue representation, and accompanying models, against several dialogue modelling tasks, including next label prediction, response generation, and structure representation.

With our evaluation of CAMS we show that it is both reproducible, and inherently learnable, even for novice annotators. And further, that it is most intuitively applied to task-oriented dialogues. During development of our ML classifiers we determined that, in most cases, input and architectural choices are equally applicable to DA and AP classification. We evaluated our classification models against CAMS-KVRET, and achieved high test set classification accuracy for all label components of the corpus. Additionally, we were able to show that, not only is our model capable of learning the semantic and structural aspects of both the DA and AP components, but also that AP are more predictive of future utterance labels, and thus representative of the overall dialogue structure. These findings were further supported by the results of our next-label prediction and response generation experiments. Moreover, we found AP were able to reduce the perplexity of the generative model. Finally, by using χ^2 analysis to create dialogue structure graphs, we demonstrate that AP produce a more generalised and efficient method of dialogue representation. Thus, our research has shown that integrating DA with AP, into AP-type labels, captures the semantic and syntactic structure of an interaction, in a format that is independent of the domain or topic, and which benefits the computational modelling of task-oriented dialogues.

Acknowledgements

First and foremost I would like to thank Prof. Jim Smith and Dr. Steve Battle for their continued support and supervision for this PhD.

I would also like to extend my thanks to colleagues who have supported and encouraged along the way, including Prof. Larry Bull, Dr. Mehmet Aydin, Dr. Benedict Gaster, Dr. Chris Simons, Dr. Elias Pimenidis, Dr. Charlotte Selleck, and Dr. Lindsey Gillies.

Finally, thanks to my family and friends for their ongoing support and distraction.

Contents

List of Tables viii

List of Figures ix

Acronyms and Abbreviations x

1 Introduction 1
- 1.1 Motivation and Research Problem . . . 2
- 1.2 Research Questions, Hypothesis and Objectives . . . 4
  - 1.2.1 Research Questions . . . 4
  - 1.2.2 Hypothesis . . . 5
  - 1.2.3 Objectives . . . 5
- 1.3 Contribution of Thesis . . . 5
- 1.4 Thesis Structure . . . 6

2 Literature Review 7
- 2.1 Conversation Analysis and Dialogue Acts . . . 7
  - 2.1.1 Adjacency Pairs . . . 9
    - 2.1.1.1 Sequence Expansion . . . 10
  - 2.1.2 Dialogue Acts . . . 14
    - 2.1.2.1 Dialogue Act Taxonomies . . . 16
  - 2.1.3 Summary . . . 18
- 2.2 Dialogue Act Classification . . . 19
  - 2.2.1 Dialogue Act Corpora . . . 20
  - 2.2.2 Unsupervised, Supervised, and Probabilistic . . . 22
  - 2.2.3 Neural Architectures and Language Models . . . 23
    - 2.2.3.1 Input Sequence Processing . . . 25
    - 2.2.3.2 Word Embeddings . . . 27
    - 2.2.3.3 Sentence Encoding . . . 27
    - 2.2.3.4 Context Encoding . . . 30
    - 2.2.3.5 Classification . . . 33
- 2.3 Adjacency Pair Identification and Dialogue Structure . . . 33
  - 2.3.1 Dialogue Structure . . . 36
- 2.4 Task-oriented Dialogue Modelling . . . 37
  - 2.4.1 Task-oriented Dialogue . . . 37
  - 2.4.2 Computational Dialogue Modelling . . . 37

3 Methodology 39
- 3.1 Schema Development, Data Collection, and Preparation . . . 40
  - 3.1.1 Conversation Analysis Modelling Schema . . . 40
  - 3.1.2 Schema Evaluation . . . 41
  - 3.1.3 Corpora Annotation . . . 41

3.1.4 Other Dialogue Act Annotated Corpora . . . 42
3.2 Dialogue Classification Systems . . . 42
3.2.1 Phase 1: Sentence Encoding . . . 42
3.2.2 Phase 2: Context Encoding . . . 43
3.2.3 Phase 3: AP and Multi-label Identification . . . 44
3.2.4 Training and Evaluation . . . 44
3.2.4.1 Significance Testing . . . 45
3.3 Dialogue Structure Evaluation and Analysis . . . 45
3.3.1 Next-Label Prediction . . . 46
3.3.2 Response Generation . . . 46
3.3.3 Analysis of Dialogue Structure . . . 48

4 Conversation Analysis Modelling Schema and Corpora Annotation 50
4.1 CAMS Definition and Guidelines . . . 50
4.1.1 Dialogue Acts . . . 50
4.1.2 Adjacency Pairs . . . 51
4.1.3 Adjacency Pair Types . . . 53
4.2 CAMS Evaluation . . . 53
4.2.1 Dialogue Selection . . . 54
4.2.2 Participant Selection . . . 56
4.2.3 Inter-annotator Agreement . . . 56
4.2.3.1 Coefficient Selection . . . 57
4.2.3.2 Coefficient Evaluation . . . 58
4.2.4 Timing and Rating Measures . . . 59
4.2.4.1 Annotation Timing . . . 59
4.2.4.2 Annotation Confidence . . . 59
4.2.5 Results and Discussion . . . 59
4.2.5.1 Inter-annotator Agreement . . . 59
4.2.5.2 Task-oriented and Non-task-oriented Dialogues . . . 61
4.2.5.3 Corpora Dialogues . . . 64
4.2.5.4 AP Label Agreement . . . 65
4.2.5.5 Expert Annotators . . . 67
4.2.5.6 Annotation Confidence Scores . . . 68
4.2.5.7 Annotation Time . . . 69
4.2.5.8 CAMS Evaluation Summary . . . 70
4.3 CAMS-KVRET Annotation . . . 72
4.3.1 CAMS-KVRET Overview . . . 73

5 Dialogue Act Identification 75
5.1 Sentence Encoding for Dialogue Act Classification . . . 75
5.1.1 Dialogue Act Corpora . . . 75
5.1.2 Sentence-encoders . . . 76
5.1.3 Input Sequence Representations . . . 78
5.1.3.1 Letter Case and Punctuation . . . 78
5.1.3.2 Vocabulary Size . . . 79
5.1.3.3 Sequence Length . . . 81
5.1.3.4 Input Sequences Comparison . . . 83
5.1.4 Word Embeddings . . . 83
5.1.5 Supervised Models . . . 85
5.1.6 Language Models . . . 86
5.1.7 Sentence Encoding Summary . . . 87
5.2 Context Encoding for Dialogue Act Classification . . . 89

5.2.1 Context-encoders 89
5.2.2 Context Utterances 90
5.2.2.1 Previous Context 91
5.2.2.2 Full Context 93
5.2.3 Speakers and Labels 95
5.2.3.1 Speakers 95
5.2.3.2 Labels 96
5.2.3.3 Speaker and Label Combinations 97
5.2.4 Classifiers 99
5.2.5 Context Encoding Summary 99

6 Adjacency Pair Identification 101
6.1 CAMS Single-labels 101
6.1.1 The Benefit of Context Utterances for CAMS 101
6.1.2 The Influence of Speakers and Labels 105
6.1.3 Evaluating CAMS Single-label Classification 107
6.2 CAMS Multi-labels 111
6.3 CAMS Classification Summary 115

7 Dialogue Structure Evaluation and Analysis 117
7.1 Next-Label Prediction 117
7.2 Response Generation 119
7.3 Analysis of Dialogue Structure 124

8 Conclusion 130
8.1 Evaluation of Objectives and Research Hypotheses 130
8.1.1 Hypothesis 1 130
8.1.2 Hypothesis 2 131
8.1.3 Hypothesis 3 132
8.2 Challenges and Limitations 133
8.3 Applications and Further Work 134

Appendices 137

Appendix A CAMS Label Definitions 138
A.1 Adjacency Pairs 138
A.1.1 Base Pairs 139
A.1.2 Expansions 139
A.1.3 Minimal-expansions 140
A.2 Dialogue Acts 140
A.2.1 Information-seeking Functions 141
A.2.2 Information-providing Functions 142
A.2.3 Commissive Functions 143
A.2.4 Directive Functions 144
A.2.5 Feedback Functions 144
A.2.6 Time Management Functions 145
A.2.7 Own and Partner Communication Management Functions 145
A.2.8 Social Obligations Management Functions 145

Appendix B Inter-annotator Agreement **147**
B.1 Agreement Coefficients . . . 147
B.1.1 Unweighted Coefficients . . . 147
B.1.1.1 Pi . . . 148
B.1.1.2 Kappa . . . 148
B.1.1.3 Multi-Pi . . . 149
B.1.1.4 Multi-Kappa . . . 149
B.1.2 Weighted Coefficients . . . 150
B.1.2.1 Alpha . . . 151
B.1.2.2 Beta . . . 151
B.2 Weighted Coefficient Distance Functions . . . 151
B.2.1 Dialogue Act Distance Function . . . 152
B.2.2 Adjacency Pair Distance Function . . . 153
B.2.3 AP-type Distance Function . . . 154
B.3 Coefficient Selection . . . 154
B.4 Coefficient Evaluation . . . 156

Appendix C Alpha vs Beta **159**

Appendix D CAMS-KVRET Label Distributions **162**

Appendix E Model Hyperparameters **163**

Appendix F Model Variants Results **165**
F.1 Supervised Model Variants . . . 165
F.1.1 Multi-layer and Bi-directional Models . . . 165
F.1.2 Attentional Models . . . 166
F.2 Context Model Variants . . . 166

Appendix G CAMS Classification Results **168**
G.1 AP Identification Results . . . 168
G.2 Next-label Results . . . 170

Appendix H Dialogue Structure Evaluation and Analysis **171**

Bibliography **193**

List of Tables

2.1 Adjacency Pair Type Relations . . . 10
2.2 DiAML Categories . . . 18
2.3 Unsupervised, Supervised, and Probabilistic DA classification Studies . . . 24
2.4 Single-sentence ANN DA Classification Studies . . . 28
2.5 Contextual ANN DA Classification Studies . . . 34

3.1 χ^2 Contingency Table . . . 48

4.1 CAMS DA Categories . . . 51
4.2 Dialogue Sets . . . 55
4.3 Task and Non-task Label Assignments . . . 62
4.4 Annotator Confidence Scores . . . 68
4.5 Annotation Time . . . 70
4.6 Ordered Annotation Time . . . 70
4.7 CAMS-KVRET General Statistics . . . 73
4.8 CAMS-KVRET Set Statistics . . . 73

5.1 Overview of SwDA and Maptask . . . 76
5.2 Letter Case and Punctuation . . . 79
5.3 Punctuation and No Punctuation F1 Scores . . . 80
5.4 Vocabulary Size . . . 80
5.5 Sequence Length . . . 82
5.6 Input Sequence Comparison . . . 84
5.7 Word Embeddings . . . 84
5.8 Supervised Models . . . 86
5.9 Language Models . . . 87
5.10 CAMS-KVRET Baselines . . . 90
5.11 CAMS-KVRET and Maptask Previous Context . . . 91
5.12 SwDA Previous Context . . . 92
5.13 Full Context . . . 93
5.14 Speaker Encoders . . . 96
5.15 Label Encoders . . . 97
5.16 Speakers and Labels . . . 98
5.17 Classifiers . . . 99
5.18 Context Models . . . 100

6.1 CAMS-KVRET Single-sentence Baselines . . . 102
6.2 AP Contextual and Non-contextual F1 Scores . . . 103
6.3 DA Contextual and Non-contextual F1 Scores . . . 105
6.4 Speakers and Labels . . . 105
6.5 AP Label and No-label F1 Scores . . . 107
6.6 CAMS-KVRET Single-label Models . . . 108

6.7 Gold-standard vs Predicted Labels . . . 110
6.8 CAMS-KVRET Multi-label Models . . . 113
6.9 Single and Multi-label Prediction Probabilities . . . 114

7.1 Next-label Prediction . . . 118
7.2 Sentence Similarity Metrics . . . 121
7.3 Sentence Similarity Error Analysis . . . 122
7.4 Perplexity Metrics . . . 123
7.5 Similarity Edit Distance . . . 127
7.6 Global and Local Efficiency . . . 128

C.1 AP and DA JSD . . . 159
C.2 Annotator Label χ^2 . . . 160

D.1 CAMS-KVRET DA Label Distributions . . . 162
D.2 CAMS-KVRET AP Label Distributions . . . 162

E.1 Supervised Sentence Encoder Hyperparameters . . . 163
E.2 Language Model Sentence Encoder Hyperparameters . . . 164
E.3 Context Encoder Hyperparameters . . . 164

F.1 Multi-layer Recurrent Models . . . 166
F.2 Context Model Variants . . . 167

G.1 CAMS-KVRET Baseline Comparison . . . 169
G.2 CAMS-KVRET Full Multi-label Models Results . . . 169
G.3 Next-label Baseline Comparison . . . 170
G.4 Next-label Multi-label Results . . . 170

H.1 Word Overlap Metrics . . . 171

List of Figures

2.1 Generic DA Classification Architecture 25
2.2 Generalised Sentence Encoder Architecture 29
2.3 Generalised Contextual DA classification Architecture 31
2.4 Conversational Map 35

4.1 Software Annotation Tool 54
4.2 Dialogue Set IAA 60
4.3 Task and Non-task IAA 61
4.4 Corpora IAA 64
4.5 Corpora Suffix-only IAA 66
4.6 Expert IAA 67
4.7 Corpora Annotator Confidence Scores 69
4.8 Annotation Time Distribution 71
4.9 CAMS-KVRET DA and AP Distribution 74

5.1 Vocabulary Size 81
5.2 Sequence Length 82
5.3 DCNN Word Embeddings 85
5.4 Context Combinations 94

6.1 AP and DA Context Combinations 102
6.2 AP Contextual and Non-contextual Predictions 104
6.3 AP and DA Label Context 106
6.4 CAMS-KVRET Baseline Comparison 109
6.5 CAMS Multi-label Classification Architecture 112

7.1 Next-label Baseline Comparison 118
7.2 Next-label Full and Partial Context 120
7.3 AP and DA Directed Dialogue Structure Graphs 124
7.4 AP and DA SimRank Similarity 125
7.5 AP and DA Per-task Directed Dialogue Structure Graphs 127

B.1 Coefficient Cube 152
B.2 DA Relationship Information-transfer Sub-tree 153
B.3 DA Relationship Graph 158

G.1 DA Contextual and Non-contextual Predictions 168

H.1 AP Directed Dialogue Structure Graph 171
H.2 DA Directed Dialogue Structure Graph 172
H.3 AP-type Directed Dialogue Structure Graph 173
H.4 AP-type Per-task Directed Dialogue Structure Graph 174

Acronyms and Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
AMI	Augmented Multi-party Interaction Corpus
AP	Adjacency Pairs
AP-type	Adjacency Pair Types
BOW	Bag of Words
BSR	Bayesian Signed-rank
CAI	Conversational Artificial Intelligence
CA	Conversation Analysis
CAMS	Conversation Analysis Modelling Schema
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
DA	Dialogue Acts
DAMSL	Dialogue Act Markup in Several Layers
DiAML	Dialogue Act Markup Language
DIT	Dynamic Interpretation Theory
DL	Deep Learning
DRL	Deep Reinforcement Learning
DST	Dialogue State Tracking
FFNN	Feed Forward Neural Network
FPP	First-pair-part
GRU	Gated Recurrent Unit
HCI	Human Computer Interaction
HMM	Hidden Markov Model
IAA	Inter-annotator Agreement

IPA	Intelligent Personal Assistants
KVRET	Key-Value Retrieval Networks Corpus
LM	Language Model
LSTM	Long Short-term Memory
LSA	Latent Semantic Analysis
ML	Machine Learning
MRDA	Meeting Recorder Dialogue Act Corpus
NHST	Null Hypothesis Significance Testing
NLP	Natural Language Processing
NLU	Natural Language Understanding
NLG	Natural Language Generation
POS	Part of Speech
RNN	Recurrent Neural Network
Seq2seq	Sequence to Sequence
SPP	Second-pair-part
SGD	Speech-generating Devices
SwDA	Switchboard Dialogue Act Corpus
SWBD-DAMSL	Switchboard Shallow-Discourse Function Annotation
SVM	Support Vector Machines
TCU	Turn-constructional Units
TRP	Transition-relevance Place
Tukey-HSD	Tukey's Honest Significant Difference
VA	Virtual Assistant
VEA	Virtual Employee Assistant

Chapter 1

Introduction

With regard to *computing* machines, it was Turing who initially proposed, and popularised, the idea of a 'thinking' machine conversing with human subjects (Turing, 1950). Though the field of Natural Language Processing (NLP) has grown to encompass a wide range of topics, from Sentiment Analysis to Machine Translation (Eisenstein, 2018), the goal of humans using natural language to interact with computers - Conversational AI (CAI) - has been a long-term and prominent one. CAI has since become a ubiquitous part of modern life, with Virtual Assistants (VA), such as Google Assistant, Apple's Siri, and Amazon's Alexa estimated to reach 1.8 billion users worldwide by 2021 (Budzinski, Noskova, and Zhang, 2019). This growth has been largely fuelled by advances in Machine Learning (ML), and associated NLP applications, which has not only improved the capabilities of CAI, but increased the number of ways in which it can be developed and deployed in the real world. The 'traditional' linguistic, or rule-based, methods of building CAI are of the same ilk that Weizenbaum (1966) used to develop ELIZA, and still commonly used. Rule-based CAI relies on predefined if-then logic, and certain language conditions can be created to identify word order, synonyms, and common phrases, to formulate desired responses. Conversely, ML provides several options for a purely data-driven approach to CAI development. Artificial Neural Networks (ANN), and Deep Reinforcement Learning (DRL), both solve the problem in an end-to-end fashion. That is, using a large corpus of conversational data, they are trained to automatically produce appropriate responses to a user's input. On the other hand, a hybrid approach offers the best of both these worlds while overcoming some of their respective disadvantages. It increases transparency and reduces development cost, by allowing developers to specify the desired conversational flow and response type, without the need to define all possible phrasings and synonyms. At the same time, it can be built with little or no data, and still leverage NLP components (speech recognition, user-intent classification, named entity recognition (NER), and so on) to provide smarter and more flexible responses.

However, one characteristic shared by each of these approaches to CAI, is that the method in which they model dialogue bears little resemblance to the way humans organise conversational interactions. The way utterances are represented, interpreted, and generated are determined by the necessities of the chosen technique and *not* those used by humans during natural conversation. This research proposes a new method of representing dialogue, for the purpose of computational modelling, which draws inspiration from the study of human conversational structures, Conversation Analysis (CA).

In this introductory chapter, we describe the motivation for the proposed approach and specify the research problem investigated. This is followed by outlining the research aim, the hypotheses, objectives and contributions, and finally the structure of this thesis.

1.1 Motivation and Research Problem

Human conversational interactions are, naturally, a complex phenomenon. When we take part in such interactions, we may utilise a range of visual, verbal, and linguistic cues to interpret the intentions of other participants, formulate responses and organise turns of talk (Goodwin, 1981). Even when considered solely in an audio or text-based form, the utterances of an interaction cannot be fully understood on an individual basis, but rather must be interpreted within the wider context in which they were produced (Ekman and Scherer, 1984). In other words, because utterances are not produced in isolation, to fully understand the *meaning* of a dialogue, or section of dialogue, we must also have some understanding of its *structure*. The question of how such intricate conversational structure, and meaning, can be represented in a computationally practical format, remains an open problem within NLP, and is a primary motivation for this research. A further motivation, and as previously mentioned, is that current dialogue modelling techniques bear little resemblance to the mechanisms that humans use to organise talk. In rule-based, and hybrid approaches, the dialogue model is hand-crafted by the developer. As such, the systems conversational behaviour is typically *reactive* to the user utterance or dialogue state (Matějů et al., 2021). ML and statistical approaches, on the other hand, are purely data-driven. Fundamentally, the dialogue model is only as flexible, or generalisable, as the data upon which it was trained, and the developer's agency over dialogue strategy is removed. Thus, NLP methods of representing dialogue syntax, and semantics, are often dependant on, or bespoke to, the target domain, problem, or the available data (Shum, He, and Li, 2018). We propose a method of representing the meaning and structure of dialogue that is not only domain agnostic, but also independent of the constituent words of the utterances themselves. It is therefore not constrained by the domain to which it is being applied, but instead, provides a 'high-level' dialogue model based on the empirical study of human conversational patterns.

Currently, the predominant approach to representing dialogue semantics for the purpose of NLP, is the use of Dialogue Acts (DA). Originating from John Austin's 'illocutionary act' theory (Austin, 1962), and later developed with John Searle's 'speech acts' (Searle, 1969), though the term *dialogue act* was introduced by Bunt (1978). In speech act theory, speakers produce utterances in a conversation in order to perform actions, for example, a *question*, *statement*, or *greeting*. A DA, is therefore the semantic content and communicative function of a single utterance of dialogue, and these can be defined with an appropriate set of DA labels. The utility of DA, as a set of labels for a semantic interpretation of a given utterance, has led to their use in many NLP applications (Jurafsky and Martin, 2017; McTear, Callejas, and Griol, 2016). In dialogue management systems they have been used as a representation of user turns and intent (Firdaus et al., 2020; Li et al., 2017), and as a set of possible system actions as a means of dialogue state tracking (DST) (Keizer and Rieser, 2017; Cuayáhuitl et al., 2016; Ge and Xu, 2015; Griol et al., 2008). For spoken language translation Kumar et al. (2008), utilised the contextual information provided by DA's to improve accuracy in phrase based statistical speech translation. They have also been used to analyse the structure of dialogue within the intelligent tutoring domain (Boyer et al., 2010a, 2009a), and everyday conversations (Iseki, 2019).

While DAs provide valuable semantic and intentional information, they naturally consider utterances as an isolated unit, even if the DA may imply some future or past action. In so doing, they fail to recognise the sequential nature of interactions, and the influence that both context and position have, on the production and meaning of an utterance (Clift, 2016; Ekman and Scherer, 1984). As Clift (2016), points out, *"the form of an utterance alone cannot necessarily be relied upon to deliver how it is understood by its recipient"*. Consider the use of "Okay" in the following examples. In the first instance speaker B uses "Okay" in response to a question. In the second instance speaker A uses "Okay" as confirmation that a response has been heard and understood.

1	A: How are you? B: Okay	2	A: Do you need help with that? B: No thank you. A: Okay

What is needed, then, is a method of representing not just the semantics of single utterances but the context within which they were produced and their contribution to the interaction as a whole. For this we turn to the study of human conversation. CA is an area of sociological research that is oriented towards understanding the organisational structure of talk, and aims to define, and analyse, constructs that facilitate turn-taking in human conversations (Sacks, Schegloff, and Jefferson, 1974). Further, CA theory can be applied to any conversation, and therefore, offers some insights into domain agnostic methods of capturing the structure of dialogue. Some key principles of CA are: that turns of talk have some organisational structure; that the structure itself has a descriptive quality for the utterances produced; and in turn, helps to shape the future utterances of the interaction (Sidnell, 2010; Schegloff, 2007). Within CA, the predominant method by which this structure is defined is the concept of an Adjacency Pair (AP). AP arose from the observation that many turns at talk occur as pairs; a greeting is followed by another greeting, and a question followed by an answer. Liddicoat (2007), provides a definition of the basic AP features which are, i) they comprise of two turns, ii) by different speakers, ii) that take place one after the other (in their basic form), and iv) are differentiated into pair types. In an AP the initial turn is called the *First Pair Part* (FPP), and initiates an exchange, the second turn is a *Second Pair Part* (SPP) which is responsive to the prior FPP. To elaborate on the third point, talk does not always occur in neatly ordered pairs. To account for this AP also include the concept of *sequence expansion* (or just expansion) pairs. Expansions allow for talk which is made up of more than a single AP, yet is still constructed and understood as performing, or oriented around, the same basic action. They are constructed in relation to a base sequence of a FPP and SPP in which the core action under way is achieved. To further elaborate on the final point of Liddicoat (2007), AP may also be 'type related'. That is, not every SPP can properly follow any FPP, but rather, the SPP must be of the appropriate type according to the initiating FPP. A question could be followed by an answer, but not a greeting, an offer accepted or declined, and so on (Schegloff, 2007). Thus, in producing a FPP, a speaker determines the relevance of a particular (range of) 'next actions' for the subsequent speaker (Clift, 2016; Liddicoat, 2007). In terms of computational dialogue modelling, this *pair-type* relation has the useful property of limiting the range of possible SPP responses to a given FPP, and further, adds a predictive quality in determining future valid responses given an initiating FPP.

From these observations we may now start to draw some connections between AP and DA. And further, to outline how the concept of combining relevant theories from within the CA, and DA literature, forms the basis of this research.

1. Both DA and AP consider talk as a sequence of actions. DA may be considered single-utterance label defining the action, or intention of the speaker. AP on the other hand, model the sequential nature of these actions through pair-type relationships.

2. The *type* of a single FPP, or SPP, is effectively a DA. Therefore, it is possible to unify these two concepts, DA and AP, from their related, but distinct, linguistic sub-fields of speech act theory and CA. That is, we may use the well defined, and descriptive, DAs as a *type* label for each component of an AP (FPP, SPP, and expansions). This extends the concept of pair-type relation between AP, into a much richer, expressive, and fine-grained, method of labelling dialogue utterances with *AP-types*.

3. The individual weaknesses and strengths of DA, and AP, compliment each other when used in combination. DA provide useful semantic and intentional information for single utterances of dialogue, but fail to convey structural or contextual detail. AP, on the other hand, indicate the structural and contextual relationship between dialogue utterances, yet their pair-type relationships are enhanced through a more descriptive set of DA labels.

Thus, we propose an approach to task-oriented dialogue modelling, which combines AP with DA, into AP-types. The AP-types express the general structure of an interaction, while leveraging the descriptive power of the DA for individual utterances. Or viewed another way, we can consider DA labels as descriptions of the *intra-utterance* features of a dialogue, while AP represent the *inter-utterance* features. We will show that integrating DA with AP, into AP-type labels, captures the semantic and syntactic structure of an interaction, in a format that is independent of the domain or topic, and which facilitate the computational modelling of task-oriented dialogues.

Note that while CA, and DA, theory is applicable to any form of interaction, not only task or goal driven, we have chosen to limit the scope of this research to task-oriented dialogues. This is because they are more likely to have a well defined structure, that is not merely linear sequences of utterances, or a collection of question-answer pairs (Grosz, 2018), and because task-oriented dialogues are a more widely studied domain for computational dialogue modelling (Matějů et al., 2021; Shum, He, and Li, 2018) (see section 2.4). Further, as with so many AI and ML paradigms that are, at least initially, based on natural phenomenon or processes, this CA inspired approach to dialogue modelling is directly influenced by the study of 'natural' human conversational patterns. The utility of CA for modelling Human-computer Interaction (HCI) has been noted before (Luff, Gilbert, and Frolich, 1990; Norman and Thomas, 1990; Robinson, 1990), yet, to date it has received relatively little attention within the NLP and dialogue modelling literature. We therefore intend to explore whether this human-centric approach can be augmented with the current dialogue modelling techniques we have previously discussed, and to what extent this proves beneficial.

1.2 Research Questions, Hypothesis and Objectives

The primary aim of this research is to develop and evaluate a CA inspired computational model for task-oriented dialogues; by combining the descriptive power of DA for individual utterances, with the structural and contextual information provided by AP, to form AP-types, we hope to produce richer and more expressive representations of dialogue. In the following we establish the research questions, hypothesis and objectives that form the basis of this research.

1.2.1 Research Questions

In this research, we established the following research questions:

Q1 Are CA representations suitable for task-oriented and non-task-oriented dialogue?

Q2 What algorithmic process can effectively, and automatically, label task-oriented dialogue with AP and DA?

Q3 Can CA be used as a computationally compatible method of representing task-oriented dialogue structure?

Q4 To what extent do CA representations benefit computational modelling of task-oriented dialogue?

1.2.2 Hypothesis

Motivated by the research aims and questions, the following outlines the research hypotheses that are assessed throughout this thesis:

H1 CA theories on the structure of dialogue can be incorporated with DA as a method of effectively representing task-oriented dialogue for computational modelling purposes.

H2 Existing text classification methods can be adapted to automatically label task-oriented dialogues with DA and AP structure.

H3 Dialogues labelled with DA and AP provide a more syntactically and semantically rich method of dialogue representation than existing methods.

1.2.3 Objectives

Based on the above, we have established the following objectives of our research:

O1 Develop and evaluate a CA annotation schema that defines DA and AP, which combine to form AP-types.

O2 Annotate a task-oriented corpus with the CA schema to produce a gold-standard CA labelled corpus.

O3 Design and develop a novel method to automatically label task-oriented dialogue with the CA schema.

O4 Evaluate proposed method on gold-standard CA labelled corpus.

O5 Evaluate proposed method against different aspects of dialogue modelling.

1.3 Contribution of Thesis

The contributions of this research are summarised as follows:

C1 Developed the Conversation Analysis Modelling Schema (CAMS). A novel dialogue annotation schema, that combines the CA concept of AP, with DA, to form AP-types, for the purpose of computational task-oriented dialogue modelling.

C2 Constructed a gold-standard corpus of task-oriented dialogues annotated with the CAMS. We annotated the Key-value Retrieval Network dataset (Eric and Manning, 2017a) with CAMS to produce a first-of-its-kind, AP-type labelled corpus, CAMS-KVRET, and made it available to the research community.

C3 Conducted detailed investigations of input representations and architectural considerations with respect to the DA classification task. Our findings are more comprehensive than any previously reported in the literature and in several cases are applicable to other short text classification tasks.

C4 Developed and refined several ML models capable of automatically labelling segments of dialogue with CAMS labels. Both our single-label and multi-label models are the first applications of DL algorithms to identify AP labels, and the latter model is capable of simultaneously applying all label components of CAMS.

In meeting our previously stated objectives, and in contribution to this work, we have published the following articles:

1. Duran, N., Battle, S. and Smith, J. (2022) Inter-annotator Agreement Using the Conversation Analysis Modelling Schema for Dialogue, *Communication Methods and Measures*
The work in this paper is presented in chapter 4.

2. Duran, N., Battle, S. and Smith, J. (2021) Sentence encoding for Dialogue Act classification. *Natural Language Engineering*
The work in this paper is presented throughout the first section (5.1) of chapter 5.

3. Duran, N. and Battle, S. (2018) Conversation Analysis Structured Dialogue for Multi-Domain Dialogue Management. *The International Workshop on Dialogue, Explanation and Argumentation in Human-Agent Interaction (DEXAHAI)*

4. Duran, N. and Battle, S. (2018) Probabilistic Word Association for Dialogue Act Classification with Recurrent Neural Networks. *EANN 2018*

1.4 Thesis Structure

This thesis is divided into 8 chapters. In this first chapter, we have explained the research motivation and problem and then outlined the aim, hypotheses, objectives and contributions of the work. The rest of this thesis is organised as follows:

- Chapter 2 provides a comprehensive literature review of CA and DA, followed by an overview of previous dialogue annotation studies and procedures, and finally, a discussion of task-oriented dialogue modelling.

- Chapter 3 describes the research framework that we have designed, and which aligns with the research questions and objectives discussed above. First, we discuss schema development, corpus annotation, and data collection. Next, we provide an overview for the development and evaluation of our automatic dialogue classification architectures. Finally, we outline our approach for evaluating the resulting dialogue representations in terms of dialogue management and modelling structure.

- Chapter 4 introduces the process used in the first stage of the research framework, developing and evaluating a CA annotation schema, and applying said schema to create a gold-standard CA labelled corpus.

- Chapter 5 describes the processes used in the second stage of our framework. We investigate several text pre-processing and input representation considerations, as well as single sentence and contextual model architectures, for the purpose of DA classification.

- Chapter 6 continues work on the second stage of our framework. Here we apply and evaluate the dialogue classification architecture developed in the previous chapter to automatically label task-oriented dialogue with our proposed method. Including, the as yet largely unexplored task of identifying AP, and AP-types.

- Chapter 7 outlines three experiments that comprise the final stage of our framework. We evaluate our proposed method in terms of response selection and generation for dialogue management, and use χ^2 analysis to produce dialogue structure graphs in order to consider structural qualities of our dialogue representations.

- Chapter 8 summarises our work and evaluates the findings with respect to our stated hypothesis, objectives and research questions. We also discuss possible limitations, the applications of our findings, and future directions of this research.

Chapter 2

Literature Review

Within the domain of Sociolinguistics, both CA, and DA, represent an extensive body of research. In the first section of this chapter we review these two fields, and in particular, those aspects that are relevant to the aims and objectives of this research. In relation to **O1**, we initially discuss the background of CA, and its key component of AP, followed by Speech Act Theory, and the development of DA. With respect to **O2**, we also outline several widely studied DA taxonomies. Regarding **O3**, **O4**, and **O5**, in sections 2.2 and 2.3, we review current approaches to DA classification and the identification of AP or dialogue structure. Finally, in section 2.4 we discuss the problem domain and define some key terms with respect to this research and **O5**.

2.1 Conversation Analysis and Dialogue Acts

Historically, conversation has received a great deal of attention, and been the subject of intellectual inquiry, for centuries. Yet, until the middle of the 20th century, much of what was written is *prescriptive* in nature, and describes a set of rules for what a conversation should be, or what makes a good 'conversationalist' (Burke, 1993). This approach frames conversation as an elite activity, a skill that must be taught, and learned. Of course, in reality, it is an everyday activity, with a fundamental role in human social life (Liddicoat, 2007). In contrast, CA is a *descriptive* approach to the study of talk as interaction, which grew from the ethnomethodological tradition developed by Garfinkel (1967). Ethnomethodology studies the *"common sense resources, practices and procedures through which members of a society produce and recognise mutually intelligible objects, events and actions"* (Liddicoat, 2007). From this viewpoint, conversation can only be understood by studying actual instances of naturally occurring social interaction, and it is this notion that was taken up by Harvey Sacks in the early 1960s (Coulter, 1995). CA later emerged from its sociological background – through the work of Harvey Sacks, Emmanuel A. Schegloff, and Gail Jefferson, in the late 1960s and early 1970s (Liddicoat, 2007; Sacks, Schegloff, and Jefferson, 1974; Schegloff and Sacks, 1973) – as an independent area of enquiry intended to understand the organisational structure of talk. The CA approach to the study of conversation is characterised by the idea that talk is an activity through which speakers accomplish their communicative goals, or actions, and that this is achieved through the orderly nature of talk. Clift (2016), considers these two characteristics of CA as, *"the two things from which all else follows"*. That is, *action* – the things we do with words – and *sequence*, or *sequence organisation* – *"the ways in which turns-at-talk are ordered and combined to make actions take place in conversation, such as requests, offers, complaints, and announcements"* (Schegloff, 2007). From these 'two things' we can begin to consider the association between DA theory, and CA. DA are, naturally, considered conversational actions, or communicative functions, for *single* utterances of dialogue. The

CA view differs, in that, the actions are an emergent property of the order, or sequence, in which *multiple* utterances are produced. In fact, from the above quote by Schegloff (2007), *request* and *offer* are commonly featured labels in DA annotation schema (Bunt, 2012; Shriberg et al., 2004; Jurafsky, Shriberg, and Biasca, 1997), and so there are clear similarities between the kinds of actions these two paradigms attempt to describe. Thus we have two different approaches, one focused on actions, the other, oriented towards the sequences that make up dialogue. It is these two notions that motivate the union of DA, with CA, that underpins this research. Specifically, DA theory provides a more direct and immediate description of the current actions being undertaken during talk, while CA is better suited to express the sequential, orderly, relationship between utterances, and hence the structure of the interaction. DA theory is discussed further in section 2.1.2, however, we first explore further the methods with which CA ascribes order within dialogue.

Psathas (1995), elaborates on the core assumptions of CA, and how order is produced:

1. *Order is produced orderliness.* Order does not occur on its own, or pre-exist the interaction, but rather, as the result of the coordinated practices of the participants.

2. *Order is produced, situated and occasioned.* Order is produced by the participants themselves for the conversation in which it occurs.

3. *Order is repeatable and recurrent.* Patterns of orderliness, not only in the talk of an individual speaker, but across groups of speakers, are repeated.

The first two points appear self-evident, and imply that, while CA assumes there is order within a given conversation, it is not generalisable across *all* conversations (Wooffitt, 2005), but rather produced by the speakers as the interaction develops. Of course, if this were not the case, we may find conversation a rather inflexible, unproductive, and tedious method of communication. However, the final point suggests that there are repeated patterns that are common between speakers, and by extension, between different conversations. In other words, the order that is produced is the result of a shared understanding between speakers for how conversations should be structured. A key goal of CA then, is to identify, and analyse, the activities performed by speakers in order to produce the orderly patterns observed through successive utterances of an interaction (Wooffitt, 2005; Luff, Gilbert, and Frolich, 1990). In order to achieve this, CA has a number of commonly employed analytical tools at its disposal, the most prominent of which are outlined in the following:

- *Turn-taking.* Initially this may sound similar to the concept of an AP. However, this component of CA is concerned with the social organisation that facilitate the most intuitive and commonplace unit of conversation, taking turns, that was initially outlined by Sacks, Schegloff, and Jefferson (1974). Though an obvious component of conversation, the manner in which we transition from one speaker to the next, with minimal silence between turns, and with little overlapping speech (in most cases), is remarkably complex. Central to the Sacks, Schegloff, and Jefferson (1974), view is that a speaker, upon initiating a turn, has the primary right to speak and that the transfer of speakership only becomes a possibility at certain junctures.

- *Repair.* When people talk together they frequently encounter problems of hearing, speaking, and understanding. The concept of repair refers to an organised set of practices through which participants are able to address and potentially resolve such problems (Liddicoat, 2007; Schegloff, Jefferson, and Sacks, 1977). Repair can be initiated by either the speaker or the recipient, and similarly the repair itself can be done by by either the speaker of the trouble source or someone else (Liddicoat, 2007). In either case however, repair is treated as a 'priority activity', which takes precedence over other rules of turn taking (Clift, 2016).

- *Preference Organisation.* The concept of preference refers to the different ways in which a conversational action may be achieved, and is closely linked with social and cultural norms. For example, the acceptance of an invitation is a *preferred* action, while declining is *dispreferred.* Yet, there is a further, more subtle, social element to preference organisation which suggests that dispreferred turns must be constructed in different ways. For instance, immediately declining an invitation may be considered rude, whereas delaying the declination response appears less so (Liddicoat, 2007). Preference organisation, therefore, typically operates over the SPP of AP, while the FPP proposes agendas or constraints on the next turn, which must be accepted or resisted (Clift, 2016).

Of course, each of these components of CA can be recognised as having a crucial role in social interaction. Certainly, they would also be interesting avenues of research for future CAI and computational modelling applications. However, while each of these aspects describe ways in which speakers produce orderly patterns within conversation, we exclude them from this research for a number of reasons. Primarily, they are all intrinsically linked with social or environmental cue's which are either difficult to incorporate, or entirely irrelevant, for most computational modelling purposes; speaker intonation, gaze, and pauses in speech require additional sources of input, and methods of representation, beyond the scope of the proposed text-based computational model. These components also relate to localised phenomenon within a conversation, typically two utterances for repair and preference organisation, or the transition between them for turn-taking, and are therefore too limited in scope to incorporate into a generalised representation of dialogue. Further, they may all be represented, or are directly associated, with AP, and would therefore be more effectively implemented via a higher-level AP-based computational model. Thus, we choose instead to focus on the primary method of sequence organisation within CA, the AP, and discuss this component in detail in the following section.

2.1.1 Adjacency Pairs

The concept of the AP begins with the observation that each utterance is associated with what comes prior, and what comes next (Schegloff, 2007, 1968); a notion that differs from that of linguistics and psychology, which have generally focused on the composition of the singular utterance in terms of a phrase or sentence. CA views the positioning of an utterance as fundamental to the understanding of its meaning, and the significance of it as an action. This view facilitates an understanding of social actions as positioned, either to initiate a possible sequence of action, or to respond to an already initiated action within the sequence (Sidnell and Stivers, 2013, Chapter 10). The minimal sequence in interaction consists of two paired utterances: the adjacency pair, and Schegloff and Sacks (1973), defined the characteristics of the AP in the following. They are:

1. Composed of two turns.
2. Produced by different speakers.
3. Adjacent to one another (in their basic unexpanded form).
4. Sequentially ordered into *First Pair Part* (FPP) and *Second Pair Part* (SPP).
5. Type related, so that a particular FPP suggests the relevance of a particular SPP, or a range of SPP.

The first two features in this list are straightforward, but the latter three warrant further explanation. Firstly, it is typically the case that the two turns of an AP occur immediately adjacent to one another, with no intervening talk. However, in some cases talk can come between the two turns via *insert-expansions*, discussed in the following section. Secondly, the two turns which make up an AP are ordered so that one always occurs first (FPP), initiating an exchange, and the other occurs second (SPP), and is responsive to the prior FPP; for example, a question always precedes its answer. Thirdly, the FPP and SPP of an AP are pair-type related. That is, when a FPP initiates a sequence, not every SPP can properly follow the FPP, but rather the second must be of an appropriate type for the action initiated by the first (Schegloff, 2007). For example, a question followed by an answer, or an offer may be accepted, declined, or perhaps an alternative suggested (conditionally accepted). Table 2.1 shows a number of common type-related AP, though is by no means exhaustive list (Sidnell and Stivers, 2013).

Table 2.1: Adjacency pair type relations.

FPP Action	**SPP Action**
Summons	Answer
Greeting	Greeting
Invitation	Acceptance/declination
Offer	Acceptance/declination
Request for action	Granting/denial
Request for information	Informative answer
Accusation	Admission/denial
Farewell	Farewell

In each of the examples in table 2.1 the FPP initiates some action and makes some next action relevant. The SPP responds to the prior turn and completes the action that was initiated, and together they accomplish the action (Liddicoat, 2007). Schegloff (1968), referred to this concept as *conditional relevance*, meaning, given the production of a FPP by a speaker, the SPP from the next speaker is immediately relevant and expectable.

The relationship between the two turns in an AP is not only *prospective*, but also *retrospective*, in that, a SPP also signifies its speaker's understanding of the FPP to which it responds. Thus, AP allow for a framework of understanding that is constructed and maintained on a turn-by-turn basis (Sidnell, 2010). Participants in conversation orient to this basic structure when constructing orderly sequences of talk and setting up expectations about how talk will proceed. The basic two-turn AP sequences described here are the primary building blocks of interaction, what Schegloff (2007) calls the 'base pair', and are common in the opening and closing sections of conversations and other types of interaction; as in the following simple telling-accept example from Liddicoat (2007).

JOHN: I've jus' finished by las' exam.
BETTY: That's great.

2.1.1.1 Sequence Expansion

The basic two-turn AP sequences are notable for their presence in most interactions but also because many other sequences are built around them. To account for more complex dialogue structures AP also include the concept of sequence expansion, which allows the construction of sequences of talk that are made up of more than one AP, while still contributing to the same basic action (Liddicoat, 2007). Sequence expansion is constructed in relation to a base sequence of a FPP and SPP in which the core action under way is achieved and are themselves AP, in that they consist of a FPP and SPP, with all the associated characteristics previously discussed. There are three types of expansion pairs: *pre-expansion*, *post-expansion*, and *insert-expansion*, which respectively take place before the FPP, after the SPP, or between the first and second pair parts.

Pre-expansion The first place at which a base pair may be expanded is before the occurrence of a base FPP. Pre-expansions are preparatory, or preliminary, to some other projected work, or action, to be implemented by the FPP of the base AP (Sacks, 1995; Schegloff, 1979). Thus, participants in pre-expansion sequences orient themselves towards a base AP which may subsequently develop. The initial turn of a pre-sequence, such as a pre-invitation, does two things: it suggests the possibility that a base FPP (for example, an invitation) will be produced; and indicates the production of a SPP, in response to the pre-invitation, is now relevant. From this response the expected occurrence of the base FPP (the invitation) is made relevant and the base pair is produced, as in the following example (Schegloff, 2007).

A:	What you doing?	***FPP-pre***
B:	Not much.	***SPP-pre***
A:	Wanna drink?	***FPP-base***
B:	Sure.	***SPP-base***

Like the example above, most pre-expansions are ‘type-specific’, in that they project towards a specific base FPP; for example, pre-invitations (“hey, are you busy tonight?”), pre-announcements (“Guess what happened to me?”), or pre-requests (“You wouldn’t happen to be going my way would you?”) (Sidnell, 2010). However, there is one pre-sequence that is not intended to reference the base pair action that it precedes, and is instead a generic pre-sequence that can be used to begin any kind of talk. The summons-answer sequence is simply designed to gain the attention of the recipient, for example “hey”, “excuse me”, or simply uttering their name (Liddicoat, 2007). One final point on pre-expansions, and which is in fact common to all expansion types, is that there are no restrictions on the number that may be produced. While there are some forms of pre-sequences that are more likely to come first, such as a summons-answer, multiple pre-expansions can be initiated and concluded prior to the FPP of a base AP (Schegloff, 2007).

Insert-expansion As alluded to in the previous section, the third characteristic of an AP, that the FPP and SPP are adjacent to one another, need not always be the case. In some instances a sequence may be *inserted* between the first and second pair parts of an AP (Schegloff, 1972). Insert-expansions interrupt the activity previously underway but are still relevant to that action and allows the second speaker (who must produce the SPP) to do interactional work relevant to the SPP. Once the sequence is completed the SPP once again becomes relevant as the next action. For example, a question (FPP-base) could be followed by a question (FPP-insert), to elicit information required to better answer the initial question. The insert-expansion is then concluded before completing the original base pair, as in the following example.

A:	Do you know the directions to the zoo?	***FPP-base***
B:	Are you driving or walking?	***FPP-insert***
A:	Walking.	***SPP-insert***
B:	Get on the subway...	***SPP-base***

Insert-expansions can be divided into two categories, *pre-seconds* and *post-firsts*, depending on the interactional activity they address (Schegloff, 2007). Pre-second insert-expansions, such as the example above, are type-specific, just as some of the pre-expansions discussed previously. That is, they are preliminary to a particular type (or types) of SPP, which is made relevant by the type of FPP to which it is responding. On the other hand, post-first insert-expansions are retrospective in that they are intended to 'repair' problems of hearing or understanding from the preceding talk, for example "sorry what?", or "huh?". Insert-expansions, like pre and post-expansions, have no restrictions on the number that may occur between the first and second pair part of an AP. So, for example, a post-first and pre-second insert-expansion may both take place, one after the other, within a base AP. Additionally, *all* AP may take insert-expansions, including expansions themselves. This implies, not only that base AP may take insert-expansions, but so too can pre, post and insert AP. The latter case results in a kind of 'nested' insert-sequence, where each subsequent level fulfils the same kind of interactional role for its enclosing insert-expansion as the pre-seconds or pre-firsts described previously. Though, Schegloff (2007, p. 110), notes that there is rarely more than a second level of nesting that occurs between the first and second pair parts of the base AP, as in the following example.

A:	Do you know the directions to the zoo?	***FPP-base***
B:	Are you driving or walking?	***FPP-insert$_1$***
A:	Which is faster?	***FPP-insert$_2$***
B:	Walking.	***SPP-insert$_2$***
A:	I'll walk then.	***SPP-insert$_1$***
B:	Get on the subway...	***SPP-base***

Post-expansion Once a base SPP has been completed, the sequence itself is potentially complete. The action initiated by the FPP is accomplished and a new action could begin, or the interaction concluded. However, it is possible for sequences to be expanded after their SPP with talk that is recognisably associated with the preceding sequence, a post-expansion (Liddicoat, 2007). In this respect Schegloff and Sacks (1973), consider sequences like a turn, a conversation, or any other structured unit that does not just end, but has a recognisable form of closure. Schegloff (2007), identifies two sorts of post-expansion: *minimal* and *non-minimal*.

Beginning with non-minimal post-expansions; these are made up of first and second pair parts and therefore behave much like all the AP previously discussed. These expansions are designed to project talk at least one further turn and can take many different forms. For example: repairing problems of hearing or understanding which may occur just as easily in SPPs as FPPs; rejecting, challenging, or disagreeing with a dispreferred SPP; reworking a FPP as a consequence of a dispreferred SPP; clarifying or confirming a SPP; and so on (Liddicoat, 2007; Schegloff, 2007). Hoewever, perhaps the most common and easily recognisable non-minimal post-expansions are sequence-closings, such as 'goodbyes', 'thankings', and 'acknowledgements', such as the following example.

A:	What is the weather like today?	***FPP-base***
B:	Forecast for cloudy skies today.	***SPP-base***
A:	Okay.	***FPP-post***
B:	No problem.	***SPP-post***

On the other hand, minimal post-expansions, or 'sequence-closing thirds' (Schegloff, 2007), are unique in relation to the expansion types discussed so far. Firstly, they consist of only *one* additional turn that takes place after a SPP. Secondly, they are not intended to project any further talk beyond itself. Minimal post-expansions, then, offer a reaction to the SPP response, but this reaction does not initiate a new sequence (AP) (Sidnell and Stivers, 2013). Responses are typically short (*minimal*) and designed to propose, or move towards, a sequence-closing, or conveying that the response to the action was adequate, for example, "Oh", "Okay", "Great", and so on. This can be illustrated using the above example, where speaker A's second utterance ("Okay") could reasonably be interpreted as a sequence-closing, with no further need for the production of a SPP by speaker B.

With these descriptions of AP in mind, we can begin to outline their utility within the context of computational dialogue modelling, NLP, and dialogue systems. Beginning with the first four of the five characteristics of AP defined at the beginning of this section; that they are composed of two turns by different speakers, adjacent, and sequentially ordered. These are indicative of structured, orderly patterns within talk, that are recurrent, and emerge from instances of natural conversation. Turns are either initiating, or responsive to, a particular desired action or conversational goal, and that they are produced by different speakers signals clear occasions for turn-taking. Together, these provide indications of the participants' (user) intent and the current dialogue state. Sequence expansions develop upon this basic structure and facilitate more complex sequences, beyond the simple two-turn AP, that are a necessary requirement for sophisticated dialogue modelling. As we have seen, expansions also fulfil important interactional roles in their own right, such as work preliminary to the action initiated by a base AP (pre), repair or elaboration required to complete an AP (inserts), or reworking, clarifying, and closings of previous first and second pair parts (post). The fifth characteristic, that AP are type-related, is particularly advantageous when considered from a computational perspective. The production of a given FPP type signifies the relevance, or expectation, of a particular set (or subset) of SPP types, as illustrated in table 2.1. Thus, system response selection is simplified by reducing the set of all possible responses to just a few types, or conversely, indicating a range of expected user responses for a given system utterance. Similarly the type-specific properties of sequence expansions contribute further information as to the nature of the current or future utterances, for example, the prospective pre-seconds, or retrospective post-firsts, for insert expansions. These qualities not only assists the process response generation, but also provides a flexible form of conversational flow, identifying dialogue state, and user intent induction.

This knowledge can be formulated in terms of structures to be employed in interactional sequences between a system and the user (Norman and Thomas, 1990). It is also important to note, that these features are not based on any theoretical speculation as to the 'nature' of conversation. Rather, they are insights about the character of conversational interaction that are obtained from empirical investigation (Wooffitt, 1990). However, despite these advantageous qualities, AP alone are not descriptive enough for a robust computational model of dialogue. Most notably, while there are extensive descriptions of many commonly occurring types of AP (Sidnell and Stivers, 2013; Liddicoat, 2007; Schegloff, 2007), there exists no formal, definitive, list of all possible types. Indeed, the existence of such a list would suggest only a limited set of such pairs are ever enacted, a notion too inflexible for the complex nature of conversations. Further, these type-relations are typically highlighted to fulfil the current analytic objectives of the CA practitioner, and no formal annotation conventions exist. For computational purposes then, we have a well founded set of structures for describing conversational actions across multiple utterances, and are yet lacking methods for explicitly, and flexibly, representing what those actions are. For this we turn to DA.

2.1.2 Dialogue Acts

While CA is regarded as a sociological approach to analysing conversation, Speech Act Theory has its origins in the philosophical study of meaning and how speakers intentions are expressed in language. The notion that uses of language can, and often do, have a 'character of action' was largely unrealised by the study of language before the 20th century (Smith, 1990). Wittgenstein (1953), posited that the *meaning* of a word is systematically related to its *use* in language, and this influenced Grice (1975), Searle (1969), and Austin (1962), among others, to form a new philosophical trend devoted to ordinary language analysis (Vanderveken and Kubo, 2001). Austin (1962), observed that in *saying* something the speaker is normally also *doing* something – making a request, promise, offer, apology, and so forth – or in Austin's terminology, performing an 'illocutionary act'. According to Austin, in performing an illocutionary act the speaker may also be performing: a 'locutionary act', uttering words with a certain sense of reference, and a 'perlocutionary act', that is, the effect the utterance has on the hearer, to convince, please, amuse, influence, and so on (Geis, 1995). With 'speech acts' Searle (1969), later developed upon Austin's theory and proposed five 'illocutionary points' that speakers can achieve on propositions in an utterance, they are: the *assertive*, *commissive*, *directive*, *declaratory*, and *expressive*. As described by Vanderveken and Kubo (2001), speakers achieve the *assertive point* when they represent how things are in the world, the *commissive point* when they commit themselves to doing something, the *directive point* when they make an attempt to get the recipient to do something, the *declaratory point* when they do things in the world at the moment of the utterance solely by virtue of saying that they do, and the *expressive point* when they express their opinions about objects and facts of the world. Another important insight from speech act theory is that the performance of a speech act requires that certain conditions be fulfilled. Of course these conditions will differ in form and number depending on the requirements of a given speech act. However, they do provide the intuition that speech acts convey not only the communicative goal, or action, of the speaker, but also the expectation of a certain effect on the addressee (akin to Austin's perlocutionary act). For example, to make them aware of the speakers presence (greeting), make certain information available to them (inform or statement), request or command and action, and so on. McTear, Callejas, and Griol (2016), referencing Searle (1969), provide an example for an utterance to be intended as a command by a speaker and understood as such by an addressee, the following conditions are required:

1. The utterance is concerned with some future act that the hearer should perform.
2. The hearer is able to do the act, and the speaker believes that the hearer can do the act.
3. It is not obvious to the speaker and hearer that the hearer will do the act in the normal course of events.
4. The speaker wants the hearer to do the act.

Thus, philosophers such as Austin (1962), and Searle (1969), reconceptualised speech as 'actions' and attempted to describe how spoken utterances can be classified according to a finite (and relatively limited) set of functions (Thornbury and Slade, 2006). It should be noted that *communicative acts* (Allwood, 1976), *conversation act* (Traum and Hinkelman, 1992), and *conversational moves* (Carletta et al., 1997), among others, are all broadly synonymous with speech acts.

The term *dialogue act* was introduced by Bunt (1978), and is the more frequently used term within contemporary Computer Science and NLP research, while perhaps also being the most generic in the context of dialogue (Traum, 2000). Bunt (2000), explains that, while speech act theory inspired the action-based approaches to language, and is a useful conceptual framework for human-computer dialogue, there are several points where it is not satisfactory for application to real dialogue or dialogue system design:

1. In speech act theory utterance interpretation is the assignment of an illocutionary force and propositional content, yet it is unclear exactly which illocutionary forces should be distinguished, and why. The illocutionary points defined by Searle (1969), do provide a taxonomic grouping of the basic semantic concepts. However, Bunt (2000), argues that a notion of *communicative functions* is required, which establish semantic definitions in terms of dialogue context changes.

2. Speech act theory considers that each utterance corresponds with a single illocutionary act, which may be functionally ambiguous. For example, Searle (1979, ch 2), demonstrates that “Can you pass me the salt?”, is both a question about the hearer’s ability to pass the salt and request to pass the salt. In contrast Bunt (2000), believes that communication has many ‘dimensions’ that a speaker can address simultaneously, and that some utterances can be considered to have several functions at the same time.

3. Crucially, although speech act theory naturally considers the interactive use of language, certain characteristics of spoken dialogue are somewhat overlooked. Common phenomenon, such as the use of feedback utterances (“Okay”, “Uh-huh”, “Hm”, and such like), hesitations, self-corrections, greetings, apologies, and so on. Speech act theory unhelpfully classifies all of these as ‘expressives’.

4. Finally, Bunt (2000), suggests that for application in the design of dialogue systems a formalised theory is needed, which takes into account the types of communicative acts that are relevant in the situation where the system is to be used. Moreover, that such a theory should be based on general principles like speech act theory, and also acknowledge that the set of communicative action types to be considered depends on the social environment, the linguistic community, the use of media, the kind of task, and so on.

These considerations illustrate some conceptual ‘gaps’ in traditional speech act theory and in part prompted the development of Dynamic Interpretation Theory (DIT) (Bunt, 1990), and by extension DA theory. Inspired by the study of spoken human-human *information exchange dialogues* (or simply information dialogues), DIT aims to identify fundamental principles of dialogue for the purpose of understanding natural dialogue phenomena, and for designing effective computer dialogue systems (Bunt, 2000). Information dialogues, that facilitate the exchange of factual information, have obvious practical applications for human-computer interaction. However, Bunt (2000, 1990), argues that *all* communication relies on conveying intentions and information, and therefore dialogues with more complex purposes will also require the concepts needed for information dialogues. Thus, it is important to make some distinctions between speech act theory, which is primarily rooted in the philosophy of language, and DA theory, which is a data driven approach to the computational modelling of language (Bunt, 2009); hence the widespread adoption of DA theory within NLP and dialogue system research (Eisenstein, 2018; Jurafsky and Martin, 2017).

A fundamental principle of DIT is the way meaning in communicative behaviour is described in terms of information-state-updates, or context changes within the interaction. In this view, DA are semantic concepts that are defined by the way it is intended to affect the information-state of an addressee. For instance, if the utterance “Do you know what time it is?”, is interpreted as a question, then the addressee’s information-state is updated with the information that the speaker does not know what time it is and would like to know. On the other hand, if the utterance is interpreted as reprimanding the addressee for being late, then the addressee's information-state is updated to include the information that the speaker does know what time it is. From this example we can define DA in terms of two components: i) its *communicative function*, what the speaker is trying to achieve (question or reprimand), and ii) the *semantic content*, which describes the information that is being addressed and should be used to update the information state – the entities, their properties, and relations that are referred to (Bunt, 2009). DIT defines a range of communicative function categories for DA, and this was later developed into a more general purpose taxonomy in DIT^{++} (Bunt, 2009). Additionally, as alluded to in point 2 above, DIT considers DA to be multidimensional. That is, DA are not mutually exclusive, and each utterance of dialogue may be assigned more than one DA to indicate that it is performing more than one communicative function. These concepts, and DIT in general, have proved to be highly influential to development of DA theory, and appear in various forms within the DA taxonomies discussed in the next section.

2.1.2.1 Dialogue Act Taxonomies

In contrast to CA, where there appears to be consensus on the conventions for annotation of the various phenomena, for DA a wide range of taxonomies have been proposed for different kinds of dialogue activities. For example, casual conversations (Jurafsky, Shriberg, and Biasca, 1997), classroom interactions (Sinclair and Coulthard, 1974), clinical research mediation (Hoxha et al., 2016), direction following (Carletta et al., 1997), multi-party meetings (Mccowan et al., 2005), and task scheduling (Alexandersson et al., 1997). Yet, despite their popularity, there have been relatively few attempts at formalisation (see Poesio and Traum (1998), for one example), and are often reduced to informal, intuitive concepts which lack proper definition (Bunt, 2011). Comparing the distributions of DA types across different domains, schemes, and corpora, Traum (2000), found that taxonomies for different tasks or genres of dialogue tend to be quite different; even within similar task-oriented dialogue domains, such as task scheduling and direction following. Though, to some extent this is expected given that different tasks will have different frequencies of DA types. The large number of DA taxonomies has led to a similarly large number of corpora annotated with DA. Indeed, corpora that are created to aid in the development of dialogue systems, or other NLP tasks, are frequently accompanied by their own bespoke task-related set of DA (Asri et al., 2017; Williams, Raux, and Henderson, 2016; Weston et al., 2015). The diversity in taxonomic approaches and the associated corpora has led to confusion over terminology and conceptual definitions, and problems reusing annotated corpora because it may be difficult to determine if equivalent categories exist between any two schema, or indeed if any functionally equivalent categories exist at all (Mezza et al., 2018; Bunt, Fang, and Liu, 2013; Bunt, 2011). Within the context of ML and NLP applications, models developed with a particular corpus, and for a given task, may be highly domain or task-specific. Thus, any kind of comparison between corpora and their labelling scheme (such as that of Traum (2000)), or data produced through analysis or experimentation, is made much more challenging. Here we provide an overview of two prominent DA taxonomies that are frequently used within NLP and DA classification research (see section 2.2), and are therefore relevant to this work.

DAMSL Dialogue Act Markup in Several Layers (DAMSL) (Allen and Core, 1997) is one of the most widely used DA taxonomies in computational linguistics and was primarily developed for two party task-oriented dialogues in which participants collaborate to solve a problem. The scheme has been applied in the annotation of both the TRAINS corpus (Allen et al., 1996), and the widely studied Switchboard corpus using the more elaborate

Switchboard Shallow-Discourse Function Annotation (SWBD-DAMSL) scheme (Jurafsky, Shriberg, and Biasca, 1997) (see section 2.2.1), among others. DAMSL defines four main categories of DA which indicate some aspect of the utterance itself, summarising the intentions of the speaker and the content of the utterance:

1. Communicative Status - records whether the utterance is intelligible and whether it was successfully completed.
2. Information Level - a characterisation of the semantic content of the utterance.
3. Forward Looking Function - how the current utterance constrains the future beliefs and actions of the participants and affects the discourse.
4. Backward Looking Function - how the current utterance relates to the previous discourse.

Note that the Information Level, Forward, and Backward Looking Function categories are effectively synonymous with the concepts of *semantic content* and *communicative function* from our previous definition of DA (Bunt, 2009). However, that DAMSL separates the forward and backward looking functions highlights an additional feature of DAs that is akin to the initiation and response relationship of AP that was discussed previously. That is: they may be *prospective*, and are intended to have a particular affect on the recipient, such as creating expectations of a certain action or response-type (*questions*, *statements*, *greetings*, and so on; or they may be *retrospective*, if they are responsive to a previous utterance (*agreements*, *answers*, and so on). Therefore, while DA generally considered to be representations of a single utterance, they may also convey the structure or relationship between localised utterances of dialogue.

DiAML The large number of bespoke DA taxonomies, and lack of formalisation, motivated the recent development of the ISO standard 24617-2 “Semantic annotation framework, Part 2: Dialogue acts” (British Standards Institution, 2012; Bunt et al., 2012), which defines the Dialogue Act Markup Language (DiAML), which originated from DIT (Bunt, 2009, 1990). DiAML is intended as an application-independent annotation scheme that is empirically and theoretically well founded, that can be used to annotate typed, spoken, and multimodal dialogue (where speech is used in combination with nonverbal behaviour) via human or automatic annotation methods (Bunt et al., 2012). As yet no corpora have been fully annotated with DiAML scheme, though several attempts have been made to map existing data to the scheme (Mezza et al., 2018; Petukhova, Malchanau, and Bunt, 2014; Fang, Bunt, and Cao, 2012).

DiAML accommodates all of the characteristics of DA examined so far and each of these can be represented within the DiAML notation for a single utterance. That a given utterances DA (in DiAML parlance, a *functional segment*) can have a relation to a previous, or future, DA is referred to as a *functional dependency*. The notion of multidimensionality is represented as the *dimension*. For example, the Task dimension, or Turn, Time, and Social Obligations Management dimensions. DA can also have additional *qualifiers* which encode the various ways in which a speaker can specify certain conditions, qualifications, or feelings accompanying a DA (Bunt, 2017). However, the core set of DA are defined within categories according to their *communicative function*, of which there are two classes: *dimension-specific* (such as Pause, Apology, and Take Turn), which can only be used in one specific dimension, and *general-purpose* ones, like Question, Answer, Offer, and Instruct, which can be combined with any kind of semantic content and form a DA in the corresponding dimension (Bunt et al., 2012). Table 2.2 shows the categories of communicative functions and their associated DA labels, organised into the general-purpose and dimension-specific classes.

Table 2.2: The DiAML categories of communicative functions and associated DA labels.

Communicative Function	DA Labels
General-purpose	
Information-seeking	Question, setQuestion, choiceQuestion, propositionalQuestion, checkQuestion
Information-providing	answer, inform, correction, agreement, disagreement, confirm, disconfirm
Commissive	promise, offer, addressRequest, acceptRequest, declineRequest, addressSuggest, acceptSuggest, declineSuggest
Directive	suggest, request, instruct, addressOffer, acceptOffer, declineOffer
Dimension-specific	
Feedback	autoPositive, alloPositive, autoNegative, alloNegative, feedbackElicitation
Turn Management	turnAccept, turnAssign, turnGrab, turnKeep, turnRelease, turnTake
Time Management	stalling, pausing
Owner and Partner Management	completion, correctMisspeaking, selfError, retraction, selfCorrection
Discourse Structuring	interactionStructuring, opening
Social Management	initialGreeting, returnGreeting, initialSelfIntroduction, returnSelfIntroduction, apology, acceptApology, thanking, acceptThanking, initialGoodbye, returnGoodbye

2.1.3 Summary

As we have seen, DA, and the various taxonomies, provide an empirically well founded method of expressing the intended communicative function (or action), and semantic content, of a single utterance of dialogue; hence, their widespread use in many NLP applications (Jurafsky and Martin, 2017; McTear, Callejas, and Griol, 2016). However, while DA's provide valuable semantic and intentional information, as Clift (2016) points out, *"the form of an utterance alone cannot necessarily be relied upon to deliver how it is understood by its recipient"*. For example, "Is that your coat on the floor?", said by parent to child, is not the propositional question that its form suggests, but rather a directive to pick the coat up. In a further criticism of DA theory, Geis (1995) and Patten, Geis, and Becker (1992), illustrate that conversational actions, such as making a request, issuing an invitation, conveying information, and so on, are not normally concluded in a single turn of conversation. Instead, these kinds of actions are completed over several turns and should therefore be viewed as properties of conversational sequences, rather than individual utterances. Indeed, Searle (1992) himself noted that speech act theory does not provide a promising foundation for the account of conversation. It is perhaps surprising then, given these limitations, that DA have been so widely adopted within the field of NLP and dialogue systems. Yet, DA are uniquely suited to explicitly, and flexibly, represent the individual actions of utterances within the wider structure of conversational action provided by AP. Thus, we propose that DA and AP theory can be unified into a single taxonomy for describing the structure and meaning for sequences of utterances within conversations. In this view, DA can be considered the *type* label for an individual utterance, and when combined with AP, accommodate the pair-type relations associated with the component utterances of an AP. Or, viewed another way, we can consider DA labels as descriptions of the *intra-utterance* features of a dialogue, while AP represent the *inter-utterance* features. In this way we can overcome the aforementioned drawbacks of DA, as a method of representing contextual or structural meaning of dialogue, and those of AP, lacking any formalisation and flexibility for the crucial pair-type relationships between utterances.

It should be noted that the utility of CA and AP for the purpose of HCI (Luff, Gilbert, and Frolich, 1990), and NLP (Jurafsky and Martin, 2017; McTear, Callejas, and Griol, 2016; Patten, Geis, and Becker, 1992) has been recognised previously. However, thus far this has primarily been an acknowledgement of the benefits of a conversational agent, or dialogue-state model (Jurafsky and Martin, 2017), that emulates or recognises the processes of human conversation. For example, determining the type of response the system should make to a users utterance (McTear, Callejas, and Griol, 2016), or initiating repair (Alloatti, Caro, and Bosca, 2020). To the best of our knowledge, no attempt has been made to directly incorporate CA theory into a computational model of dialogue, nor to combine DA and AP into a more descriptive representation. Instead, dialogues are generally represented as a sequence of DA alone (Griol et al., 2014, 2008).

2.2 Dialogue Act Classification

The task of automatic DA classification, sometimes referred to as short text classification, simply put, is the task of assigning a DA label, or labels – from a chosen DA taxonomy – to a given input sentence, or sequence of sentences. Typically, input sentences are utterances from human-human or human-machine dialogues, but DA classification has also been applied to emails and forum posts (Wang et al., 2011; Jeong, Lin, and Lee, 2009), for example, Wikipedia discussions (Jamison and Gurevych, 2014). Within the field of NLP many applications have been developed that utilise the automatic identification, or classification, of DA. Most prominently, within dialogue management systems, they have been used as high-level representations for user intents, system actions and dialogue state (Firdaus et al., 2020; Liu et al., 2018; Cuayáhuitl et al., 2016; Wen et al., 2016; Ge and Xu, 2015; Griol et al., 2008). DAs have also been applied to: spoken language translation (Sridhar, Narayanan, and Bangalore, 2009; Kumar et al., 2008; Reithinger et al., 1996); team communication in the domain of robot-assisted disaster response (Anikina and Kruijff-Korbayova, 2019); and understanding the flow of conversation within therapy sessions or psychiatric interviews (Bifis et al., 2021; Lee et al., 2019).

With respect to this research, specifically objectives **O3** and **O4**, DA classification is of particular interest. It has received a considerable amount of attention within the literature, and of course, this is directly relevant to the identification of DAs that is intrinsic to our proposed approach. However, given that the task is to classify utterances of dialogue with a chosen set of labels, we surmise that there is no fundamental difference between DA and AP, especially considering the contextual approaches to DA classification discussed in section 2.2.3. Thus, much of the previous work on DA classification will *also* be relevant to the identification of AP. This is particularly valuable, given the comparatively sparse literature on the automatic identification of AP (see section 2.3). In the following we provide an overview of DA labelled corpora, before discussing approaches to automatic DA classification in the subsequent sections.

2.2.1 Dialogue Act Corpora

As discussed in section 2.1.2.1, a large number of DA taxonomies have been developed for various kinds of dialogue activities, and this has led to a similarly large number of corpora annotated with DA. Typically, the corpora and their accompanying DA schema are created to aid in the development of dialogue systems for a particular task, for example, question answering (Weston et al., 2015), or information retrieval and customer services (Asri et al., 2017; Williams, Raux, and Henderson, 2016). However, several corpora of human-human interactions, in both task-oriented and non-task-oriented domains, have been annotated with DA. The following provides a brief overview of several that are the most widely studied within DA classification research.

HCRC Maptask Developed at the Human Communication Research Centre (HCRC) at the University of Edinburgh, the HCRC Maptask corpus (Anderson et al., 1991; Thompson et al., 1991), is intended to facilitate the study of natural dialogues from many different perspectives in a task-oriented cooperative problem solving domain. It contains 128 conversations that were produced by pairs of participants, 64 total, who worked collaboratively on a map annotation task. In the task, both participants have a map which is only visible to them and not to the other participant. Each map consisted of an outline and around a dozen labelled features, for example, "white mountain" or "oak forest". Most features are common to the two maps, but not all, and the participants were informed of this. One map had a route drawn in, the other did not. The task was for the participant without the route (the instruction follower) to draw one on the basis of discussion with the participant with the route (the instruction giver) (Thompson et al., 1991).

The transcribed dialogue utterances were annotated with one of 12 different *conversational moves*, or DA, and these are primarily either initiations or responses, according to their purpose (Carletta et al., 1996). Initiating moves set up the expectation of a response, such as: instructions (*instruct*); stating information (*explain*); confirming information (*check*); checking attention, agreement or readiness (*align*); "yes" or "no" questions (*query-yn*); and "who", "what" "where" type questions (*query-w*). Responsive moves fulfil the expectations of initiations by: acknowledging a move is understood (*acknowledge*); replying "yes" (*reply-y*); replying "no" (*reply-n*); replying in a form that is not "yes" or "no" (*reply-w*); and a reply in which the speaker includes information beyond what was strictly asked (*clarify*). In addition to the initiation and response moves, the (*ready*) moves occur when one section of the task (and thus dialogue) is complete and signify readiness for more instructions, for example "OK" or "right". Maptask was annotated with an additional higher-level grouping of these conversational moves, known as *conversational games*. That is, a set of utterances starting with an initiation and encompassing all utterances up until the game has been fulfilled. For example, a request for information has been completed or abandoned.

The Maptask coding scheme is a good example of a task-specific set of DA that are appropriate for the kind of interactions found in Maptask dialogues, but do not generalise to *all* task-oriented, or even general, conversations. Nevertheless, the Maptask corpus has been used in several DA classification studies (Wan et al., 2018; Tran, Haffari, and Zukerman, 2017; Tran, Zukerman, and Haffari, 2017), where it has proven to be one of the most difficult datasets to achieve high accuracy for the task.

A:	okay	***Align***
A:	and then we're going to turn east	***Instruct***
B:	mmhmm	***Acknowledge***

Switchboard The Switchboard corpus of spontaneous conversational speech was automatically gathered by Texas Instruments for the purpose of training and testing speech processing algorithms (Godfrey, Holliman, and McDaniel, 1992). It contains 2500 naturalistic conversations by 500 paid participants from around the U.S, and consists of telephone conversations between two participants, who did not know each other, and were assigned one of 70 topics to discuss. Jurafsky, Shriberg, and Biasca (1997), later annotated a subset of the Switchboard corpus using a slightly modified version of DAMSL (SWBD-DAMSL), to form the Switchboard Dialogue Act corpus (SwDA). The SwDA corpus contains 1,155 conversations, comprising 205,000 utterances, and 42 unique DA labels. The SWBD-DAMSL label set is multidimensional; approximately 50 basic labels (such as, Question, Statement) could each be combined with additional dialogue function information, for example, Task-Management and Communication-Management. However, the many possible combinations were reduced to a set of 42 mutually exclusive DA labels and annotation was carried out over a three-month period in 1997 by eight linguistics graduate students at CU Boulder. Inter-annotator agreement (IAA) for the 42-label set used was 84%, resulting in a Kappa statistic of 0.80 (Cohen, 1960). Though, it should be noted that the DA labels within SwDA are highly imbalanced, with the three most frequent DA (Statement-opinion, Statement-non-opinion, and Acknowledge (Backchannel)) comprising ~70% of all labels. Further, the training (1,115 conversations) and test (19 conversations) data split suggested by Stolcke et al. (2000) is also imbalanced with regards to the label distributions, and yet has been widely adopted throughout the literature. Nevertheless, SwDA is one of the largest and most frequently studied non-task-oriented corpora within DA classification research (Colombo et al., 2020; Bothe et al., 2018a; Wan et al., 2018; Cerisara, Král, and Lenc, 2017; Papalampidi, Iosif, and Potamianos, 2017; Tran, Haffari, and Zukerman, 2017; Ribeiro, Ribeiro, and De Matos, 2015; Stolcke et al., 2000).

A:	I'm more out in the suburbs,	***Statement-non-opinion***
A:	but I certainly work near a city.	***Statement-non-opinion***
B:	Okay,	***Acknowledge***

MRDA The Meeting Recorder Dialogue Act (MRDA) corpus (Shriberg et al., 2004), was developed as part of the International Computer Science Institute (ICSI) Meeting Recorder Project (Janin et al., 2004). It is intended to provide a resource for studying the complex discourse phenomenon present in meetings, such as regions of high speaker overlap, complicated interaction structures, abandoned or interrupted utterances, and other turn-taking and discourse-level features. It contains transcriptions of 72 hours of speech from 75 naturally-occurring multi-party meetings with 53 unique speakers and an average of ~6 speakers per meeting. In total ~108,000 utterances were labelled with a modified version of the SWBD-DAMSL (Dhillon et al., 2004), with three different levels of granularity. The *specific* set contains 52 labels, most, but not all, are compatible with the SWBD-DAMSL scheme. The *general* set contains 12 labels that are generalised versions of the specific set, and the *basic* set contains just 6 labels that are generalised versions of the general set. The MRDA corpus is unique because it was also annotated with AP and, to the best of our knowledge, is the only large corpus created for the purpose of computational modelling or analysis that is annotated with both DA and AP. Unfortunately, only some FPP and SPP utterance pairs were labelled with AP, and the schema contains no concept of expansion. AP annotations are therefore quite sparse within the corpus making it unsuitable for our purpose without complete re-annotation. Regardless, MRDA is the largest task-oriented DA labelled corpora and is frequently studied within DA classification research, typically using the basic label set (Colombo et al., 2020; Ribeiro, Ribeiro, and De Matos, 2019; Chen et al., 2018; Kumar et al., 2017; Ortega and Vu, 2017; Lee and Dernoncourt, 2016).

A:	okay	***FloorGrabber***
A:	some some introductions are in order.	***Statement***
B:	oh okay.	***Statement***

2.2.2 Unsupervised, Supervised, and Probabilistic

Much of the extensive earlier literature on automatic DA classification focuses on conventional (non-ANN) supervised, unsupervised, and probabilistic approaches. Some of the earliest examples applied statistical methods to the problem. For example, Reithinger and Klesen (1997), formulated the problem as:

$$DA = \underset{\acute{D}}{\operatorname{argmax}}\, P(\acute{D}|W) \tag{2.1}$$

Where $\acute{D}$ is the set of probabilities that each DA is associated with the string of words W, which consists of varying length n-grams. The argmax component then selects the most probable DA, of all possible labels, given the range of probabilities in $\acute{D}$. They additionally noticed that including knowledge about the previous dialogue history H, the predicted DA, improved classification by 3%. Their final classifier was therefore:

$$DA = \underset{\acute{D}}{\operatorname{argmax}}\, P(\acute{D}|W)P(\acute{D}|H) \tag{2.2}$$

Using this approach Reithinger and Klesen (1997) achieved classification accuracy of 67% on the German, and 74% on the English, versions of the VERBMOBILE-2 corpus (Alexandersson et al., 1997). Similar use of Bayesian classifiers with n-gram features was employed by Grau et al. (2004), who investigated the effect of different smoothing methods, and both Louwerse and Crossley (2006) and Webb and Hepple (2005), explored the association between DAs and occurrences of varying length n-grams within the utterance. Keizer (2001), alternatively formulated the problem using a Bayesian Network with three components: the belief state (of the previous DA), the DA of the current utterance, and the linguistic features of continuation patterns or question marks. With a similar approach Král, Pavelka, and Cerisara (2008) instead modelled the position of each word or n-gram within the utterances, as well as prosodic information.

A similar, and perhaps one of the most common, statistical methods applied to the problem, is Hidden Markov Models (HMM) combined with a range of techniques for creating textual feature representations and generating predictions. For example, Ries (1999) created a hybrid HMM with an ANN classifier and Julia and Iftekharuddin (2008) and Surendran and Levow (2006) both combined Support Vector Machines (SVM) with HMM using text and acoustic features. The former created an ensemble of HMM and SVM classifiers for each of the different feature types, and the latter used SVM for local information of the current DA and a HMM for previous DA within the sequence. Other examples of HMM hybrids, or similar statistical methods, include the use of Latent Dirichlet Allocation (Zhai and Williams, 2014), Maximum Likelihood (Boyer et al., 2010a), Maximum Entropy (Sridhar, Narayanan, and Bangalore, 2009), and Conditional Random Fields (CRF) (Quarteroni, Ivanov, and Riccardi, 2011). One of the seminal studies on DA classification and the use of HMM, applied to the SwDA corpus, explored the use of transcribed words, speech recognised words, and prosody, such as pitch, duration and energy, as input features for the classifier (Stolcke et al., 2000). Unsurprisingly they found performance was better using transcribed words (71.0%), than with the added uncertainty of automatically recognised words (64.8%). For the prosodic features they also explored the use of decision trees and ANN classifiers, with accuracies of 45.4% and 46.0% respectively. Nevertheless, the training and test data split suggested by Stolcke et al. (2000) (despite the imbalanced classes), and the 71% accuracy reached with transcribed words remains a benchmark for the SwDA corpus.

Considering other supervised, unsupervised and semi-supervised approaches. Both Jeong, Lin, and Lee (2009) and Shriberg et al. (1998) applied decision trees to the problem, the former trained an ensemble of trees on the MRDA and SwDA corpora, with Bag of Words (BOW) and n-gram features, before exploring the generalisability to DA classification of emails and forum posts. Yang et al. (2015) computed sentence similarity with word embeddings and sentence syntax before applying k-means clustering, while Ezen-Can and Boyer (2015) employ Markov Random Field clustering. Milajevs and Purver (2014) encoded distributional information of word and utterance order using BOW, n-gram and previous utterance features before applying k-nearest neighbours for classification, and Ribeiro, Ribeiro, and De Matos (2015) used the frequencies of n-grams as features with a SVM classifier. Finally, Novielli and Strapparava (2009), Serafin and Di Eugenio (2004), and Serafin, Eugenio, and Glass (2003), applied Latent Semantic Analysis (LSA) to the task, though the former also found that SVM was much more effective on the same (SwDA) task.

We can summarise these approaches in terms of two aspects, i) the representation of words and utterances, and ii) other contextual discourse information for a given utterance, for example, its positioning within the dialogue itself, the sequence of DAs, or speakers. In other words, how (or what) words compose to form the meaning of an utterance, and hence its associated DA, and the relationship between a given utterance and the surrounding utterances. This latter point is particularly salient when we recall that both DA and AP may have a forward and backward looking component. Many of these studies used conventional NLP techniques for utterance representation, or features. Primarily, varying length n-grams or BOW, but in some cases acoustic (Julia and Iftekharuddin, 2008; Surendran and Levow, 2006) or prosodic (Stolcke et al., 2000) information was used. However, most were also conducted before the advent of word embeddings, and so were unable to take advantage of these representations. In terms of dialogue context, Reithinger and Klesen (1997) found that including predictions for previous DA within the sequence improved classification accuracy, and indeed many of the HMM, and similar probabilistic approaches, incorporated this notion – predicting sequences of DA. In most cases including predictions, or previous utterance features, improves performance, though often contextual and non-contextual approaches were not compared. Table 2.3 provides an overview of several of the studies discussed in this section that report results for two of the corpora reviewed in section 2.2.1, SwDA and Maptask. It is worth noting that, for the SwDA corpus, the two studies that report higher performance than Stolcke et al. (2000) use either a much simplified labelling scheme (Yang et al., 2015), or deviate from the conventional training and test split (Cross-validation) (Ribeiro, Ribeiro, and De Matos, 2015), which significantly effects results for this corpus and makes comparison difficult. Similarly, Serafin and Di Eugenio (2004) utilised the ‘conversational game’ annotations for Maptask, which are a feature unique to this corpus. Thus, despite the wide range of approaches summarised here, many of which are far more contemporary than that of Stolcke et al. (2000), there are relatively small differences in performance between conventional NLP models when applied to the DA classification problem. More recently Deep Learning (DL) neural network approaches have yielded greater improvements, and we review these in the following section.

2.2.3 Neural Architectures and Language Models

The performance of contemporary DL neural network techniques, often based on recurrent, convolutional, and more recently Transformer architectures (Vaswani et al., 2017), have surpassed that of the more traditional NLP approaches discussed in the previous

Table 2.3: Summary of conventional Unsupervised, Supervised, and Probabilistic, DA classification Studies.

Study	Model	Features	SwDA	Maptask
Grau et al. (2004)	Naive Bayes	3-gram	66.0%	-
Webb and Hepple (2005)	Cue Phrase Selection	1 to 4-gram	69.1%	-
Stolcke et al. (2000)	HMM	n-gram	71.0%	-
Surendran and Levow (2006)	SVM and HMM	Text	-	59.1%
	SVM and HMM	Text + Acoustic	-	65.5%
Julia and Iftekharuddin (2008)	SVM and HMM	Text	-	55.4%
	SVM and HMM	Text + Acoustic	-	68.6%
Sridhar, Narayanan, and Bangalore (2009)	Maximum Entropy	-	70.4%	-
Quarteroni, Ivanov, and Riccardi (2011)	CRF	-	70.9%	-
Milajevs and Purver (2014)	KNN	Bag of bi-grams	62.1%	-
	KNN	Bag of bi-grams + Prev Utt	63.9%	-
Jeong, Lin, and Lee (2009)	Tree Ensemble	BOW + n-gram	63.7%	-
Yang et al. (2015)[†]	Embedding Similarity	-	78.6%	-
	K-means	Embedding Similarity	81.6%	-
Serafin and Di Eugenio (2004)[‡]	LSA	Game + Speaker + Prev DA	-	73.3%
	LSA	Game + Speaker	-	73.9%
Ribeiro, Ribeiro, and De Matos (2015)[+]	SVM	2 n-gram frequency	73.7%	-
	SVM	2 n-gram frequency + Prev Utt	74.9%	-

[†] Uses 10 simplified DA labels from the set of 42 SwDA labels, thus not directly comparable to other SwDA results.
[‡] The 'conversational game' features are unique to the Maptask corpus, so not generalisable to other corpora.
[+] Uses Cross-validation rather than the standard training and test split suggested by Stolcke et al. (2000), thus not directly comparable to other SwDA results.

section. Similar to the conventional NLP approaches, much of the prior work on neural networks has considered the problem in terms of i) how to appropriately represent words and utterances, and ii) how to incorporate other contextual discourse information. Thus, regardless of architectural variations, neural network models may be broadly split into two categories, referred to here as *single-sentence* and *contextual*. Single-sentence models take one utterance of dialogue as input and assign a predicted DA label for that utterance. On the other hand, input for contextual models includes additional historical or contextual information, for example, indicating a change in speaker, previous dialogue utterances or previously predicted DA labels. In some cases, the contextual information may also include *future* utterances or DA labels, in other words, those that appear after the current utterance requiring classification; though, the utility of such future information for real-time applications such as dialogue systems is questionable. Within DA classification research, it has been widely shown that including such contextual information yields improved performance over single-sentence approaches (Ribeiro, Ribeiro, and De Matos, 2019; Bothe et al., 2018a; Liu and Lane, 2017; Lee and Dernoncourt, 2016). Consequently, much of the contemporary research has focused on representing contextual information and related architectures. Yet, both single-sentence and contextual classification models share some commonalities. Primarily, that is, each input utterance, or utterances, must first be encoded into a format conducive to classification – most commonly with several Feed Forward Neural Network (FFNN) layers – or for further down-stream operations, such as combining additional contextual information (Bothe et al., 2018b; Ortega and Vu, 2017; Papalampidi, Iosif, and Potamianos, 2017; Lee and Dernoncourt, 2016; Kalchbrenner and Blunsom, 2013). In other words, the plain text input utterances must be converted into a vector representation that *'encodes'* the semantics of the given utterance. Hence, both single-sentence and contextual models tend to share a common *sentence encoding* module. Though specific implementation details may vary, most may be described with the generic DA classification architecture diagram shown in Figure 2.1. In short, the encoding module converts the plain text input sentences into the vectorised representations necessary for classification or other downstream tasks, such as concatenation with other contextual data. The following sections discuss each component of Figure 2.1 (numbered 1 through

5) in more detail.

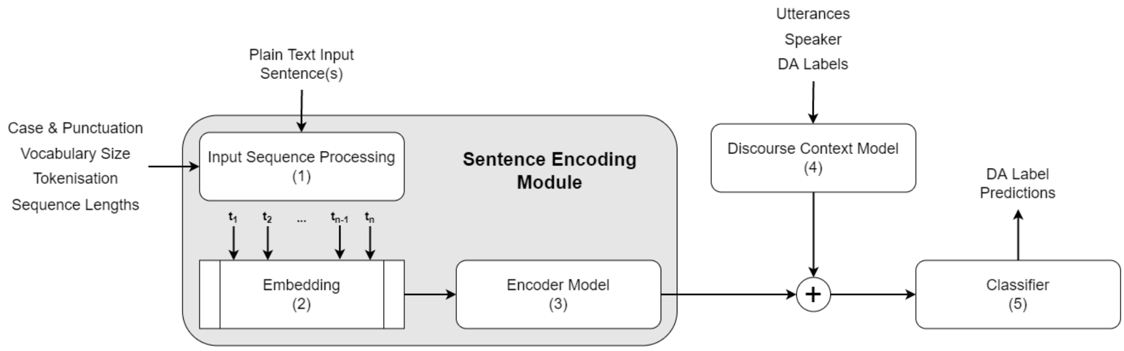


Figure 2.1: A generic DA classification architecture, including the Sentence Encoding Module (components 1-3), and example parameters (Sequence Length, Vocabulary, etc), additional context information (4), and final classifier (5).

2.2.3.1 Input Sequence Processing

The input sequence processing component (1), takes as input a plain text sentence and produces a tokenised sequence. Generally, this procedure is carried out as part of pre-processing the data prior to training, or inference. Sequence processing involves several text pre-processing steps, however, very few studies have explored the impact that different parameters may have on the resulting sentence encodings. Here, we discuss several different, in some cases optional, aspects of the sequence processing component: letter case and punctuation, vocabulary size, tokenisation and sequence length.

Letter Case and Punctuation Letter case simply refers to the optional pre-processing step of converting the letters in all words to lower case, or not, which helps to reduce 'noise' within the data. Firstly, by reducing repeated words in the vocabulary, for example, removing words that are capitalised at the beginning of a sentence and appear lower cased elsewhere. Secondly, by removing capitals from names, abbreviations, and so on. Converting all words to lower case is common practice in many NLP applications and the same appears true for DA classification (Chen et al., 2018; Wan et al., 2018; Kumar et al., 2017; Ji, Haffari, and Eisenstein, 2016).

Whether to remove punctuation, or not, is another optional pre-processing step. It seems reasonable to assume that, for the DA classification task, some punctuation marks may contain valuable information which should not be removed. Certainly, an interrogation mark at the end of a sentence should indicate a high probability that it was a question. For instance, Wan et al. (2018) removed all punctuation marks except for interrogation, Kumar et al. (2017) removed only exclamation marks and commas, and Webb and Hepple (2005) removed all punctuation. Yet, Ortega et al. (2019) found that keeping punctuation was beneficial for DA classification using the MRDA dataset, as did Żelasko, Pappagari, and Dehak (2021) for the MRDA and SwDA corpora.

In addition to case and punctuation, lemmatising words or converting them to their Parts of Speech (POS) is a frequently used pre-processing step within NLP applications. However, previous studies have shown that, whether used as additional features (Kumar et al., 2017), or replacing words entirely (Ribeiro, Ribeiro, and De Matos, 2019), they result in an unfavourable effect on performance.

Vocabulary Size Corpora often contain a large number of unique words within their vocabulary. It is common practice, within NLP and DA classification tasks, to remove words that appear less frequently within the corpus. Or in other words, to keep only a certain number – the *vocabulary size* – of the most frequent words, and consider the rest out-of-vocabulary (OOV); which are often replaced with a special 'unknown' token, such as <unk> (Wan et al., 2018; Ji, Haffari, and Eisenstein, 2016). Though a vocabulary size is often stated within DA classification studies, it is generally not accompanied with an explanation of why that value was chosen. For example, the SwDA corpus contains ~22,000 unique words (this varies depending on certain pre-processing decisions), yet different studies have elected to use vocabulary sizes in the range of 10,000 to 20,000 words (Li et al., 2019b; Raheja and Tetreault, 2019; Chen et al., 2018; Kumar et al., 2017; Ji, Haffari, and Eisenstein, 2016; Lee and Dernoncourt, 2016), while Wan et al. (2018), kept only words that appeared more than once within the corpus. Only one previous study has explored the effect of different vocabulary sizes on the DA classification task. Cerisara, Král, and Lenc (2017), conducted experiments on the SwDA corpus using different vocabulary sizes in the range 500 to 10,000. They found that, with their model, the best performance was achieved with a vocabulary size of between 1,000 and 2,000 words and that accuracy slightly decreased with larger vocabularies.

Tokenisation and Sequence Length The final stage of preparing the input sequence is that of transforming the plain text sentence into a fixed length sequence of word or character tokens. Tokenisation at the word level is the most common approach for DA classification because it enables the mapping of words to pre-trained embeddings, and hence facilitates transfer learning. Though, recently some studies have also explored character based language models (LM) (Bothe et al., 2018b), or a combination of character and word embeddings (Raheja and Tetreault, 2019; Ribeiro, Ribeiro, and De Matos, 2019). In any case, once the text has been tokenised, it is padded, or truncated, to a fixed size sequence length, or *maximum sequence length*. In cases where the number of tokens is less than the maximum sequence length, extra 'padding' tokens, such as <pad>, are used to extend the sequence to the desired size. Input sequences must be converted to a fixed length because many sentence encoding and classification models require the size of the input data to be defined before run-time, or before processing a batch of data, for example, to determine the number of iterations over the input sequence for recurrent models. The final tokenisation step is to simply map each word, or character, to an integer representation. In the case of word tokens this is typically the words' index within the vocabulary.

Choosing a sequence length equal to the number of tokens in the longest sentence in the corpus may result in the majority of sequences consisting predominantly of the padding token, and hence, increasing the computational effort without adding any useful information. For instance, the SwDA corpus has an average of ~9.6 tokens per utterance, yet the maximum utterance length is 133 tokens. On the other hand, if an input sequence is too short, the process of truncation could remove information valuable to the encoding and classification process. However, considerations around appropriate values for input sequence length are rarely discussed within the literature. To the best of our knowledge, thus far only two studies have explored the impact of different sequence lengths on the DA classification task. Cerisara, Král, and Lenc (2017), tested different sequence lengths in the range 5 to 30 on the SwDA corpus. They found the best performance was achieved using 15 to 20 tokens, with further increases not yielding any improvement. Similarly, Wan et al. (2018), using the same corpus, tested sequence lengths in the range 10 to 80 and achieved their best results with a sequence length of 40, with further increases actually reducing performance.

2.2.3.2 Word Embeddings

The embedding component (2), is often the first layer of a DA classification model. Though, this is typically not the case with many pre-trained LM, where input is simply the tokenised sentence (see Section 2.2.3.3). In contrast to the n-gram and BOW features discussed in section 2.2.2, word embeddings offer a far richer semantic and relational word representation that can be used as features for a wide range of downstream tasks, such as sequence labelling.

The embedding layer maps each word in the tokenised input sequences to higher dimensional vector representations, most frequently with pre-trained embeddings, such as Word2Vec (Mikolov et al., 2013), and GloVe (Pennington, Socher, and Manning, 2014). However, within the literature a number of studies simply state the type and dimensions of the embeddings used (Ahmadvand, Choi, and Agichtein, 2019; Li et al., 2019b; Ortega and Vu, 2017; Lee and Dernoncourt, 2016), while others have explored several different types or dimensions (Cerisara, Král, and Lenc, 2017; Papalampidi, Iosif, and Potamianos, 2017). For instance, Ribeiro, Ribeiro, and De Matos (2019) examined a number of 300 dimensional pre-trained embeddings: Word2Vec, FastText (Joulin et al., 2017), and Dependency (Levy and Goldberg, 2014), with the latter yielding the best results. In contrast, it appears 200 to 300 dimensional GloVe embeddings, are used more frequently within DA classification studies (Li et al., 2019b; Chen et al., 2018; Wan et al., 2018; Kumar et al., 2017; Papalampidi, Iosif, and Potamianos, 2017; Lee and Dernoncourt, 2016). As such, it is unclear what impact different types of pre-trained embedding, or dimensionality, choices may have on classification results. As an example, according to the results reported by Ribeiro, Ribeiro, and De Matos (2019), the difference between their best and worst performing pre-trained embeddings, Dependency and FastText respectively, is 0.66%. While the difference between FastText and Word2Vec was only 0.2%.

2.2.3.3 Sentence Encoding

The encoder model component (3), is, of course, the key aspect of the sentence encoding process. Here we discuss sentence encoders in terms of two categories; (i) models that have been trained in a supervised fashion, which is the predominant approach within DA classification research, and (ii), those that use a language model – or pre-trained language model – to generate sentence encodings, an approach which, despite widespread application to many NLP tasks, has thus far received little attention for DA classification. Table 2.4 provides a summary of several of the models and input features used within the single-sentence DA classification studies discussed in the following sections.

Supervised Models Supervised encoder models take as input a sequence of tokens that have been mapped to higher dimensional representations via the embedding layer. Input is therefore an $\boldsymbol{n} \times \boldsymbol{e}$ matrix $\mathbf{E}$, where $\boldsymbol{n}$ is the number of tokens in the input sentence (or maximum sequence length), and $\boldsymbol{e}$ is the dimension of the embedding. The encoder model itself is then typically based on either convolutional or recurrent architectures, or a hybrid of the two (Ribeiro, Ribeiro, and De Matos, 2019). Though, in each case the purpose is the same, to produce a vectorised representation of the input sentence that captures, or encodes, its semantic and communicative intent. Note that, the shape of the output vector representation is highly dependent on the model architecture and parameters, for example, the kernel size and number of filters in convolutional models, or the dimensionality of the hidden units in recurrent models. Regardless of approach, the goal of convolutional and recurrent architectures is the same, though they both consider the encoding problem from a different perspective. Broadly, convolutional models attempt to encode the important features – words or characters within the text – that are indicative of an utterances DA

Table 2.4: Summary of single-sentence neural network DA classification studies.

Study	Model	Features	SwDA	MRDA
		Supervised		
Cerisara, Král, and Lenc (2017)	Max Entropy + LSTM	BoW + Oracle Embedding	69.3%	-
	Max Entropy + LSTM	BoW + Word2vec 2-300 dim	69.1%	-
Papalampidi, Iosif, and Potamianos (2017)	LSTM	GloVe 200 dim	73.8%	-
Shen and Lee (2016)	LSTM	400 dim Embedding	69.7%	-
	LSTM + Attn (Smoothing)	400 dim Embedding	70.4%	-
Lee and Dernoncourt (2016)	LSTM	GloVe 200 dim	66.3%	82.8%
	CNN	GloVe 200 dim	67.0%	**83.2%**
Liu et al. (2017)†	CNN	200 dim Embedding + Speaker	77.1%	-
Ribeiro, Ribeiro, and De Matos (2018)	RCNN	Dependency 200 dim	74.3%	-
		Language Model		
Ribeiro, Ribeiro, and De Matos (2019)‡	ELMo	Dependency 300 dim	**77.9%**	80.7%
	BERT	Dependency 300 dim	79.2%	88.7%
	BERT	Dependency 300 dim + Character	79.3%	88.7%
Bothe et al. (2018b)	mLSTM	ConceptNet 300 dim	74.0%	-

† Does not use standard SwDA training/test split, thus not directly comparable to other SwDA results.
‡ Results reported are for the validation set, thus not directly comparable to other SwDA test set results which are typically several percentage points lower.

label, and recurrent models focus on the sequential, or temporal, relationships between the tokens of the input sequence. Certainly, both paradigms are motivated by the sound reasoning that the constituent words, and their order within the sentence, are both key to interpreting its meaning, and hence both have been extensively explored within the literature. Figure 2.2 illustrates a generalised sentence encoding architecture with an example of parallel convolutional, or recurrent, encoding layers to form the utterance matrix $\mathbf{U}$.

Convolutional approaches typically apply a collection of convolution and pooling layers, either in parallel or sequence, to the sentence embedding matrix $\mathbf{E}$. Parallel encoders apply separate convolutional layers – with varying kernel sizes k – and pooling operations, before combining the pooling outputs, while sequential encoders apply a 'stack' of convolution and pooling layers one after the other to form $\mathbf{U}$. In both cases, the output of a final pooling, or concatenation operation, over $\mathbf{U}$ is then considered the sentences vector representation $\boldsymbol{u}$. Ahmadvand, Choi, and Agichtein (2019), Liu and Lane (2017), Ortega and Vu (2017), Rojas-Barahona et al. (2016), and Kalchbrenner and Blunsom (2013), all use variations of convolutional models as sentence encoders. For example, Ahmadvand, Choi, and Agichtein (2019) employed two different convolutional models, one taking Part of Speech (POS) features and the other word embeddings as input, and Liu et al. (2017) applied two separate convolution and pooling layers, with kernel sizes of 2 and 3, before concatenating the result of the pooling operation to form the sentence vector. Kalchbrenner and Blunsom (2013) instead use a hierarchy of convolutional layers, across the same feature of all word vectors in the sentence, and increased the kernel size from 2 to 4 for each subsequent layer, resulting in a $\boldsymbol{u}$ dimensional sentence vector.

On the other hand, recurrent models process each word vector of the matrix $\mathbf{E}$ in turn. The resulting sentence representation is then either: the final hidden-state of the recurrent layer $\boldsymbol{h}$, where $\boldsymbol{h}$ is the dimensionality of the hidden units (Cerisara, Král, and Lenc, 2017); or the output of each time-step, and therefore an $\boldsymbol{n} \times \boldsymbol{h}$ matrix $\mathbf{U}$ (Papalampidi, Iosif, and Potamianos, 2017; Shen and Lee, 2016). In this latter case, $\mathbf{U}$ typically undergoes some form of dimensionality reduction, such as pooling, to reduce $\mathbf{U}$ to a single vector representation $\boldsymbol{u}$. Li et al. (2019b), Cerisara, Král, and Lenc (2017), Liu and Lane (2017), Papalampidi, Iosif, and Potamianos (2017), Tran, Haffari, and Zukerman (2017), and Shen and Lee (2016), all employed recurrent architectures; either Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997a), or Gated Recurrent Units (GRU) (Cho et al., 2014b).

However, considering previous work, it is not clear if either paradigm produces optimal

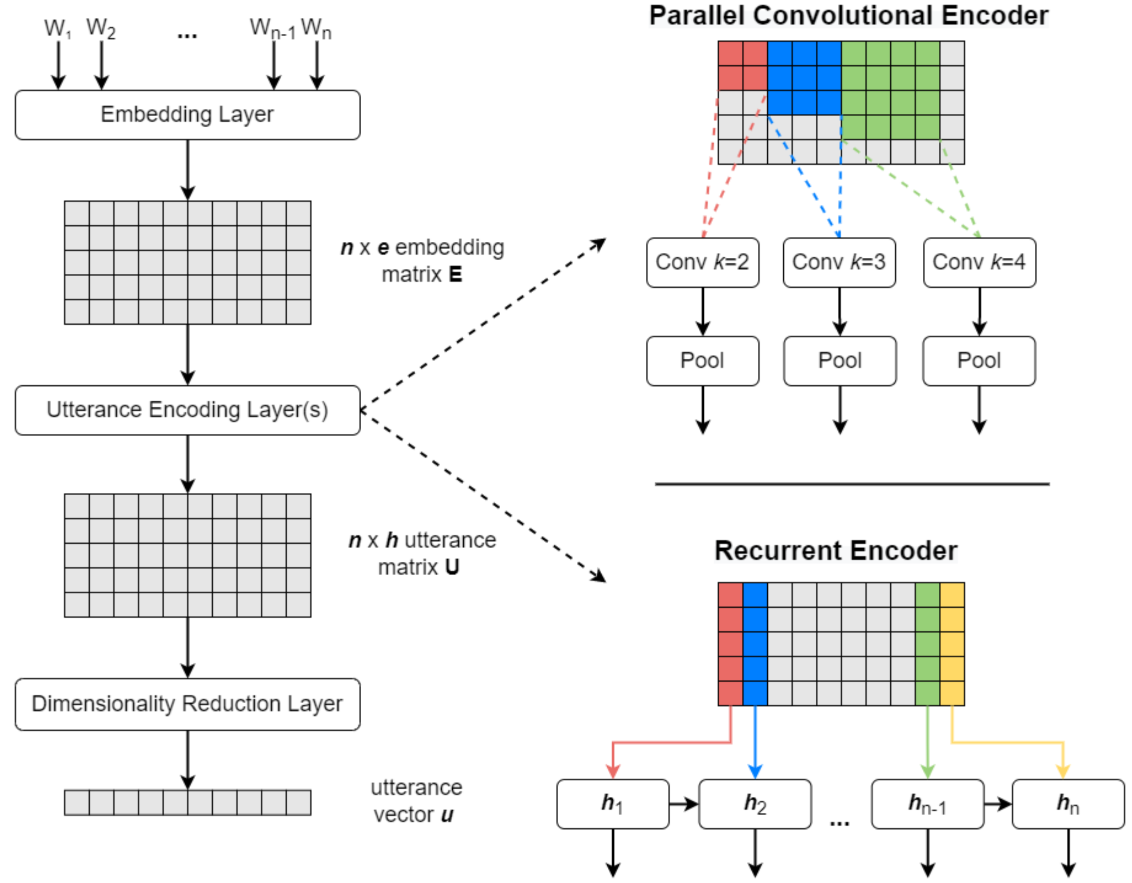


Figure 2.2: A generalised sentence encoder architecture using either, a) parallel convolutional encoder, or b) recurrent encoder layers.

sentence encodings for the DA classification task. For instance, Lee and Dernoncourt (2016) experimented with both convolutional and recurrent sentence encoders on several different corpora and found that neither approach was superior in all cases. While Ribeiro, Ribeiro, and De Matos (2019) tested a Recurrent Convolutional Neural Network (RCNN), based on the work of Lai et al. (2015), and found that it did not result in any improvement over convolutional or recurrent models. Further, several studies have explored variations, or additions, to the convolutional and recurrent architectures described above. Primarily, the use of bi-directional or multi-layer recurrent models (Ribeiro, Ribeiro, and De Matos, 2019; Bothe et al., 2018a; Chen et al., 2018; Kumar et al., 2017). Bi-directional models process the input sequence in the *forwards* and then *backwards* directions, and multi-layer models simply stack multiple recurrent layers on top of each other, with the output for a given layer, at each timestep, becoming the input for the following layer. Numerous studies have also explored the use of different attention mechanisms (Bothe et al., 2018a; Chen et al., 2018; Ortega and Vu, 2017; Tran, Haffari, and Zukerman, 2017; Shen and Lee, 2016). Though different attention mechanisms have been proposed, in various contexts, these are typically based on additive attention (Bahdanau, Cho, and Bengio, 2015), or multiplicative attention (Luong, Pham, and Manning, 2015). Yet, much of this work does not include appropriate ablation studies, thus it is not clear what impact variations of recurrent architecture, or the inclusion of attention mechanisms, may have on DA classification results.

Language Models Though the *joint-training* of LM and classifiers has been successfully applied to DA classification (Liu and Lane, 2017; Ji, Haffari, and Eisenstein, 2016), in both cases the LM is conditioned on contextual information, such as sequences of utterances, speakers or DA labels (see section 2.2.3.4). Thus, due to their ability to produce single sentence encoding, and recent prevalence within NLP, here we only discuss the *fine-tuning* approach. Using this method, a LM is first *pre-trained* on large amounts of unlabelled text data, with a language modelling objective, and then *fine-tuned* for a particular task. In essence, this form of transfer learning is very similar in concept to the use of pre-trained word embeddings, the primary difference being, that the LM itself is used to generate the embeddings and effectively treated as an embedding layer within the classification model. This method of fine-tuning has proved to be highly effective for many NLP tasks and has therefore received considerable attention within the literature. Particularly the *contextual embedding* LM, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and many others (Henderson et al., 2020; Zhang et al., 2020a; Lan et al., 2019; Yang et al., 2019; Cer et al., 2018).

However, despite their success on a wide range of NLP tasks, comparatively few studies have explored language model fine-tuning for DA classification. Both Malhotra et al. (2021) and Tavabi et al. (2021) generate contextual sentence representations with RoBERTa (Liu et al., 2019) for the purpose of DA classification in counselling conversations, and behaviour classification in motivational interviewing, respectively and Żelasko, Pappagari, and Dehak (2021) applied XLNet (Yang et al., 2019) and Longformer (Beltagy, Peters, and Cohan, 2020) to the task of joint segmentation and identification of DA's. In regards to the benchmark corpora we have discussed Bothe et al. (2018a,b), utilised multiplicative long-short-term memory (mLSTM) (Krause et al., 2016a), pre-trained as a character language model on ~80 million Amazon product reviews (Radford, Jozefowicz, and Sutskever, 2017), as a sentence encoder. While Ribeiro, Ribeiro, and De Matos (2019), explored the contextual embedding representations generated by ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019), and He et al. (2021) those of RoBERTa. These studies reported notable results and therefore the fine-tuning of language models seems a promising direction for further research.

2.2.3.4 Context Encoding

The contextual data and discourse model component (4) incorporates additional historical, or future, conversational data into the DA classification process. As previously mentioned, incorporating contextual information, such as surrounding utterances, speaker, or DA labels, has been widely shown to improve performance for the DA classification task (Colombo et al., 2020; Ahmadvand, Choi, and Agichtein, 2019; Ribeiro, Ribeiro, and De Matos, 2019; Bothe et al., 2018b; Liu and Lane, 2017; Papalampidi, Iosif, and Potamianos, 2017; Lee and Dernoncourt, 2016; Kalchbrenner and Blunsom, 2013). The utility of these kinds of contextual information is intuitive if we consider the implication of some future or past action that DA have, and the importance of context in interpreting their meaning, that was discussed in section 2.1.3. Contextual DA classification models typically employ a *hierarchical* architecture which first processes inputs at the utterance level – the sequence of utterances, speakers, or labels – and then at the discourse level, to capture the sequential relationship between the inputs and produce an encoded *dialogue segment*. Within the literature the discourse model is frequently based on recurrent architecture, and often with a bi-directional or attentional component (He et al., 2021; Colombo et al., 2020; Li et al., 2019b; Ribeiro, Ribeiro, and De Matos, 2019; Chen et al., 2018; Kumar et al., 2017; Ortega and Vu, 2017), though convolutional (Liu et al., 2017), and recurrent-convolutional models have also been explored (Ribeiro, Ribeiro, and De Matos, 2018; Kalchbrenner and Blunsom, 2013). A generalised contextual, or hierarchical, DA classification architecture is shown in Figure 2.3. Note that, speaker and label representations can be combined in *sequence* with each utterance encoding, that is, each utterance they are associate with, and prior to input to the discourse context layers (He et al., 2021; Colombo et al., 2020), or appended as a *summary* to the resulting dialogue segment encoding as extra features (Tavabi et al., 2021; Ribeiro, Ribeiro, and De Matos, 2019).

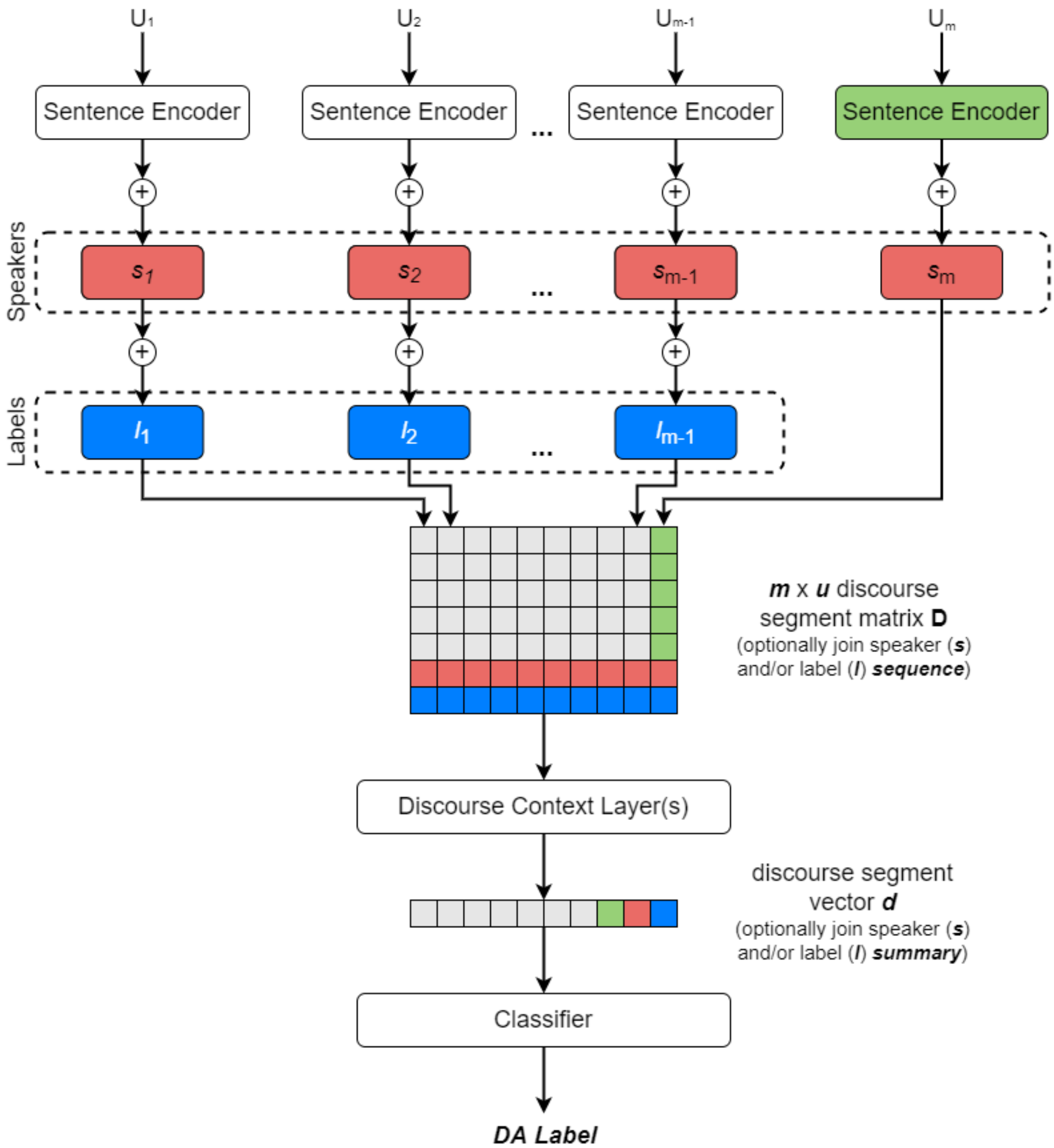


Figure 2.3: A generalised contextual, or hierarchical, DA classification architecture. U_m is the current target utterance for classification. Speaker ($\boldsymbol{s}$) and label ($\boldsymbol{l}$) information can be included as a *sequence* before input to the context encoder, or as a *summary* after context encoding.

Utterance Contextual information is most often included in the form of several utterances that surround the current classification target utterance. The predominant approach within the literature is to include context utterances that only occur prior to the current utterance, that is, they represent historical dialogue information. In most cases it has been shown that between 2 and 4 context utterances yield the best performance (He et al., 2021; Ahmadvand, Choi, and Agichtein, 2019; Ortega et al., 2019; Bothe et al., 2018b; Ortega and Vu, 2017; Papalampidi, Iosif, and Potamianos, 2017; Lee and Dernoncourt, 2016). However, several studies have also experimented with including future dialogue utterances (Ribeiro, Ribeiro, and De Matos, 2019), or even the entire conversation as input (Chen et al., 2018; Kumar et al., 2017). These latter approaches, while shown to marginally improve results, are of course only applicable in situations where the complete dialogue is available and are therefore not appropriate for dialogue systems or similar applications.

Crucially, each utterance U is encoded using the same sentence encoding model (Colombo et al., 2020; Ahmadvand, Choi, and Agichtein, 2019; Ribeiro, Ribeiro, and De Matos, 2019; Bothe et al., 2018b; Liu and Lane, 2017; Papalampidi, Iosif, and Potamianos, 2017; Lee and Dernoncourt, 2016; Rojas-Barahona et al., 2016). If the input dialogue segment contains m utterances, each is first encoded with the sentence model to produce an $\boldsymbol{m} \times \boldsymbol{u}$ matrix $\mathbf{D}$, where $\boldsymbol{u}$ is the dimension of the sentence encoder output. $\mathbf{D}$ is then the input to the discourse model and in the case of recurrent architectures each utterance encoding $\boldsymbol{u}$ is the input at each timestep. As with the recurrent sentence encoder described in section 2.2.3.3, the discourse segment vector $\boldsymbol{d}$ is then either: the final hidden-state of the context encoder layer, or the result of dimensionality reduction, such as pooling, over the output at each time-step. $\boldsymbol{d}$ can then be considered a representation of all utterances (1 to m) in the discourse segment, including the current classification target and surrounding context.

Speakers In addition to surrounding utterances, the use of *speaker* contextual data has also been investigated. For example, conditioning model parameters on a particular speaker (Shang et al., 2020; Kalchbrenner and Blunsom, 2013), concatenating change-in-speaker information to sequence representations (Liu and Lane, 2017; Li and Wu, 2016), or a summary of all previous speaker turns (Ribeiro, Ribeiro, and De Matos, 2019). Intuitively including speaker information can provide clues to the intention of a given utterance and has been shown to produce small improvements in most cases. Though as demonstrated by Shang et al. (2020), this may vary across different DA labels, as some utterances contain unambiguous lexical cues that do not benefit from speaker information.

Typically speaker information is is represented as either: a sequence of binary speaker-change or turn-taking flags, which indicates if the current utterance is produced by the same (0) or a different speaker (1) (He et al., 2021; Colombo et al., 2020; Ribeiro, Ribeiro, and De Matos, 2019, 2018; Liu and Lane, 2017; Li and Wu, 2016); or as one-hot encoded speaker identifiers, for example speaker A as [1, 0] and B as [0, 1] (Bothe et al., 2018b). In addition to the different speaker input representations, as depicted in Figure 2.3 there are several different approaches to incorporating this information into the model. The first approach aims to preserve the *sequential* relationship by joining the speaker and utterance representations, s and $\boldsymbol{u}$ respectively, *before* the sequence of utterances is processed by the discourse context model. For example: Colombo et al. (2020) employ a 'persona' GRU layer to encode the sequence of speakers into higher dimensional representations before concatenating each with $\boldsymbol{u}$; He et al. (2021) simply sum the binary sequence of speakers with $\boldsymbol{u}$; and Bothe et al. (2018b) concatenate their one-hot representations with $\boldsymbol{u}$. The second approach is to instead produce a *summary* of the sequence of speakers – either as a simple flattened sequence, or product of a pooling or recurrent layer – and later join with the discourse segment representation $\boldsymbol{d}$ as extra features prior to classification (Ribeiro, Ribeiro, and De Matos, 2019).

Labels In a similar manner to speakers, surrounding utterances DA label information may also be included, using either predicted, or 'gold standard' DA labels (Ahmadvand, Choi, and Agichtein, 2019; Li et al., 2019a; Ribeiro, Ribeiro, and De Matos, 2019; Liu and Lane, 2017; Tran, Zukerman, and Haffari, 2017; Kalchbrenner and Blunsom, 2013). Again, the benefit of including such information is apparent when considering the implication of some future or past action that DA have. Though, comparatively few studies have explored the inclusion of contextual label information, and fewer still have reported the affect that such information has. Nevertheless, Ribeiro, Ribeiro, and De Matos (2019) demonstrated that contextual label information does indeed improve performance on the SwDA corpus, and further, that using gold standard labels from the corpus resulted in a 1.38% improvement over predicted labels. Label information may be incorporated in much the same way as speaker, as a sequence of predicted labels, or actual labels from the corpus. With the exception that, the label (or prediction) for the current target utterance for classification cannot be included. Thus, as shown in Figure 2.3, $m-1$ labels can be joined in a sequential manner *before* the sequence of utterances is processed by the discourse context model, or as a summary prior to classification.

2.2.3.5 Classification

The final classification model component (5), produces DA label prediction(s) from the input discourse segment representation $\boldsymbol{d}$, or in the case if single sentence models, $\boldsymbol{u}$. Most frequently this involves one or more FFNN layer(s), where the number of output units is equal to the number of DA labels. Softmax activation produces a probability distribution over all possible labels and the final prediction is considered the label with the highest probability. However, rather than predicting each label independently, recently several studies have re-framed the problem as a sequence labelling task (Li et al., 2019b; Ortega et al., 2019; Raheja and Tetreault, 2019; Chen et al., 2018; Kumar et al., 2017), similar to the HMM of Julia and Iftekharuddin (2008), Surendran and Levow (2006), and Stolcke et al. (2000). These works instead employ a CRF as the final classification layer in order to produce a sequence of classifications for each utterance in the discourse segment (Li et al., 2019b; Ortega et al., 2019; Kumar et al., 2017). In some cases the discourse segment is much larger than the previously discussed approaches, using 100 (Raheja and Tetreault, 2019), or even the entire conversational history (Chen et al., 2018). Similarly, Colombo et al. (2020) apply a sequence to sequence (seq2seq) (Sutskever, Vinyals, and Le, 2014), or encoder-decoder, architecture to the problem. That is, the encoded discourse segment $\boldsymbol{d}$ is used to initialise the hidden state of a further recurrent *decoder* layer. The decoder is then conditioned to generate DA predictions for each utterance in the discourse segment (in this case of length 5). By conditioning the classifier, not just the encoders, on the sequential relationship between utterances, both CRF and seq2seq approaches have shown to improve overall classification accuracy over traditional FFNN layers. Thus DA classification models are likely to benefit from incorporating the sequence classification approach. Table 2.5 provides a summary of several of the models and context features used within the contextual DA classification studies discussed in the following sections.

2.3 Adjacency Pair Identification and Dialogue Structure

In contrast to DA, the automatic identification of AP has received relatively little attention within the literature. Using a corpus of human-human dialogues Boyer et al. (2009a,b) attempted to model the dialogue structure by identifying AP within the domain of intelligent tutoring systems. The corpus was first manually annotated with DA, and AP were found using the χ^2 test for independence over all sequential pairs of DA's that occurred

Table 2.5: Summary of contextual neural network DA classification studies.

Study	Model	Features	SwDA	MRDA	Maptask
Ahmadvand, Choi, and Agichtein (2019)†	CNN + Prev State	3 Utts Context	76.7%	-	-
Cerisara, Král, and Lenc (2017)	LSTM + FFNN	3 Utts Context (BoW)	72.8%	-	-
Papalampidi, Iosif, and Potamianos (2017)	LSTM + FFNN	2 Utts Context	75.6%	-	-
Shen and Lee (2016)	LSTM + Attn (Smoothing)	3 Utts Context	72.6%	-	-
Tran, Haffari, and Zukerman (2017)	Gated Attn RNN + HMM	1 Utts Context	74.2%	-	65.9%
Tran, Zukerman, and Haffari (2017)	Hierarchical Attn RNN	1 Utts Context	74.5%	-	63.3%
Lee and Dernoncourt (2016)	LSTM + FFNN	2 Utts Context	69.5%	84.1%	-
	CNN + FFNN	2 Utts Context	73.1%	84.6%	-
Liu et al. (2017)†	CNN + Bi-LSTM	3 Utts Context	76.9%	-	-
	CNN + CNN	3 Utts Context	77.2%	-	-
Ortega and Vu (2017)	CNN + LSTM + Attn	2/3 Utts Context	73.8%	84.1%	-
	CNN + Attn + LSTM	2/3 Utts Context	73.3%	84.3%	-
Kalchbrenner and Blunsom (2013)	RCNN	2 Speaker + DA Label	73.9%	-	-
Ribeiro, Ribeiro, and De Matos (2018)	RCNN	4 Utts Context	78.8%	-	-
		4 Utts + Speaker Turn + DA Label	79.4%	-	-
Wan et al. (2018)†	DMN	All Utts Context	81.5%	-	**68.5%**
Chen et al. (2018)	Bi-Gru + Bi-Gru + Attn + CRF	All Utts Context	81.3%	**91.7%**	-
Kumar et al. (2017)	Bi-LSTM + Bi-LSTM and CRF	All Utts Context	79.2%	90.9%	-
Raheja and Tetreault (2019)	Bi-Gru + Bi-Gru + Self Attn + CRF	All Utts Context	**82.9%**	91.1%	-
Li et al. (2019b)	Bi-Gru + Bi-Gru + Dual Attn + CRF	All Utts Context + Topic	78.3%	**91.7%**	-
Colombo et al. (2020)+	Seq2Seq	5 Utts Context	**85.0%**	91.6%	-
		Language Models			
Ribeiro, Ribeiro, and De Matos (2019)‡	BERT + GRU	3 Utts Context (Summary)	82.5%	89.2%	-
		All Utts Context (Summary)	82.8%	89.2%	-
		Prev DA Pred + Speaker Turn	83.2%	89.3%	-
		Prev + Future DA Pred + Speaker Turn	84.4%	89.6%	-
Bothe et al. (2018b)	mLSTM + RNN	3 Utts Context	77.3%	-	-
		3 Utts Context + Speaker ID	76.5%	-	-
He et al. (2021)	RoBERTA + Bi-GRU	4 Utts Context	82.4%	90.7%	-
		4 Utts Context + Speaker Encoding	**83.2%**	**91.4%**	-
Ji, Haffari, and Eisenstein (2016)	DrLM	Document Context	77.0%	-	-

† Does not use standard SwDA training/test split, thus not directly comparable to other SwDA results.
‡ Results reported are for the validation set, thus not directly comparable to other SwDA test set results which are typically several percentage points lower.
+ Reported validation set accuracy is the same as the test set, so it is not clear which result is reported here.

within the corpus. That is, pairs of DA that showed statistical significance ($p < 0.01$) were joined to form AP. Beginning with a set of 8 DA they were able to identify 23 AP using this method, though it should be noted that these AP are of basic two-turn AP, and therefore lack the richer representations afforded by sequence expansions. Two separate HMM were then trained on the sequence of DAs and sequence of identified AP. They found that the AP model achieved an average log-likelihood fit on the training data that was 5.8% better than the same measure achieved by the DA model, despite the input sequences containing more than twice the number of unique symbols. For the purposes of tutorial dialogue management systems they concluded that the AP model was more preferable, because its structure lends itself more readily to interpretation as a set of dialogue modes that encompass more than one dialogue move (for example, Tutor Evaluation, Student Question/Answer). This model was used in later work (Boyer et al., 2010b) in order to select appropriate responses given the dialogue context for a tutorial dialogue system. Using a similar approach Midgley, Harrison, and Macnish (2006) were also able to identify AP within the Verbmobil-2 corpus (Alexandersson et al., 1997). First the dialogues were segmented, where everything from the last utterance of one speaker's turn to the last-but-one utterance of the next speaker as a segment. Again, basic two-turn AP were identified by applying χ^2 analysis to the dialogue segments to show the most frequent DA pairs. Midgley, Harrison, and Macnish (2006) suggest that these results yield information about what is likely to happen, not just for the next utterance, but somewhere in the next segment. Further, they were able to use the χ^2 values to produce a *Conversational Map*, in the form of a directed acyclic graph shown in Figure 2.4, which depicts the transition probabilities between DA.

Several further works attempted to identify AP within online forum thread data. Exploring dependency parsing and CRF approaches Wang et al. (2011) jointly classified 'inter-post links' (relationships) between posts and the DA of each link to learn the structure of CNET forum threads, and Jamison and Gurevych (2014) used lexical pairs, in the

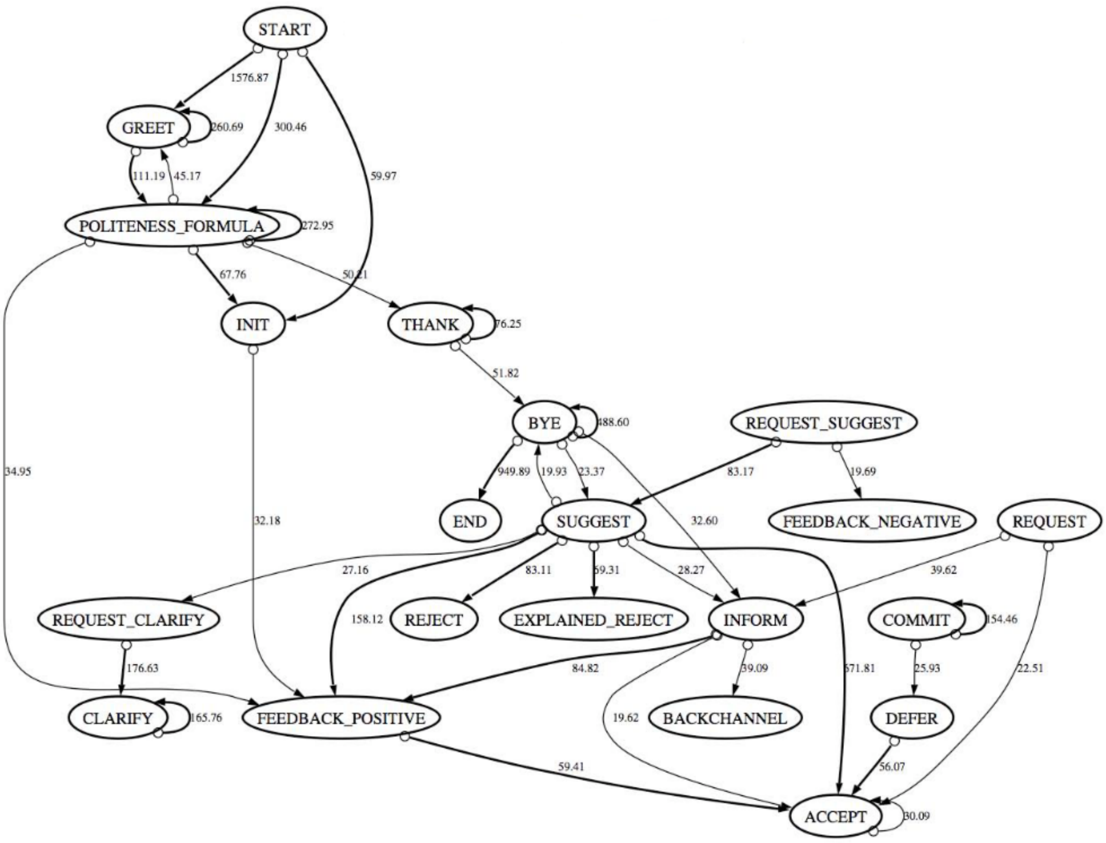


Figure 2.4: A Conversational Map (Midgley, Harrison, and Macnish, 2006), in the form of a directed acyclic graph using the χ^2 data for the 40 highest pairs. For any pair of connected nodes, the first node represents the last utterance in a speaker's turn, and the second could be any utterance in the other speaker's turn.

form of two n-grams, to identify AP within Wikipedia discussions. However, all of the aforementioned work focus only on the basic two-turn AP and thus lack the full descriptive quality that AP sequence expansions allow. In work closely related to this research, Tewari and Suna (2018) proposed the use of Sequence Organisation (AP) as one of several features used to develop a dialogue management system within the domain of social robots for assisted living. Dialogues tagged with topic, DA, sequence organisation, and keywords were organised into a tree structure which could be further used to extract rules for a distributed grammar system. Compared to rule-based, finite state, or data-driven approaches, they suggest that this method has the advantage of describing dialogue as a cooperation among agents, instead of only capturing the machine's perspective, and gaining insight into the structure of dialogues. In later work, Maitreyee (2020) discuss the drawbacks of only two-turn AP that we have previously discussed and highlight the need for flexibility via sequence expansions. To identify AP and sequence expansions within dialogues annotated with DiAML and AP, Maitreyee (2020) first extract several syntactic features in the form of POS tags as unigram and subject-object-verb tuples as tri-grams. Principal Component Analysis was then applied to the syntactic features and DAs before Hierarchical Agglomerative Clustering is used to identify AP. Using this unsupervised method Maitreyee (2020) were able to successfully identify *base*, *insert*, *pre*, and *post*

AP sequences from the corpus. Interestingly, they noted that *"the model doesn't predict the nodes to be in perfect pairs"* and therefore basic two-turn AP *"will be insufficient in extracting knowledge that is not distributed with-in pairs"*. In other words, sequence expansions, and by extension minimal expansions, must be incorporated into any CA model of dialogue. To the best of our knowledge the work of Maitreyee (2020) and Tewari and Suna (2018) is thus far the only work to explore the automatic identification of AP with sequence expansions.

2.3.1 Dialogue Structure

The literature discussed in the previous section related to the identification of AP, or AP-like structures, within dialogues. Here we briefly review several approaches to modelling or ascertaining dialogue structure from different perspectives.

Several earlier works utilise finite-state, graphs, or tree diagrams to represent the relationships, or transitions, between dialogue moves (DA). For example, the finite-state-machine of (Morelli, Brozino, and Goethe, 1991), the hierarchical plan-tree described by Alexandersson et al. (1997), or parse-tree-like structures that encapsulated the task, subtask, and DA sequences within task-oriented dialogues Bangalore, Di Fabbrizio, and Stent, 2008. HMM have also been used to identify the latent structure of task-oriented dialogues (Zhai and Williams, 2014; Chotimongkol, 2008), and in determining the relationship between dialogue structure and tutoring effectiveness (Boyer et al., 2011).

Though it is difficult to formally evaluate the performance of dialogue structure induction algorithms (Qiu et al., 2020), as with DA classification, much of these earlier approaches have now been superseded by supervised and semi-supervised ML methods. Several related works propose an unsupervised approach to dialogue structure learning. Qiu et al. (2020) and Shi, Zhao, and Yu (2019) employ a Variational RNN (the former with an additional structured attention component) to learn latent states and transitions by jointly re-generating training dialogues in several domains, including restaurant booking (Henderson, Thomson, and Williams, 2014), and simulated dialogues (Zhao and Eskenazi, 2018). In both cases Qiu et al. (2020) and Shi, Zhao, and Yu (2019) were able to re-create 'an utterance dependency tree' of different dialogue types, qualitatively evaluating their similarity to the ground truth dialogue structure, and quantitatively showing improved performance over previous HMM methods, such as that of (Zhai and Williams, 2014). Several supervised approaches have focused, not on the identification or generation of the overall dialogue structure, but instead on modelling the relation between utterances or discourse segments. For example, Li et al. (2021) who propose a 'DialoFlow' model which attempts to model the context and information flow across dialogue utterances by keeping track of the semantic influence brought about by each utterance. While Son and Schwartz (2021), using a hierarchical Bi-directional RNN similar to those discussed in 2.2.3.4, to generate Discourse Relation Embeddings that represent the relation between discourse segments in social media data.

One particular work of note for this research – in that it relates to schema which represent dialogue structure – is the 'Hierarchical Schema of Linked DA' proposed by Pareti and Lando (2019). The schema models dialogue structure as linked units of intent (DA) that take place within minimal spans of text, or *functional segments*. The DA labels are derived from DiAML and the schema is intended to, i) be independent from semantic or domain specific representations, ii) support any kind of dialogue relevant to the development of dialogue systems, regardless of the number, type, and role of the participants or the domain or topic discussed, and iii) supports extensible granularity through a set of coarse and fine grained tags hierarchically organised. Crucially, within the schema, linked DA need not necessarily be connected to the immediately preceding utterances. These features, particularly the latter, are closely related to the concept of AP-types proposed by this research, and is in contrast to much of the prior work discussed in this section which only considers relations between immediately adjacent utterances. However, while the work of Pareti and Lando (2019) outlines the schema and the annotation process of a corpus, it does not empirically establish the benefits of this representation method for downstream dialogue modelling tasks.

2.4 Task-oriented Dialogue Modelling

The following provides a concrete definition of the problem domain, as the areas of linguistics, NLP, and computational modelling to which it is directly associated. We make the distinction between task-oriented and non-task-oriented dialogue, and define the purpose of computational dialogue modelling within the context of this research.

2.4.1 Task-oriented Dialogue

There are many different terms for the process of spoken human interaction. The terms 'discourse', 'conversation', 'talk', and 'dialogue', are used in various disciplines, and thus far we have used these words interchangeably. Yet it is not the case that they always refer to three separate areas (Weigand, 1994). For example, there is no widely accepted definition of conversation as a speech event, and within the literature a range of discourse types are referred to as conversation (Warren, 2006). We will make no attempt to provide a more concrete definition of these terms, but for the purpose of this research it is important to make some distinctions between those that refer to task-oriented and non-task-oriented interactions. According to Warren (2006), a conversation, i) involves at least two participants, ii) who have equal rights in terms of initiating, interrupting and responding, iii) where topic is open-ended, with 'randomness of subject matter', and iv) which is *not* motivated by any clear pragmatic purpose. The final two points here suggest that conversation is *not* goal driven, or task-oriented, but rather, informal communication and social interaction. This definition aligns with that of Thornbury and Slade (2006), who state that, *"conversation is the informal, interactive talk between two or more people, which happens in real time, is spontaneous, has a largely interpersonal function, and in which participants share symmetrical rights"*. A dialogue, on the other hand, can be considered more goal-defined and structure oriented (Weigand, 1994). Thus, we define a task-oriented dialogue as, an interaction in which at least one participant has some predetermined goal, such as asking for directions, and engages in the conversation in order to meet that goal. Once that goal is met, or if it is unsuccessful, the interaction is concluded. In contrast, a non-task-oriented dialogue, a conversation, or general talk, is one in which no participant has a specific predetermined purpose for the interaction other than social communication. Topics may change frequently, and while information may be exchanged it is not in the pursuit of some external predetermined purpose.

While CA, and DA, theory is applicable to any form of interaction, not only task or goal driven, we have chosen to limit the scope of this research to task-oriented dialogues. This is because they are more likely to have a well defined structure, that is not merely linear sequences of utterances, or a collection of question-answer pairs. Rather, they are a structured collection of multiple utterances that can be grouped into segments of dialogue, and this structure mirrors the structure of the task (Grosz, 2018).

2.4.2 Computational Dialogue Modelling

Within NLP the phrase 'dialogue model' is sometimes used synonymously to mean dialogue management, or a dialogue system. Particularly within end-to-end neural network (Wen et al., 2017; Luan, Ji, and Ostendorf, 2016; Vinyals and Quoc, 2015), and statistical dialogue systems (Gašić, 2011), the dialogue (or conversation) model typically refers to the model learned by the algorithm. While it is true that such data-driven systems can be said to have learned a model of dialogue, and may be capable of interpreting input and generating responses, the model itself is entirely derived from the data by the algorithm. These kinds of dialogue model behave like black-boxes, and for example, the same algorithm trained independently with two different sets of data would produce two distinct dialogue models. However, the term has mostly been used to refer to components within a dialogue system that manage the conversational flow aspects of dialogue, organising the coherence, and relevance, of utterances and cooperation on the task (Schlangen, 2005). Dialogue management systems may integrate a dialogue model with other components for, task completion, a database, natural language understanding (NLU), and natural language generation (NLG) (Shum, He, and Li, 2018). From this viewpoint, dialogue modelling may also be considered more akin to the process of building grammars for sentence parsing. Given that there are differences between syntax at the sentence level and the structures found at the discourse level (Kühnlein and Piwek, 2007); this may include representing the regularities and patterns that dialogues exhibit, linguistic features, and communicative actions, or behaviour, at the semantic-pragmatic level (Petukhova, 2011). Computational dialogue modelling is therefore, the task of finding the right structural descriptions for these phenomenon and designing a formal system to represent them in a computationally compatible format.

This research is more closely related to the latter definition of dialogue modelling, in that, the concept of an AP-type may be used to describe the semantic and syntactic features that exist within dialogue. However, the purpose of applying the proposed CA model is to facilitate the development of ML algorithms that are able to automatically capture, and analyse, the structure of AP-type labelled dialogues. As such it should not be considered completely independent of the purely data-driven approaches discussed previously.

In order to more clearly define the scope of the proposed dialogue model the following elaborates on several other considerations. Firstly, Bunt (2006), and Petukhova (2011), consider DA in terms of multidimensional actions. That is, dialogue utterances may often be multi-functional, and therefore, a single DA label is not semantically rich enough to convey the meaning of all utterances. For example, "Yes, but what is it?", indicates both an understanding of what was previously said, and a request for more information. While the validity of this viewpoint is clear, it has not yet been widely adopted within the NLP literature. We have therefore chosen to avoid adding unnecessary complexity, and consider DA within our model to have only one 'dimension', or meaning. In the previous example, "Yes, but what is it?", would be considered a question, or a request, depending on the context, as this is the higher priority function of the utterance. Finally, AP and DA are, of course, not a phenomena exclusive to the English language, and are also not limited to only two-party dialogues. The proposed approach could therefore be applied to data from multiple languages, or multi-party dialogues. However, although multi-party DA labelled corpora do exist (Shriberg et al., 2004), and studies have shown the same techniques may be applied to identify DA in multiple different languages (Cerisara, Král, and Lenc, 2017), we have chosen to limit the scope of this research to two-party English dialogues.

Chapter 3

Methodology

In this chapter we provide an overview of our experimental design and methodology implemented to evaluate our objectives and hypotheses. Having proposed our novel approach to task-oriented dialogue modelling and reviewing the relevant literature, we briefly discuss methodological considerations with respect to the objectives outlined in section 1.2.3.

The initial stage of our methodology will be to develop an annotation schema, and accompanying guidelines, which defines two sets of labels, DA and AP, which combine to form AP-type labels (**O1**). Both sets of labels should be derived from, or informed by, the relevant theories from DA and CA literature, thus maintaining consistency with these well established linguistic disciplines. We evaluate the schema, to determine whether it can be reliably understood and applied, by means of an annotation study which is assessed via inter-annotator agreement (IAA) scores and various other measures. Following this, a suitable task-oriented corpus of dialogues can be selected and annotated with the CA annotation schema, which will enable us to overcome the issues of sparsely labelled AP, and limited data availability discussed previously (**O2**).

The corpus, annotated with AP-types, will facilitate the second stage of our methodology; developing an ML algorithm that is capable of automatic classification of utterances with AP-types, thus creating a CA dialogue model (**O3** and **O4**). This can initially be thought of as a short-text, or utterance classification task. That is, given a sequence of utterances, an ML algorithm that is capable of correctly identifying appropriate AP-types can be considered a model of dialogue. Additionally, because we are not rigidly defining a set of AP-type labels, but rather, allowing flexibility through two distinct sets of DA and AP labels, this process can initially be decomposed into separate DA and AP classification tasks. Though DA classification is well established within NLP (2.2), the identification of AP has received comparatively little attention, and thus far no modern ML, or DL, approaches have been applied to the task (2.3). This research, therefore, aims to extend the existing work on DA classification, and develop an ML algorithm that is capable of multi-label (DA and AP) classification over sequences of dialogue utterances.

In the final stage of our methodology the resulting dialogue model, and structure, may then be evaluated on further dialogue modelling tasks (**O5**). First we assess our model's ability to predict future utterance AP-types, given the current dialogue state. Next, we evaluate the effect of our proposed dialogue representations for system response generation using a state-of-the-art generative language model, GPT-3 (Brown et al., 2020). Finally, we evaluate the representation of dialogue structure facilitated by CAMS via dialogue graph representations produced with χ^2 analysis of the gold-standard annotations within CAMS-KVRET.

In summary, our methodology comprises of three major stages; schema development and data preparation, DA and AP classification, and evaluation of the dialogue structure. We examine each of these in the following sections.

3.1 Schema Development, Data Collection, and Preparation

3.1.1 Conversation Analysis Modelling Schema

To the best of our knowledge, and unlike DA, within the CA literature there is no specifically defined, or generally accepted, set of type-related AP labels. Indeed, this flexibility is one of the attractive properties for our approach, in that it facilitates the combination of AP with a much more extensively defined set of DA labels. Additionally, while CA annotated dialogues are available, they typically feature several properties that make them unsuitable for our purposes. Firstly, AP are not the only form of annotation a CA practitioner may wish to assign to dialogue. For example, symbols for intonation, emphasis, pace of speech, pauses, and non-verbal sounds (Schegloff, 2007, p. 265), may all be added, sometimes within a word, which makes pre-processing text for the purpose of ML much more difficult. Secondly, pre-processing issues aside, the extensive and costly process of annotation (Sidnell and Stivers, 2013, Chapter 4), at least in part, results in only a limited quantity of CA annotated dialogues being available, and many of these are not task-oriented. This is problematic when developing an automated ML-based annotation algorithm, and particularly for DL approaches, as these benefit from larger datasets. Thirdly, and most importantly, the standard annotation process for CA does not strictly require each utterance of dialogue to be labelled with an AP, or indeed any at all. Rather, within CA, the AP is viewed as a tool for the analyst, and may or may not be explicitly included in the annotations (Sidnell, 2010; Liddicoat, 2007). Of course, within the context of CA, a practitioner may want to identify specific utterances, or pairs of utterances, with AP, and any 'gaps' created by unlabelled utterances may be inconsequential. However, for the purpose of computational modelling, we require *all* utterances to be labelled with both DA and AP, to create AP-types. Any information-gaps would simply result in an incomplete model of dialogue, regardless of how inconsequential a particular unlabelled utterance may be.

With these aspects in mind we will develop our novel Conversation Analysis Modelling Schema (CAMS) (**O1**). CAMS is intended to combine concepts of DA and AP into a single annotation scheme that is able to capture the semantic and syntactic structure of a dialogue at the *inter* and *intra* utterance level, for the purpose of computational task-oriented dialogue modelling. The schema defines two sets of labels, DA and AP, which are combined to form AP-type labels. When applying the schema, the intent is to assign each utterance of a dialogue one DA and one AP label, which together are considered the AP-type label for that utterance. The AP-type labels, for a fully annotated dialogue, can then be viewed as a representation of its semantic and syntactic structure. CAMS, therefore, is an attempt to define these concepts, and how they may be applied, into a computationally compatible format where each utterance is labelled with an AP-type.

We have elected to use the DiAML (British Standards Institution, 2012; Bunt et al., 2012) as our base set of DA labels within CAMS. As discussed previously (2.1.2.1), this allows us to maintain compatibility with existing DiAML annotated dialogues, support the standardisation of DA labelling schema, and avoid creating our own bespoke set of DA labels, including the accompanying issues surrounding the large number of DA taxonomies. The AP component of CAMS will follow standard CA definitions for base pair and sequence expansion AP types. However, within CAMS we will extend the concept of minimal post-expansions, discussed in section 2.1.1.1, to apply to *all* sequence expansion types. That is, in addition to the minimal single utterance post-expansions, or 'sequence-closing thirds' (Schegloff, 2007), we will allow single utterance pre-expansions, and insert-expansions also. Thus CAMS can facilitate annotation of each utterance of dialogue and avoid the issues discussed above.

3.1.2 Schema Evaluation

With CAMS definitions and annotation guidelines in place we must evaluate whether the schema can be reliably interpreted and applied by human annotators, and whether the annotation process produces a set of meaningfully labelled dialogues (**O1**). For this purpose IAA measures can be used as a means of assessing the *reproducibility* of a coding scheme or determining the *reliability* of a produced 'gold-standard' labelled dataset. That is, determining if the schema is inherently learnable, that the labels applied to utterances are not entirely dependent on the biases of an individual annotator, and that there is a common understanding of the meaning of labels and the utterances to which they are applicable (Craggs and Wood, 2005). It should be noted, that reproducibility is a natural prerequisite to demonstrating reliability of a coding scheme. If annotators produce similar results, they likely have a similar understanding of the annotation scheme and guidelines, and that these are able to represent the desired characteristics of the data (Artstein and Poesio, 2008). Within the literature chance-corrected coefficients, which account for the probability that annotators select the same label by chance, such as Cohen's Kappa (Cohen, 1960), or Scott's Pi (Scott, 1955), are the preferable measures of IAA (Craggs and Wood, 2005; Di Eugenio, 2000; Carletta, 1996). However, weighted coefficients, such as Krippendorff's Alpha (Krippendorff, 2004), are more suitable to annotation tasks such as this, which require an element of semantic interpretation. Thus, we evaluate CAMS via an annotation study conducted with novice annotators and using two weighted agreement coefficients to assess IAA.

To evaluate the results of our IAA study we perform hypothesis testing in the form of Two-sided t-tests or Analysis of Variance (ANOVA), where appropriate. Where the results of an ANOVA reveal a significant overall effect, we perform a further Tukey's Honest Significant Difference (Tukey-HSD) post-hoc analysis, in order to determine the factors contributing to the observed effect. Due to relatively small sample sizes, we calculate the ω^2 effect size and adopt the standard ranges for interpretation, low (.01 - .059), medium (.06 – .139) and large (.14+). For t-tests we report Cohen's d effect size, with standard interpretations of small (.2), medium (.5), and large (.8+). Throughout the analysis, we use a significance level $\alpha = .05$, and, unless otherwise stated, the statistical power is $\geq .8$. In chapter 4.1 we provide more details of our schema and evaluation procedure: including, selection and evaluation of agreement coefficients, the dialogue material annotated, and selection of participants.

3.1.3 Corpora Annotation

In conjunction with the evaluation of CAMS we will annotate a suitable task-oriented corpus of dialogues (**02**). This corpus will ultimately facilitate the development of an ML algorithm that is capable of automatic classification of utterances with AP-types. While there are a large number of DA annotated corpora, comparatively few also include AP annotations. To the best of our knowledge, only the Augmented Multi-party Interaction (AMI) corpus (Mccowan et al., 2005), and MRDA corpus (Shriberg et al., 2004), have been annotated with AP and DA. However, both of these corpora only identify base AP – denoted 'a' and 'b' and numbered in order of occurrence – and therefore do not include the crucial concepts of sequence expansion. Further, because not all utterances belong to a base pair, neither of these two corpora contain AP annotations for *all* utterances. Indeed, the AP annotations are quite sparsely distributed within the data, making it unfit for our purposes. Thus, we have instead elected to annotate an existing corpus, the Key-Value Retrieval Networks for Task-Oriented Dialogue (KVRET) corpus (Eric and Manning, 2017b). KVRET was developed as a multi-turn, multi-domain dataset which contains 3,031 dialogues in three distinct domains appropriate for an in-car assistant: calendar scheduling, weather information retrieval, and point-of-interest navigation. For our purposes KVRET is therefore, i) task-oriented, with three distinct tasks, ii) large enough to meaningfully train a ML model to identify the annotated AP-types, yet small enough to annotate within a reasonable amount of time, and iii) contains slot and task information for developing task-oriented dialogue systems, and hence is applicable to future applications of this work. We annotate KVRET with a specially developed tool to produce CAMS-KVRET, which extends the KVRET format to include the DA and AP defined within CAMS. Further details of the annotation procedure and resulting corpus are presented in section 4.3.

3.1.4 Other Dialogue Act Annotated Corpora

In addition to the development of CAMS-KVRET we make use of existing DA annotated corpora to aid in the development of our ML dialogue classification models. If we consider that AP are alternative, or additional, labels for utterances of dialogue, then we can frame the identification of AP in the same manner as DA classification. Thus, using large and well studied DA annotated corpora provides performance baselines for comparison when developing our classification models. Further, it allows us to begin developing said models immediately, without being hindered by the lengthy task of corpus annotation. We use two additional corpora throughout our DA classification experiments, the Switchboard (SwDA) and Maptask corpus, discussed in Section 2.2.1. These corpora were selected primarily due to several contrasting features between them, which allows for some interesting comparisons between two quite different datasets. Firstly, SwDA contains many more utterances and has a larger vocabulary than Maptask. Secondly, the conversations within SwDA can also be considered open-domain, or non-task-oriented, while maptask is task-oriented, and therefore the type of language used, and problem domain, is contrasted between the two.

3.2 Dialogue Classification Systems

For our second stage – developing an ML algorithm that is capable of automatic classification of utterances with AP-types (**O3** and **Q2**) – we follow the same general architecture and procedure outlined in 2.2.3, and illustrated in Figure 2.1. That is, (1) input sequence representations and (2) embedding, (3) sentence encoding, (4) context encoding, and (5) classification. We group these components into three phases of experiments. The first phase, *sentence encoding*, encompasses components 1, 2, and 3. Here we focus on the various text pre-processing considerations and single-sentence classification models, without any additional contextual information, in order to explore the factors that contribute to creating high quality sentence encodings for the purpose of DA classification. Our intuition is that sentence encodings, with high information content, are a vital prerequisite for contextual models, where the primary form of input is segments of dialogue. Thus, the second phase will be *context encoding*, which extends the sentence encoding models with components 4 and 5. Here we explore different forms of contextual input (speakers, labels, and utterances), multi-sentence architectures, and sequence classification models. Once this contextual classification model is in place, for the third phase we apply it to AP identification, and further extend the architecture into multi-label classification for both DA and AP, and thus AP-types (**O4** and **Q2**).

3.2.1 Phase 1: Sentence Encoding

For our *sentence encoding* phase of experiments the primary focus is exploring the impact of different text pre-processing parameters (component 1), and embeddings (component 2), on the quality of single-sentence encodings – with regard to classification performance

– produced by various sentence encoder models (component 3). Throughout the sentence encoding experiments we use the SwDA and Maptask corpora.

Each input example consists of a single sentence, which is tokenised to form an input sequence of the maximum sequence length ($\boldsymbol{n}$). As discussed previously (2.2.3.1), there are several important, and yet within the literature often under-reported, considerations with regards to input sequence processing for DA classification. Thus, we conduct a set of *input sequence* representation experiments which explore several aspects of the sequence processing component: letter case and punctuation, vocabulary size, tokenisation and sequence length. For letter case and punctuation we investigate the impact of converting all letters in the input sequence to lower-case, or not, and similarly for punctuation keeping or removing all punctuation marks. We also explore varying vocabulary sizes, the number of most frequently occurring words to keep within the input sequence, and sequence lengths, the maximum number of tokens per sequence.

In a series of *word embedding* experiments we also examine the impact of various different pre-trained embedding types; Word2Vec (Mikolov et al., 2013), trained on 100 billion words of Google News data, GloVe (Pennington, Socher, and Manning, 2014), trained on 840 billion tokens of the Common Crawl dataset, FastText (Joulin et al., 2017), Dependency (Levy and Goldberg, 2014), which were both trained on Wikipedia data, and Numberbatch (Speer, Chin, and Havasi, 2016), which combines data from ConceptNet, Word2vec, GloVe, and OpenSubtitles. We test each these with different dimensions ($\boldsymbol{e}$) in the range [100, 300].

We evaluate all of our input sequence and word embedding experiments with respect to six supervised encoder models. Further, we assess the impact of several architectural variations of our supervised encoders, such as bi-directional recurrent models, and the addition of attention layers. Finally, we also apply ten pre-trained LMs to the sentence encoding task, in order to compare performance to the supervised approach. Details for each of these models is presented in 5.1.2.

3.2.2 Phase 2: Context Encoding

With a suitable selection of sentence encoding models in place, for our context encoding phase we extend their architecture to incorporate additional contextual information (component 4). In most cases we use the same two-layer FFNN as the final classifier component. However, we additionally explore several alternative approaches to classification that are appropriate for predicting over sequences of inputs, CRF and Seq2seq (component 5). Throughout our context encoding experiments we again use the SwDA and Maptask corpora, and additionally our annotated CAMS-KVRET corpus.

As discussed in 2.2.3.4, context encoding can be considered in terms of three different components: context utterances, speakers, and labels. We begin with the primary focus, the inclusion of *context utterances*, and experiment with varying numbers, and combinations, of historical or future utterances that surround the current target utterance for classification. Therefore, each input example is a dialogue segment of m utterances, the current utterance to be classified and one or more contextual utterances that occur immediately prior, or after, within the dialogue. Each input utterance is individually processed and encoded in the same manner, as determined by our sentence encoding phase (3.2.1), to produce an $\boldsymbol{m} \times \boldsymbol{u}$ matrix $\mathbf{D}$, where $\boldsymbol{u}$ is the dimension of the sentence encoder output. $\mathbf{D}$ is then passed to the dialogue context encoding layers to produce an encoding of the dialogue segment ($\boldsymbol{d}$).

In addition to contextual utterances we also experiment with including combinations of *speaker* and *label* information. That is, for each input utterance in m we also input the given utterances speaker, s, and so as to exclude the current classification targets label, we input $m-1$ labels, l. Both the speaker and label for a given utterance are represented as one-hot vectors. For speakers, this allows a variable number to be represented, rather than simply indicating speaker change with a binary flag. Thus, for each example, speaker inputs consist of a $\boldsymbol{m} \times |\boldsymbol{S}|$ matrix, and label inputs are an $\boldsymbol{m\text{-}1} \times |\boldsymbol{L}|$ matrix. Details for each of these is presented in section 5.2.1, along with an overview of the sequential classifier models.

3.2.3 Phase 3: AP and Multi-label Identification

For the final phase of dialogue classification we apply our context encoding model, developed in phases 1 and 2, to the task of AP classification. We repeat several of the experiments conducted during the context encoding phase, *context utterances*, *speakers*, and *labels*, with the AP and AP-type labels from our annotated CAMS-KVRET corpus. These *single-label* experiments will primarily allow us to determine the efficacy of applying modern DL approaches to this as yet unexplored problem domain. Further, we can establish whether any differences exist between the optimal parameters for DA context information (determined in phase 2) and that of AP, or AP-types. In other words, whether a different number of context utterances, speakers, or labels is optimal for AP prediction.

Finally, we again extend our context encoding model to multi-label classification. That is, an architecture that is capable of simultaneously identifying both the DA and AP of a given utterance, and hence its AP-type. We speculate that some DA may be highly correlated with certain AP, and conversely, some AP may be more often associated with certain DA. Thus we propose three novel architectures, DA First, AP First, and Parallel. Each of these use the dialogue segment vector $\boldsymbol{d}$ produced by the context encoder model, and then apply different arrangements of classifier layers. For example, in the case of DA First, classifier layers first generate predictions for the current DA label and these are concatenated with $\boldsymbol{d}$ for input into the AP classifier layers. We may then evaluate whether predictions for the current DA are beneficial for predicting AP, or vice versa.

3.2.4 Training and Evaluation

We train models for at most 15 epochs, using mini-batches of 32, and training the examples are shuffled before each epoch. Typically models converge within 10 epochs or less, but where no improvement in validation loss is observed for at least 3 epochs we use early stopping to prevent overfitting. Experiments are carried out on an 8 core i7-9800x, with 32GB of RAM, and a Titan RTX GPU with 24GB VRAM.

Generally, DA classification studies evaluate performance using the accuracy metric and so to allow comparison with previous work, we also use accuracy to evaluate our models. In order to account for the effects of random initialisation and non-deterministic nature of the learning algorithms, results reported are the average (μ), and standard deviation (σ), of the accuracy obtained by training and testing the model for 10 runs. However, in order to compare between different approaches or configurations, as appropriate we also report per-label F1 scores, macro averaged or weighted average per-label F1 scores, and top-k accuracy. The latter being the accuracy obtained if the correct label appears in the top-k predictions, and we choose a value of $k = 3$. Results for the validation set are the highest validation accuracy achieved over all epochs. To obtain results on the test set, we first load the model weights from the point at which validation loss was lowest during training, before applying it to the test set. Therefore, results for the test set were obtained using the model that achieved the best performance on the validation set during training. Throughout our experiments, while tuning hyperparameters, or making comparisons between different models, we only consider the mean validation accuracy scores.

3.2.4.1 Significance Testing

Within much of the previous DA classification literature, results reported for different models and parameter combinations often amount to very small differences in performance, usually in the region of 1-2% accuracy or less. Yet, even where results are the average over multiple runs, it is difficult to draw firm conclusions from such small differences. Thus, in order to determine if the reported mean accuracies are indeed significant, or not, we perform additional hypothesis testing. However, it is acknowledged that applying NHST, can be problematic in the context of machine learning problems (Demšar, 2008; Bouckaert, 2003; Dietterich, 1998; Salzberg, 1997), and that the lack of independent sampling when using the same training and test data split may lead to an increased probability of type I errors (Dietterich, 1998). With this in mind, the following outlines our approach to significance testing; based on the recommendations of Demšar (2006), and employing the Bayesian techniques of (Benavoli et al., 2017).

Wherever we make direct comparisons between two, or more, classifiers results, we employ the Bayesian Signed-rank (BSR) test (Benavoli et al., 2017). The BSR test was introduced by Benavoli et al. (2017), specifically to avoid *"the pitfalls of black and white thinking"* that accompany NHST, by analysing the likelihood that observations are significantly different. In essence, the BSR test uses Monte Carlo sampling to generate a large number of samples (50,000) from the posterior distribution – our set of results over 10 runs. These samples can then be used to calculate the probability that, for example, classifier A performs better than B, $P(A > B)$, and conversely whether B performs better than A, $P(B > A)$. Further, with BSR tests we can select what Benavoli et al. (2017) term, a *"region of practical equivalence"*, or *rope*. That is, a region, or value, within which we consider A and B to be equivalent. Though, selecting a value of rope is dependant on the properties of the domain and the practitioners judgement for what constitutes practical equivalence. Thus, for any two classifiers, A and B, we are given $P(A > B)$, $P(A == B)$, and $P(B > A)$, and we are able to make a more nuanced interpretation of results than would be possible with p-values alone. Indeed, BSR tests allow us to answer the question, *"what is the probability that the performance of two classifiers is different (or equal)?"*, unlike a p-value, which simply represents the probability of getting the observed (or larger) differences, assuming that the performance of the classifiers are equal (H_0) (Benavoli et al., 2017). Further, it does not require the same independence, or distribution, assumptions that many NHSTs do, and is therefore an entirely alternative, method of evaluating the differences between two classifiers.

Throughout these experiments we consider a result to be statistically *significant* if $P(A > B)$ or $P(B > A) \geq 0.8$, and statistically *equivalent* if $P(A == B) \geq 0.8$. Given the marginal differences in performances that are often reported within the DA classification literature (for example, shown in table 2.5), we select a value of 0.5% for rope. By reporting and discussing the most relevant probabilities produced by BSR tests, we hope to alleviate some of the concerns surrounding the potential issues of NHST discussed earlier, allow the reader to draw conclusions about the extent of the significance of the result, and in so doing, establish more confidence in our reported conclusions.

3.3 Dialogue Structure Evaluation and Analysis

In our final methodological stage we evaluate, and analyse, our ML model and proposed method of dialogue representation against several dialogue modelling related tasks (**O5** and **Q4**). First we evaluate our models ability to predict the next likely DA, AP, and AP-types given the current dialogue state. Relatedly, we use a generative language model, GPT-3, to produce appropriate responses given a segment of dialogue and next label as a prompt. Together these two tasks can be considered analogous to two key components of dialogue management systems, that is, the dialogue policy and NLG respectively. Dialogue policy, refers to the selection of next system actions based on the current dialogue state, while the NLG component, given the DA generated by the dialogue policy, maps the act to a natural language utterance (Dai et al., 2020; Zhang et al., 2020b; Chen et al., 2017). Finally, we analyse dialogue graph representations produced with χ^2 analysis of the gold-standard annotations within CAMS-KVRET.

3.3.1 Next-Label Prediction

Our *next-label prediction* experiments are intended to approximate dialogue policy learning for a dialogue management system (or at least an important component of it). We utilise the same contextual single-label and multi-label models developed during the latter part of our second stage. However, we alter the training objective such that the current target for prediction is the *next* label (DA, AP, or AP-type) that is likely to occur in the sequence, given the current segment of dialogue. In other words, given a dialogue segment of m utterances, and optionally m speakers and labels, the classification target becomes the $m+1^{th}$ label. Thus, the model is only provided historical information and is not presented with the utterance, or speaker, for which it is making predictions.

Aside from the new training objective we train and evaluate the models in the same manner as our previous ML experiments discussed in section 3.2. Again we report the top-k accuracy because there may not always be a single objectively correct label for any given sequence of dialogue. On the contrary, there may be several valid response types (Feng et al., 2021; Zhang, Ou, and Yu, 2020). Therefore, knowing the 3 most likely response types the model has selected, how often the correct label appears within them, and the magnitude of difference between probabilities of any two potentially valid labels, may yield valuable insights into the application of our approach to dialogue system policy learning.

3.3.2 Response Generation

The *response generation* experiments are intended to evaluate the extent to which our proposed dialogue representation method affects the generation of appropriate natural language system responses in a dialogue management scenario. That is, given a dialogue segment of m utterances, speakers, and labels (the dialogue state), and the $m+1^{th}$ response type (DA, AP, or AP-type) produced by the dialogue policy, the NLG component must generate a suitable response. Given the relatively small size of our CAMS-KVRET dataset we opted to utilise the remarkable NLG abilities of GPT-3, which is specifically intended for few-shot learning (Brown et al., 2020). We fine-tune GPT-3 and the generated responses are then evaluated with a range of well-known language modelling and generation metrics (Yeh, Eskenazi, and Mehri, 2021; Finch and Choi, 2020; Sharma et al., 2017; Liu et al., 2016):

1. *BLEU* (Papineni et al., 2002), and *METEOR* (Banerjee and Lavie, 2005), were developed to automatically evaluate the results of machine translation given some reference sentences. The BLEU metric compares the number of common n-grams, typically 1 to 4, over all the candidate and reference utterances in the corpus. METEOR first calculates the alignment (exact matches, as well as stem, synonym, and paraphrase matches) between the candidate and the reference utterances by mapping each uni-gram in the candidate sentence to 0 or 1 uni-gram in the reference sentence. The METEOR F1 is then computed, which is is the harmonic mean between precision and recall alignment, with the weight for recall 9 times as high as the weight for precision. Similarly, *ROUGE* (Lin, 2004) is a set of metrics proposed for the automatic evaluation of summaries. ROUGE-N calculates the n-gram co-occurrence,

while ROUGE-L is an F-measure based on the longest common sub-sequence between the candidate and reference utterances.

Some have questioned the suitability of these *word-overlap* metrics for evaluating NLG in the context of dialogue systems (Deriu et al., 2020; Liu et al., 2016). Either because they do not correlate well with human evaluation, or due to the larger number of grammatically correct responses that can be generated within dialogue, as opposed to translation. Nevertheless, it has been shown that these metrics have a stronger correlation with human judgements in the task-oriented setting compared to the non-task-oriented setting (Sharma et al., 2017), and may therefore provide some useful metrics for evaluating our task.

2. *Semantic Similarity* of generated sentences with respect to the reference sentences within the corpus. We calculate cosine distance for each sentence pair using the contextual embeddings generated by the RoBERTa-large model (Liu et al., 2019). As suggested by Sharma et al. (2017) and Liu et al. (2016), we also evaluate several different metrics using pre-trained GloVe embeddings:

    - *Embedding average* (Mitchell and Lapata, 2008; Foltz, Kintsch, and Landauer, 1998) is the most common method for computing the meanings of phrases. It creates sentence-level embeddings by averaging the vector representations of their constituent words, which are then compared using cosine distance.
    - *Greedy matching* (Rus and Lintean, 2012) does not compute sentence-level embeddings. Instead, given two sequences r and $\hat{r}$, each token $w \in r$ is 'greedily' matched with a token $\hat{w} \in \hat{r}$ based on the cosine similarity of their word embeddings (greedily, because the maximum similarity is chosen). The total score is then averaged across all words. The intention is to favour responses with key words that are semantically similar to those in the reference sentences.
    - *Vector extrema* (Forgues et al., 2014) takes the most extreme value (high or low) amongst *all* word vectors in the sentence and uses that value to produce a sentence-level embedding. Again, similarity between generated and reference sentences is calculated using cosine distance. This approach is intended to prioritise informative words over common ones, because words that appear in similar contexts will be close together in the vector space.

3. *Perplexity* can be considered a measure of a language models uncertainty when predicting the next word w_{t+1}, given the previous words in the sequence $(w_1, ..., w_t)$. Perplexity is closely linked to the models *entropy* (Brown et al., 1992). That is, the similarity between the distribution the model has learned from the training text Q, and the actual distribution P. The cross-entropy $H(P, Q)$ is then defined as:

$$H(P, Q) = -\sum_{i=1}^{t} P(w_i) log Q(w_i) \tag{3.1}$$

Perplexity is then simply:

$$PPL(P, Q) = 2^{H(P,Q)} \tag{3.2}$$

Perplexity does not inform us of the quality, nor topical relevance, of the generated utterance. However, it *does* provide us with a kind of 'confidence' value for the language model when generating responses. In other words, a the lower perplexity indicates a higher confidence that the correct words are being generated for the current dialogue context.

3.3.3 Analysis of Dialogue Structure

In order to analyse and evaluate the dialogue structure representations facilitated by the different label types within CAMS we adopt a similar technique to that of Midgley, Harrison, and Macnish (2006). Using a modified χ^2 analysis (Schutze, Hull, and Pedersen, 1995), Midgley, Harrison, and Macnish (2006) produced dialogue structure graphs showing the relationship between pairs of utterances within dialogue. Thus, we adopt the same technique using the gold-standard label annotations within the CAMS-KVRET corpus. For our purposes the χ^2 test allows us to determine if the observed frequency of a given label (A) being followed by another (B) can be attributed to random chance, or whether their co-occurrence is statistically significant. We apply the χ^2 analysis in two ways. Firstly, for DA and AP, we calculate χ^2 for all label pairs by creating a contingency table with counts for the number of times an utterance with label A (or not) is followed by an utterance with label B (or not), as in Table 3.1.

Table 3.1: χ^2 contingency table for a pair of labels A and B.

	$U_i = A$	$U_i \neq A$
$U_{i+1} = B$	AB	$\neg AB$
$U_{i+1} \neq B$	$A\neg B$	$\neg A\neg B$

Next, these counts are used in the following equation, where N is the total number of utterances, to produce a χ^2 value for each label pair:

$$\chi^2 = \frac{N(AB \times \neg A\neg B - A\neg B \times \neg AB)}{(AB + A\neg B)(AB + \neg AB)(A\neg B + \neg A\neg B)(\neg AB + \neg A\neg B)} \tag{3.3}$$

The resulting χ^2 values can then be used to create a directed dialogue structure graph for the given label type. Where each node is represented by a label and edges represent the transition from one utterance with that label to another, with the edge *weight* determined by the χ^2 value. To produce AP-type graphs the process is similar to that of AP, except we calculate *separate* χ^2 values based on the frequency that one AP label follows another (or not), *for each DA label*. Therefore, within the AP-type graph, between any pair of nodes (AP) there may be multiple edges, each representing a DA and weighted by its χ^2 value. Note that we calculate the χ^2 critical value (χ^2_{crit}) with a significance level $\alpha = .05$ and only keep edges where $\chi^2 > \chi^2_{crit}$. Thus only statistically significant transitions are represented in the graphs.

The resulting dialogue structure graph can then be evaluated in terms of the similarity between adjacent nodes, or the graphs themselves, and also measuring how efficiently they exchange information:

1. *SimRank Similarity* is a measure of structural-context similarity between nodes, which states that *"two objects are similar if they are related to similar objects"* (Jeh and Widom, 2002). For any pair of nodes a and b in a graph, SimRank similarity is calculated as the average similarity between in-neighbours of a and in-neighbours of b, and produces a score in the range [0, 1], where 0 represents the nodes are completely dissimilar and 1 identical. Thus, if SimRank similarity provides an additional measure of local and structural relationships between pairs of nodes, we may use the scores to identify pairs of labels that frequently occur at similar positions within a sequence of utterances. For example, one might expect *FPP-insert* and *SPP-base* to have a high similarity score because they are both likely to follow (be referenced by) a *FPP-base*.

2. *Similarity Edit Distance* determines the minimum sequence of edit operations on nodes or edges (insertion, deletion, or substitution), in order to transform one graph into another (Abu-Aisheh et al., 2015). For any two graphs $g1$ and $g2$, the algorithm produces an integer value which represents the number of node or edge edit operations that are required to transform $g1$ into $g2$. We can therefore use the similarity edit distance to compare the dialogue structure representations facilitated by the different label types within the schema. In particular, for each label type we can compare the dialogue structure graphs for the different tasks within the KVRET corpus, *weather*, *navigation*, and *scheduling*. Thus we can determine whether the representations are highly similar between the tasks, or not. High similarity would suggest a more generalised representation, while low similarity indicates the representation is task-specific.

3. *Efficiency* is a measure of how efficiently information is exchanged between nodes of a graph (Latora and Marchiori, 2001). The efficiency of two nodes i and j is inversely proportional to the shortest distance (d) between them $\epsilon_{ij} = {}^{1}/_{d_{ij}}$. *Global* efficiency is the average efficiency of all node pairs within the graph, and *local* efficiency is the average efficiency of local sub-graphs – node i and all nodes adjacent to i. Note that, the local efficiency around a node with fewer than two neighbours is taken to be 0, and when there is no path in the graph between i and j, $d_{ij} = +\infty$ and therefore $\epsilon_{ij} = 0$. Thus, we may use these measures of global and local efficiency as an indication of the effectiveness of a given label types dialogue representation at the macro and micro level. In other words, high global efficiency suggests a representation of the entire dialogue where, for example, the interaction can be conducted in fewer steps. On the other hand, high local efficiency is a measure of fault tolerance – the efficiency of communication between the neighbours of i when i is removed.

Chapter 4

Conversation Analysis Modelling Schema and Corpora Annotation

In this chapter we first provide an overview of our proposed CAMS, including the set of labels and brief annotation considerations. Then, in Section 4.2, we evaluate CAMS by means of an annotation study that was conducted with novice annotators, assessed via two IAA coefficients (Alpha and Beta), and several other quantitative and qualitative measures (**O1** and **Q1**). Finally, in Section 4.3, we provide details of the annotation process that was used to develop CAMS-KVRET, a multi-turn, multi-domain dataset annotated with CAMS (**O2**).

4.1 CAMS Definition and Guidelines

The schema defines two sets of labels, DA and AP, which are combined to form AP-type labels. When applying the schema, the intent is to assign each utterance one DA and one AP, which together are considered the AP-type for that utterance, and capture the semantic and syntactic structure of a dialogue at the *inter* and *intra* utterance level. The AP-type labels, for a fully annotated dialogue, can then be viewed as a representation of its semantic and syntactic structure. It should be noted that the concept of a *typed AP* is a key feature of AP present within the CA literature (Clift, 2016; Sidnell, 2010; Liddicoat, 2007; Schegloff, 2007). However, the standard annotation schemes for CA do not strictly require each utterance of dialogue to be labelled with an AP. Additionally, CA annotation often includes non-verbal sounds, pauses and other types of disfluencies. Gaps in annotations, where utterances are not labelled with AP, and other forms of non-verbal annotation, for example 'breathing', are generally undesirable for computational purposes. CAMS, therefore, is an attempt to define these concepts, and how they may be applied, into a computationally compatible format where each utterance is labelled with an AP-type. The following sections provide an overview of DA, AP, and AP-types, their respective sets of labels defined within the schema, and some brief guidelines for applying the schema to dialogue.[1]

4.1.1 Dialogue Acts

As discussed in 3.1.1, we have elected to use DiAML as our base set of DA labels within CAMS. For the initial iteration we adopt a subset of 27 DiAML labels that were most relevant to task-oriented dialogues. Though, this could simply be extended to include the full range of DiAML labels. As shown in table 4.1, they are grouped by their communicative function (though this differs slightly from the original DiAML organisation):

[1] Full label definitions and annotation guidelines are shown in Appendix A.

Table 4.1: The CAMS categories of DA labels, organised by their associated communicative function.

Communicative Function	DA Labels
Information-seeking	setQuestion, choiceQuestion, propositionalQuestion, checkQuestion
Information-providing	answer, inform, correction
Commissive	offer
Directive	suggest, request
Feedback Positive	accept, conditionalAccept, agree, confirm, feedbackPos
Feedback Negative	decline, disagree, disconfirm, feedbackNeg
Time and Communication	stalling, retraction
Social Management	greeting, goodbye, thanking, acceptThanking, apology, acceptApology

Information-seeking, information-providing, commissives, directives, feedback, time management, communication management, and social obligations management. Note that, we have also made several small changes to the conventional DiAML DA names, outlined below. However, we were careful to maintain backwards compatibility, in that, the CAMS labels can simply be expanded back to the original DiAML.

- Within DiAML, the labels *autoPositive* and *autoNegative* represent positive or negative understanding of the previous utterance, for example “Okay”, or “What?”. Within CAMS we have converted these into the slightly more intuitive labels of *feedbackPos* and *feedbackNeg*.
- For simplicity we have also collapsed several groups of labels into a smaller subset. Our reasoning and method is similar to that of Pareti and Lando (2019). That is, because responsive DA are linked via AP within CAMS, it should be unnecessary to specify their type. Specifically, *acceptRequest*, *acceptSuggest*, and *acceptOffer* have been collapsed to simply *accept*; *declineRequest*, *declineSuggest*, and *declineOffer* to *decline*; and *addressRequest*, *addressSuggest*, and *addressOffer* – which normally signify the acceptance of a request, suggestion, or offer possibly depending on certain conditions – to *conditionalAccept*.
- Similarly, *initialGreeting* and *returnGreeting* have been collapsed to *greeting*, and *initialGoodbye* and *returnGoodbye* have been collapsed to *goodbye*

4.1.2 Adjacency Pairs

Within CAMS we utilise the full range of AP discussed in section 2.1.1. That is, base pairs, sequence expansions, and minimal expansions. With the exception of extending minimal expansions (discussed below), we make no alterations to the meaning or application of AP within CAMS, and they are therefore applied in the same manner outlined within the CA literature. In the following we provide a brief overview of each AP category, their respective labels, and annotation guidelines:

Base CAMS defines two base AP, with the characteristics defined by Schegloff and Sacks (1973) and Schegloff (1968). Within CAMS these are denoted *FPP-base* and *SPP-base*.

Once a FPP-base has initiated a sequence, that sequence must be concluded with a SPP-base before any further base-type sequences are initiated. In other words, a base-type sequence may not be 'nested' within another base-type sequence.

Pre-expansions are denoted *FPP-pre* and *SPP-pre*. Pre-expansions must be initiated and concluded prior to any base sequence and are not permitted within base-type sequences. Once a FPP-pre has initiated a sequence, that sequence must be concluded with a SPP-pre before any further sequences. That is, a Pre-expansion sequence may not be 'nested' within another Pre-expansion sequence.

Post-expansions are denoted *FPP-post* and *SPP-post*. As with Pre-expansions, Post-expansions must be initiated and concluded after any base sequence and are not permitted within base-type sequences. Once a FPP-post has initiated a sequence, that sequence must be concluded with a SPP-post before any further sequences, and may not be 'nested' within another Post-expansion sequence.

Insert-expansions are denoted *FPP-insert* and *SPP-insert*. Insert-expansions are the only expansions permitted within a base-type sequence, and they are not permitted outside of a base-type sequence, such as a Pre-expansion. However, unlike base, pre, and post type sequences, Insert-expansions *are* permitted to be 'nested', provided they abide by the usual constraints for AP: different speakers for FPP and SPP, not overlapping, and so on.

Minimal-Expansions Because dialogue does not always contain even numbers of utterances, there are also single-utterance *minimal-expansions*, for utterances that do not belong to conventional AP. CAMS defines three types of minimal-expansion *Pre*, *Post*, and *Insert*, which behave in a similar manner to their expansion counterparts. That is, they must be produced before, after, or inside a base sequence. These are closely related to the idea of minimal post-expansions (Sidnell and Stivers, 2013; Schegloff, 2007), in that they are not designed to project any further sequences of talk, but rather open, close or add to sequences respectively. The primary role is to allow for additional turns that behave as expansions but consist only of one turn, and allows CAMS to account for instances where, for example, a single utterance opens the interaction before the base pair is initiated, or a single utterance takes place between the initiation and conclusion of a base pair (or insert pair). There is no restriction on speaker order for minimal-expansions, which allows the same speaker to produce more than one utterance of different types in succession, or for a speaker to produce one utterance that does not belong to (initiate or conclude) an AP. However, they should abide by their semantic intent. For example, a pre-minimal-expansion should be relating to a future base sequence, a post-minimal-expansion to a previous base sequence and an insert-minimal-expansion within a sequence.

In summary, there are 11 AP in the schema and the set includes: Two labels for the base pair, FPP-base and SPP-base. Six labels for expansion pairs. That is, FPP and SPP for pre, post and insert expansions, as described by Liddicoat (2007) and Sidnell (2010). And three labels for minimal expansions, pre, post, and insert. Within CAMS, for all AP sequence-types: base, pre, insert, and post expansions the following rules apply:

1. The speaker of a FPP must be different to the speaker of a SPP.
2. A FPP must be an initiation of a sequence and the SPP a response to that initiating FPP.

3. Once a FPP has initiated a sequence, that sequence must be concluded with a SPP of the same type.

4. The initiation and conclusion of two different sequences may not overlap each other.

For example, the following sequence is not permitted. Firstly, the base pair is concluded in utterance 3 by the same speaker that initiated it in utterance 1, thus violating 1 above. Secondly, the insert expansion initiated in utterance 2 is not concluded before the base sequence (that it is inserted within) is completed, violating 4 above.

A:	Utterance 1	***FPP-base***
B:	Utterance 2	***FPP-insert***
A:	Utterance 3	***SPP-base***

4.1.3 Adjacency Pair Types

In CAMS an AP-type is simply the product of one AP label, and one DA label, for an utterance of dialogue. The combination of these two labels is considered an AP-type label. Due to the large number of possible combinations, and to allow flexibility, the schema does not explicitly define all valid DA and AP combinations. Instead, annotators should consider the meaning and context within which the individual labels being applied produce AP-types. The following shows a previous example, now fully labelled with both AP and DA, to create AP-types. In the example, *propQ* (propositionalQuestion) is a question that implies, but does not necessitate, a 'yes' or 'no' answer, and a *choiceQ* (choiceQuestion) where the speaker provides a list of alternatives with the assumption that the addressee knows which one is true, or will select one. The alternative question-type labels are: *setQuestion*, which corresponds to what is commonly termed a 'WH-question' in the linguistic literature, that is, questions that typically begin with words such as, 'Who', 'What' or 'How'; and *checkQuestion*, which is produced by the speaker in order to know whether a proposition is true.

A:	Do you know the directions to the zoo?	***FPP-base - propQ***
B:	Are you driving or walking?	***FPP-insert - choiceQ***
A:	Walking.	***SPP-insert - answer***
B:	Get on the subway...	***SPP-base - answer***

4.2 CAMS Evaluation

The following outlines details of the annotation procedure that was conducted to assess CAMS with respect to: (1) the extent to which multiple annotators agree when applying the schema to dialogue, the IAA, (2), its suitability for application to both task-oriented and non-task-oriented (general talk) dialogues, and (3), evaluate additional characteristics of the material, or annotator behaviours, which may affect application of the schema and the resulting agreement scores. These objectives are intended to establish whether CAMS is comprehensively and explicitly defined, such that it can be reliably applied by multiple annotators, and that it is generalisable to any conversation type, topic, or domain, in order to create corpora annotated with labels that express the syntactic and semantic structure (**O1** and **Q1**).

The study participants were asked to label 5 dialogues, containing both task and non-task-oriented conversations, using a specially developed software annotation tool[2] (figure

[2]The annotation tool, an example of dialogue for each corpus, and all data generated by this study is available at: github.com/NathanDuran/CAMS-Dialogue-Annotation

4.1). In total, 15 participants took part in the study (see 4.2.2), and each was assigned one of 5 different sets of dialogue for annotation (see 4.2.1). The dialogue sets were evenly distributed among the participants, resulting in 3 annotators per set. The first dialogue in each set is a practice dialogue, followed by the 4 dialogues in their respective set (2 task-oriented and 2 non-task-oriented). The latter 4 dialogues were shown to participants in a random order to encourage independent annotation, and mitigate any learning effect of the software, or schema, on annotation results. The participants were given one hour to annotate all dialogues and had no previous training using the annotation tool or CAMS. Upon completion of each dialogue, participants were asked to rate, by means of a Likert Scale, how well their annotations fit the data. Timing data was also collected during the annotation process, which recorded how long participants spent annotating each utterance of dialogue. The timing and rating data were used, in addition to the calculated IAA, for further analysis of the manner in which annotators apply the schema, and comparison of task and non-task-oriented dialogues. The following discusses the evaluation measures, and the selection of participants and dialogues in more detail.

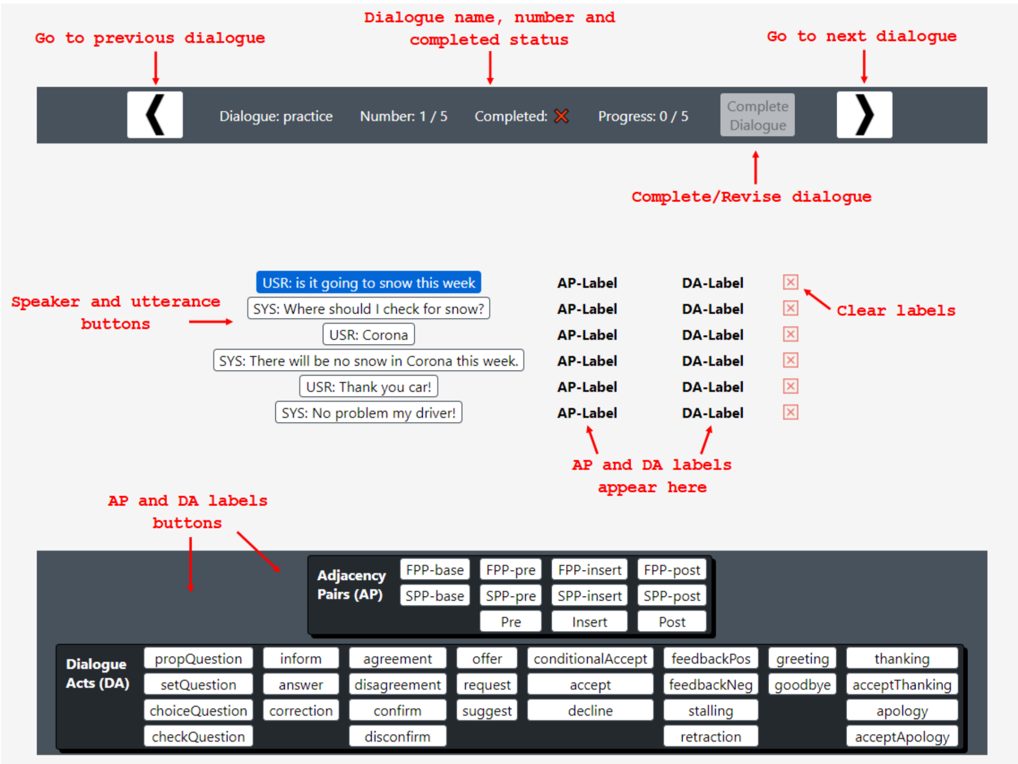


Figure 4.1: Annotation screen of the software annotation tool.

4.2.1 Dialogue Selection

A key objective of this study is to assess CAMS when it is applied in both task-oriented and non-task-oriented settings, as defined in 2.4. The dialogues selected for this study are therefore representative of these two groups, and in order to provide a more representative selection between them, dialogues were chosen from 4 different corpora, with varying numbers of utterances, participants and formats.

In total 20 dialogues were chosen, 5 from each corpus. These were then split into 5 dialogue sets, each containing one dialogue from each corpus, and grouped in order to keep the total number of utterances in each set roughly equivalent. Additionally, each

Table 4.2: Summary of dialogues, and number of utterances, per dialogue set. Total column includes 6 utterances for the practice dialogue.

Set	KVRET	Utts	bAbI	Utts	CABNC	Utts	SCoSE	Utts	Total
1	test 28	7	test 290	7	KB7RE015	9	mammoth	19	48
2	test 52	8	test 428	7	KBKRE03G	6	clone	19	46
3	test 96	4	test 555	5	KDARE00G	4	accident	29	48
4	test 129	6	test 564	5	KE2RE00Y	4	hunter	25	46
5	test 102	4	test 894	5	KBERE00G	5	tipsy	26	46
$\boldsymbol{\mu}$		5.8		5.8		5.6		23.6	46.8

set contained the same short practice dialogue, selected from the KVRET corpus. The practice dialogue is intended to mitigate any learning effect associated with the annotation software, and also provide a control dialogue annotated by each participant regardless of the dialogue set they are assigned. Table 4.2 provides an overview of each dialogue set used within the study. Next is a brief overview of each corpus.

KVRET As discussed in section 3.1.3, for **O2** we aim to annotate the KVRET corpus (Eric and Manning, 2017b) with CAMS. Therefore, we also include KVRET as one of our task-oriented corpora for the annotation study. The dialogues were randomly selected from the 304 dialogues in the KVRET test set.

bAbI The Dialogue bAbI Tasks data is a subset of the bAbI project by the Facebook AI Research group (Weston et al., 2015). The set of 6 tasks are designed to test end-to-end dialogue systems in the restaurant booking domain. The dialogues used for this study were randomly selected from the 100 dialogues in the bAbI task 1 test set. Each dialogue follows a similar format. First greetings are exchanged, and the automated system asks the user what it can help them with. The user states their preference of cuisine, location, price range, and number of diners, and in some cases extra system turns clarify these preferences.

CABNC The Jeffersonian Transcription of the Spoken British National Corpus is a conversation analytic re-transcription of naturalistic conversations from a sub corpus of the British National Corpus (Albert, Ruiter, and Ruiter, 2015). It contains 1436 conversations with a total of 4.2 million words. There is a wide range in the number of utterances within the CABNC dialogues, in many cases hundreds or thousands of utterances. In order to, as much as possible, maintain a similar number of utterances across all dialogues and dialogue sets, and due to time constraints, those used for this study were randomly selected from dialogues with less than 10 utterances.

SCoSE The Saarbrucken Corpus of Spoken English consists of 14 transcribed dialogues of general talk on a range of topics between two or more participants (Norrick, 2004). As with the CABNC corpus, due to the large number of utterances, and time constraints, those chosen for this study were the 5 dialogues with the fewest utterances. In our set, the *mammoth*, *clone*, and *accident* dialogues take place between up to three undergraduate students sharing an apartment, while *hunter*, and *tipsy* take place between Helen and her three adult daughters before a late-afternoon Thanksgiving dinner.

4.2.2 Participant Selection

The study participants comprised of 15 undergraduate students from the 1st year of an English Language and Linguistics course. For 5 weeks prior to the study the participants received instruction on CA and AP as part of their linguistics syllabus. However, we also wanted to assess how intuitive the schema is to apply with only minimal prior knowledge. Given its purpose is for computational dialogue modelling, CAMS should ideally be usable by as wide a range of people as possible. For example, Conversation Analysts, Computer Scientists, Computational Linguists, and other NLP practitioners, who either already have some familiarity with CA and AP, or who simply intend to follow the annotation guidelines and label definitions. This is particularly important when considering the application of the schema for further annotation tasks, such as creating large datasets for training and evaluating deep-learning NLP models. Therefore, our participants were not provided any specific instruction regarding CAMS and did not receive any training in its application. As such, participants could reasonably be considered novice annotators, in that, they had some prior knowledge of CA theory but no previous experience in annotation or applying CAMS. The selection of Linguistics students as annotators was largely for pragmatic reasons:

1. While DA labels could be considered somewhat intuitive, even for novice annotators, AP require some level of previous CA knowledge. Therefore, conducting a large-scale crowed-sourced annotation experiment, where we cannot guarantee any prior understanding of CA concepts, would be inappropriate.

2. Even though expert annotators are more likely to produce high agreement (Nowak and Rüger, 2010; Geertzen, Petukhova, and Bunt, 2008; Snow et al., 2008), the number of available expert annotators is limited. Further, both Krippendorff (2004), and Carletta (1996), argue that, for discourse and dialogue annotation schemes there are no real experts, and that what counts is how totally naïve annotators manage based on written instructions. While using naïve annotators is not appropriate here, the use of non-expert annotators should still provide some insight into the clarity of the CAMS label definitions and annotation guidelines.

3. Bayerl and Paul (2011), suggest using annotators with the same level of domain expertise. Using participants from the same student cohort, with a similar level of experience, should therefore reduce external factors which may influence the interpretation of the schema definitions and guidelines.

4.2.3 Inter-annotator Agreement

IAA measures can be used as a means of assessing the *reproducibility* of a coding scheme or determining the *reliability* of a produced 'gold standard' labelled dataset. Given that the focus of this study is the labelling schema itself, the purpose of measuring IAA refers to the former. That is, determining if the schema is inherently learnable, that the labels applied to utterances are not entirely dependent on the biases of an individual annotator, and that there is a common understanding of the meaning of labels and the utterances to which they are applicable (Craggs and Wood, 2005). The following provides a brief overview of our selection of IAA coefficients and evaluation criteria. Appendix B contains full definitions of all IAA coefficients discussed here, further discussion of coefficients selection and evaluation criteria, and formulations of our weighted agreement distance functions for DA, AP, and AP-types.

4.2.3.1 Coefficient Selection

Within the literature chance-corrected coefficients, which account for the probability that annotators select the same label by chance, such as Cohen's Kappa (Cohen, 1960), or Scott's Pi (Scott, 1955), are the preferable measures of IAA (Craggs and Wood, 2005; Di Eugenio, 2000; Carletta, 1996). However, for some annotation tasks it does not make sense to treat all disagreements equally. For example, the DA *choiceQuestion* and *checkQuestion* are semantically more similar than *request* and *accept*. Both Pi and Kappa are limited in such circumstances because they only consider identical labels for agreement. A solution to this problem is the use of weighted agreement coefficients, which consider the magnitude of disagreement between assigned labels. Weighted coefficients use a distance function (for the distance functions used within this study see B.2), which returns a value in the range [0, 1] representing the similarity between an arbitrary pair of labels. 0 indicates the two labels are identical and 1 indicates they are completely dissimilar. This value is then used to weight pairs of assigned labels, penalising those that are more dissimilar. Cohen (1968), proposed a weighted variation of Kappa for two annotators. More frequently used however, and more appropriate for this study, is Alpha (Krippendorff, 2004), and the Beta statistic, proposed by Artstein and Poesio (2005b).

Both Alpha and Beta are calculated from the observed and expected *disagreements*, rather than the agreement of Kappa and Pi. The ratio of observed (D_o) and expected (D_e) disagreement is subtracted from 1 to produce the final agreement value:

$$\alpha, \beta = 1 - \frac{D_o}{D_e} \tag{4.1}$$

Where Alpha and Beta differ, is in their estimations of the distribution of assigned labels for an annotator operating only by chance, that is, how D_e is estimated. When calculating D_e, Alpha estimates disagreement on the basis that each annotator assigns labels with the same distribution and therefore considered an *unbiased* coefficient, whereas Beta is *biased*, in that it calculates D_e from the observed distribution of individual annotators.

With respect to CAMS, the DA within the schema can be grouped into semantically similar communicative functions (Bunt, 2011), such as, information seeking and information providing. Further, some utterances can be thought of as *multidimensional* (Bunt, 2006), that is, they could be assigned two equally valid DA labels (or arguably both). A similar semantic grouping is also true for AP, where, for example, FPP-insert and SPP-insert are more closely related to an insert-expansion than AP from the Pre and Post groups. It seems reasonable to treat assignments that belong to different expansion types more seriously than those from the same group. As with DA, there is also an element of subjective interpretation involved when assigning AP labels. For example, identifying which utterances represent the 'core action' for a given sub-sequence of dialogue, and therefore should be assigned base-type labels, and those that should be considered expansions. The above, and the use of weighted agreement for DA annotation by (Geertzen and Bunt, 2006), indicates the use of weighted agreement measures, such as Alpha and Beta, are the appropriate choice for DA and AP annotation because the labels are not equally distinct from each other.

What is less clear, however, is the choice between these two coefficients. There has been much debate on this matter, largely concerning the so called 'bias problem' discussed by Krippendorff (2004) and others (Di Eugenio and Glass, 2004; Byrt, Bishop, and Carlin, 1993; Zwick, 1988). Though biased measures, such as Kappa and Beta, estimate expected agreement on the basis of individual annotator label distributions, they fail to account for unequal distributions *between* annotators, and effectively discount some of the disagreement resulting from different annotator distributions. Thus, for a fixed observed agreement, when annotators produce unequal distributions for the available categories –

when bias is present – the values of biased coefficients will *exceed* those of non-biased coefficients. However, Artstein and Poesio (2005b) point out that in practice the difference between biased and non-biased measures often doesn't amount to much, and that bias is a source of disagreement in its own right. Further, as stated by Di Eugenio and Glass (2004), the biased and non-biased paradigms reflect distinct conceptualisations of the problem, and in agreement with Artstein and Poesio (2008), the choice should depend on the desired interpretation of chance agreement. Yet, Di Eugenio and Glass (2004), also believed a bias coefficient is more appropriate for discourse and DA tagging, because it is questionable to assume equal annotator distributions for discourse and dialogue, and instead suggested reporting bias and unbiased coefficients together. Here a similar approach is taken, and both Alpha and Beta will be reported.

4.2.3.2 Coefficient Evaluation

Unfortunately, the question of what constitutes reliable agreement when interpreting agreement coefficients seems to be an unanswered question. The principal approach is based on a range of values proposed by Landis and Koch (1977), where below zero is considered 'Poor' agreement, and values between 0 and 1 are separated into five ranges: *Slight* (.0 - .2), *Fair* (.21 - .4), *Moderate* (.41 - .6), *Substantial* (.61 - .8), and *Perfect* (>.81). In Computational Linguistics, it is generally accepted that values of > 0.8 can be considered 'good reliability', and values in the range [.67, .8] allow for 'tentative conclusions to be drawn' (Krippendorff, 2004; Carletta, 1996). Though it is acknowledged that, as with the Landis and Koch (1977) values, because of diversity in both the phenomena being annotated and the applications of results, these ranges are not suitable in all cases (Craggs and Wood, 2005; Di Eugenio and Glass, 2004; Krippendorff, 2004; Carletta, 1996). This is especially true for annotation tasks such as this, where there is a degree of subjectivity in choosing an appropriate label, where some prior subject-specific knowledge is required, and notably for AP, prefect agreement will generally require annotators to agree on two (or more) labels, rather than one for DA. Indeed, it has been shown that achieving even the minimum 0.67 value is extremely difficult for discourse annotation (Poesio and Vieira, 1998; Hearst, 1997). Furthermore, in the presence of bias, a biased coefficient will always be larger than a non-biased one, and for this reason Geiß (2021) suggests that applying the same range of values is not appropriate, because they warrant different interpretations. Unfortunately, to the best of our knowledge no alternative scale for interpreting biased coefficients has been proposed. We therefore choose to evaluate both coefficients, Alpha and Beta, with respect to the ranges typically adopted throughout the literature; with the caveat that, for Beta it is necessary to be cautious when drawing conclusions if there is a significant difference between the two coefficients. In agreement with Artstein and Poesio, 2008; Craggs and Wood, 2005, choosing an agreement threshold should not be the sole measure upon which an annotation schema, or labelled corpus, should be considered reliable, and instead, the methodology for calculating reliability should be thoroughly communicated, so that conclusions can be drawn based on the characteristics and motivations of the particular study. The following annotation methodology considerations were suggested by Krippendorff (2004, ch. 11), and reiterated by (Artstein, 2018):

1. Annotators must work independently, so agreements come from a shared understanding not through discussion.
2. Annotators should come from a well-defined population, so that researchers are aware of previous knowledge or assumptions they bring to the annotation process.
3. Annotation instructions should be exhaustively formulated, clear and contain step-by-step instructions on how to use it.

4.2.4 Timing and Rating Measures

The annotation tool collected additional utterance annotation timing and label confidence data for each annotator. The purpose is to augment the comparison between task-oriented and non-task-oriented dialogues, and the different label types within the schema, that would not be possible with agreement coefficient data alone. It also provides additional insight into the participants annotation behaviour, such as a change in confidence, or the amount of time spent selecting labels, which may indicate how well annotators are able to learn and internalise the annotation scheme.

4.2.4.1 Annotation Timing

The annotation software allows users to select an utterance of dialogue, which is then highlighted to signal it is the 'target' for annotation. With an utterance selected, the user chooses a single DA and AP label to assign by clicking on their respective buttons. An utterance is considered *labelled* when it has been assigned one of each label type. At which point the software automatically selects the next unlabelled, or partially labelled, utterance. The time taken to annotate an utterance is measured as the total time the utterance is selected and unlabelled. This time is cumulative, so if a previously assigned label is removed, so that a different label can be selected, or it is unselected and re-selected later, any further annotation time is added to the previous total.

4.2.4.2 Annotation Confidence

Once a dialogue is fully labelled users are presented with a questionnaire screen. Here, they are asked to rate how well their assigned labels fit the dialogue in question. Ratings are provided by means of a Likert Scale between 1 (*not at all*) and 7 (*perfectly*). There are 3 questions, one for each label type; and the prompts emphasise the purpose of these label types. For example, how well the DA describe the communicative *meaning* of the utterances, AP the *structure*, and for AP-types, how well they combine to convey both structure and meaning. Since users must label every utterance, they are also given the option to highlight any cases where they felt certain labels did not adequately describe the utterance, or selection of utterances.

4.2.5 Results and Discussion

In this section the results of the annotation procedure are presented and some of the observations that arise are discussed. We begin with the IAA measures, firstly for each set of dialogue, before examining agreement for task and non-task-oriented dialogues, and each corpus. We then report the results for annotator confidence and timing data respectively.

4.2.5.1 Inter-annotator Agreement

IAA was calculated for the Alpha and Beta coefficients from the recorded annotations for each dialogue set. Figure 4.2 shows agreement values for each label type (DA, AP, and AP-type), and the overall mean agreement for each coefficient.
Figure 4.2 and subsequent statistical analysis shows that:

- According to the Landis and Koch (1977), scale we find that agreement for the Beta metric is 'substantial' for DA (.74) and AP-types (.67), and 'moderate' (.6) for AP alone. Using the range [.67, .8], (Krippendorff, 2004; Carletta, 1996), we find that only DA and AP-type labels are able to reach this threshold for the Beta coefficient.

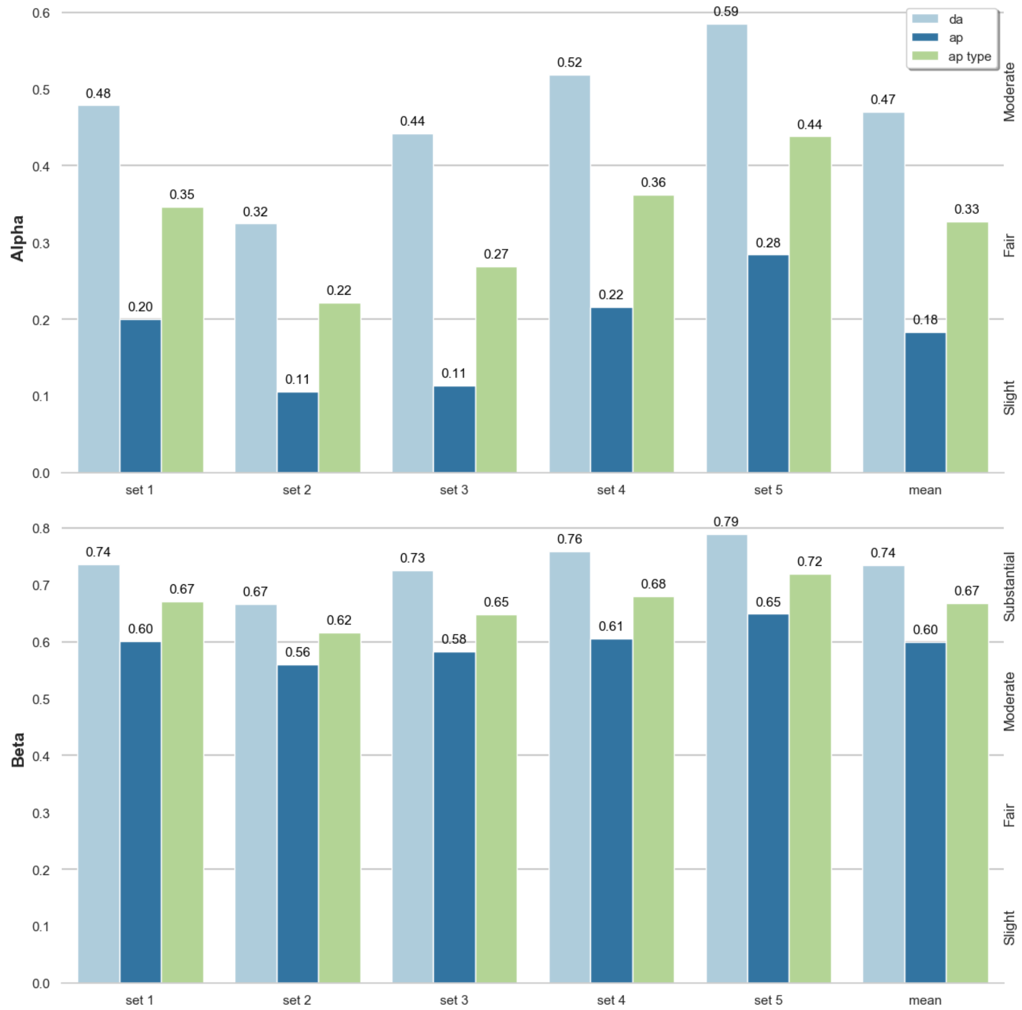


Figure 4.2: Alpha and Beta IAA values for each dialogue set.

- The Alpha metric produces the same pattern, but with lower values of agreement. DA agreement is 'moderate' (.47), while AP are 'slight' (.18), and AP-types 'fair' (.33). Comparing Alpha and Beta values, for each label type, show these are all significantly different ($p < .001, d > 1$). Possible reasons for this are explored further in appendix C.

- ANOVA over the label types (DA, AP, and AP-type) for each metric showed large effect sizes ($\omega^2 = .186$ and $\omega^2 = .179$ for Alpha and Beta respectively). Post-hoc analysis, reveals that this arose almost wholly from the AP:DA difference ($p < .001$) for both metrics.

Overall, we see a considerable difference between the values of Alpha and Beta. Though it is less pronounced for DA labels, with a mean difference of 0.27, than it is for AP, and AP-types, which differ by 0.42 and 0.34 respectively. These differences indicate that annotators had very different proclivities when assigning labels, and this bias has *increased* the values of Beta with respect to Alpha. In the case of AP this increase amounts to two full thresholds on the Landis and Koch (1977) scale, from 'slight' to 'moderate', and we therefore recommend that this is considered before drawing any conclusions of reliability from the Beta agreement values alone. However, that this difference is less for DA, and greater for AP, suggest that individual annotator distributions were more similar when assigning DA labels and less similar for AP labels. In other words, we see a higher degree of idiosyncratic interpretation between the annotators when selecting AP labels, and this is reflected in the difference between the two coefficients. This observation is discussed further in 4.2.5.4 and appendix C.

4.2.5.2 Task-oriented and Non-task-oriented Dialogues

A primary focus of this study is to investigate the extent to which the schema can be applied to different types of dialogue. Annotated dialogues were therefore split into their respective task and non-task-oriented groups, and again agreement was calculated using Alpha and Beta for each label type. Figure 4.3 shows the resulting agreement values for each dialogue group, and the practice dialogue:

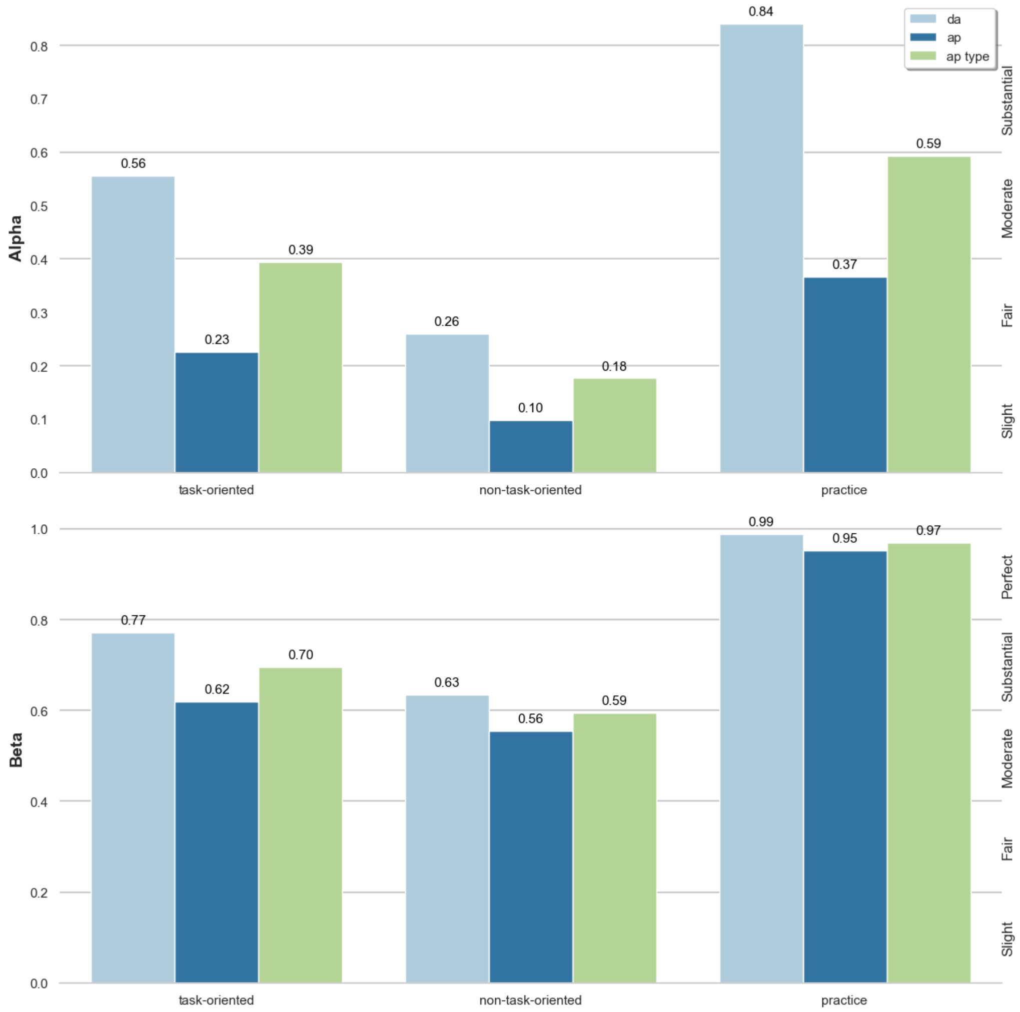


Figure 4.3: Alpha and Beta Agreement values for task and non-task dialogues.

- On the practice dialogue, the Beta metric reports 'perfect' agreement for all three groups of labels on the Landis and Koch (1977) scale (Beta > .95).
- For the Alpha metric, agreement on the practice dialogue is again 'perfect' for DA (.84), and high for the AP-types (.59) but lower for just the AP labels (.37).

Table 4.3: Label assignments by users 5 and 10 for a task (KVRET) and non-task (CABNC) dialogue.

CABNC (KBERE00G)	User-5	User-10
A1: Can you turn that radio off I want to listen to the phone in.	FPP-base propQuestion	FPP-pre request
B1: I got the whatsname on.	SPP-base decline	FPP-insert inform
A2: What What.	FPP-post stalling	SPP-insert feedbackNeg
B2: The whatsname Don't ask me I du n no what it's called.	SPP-post confirm	SPP-base answer
A3: What do you want that on for I'm trying to listen to the radio I want to listen to the phone in.	FPP-post feedbackNeg	Insert disagreement
KVRET (Test 102)	**User-5**	**User-10**
C1: Can you find out the date and parties attending my dinner?	FPP-base setQuestion	FPP-pre propQuestion
D1: Your dinner is on Tuesday with your sister.	SPP-base answer	SPP-base inform
C2: Thanks.	FPP-post thanking	FPP-post thanking
D2: you're welcome	SPP-post acceptThanking	SPP-post acceptThanking

- These practice results are consistently higher than the main results, possibly because there are more annotators, and (as will be seen later) due to the nature of the KVRET corpus.
- Agreement was consistently higher for task-oriented dialogues for all label types, and both coefficients. Overall these differences are statistically significant ($p < .001, d > 1$) for both Alpha and Beta. Only when looking at just the AP labels, is the task vs. non-task distinction not statistically significant ($p = .07, d = .86$ and $p = .56, d = .9$ for Alpha and Beta respectively).

Again, overall, the differences between the two coefficients is high in most cases, and consequently we advise caution when interpreting the Beta values with respect to typical agreement thresholds. However, it is worth noting that for DA labels the difference on the task-oriented dialogues (0.19), and the practice dialogue (0.15), is much smaller than previously observed. Therefore, we can conclude that, not only is agreement higher, but individual annotator distributions were more similar.

To examine the difference between the task-oriented and non-task-oriented groups further, table 4.3 shows the assignments produced by two annotators, users 10 and 5, for a task (KVRET) and non-task (CABNC) dialogue. We selected users 10 and 5 for this analysis because both exhibit a competent understanding of CAMS and its application. Yet as we will see, their differing interpretations of the CABNC dialogue led to negative agreement values. On the other hand, for the KVRET dialogue they reached near perfect agreement. Thus, this pairing provides clear insight into the properties of task-oriented and non-task-oriented dialogues that contribute to the observed differences in agreement between these groups, even between annotators who demonstrate a similar understanding of the annotation scheme. Additionally, both annotators made some small errors in assigning AP or DA. We highlight these assignments here and explore some of these observations further in section 4.2.5.4.

Firstly, we can see both annotators assign an invalid AP label to utterance A3; user-5 begins a FPP-post without a closing SPP, and user-10 places an insert label *outside* of a FPP/SPP base pair. User-10 also incorrectly begins a FPP-pre (A1) and closes with a SPP-base (B2), a pattern that is repeated in the KVRET dialogue. There are also some minor misuses of DA. In particular, user-5 assigns 'stalling' to (A2), which represents a speakers need for a little extra time to construct their contribution, for example, "Let me see..." or "Umm...". Given the nature of the following utterances, a question-type DA, or user-10's assignment of negative feedback, is more appropriate. However, the assignment of negative feedback for A3 is certainly incorrect, as this DA represents the speakers mishearing, or misunderstanding, of the previous utterance; a conclusion that is not borne out by its content.

Regarding AP, the main source of disagreement with the CABNC dialogue is what constitutes the core action or communicative goal, and thus should be assigned as base-type AP, and what utterances contribute to, or support, this action, and should therefore be expansions. Both correctly identify the core action as a request to turn the radio off in A1. However, user-5 considers this action complete with the refusal to do so in B1, and the following two utterances are merely clarifying the meaning of “whatsname”. On the other hand, user-10 considers that the response in B1 was a mishearing, or misunderstanding, by A and that this requires the insert pair before the action is completed in B2. Clearly these two interpretations led to significant disagreement between the two annotators and is largely driven by the ambiguity of certain utterances within the transcription, particularly A2. If A2 were instead “the what?”, or “who?”, then user-5’s interpretation is preferred, or alternatively, “sorry what?”, might suggest user-10’s understanding was correct. Unfortunately, “what what” lends itself to both these possibilities and hence the alternative interpretations. This is also reflected in the negative agreement scores between these two annotators, with an Alpha of -.1, and a Beta of -.05. For the KVRET dialogue there is no such ambiguity in which utterances make up the core action, and this resulted in ‘perfect’, or near perfect, agreement of .8 and .77 for Alpha and Beta respectively.

For DA we again see considerable disagreement for the CABNC dialogue, and this is largely driven by the alternative interpretations previously discussed. Of note, however, is the assignments of a ‘propositional question’ and a ‘request’ for utterance A1. Even though it is posed as a question, this statement is an indirect way of requesting that the radio be turned off, and therefore user-10’s assignment is more suitable (Bunt, 2017). Yet, it is easy to see how a propositional question, which suggests a positive (accept) or negative (decline) answer, is a reasonable alternative interpretation. Interestingly, despite the similar form of utterances A1 and C1, neither annotator assigned the same DA label. These dialogues were not presented in the order shown here, but this does indicate a change, or inconsistency, in interpretation; perhaps influenced by the presence of an interrogation mark in C1 which implies a question-type DA is appropriate. For the CABNC dialogue we again see negative agreement, -.03 and -.06, and for the KVRET dialogue substantial agreement of .79 and .76 for Alpha and Beta respectively.

From these results we can see that, while there is some incorrect usage of both AP and DA, the main source of disagreement stems from difficulties interpreting the non-task-oriented dialogue. The two alternative views discussed above suggest two different sets of AP assignments, depending on where one considers the core action to have been completed, and this is largely driven by the ambiguity of utterance A2 observed above. Macagno and Bigi (2018), referred to this phenomenon as ‘imaginary ambiguity’, that is, a particular utterance can have multiple distinct interpretations for the intended effect on the recipient depending on the context. In this case, A2 is interpreted differently depending on the reading of B1 as a refusal, or misunderstanding. This kind of *meaning multiplicity* (Boxman-Shabtai, 2020) may arise, at least in part, from the nature of transcribed material of natural conversations, where social cues, such as prosody, intonation, and body language, are lost. Indeed, Collins, Leonard-Clarke, and O’Mahoney (2019), were able to show that disfluencies in speech can have very different meanings when presented in spoken and written form, and we surmise that this is also true of illocutionary ambiguous utterances. As noted by Green, Franquiz, and Dixon (1997), “*a transcript is a text that ‘re’-presents an event; not the event itself*”, thus information is inevitably lost. In any case, these differing interpretations are a clear example of bias on the part of individual annotators, and have therefore contributed to the inflation of the Beta coefficient, and its divergence from Alpha. On the other hand, for the task-oriented dialogue there is a clear delineation between the core action and the remaining 'thanking' utterances. This concurs with the work of Grosz (2018), who established that task-oriented dialogues are structured, with multiple utterances grouping into a dialogue segment, and their structure mirrors the structure of the task. This characteristic simplifies the identification of AP and we therefore see much higher agreement and lower bias.

4.2.5.3 Corpora Dialogues

An additional factor which may contribute to the observed difference in agreement between the task and non-task dialogue groups is the number of utterances in each dialogue. Dialogues in the SCoSE corpus contain an average of 23.6 utterances, around half of the total number of utterances in each dialogue set, and may therefore be contributing a disproportionate amount of agreement (or disagreement) to the overall values. Hence Figure 4.4 breaks the comparison into different corpora. A further ANOVA and post-hoc analysis of agreement between pairs of corpora, was performed for each label type and coefficient:

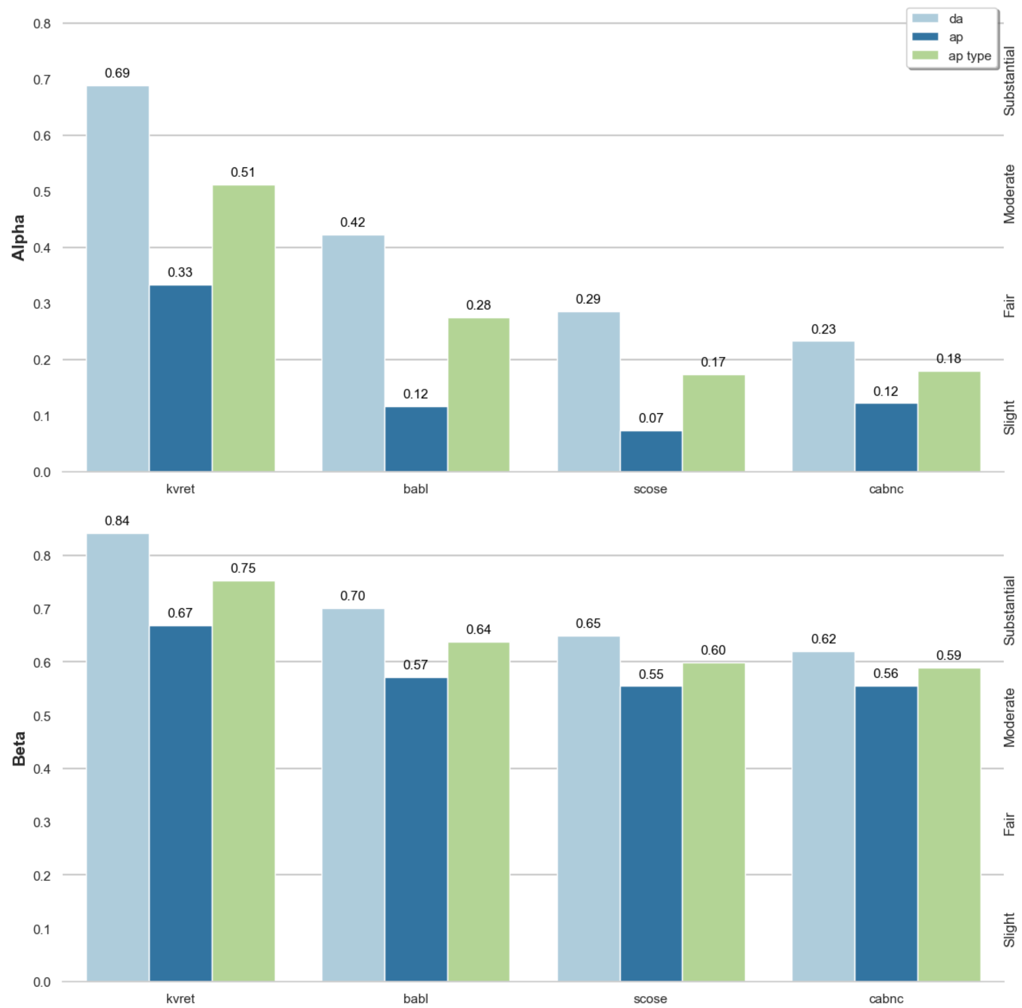


Figure 4.4: Alpha and Beta Agreement values for each corpus.

- The post-hoc analysis reveals that there is no significant difference in agreements ($p = .9$) between the two non-task-oriented corpora, CABNC and SCoSE, for both Alpha and Beta coefficients, despite a mean utterance length of 5.6 and 23.6 respectively. This is also the case when comparing the bAbI corpus (mean utterance length

5.8) and the non-task-oriented corpora. Therefore, it is unlikely that the number of utterances is contributing to the observed differences in agreement between the groups.

- Predominantly, the statistically significant results are for DA and AP-type labels between KVRET and the other corpora. This indicates that the difference in agreement values are a product of higher agreement for the KVRET corpus, rather than a difference between the groups. Certainly, agreement is higher on the KVRET corpus, for all label types and both agreement coefficients.

- These results also provide some insight into the previous observation, that there is no significant difference in agreement for AP labels between the groups. Only the KVRET and SCoSE comparison for the Alpha metric produced a significant result ($p = .028$) and in all other cases we still see no statistical difference for AP labels.

These results show that, once more, there is a large difference between Alpha and Beta, and this is greater for AP than DA, hence a larger degree of idiosyncratic interpretation between the annotators. However, in accordance with the previous remarks, this bias is lower for the KVRET corpus than it is for the other three. Thus, while agreement for DA is higher for both task-oriented corpora, for AP we see no difference in agreement between the bAbI corpus and the two non-task-oriented corpora.

4.2.5.4 AP Label Agreement

As previously observed, there appears to be no significant difference in agreement for AP labels between the task and non-task dialogue groups, and further, that much of this is caused by the negligible difference between the bAbI, CABNC and SCoSE corpora. Manual inspection of the annotations revealed that a considerable amount of confusion seemed to arise around the valid use of FPP and SPP for AP. Often annotators would assign a SPP to initiate a sequence (rather than a FPP), or fail to create a valid sequence entirely, for example, by assigning a FPP without an accompanying SPP. This observation was explored further using an adjusted AP distance function, which ignores the AP prefix (FPP/SPP), and instead *only* considers the difference between the AP *base* or *expansion types* (pre, post, and insert). The 'suffix-only' distance function treats all labels as equally distinct, with a distance of 1 for non-identical labels, and 0 otherwise. For example, two *insert* type labels (FPP-insert, SPP-insert or insert) would have a distance of 0 between them, but a distance of 1 with all other AP label types. Therefore, the suffix-only distance function should indicate the extent to which annotators misunderstanding of the valid use of FPP and SPP labels contributed to the observed AP agreement values. Figure 4.5 shows the agreement values that were recalculated for using the suffix-only distance function.

- Using the suffix-only distance function both task-oriented corpora show improved agreement for AP labels, with a minimal improvement for the KVRET corpus but a considerable improvement for bAbI. For Alpha the bAbI agreement doubled from .12 to .24, and Beta shows an increase from .57 to .62.

- Both non-task-oriented corpora show a decrease in AP agreement, though, again the effect is greater for the Alpha coefficient, with a decrease of .05 and .07 for SCoSE and CABNC respectively, compared to .01 and .04 for Beta.

- Post-hoc analysis reveals there is now no longer a significant difference in AP-type labels when comparing the KVRET and bAbI corpora ($p = .181$ and $p = .193$, for Alpha and Beta respectively).

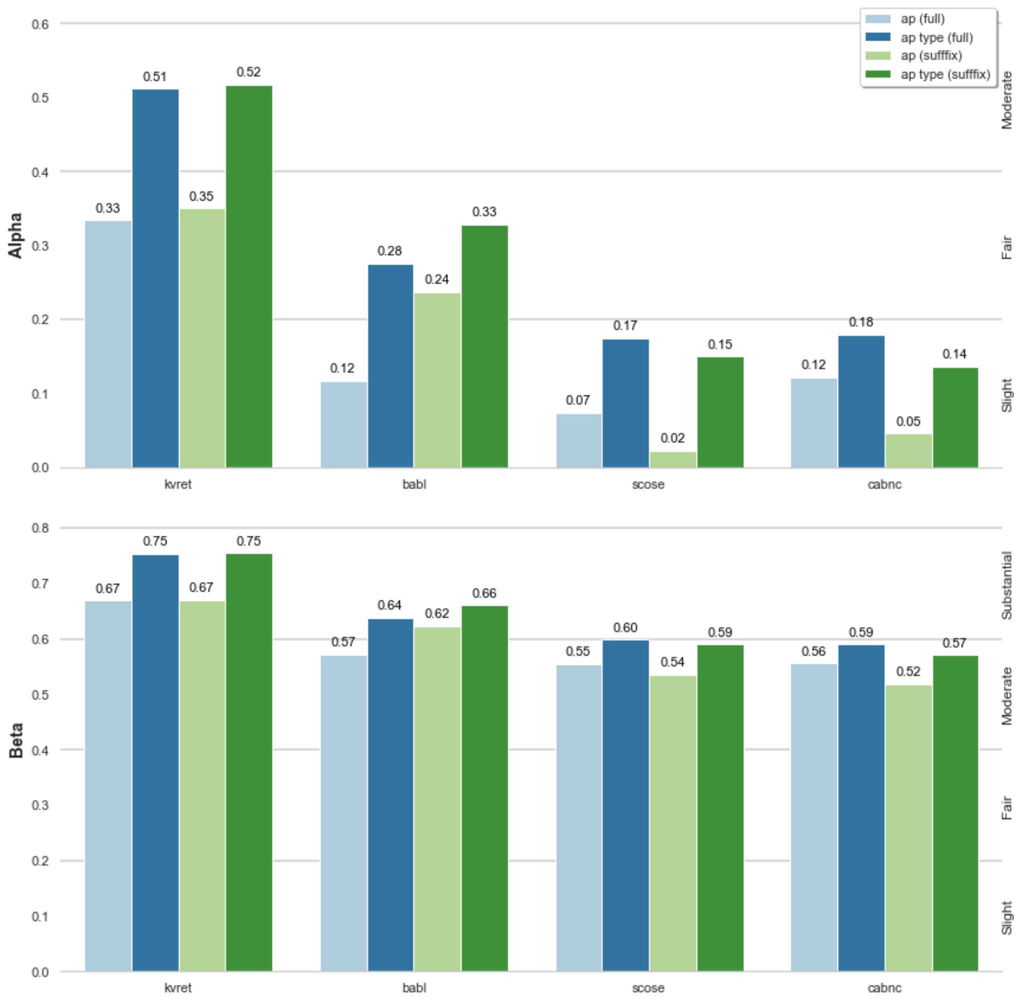


Figure 4.5: Corpora agreement values calculated with the suffix-only AP distance function.

This indicates that, when annotators misunderstanding of the valid use of FPP and SPP is not considered, they tend to more often agree on the *base* and *expansion types* of AP labels for task-oriented dialogues. Whereas, for non-task-oriented dialogues the opposite is true, with a *decrease* in agreement that suggests annotators rarely agree on the AP *base* or *expansion types*. Perhaps unsurprisingly, this suggests that the structure of non-task-oriented dialogues is less well defined, and open to more subjective interpretation, than that of task-oriented dialogues. It may also offer explanation for the lack of significant difference in AP agreement, and high bias, that was previously observed. Using a two-sided t-test to compare the suffix-only agreement scores for AP labels between the task and non-task groups now results in a statistically significant difference for Alpha and Beta ($p = .0028, d > 1$ and $p = .0089, d > 1$, respectively). Therefore, the incorrect usage of FPP and SPP was *reducing* agreement for task-oriented dialogues, while for non-task dialogues *increasing* agreement, and 'evening out' AP agreement values between the groups. These results also suggest that using non-expert annotators may not be suitable for this task, as many seem to lack a clear understanding of the proper use of AP, or alternatively, more training beforehand may help to improve understanding in this regard. It is also possible that some of the confusion was caused by the similarity between FPP and SPP, with only one-character difference between the two labels.

4.2.5.5 Expert Annotators

As we have shown, our novice annotators individual interpretations, and misunderstanding of the correct usage of AP, have led to overall lower agreement for AP when compared to DA. Additionally, difficulties identifying the core action of the non-task-oriented dialogues have contributed to lower agreement scores for those dialogues. For these reasons, here we briefly compare the IAA between two 'expert annotators', the author and a linguistics lecturer. Agreement was calculated for a single set of dialogues, with results shown in figure 4.6.

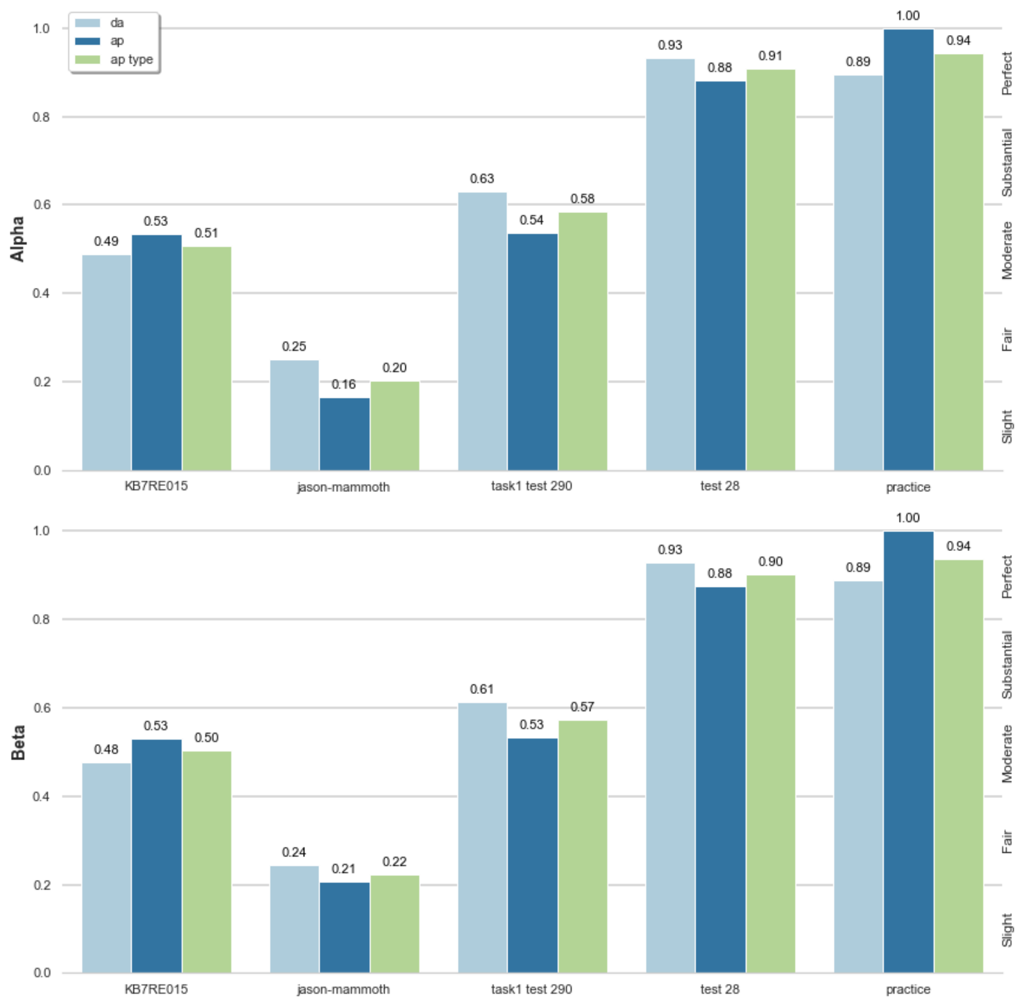


Figure 4.6: IAA scores between two expert annotators for the dialogues in set 1.

- Regarding the observed difference between task and non-task-oriented dialogues we can see the same general pattern. Agreement is higher for the task-oriented dialogues, with both KVRET dialogues (*test 28* and *practice*) reaching 'perfect' agreement for all label types, and bAbI (*task 1 test 290*) reaching 'moderate' for AP and AP-types, to 'substantial' for DA. For non-task dialogues, SCoSE (*jason-mammoth*) reaches only 'slight' to 'fair' agreement, while CABNC shows 'moderate' agreement for all label types.

- Most importantly, we see very small differences between the scores for Alpha and

Beta, typically 0.1 to 0.5 at most. This indicates that the label distributions, and interpretations, of both annotators was very similar, hence bias is lower and the two coefficient scores start to converge.

- In comparison to previous results we also see smaller differences between DA and AP, particularly for Alpha. This shows that, unsurprisingly, the expert annotators more appropriately applied the AP labels, and avoided the misunderstanding and incorrect usage of our novice annotators.

These results confirm that the schema is indeed more easily applied to task-oriented dialogues. That expert annotators more correctly applied the AP labels, and produced more similar distributions, is unsurprising. However, it does confirm the difficulties encountered by our novice annotators, and the impact on agreement discussed above. Encouragingly, the agreement for all labels types and both coefficients was extremely high for both KVRET dialogues, which indicates it is a viable corpus for our purpose of annotating with CAMS.

4.2.5.6 Annotation Confidence Scores

Analysis of participants confidence scores supports some of the observations from the previous sections. Overall, annotators reported a higher confidence in their assigned labels for task-oriented dialogues than for non-task-oriented dialogues (table 4.4), which coincides with the higher agreement for task-oriented dialogues observed in our previous results. Notably, although the mean confidence between labelling tasks differed, the standard deviation of confidences range between 0.64 and 1.31, in other words, less than two Likert scale points. The difference in confidence between task and non-task was significant overall ($p < .001$) for the AP-type labels and both AP, and DA.[3]

Table 4.4: Mean and standard deviation of confidence scores by label type, corpus, and dialogue type.

	KVRET		**bAbI**		**Task**		**SCoSE**		**CABNC**		**Non-task**	
Type	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
DA	5.06	1.03	4.53	0.99	4.8	1.03	4	1.31	4.13	0.64	4.07	1.01
AP	5.27	1.09	4.13	0.99	4.7	1.18	3.93	1.16	4	0.85	3.97	0.99
AP-Type	4.87	0.99	4.53	0.96	4.7	0.95	3.67	0.98	3.8	0.68	3.73	0.83
Overall	5.07	1.03	4.4	0.96	4.73	1.05	3.87	1.14	3.98	0.72	3.92	0.95

If we again examine confidence scores with respect to each corpus, we also see a result similar to that for agreement values. That is, confidence is highest for the KVRET corpus and lowest for SCoSE, with the other task-oriented corpus being marginally higher than CABNC in most cases (figure 4.7). For each label type, an ANOVA over confidence scores per-corpora concur with those of agreement. Overall results are significant ($p \leq .027$), and effect size is large for AP and AP-types ($\omega^2 > .14$), and medium for DA ($\omega^2 = .1$).[4] Post-hoc analysis shows the only place we see significant differences is between KVRET and the other corpora, particularly with AP. Similarly, the difference between the two non-task-oriented corpora and bAbI is statistically non-significant in all cases. This indicates that, as with agreement, the division is not necessarily between task and non-task-oriented dialogues, but primarily between KVRET and the other three corpora.

[3] Due to the small sample size of confidence scores (one score per-label) the resulting statistical power for AP and DA is .72, and .77 respectively.

[4] The resulting statistical power for DA is .72.

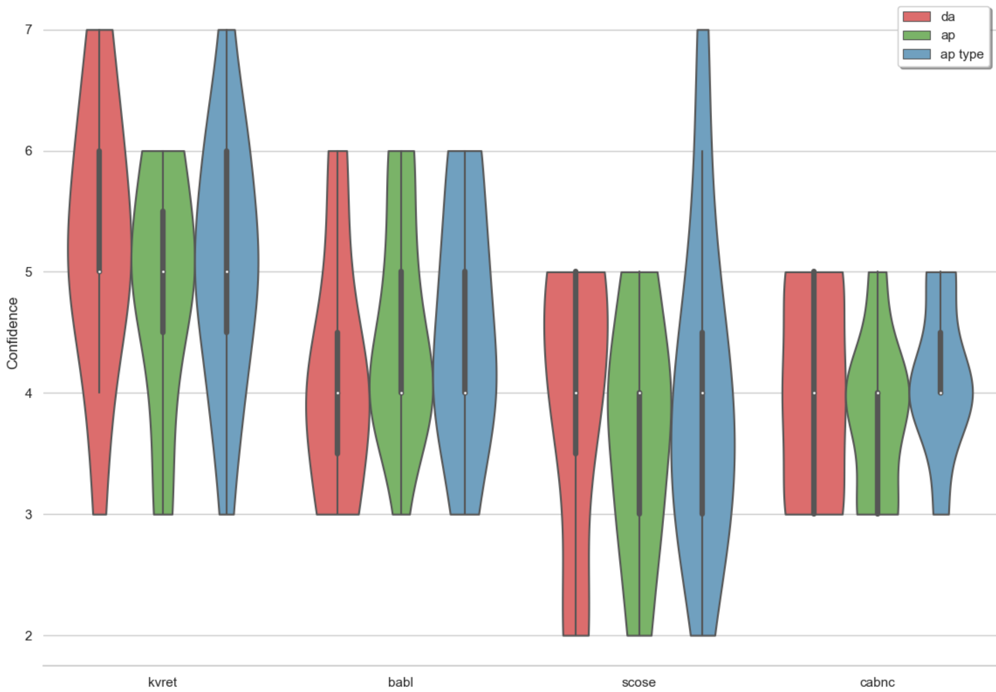


Figure 4.7: Reported annotator confidence scores for each dialogue and label type.

These results show that there is a remarkable similarity between the annotators reported confidence scores and the resulting agreement values. When considered from the perspectives of task and non-task-oriented dialogues, individual corpora, and different label types, where higher confidence was reported, agreement was also higher. Annotators were therefore quite good at assessing how well their assigned labels fit the data, reporting higher confidence for dialogues where appropriate labels, or dialogue structure, was more intuitive, and lower confidence on the less structured dialogue types. This also suggests that incorporating confidence scores could be a valuable resource assessing labelling accuracy. Kazai (2011), showed that annotators who rated the task easier also had a higher accuracy. While Oyama et al. (2013), used self-reported confidence scores, along with their assigned labels, to estimate the 'true' labels using the expectation-maximization (EM) algorithm.

4.2.5.7 Annotation Time

The time participants took to completely annotate each utterance was also recorded. Because participants likely spent some time reading utterances and considering labels at the beginning of each dialogue, here all reported times are the average time taken, in seconds, to annotate an utterance for that dialogue. Unlike agreement values and confidence scores, utterance times reveal that there is little difference between task and non-task-oriented dialogues, or the different corpora, as shown in table 4.5. Therefore, despite reporting lower confidence for non-task-oriented dialogues, and the SCoSE corpus also containing around 4 times as many utterances, this did not seem to affect the average amount of time spent annotating those dialogues.

If we instead look at the average utterance time in the order dialogues were annotated, regardless of the specific dialogue, we see that annotation habits do indeed change over time. Figure 4.8 and table 4.6 show that, for all participants, annotation time became

Table 4.5: Mean and standard deviation of utterance annotation time (seconds) per corpus and dialogue type.

	KVRET	bAbI	Task	SCoSE	CABNC	Non-task
μ	24.62	33.57	29.09	25.56	36.69	31.13
σ	8.94	19.05	15.31	11.09	24.16	19.36

faster as they progressed through the task, starting with an average of 77.89 seconds for the practice dialogue and ending with 19.81 seconds by dialogue 4. And further, that the variance between participants times also grew smaller over time, moving from a standard deviation of 27.52 on the practice dialogue, to just 6.03 on dialogue 4. These results seem to show a clear learning-effect, which echoes the results of Aulamo, Creutz, and Sjöblom (2019), where participants start with slow annotation speed, then, after a period of familiarisation with the task, speed is increased and maintained for the remaining time. It may also be valuable to determine if there is a similar change in agreement over time, as annotators became more familiar with the schema and tool. Unfortunately, because all but the practice dialogue were shown in a random order for each participant, it is not possible to show that data and it will be left for future work. However, given that the practice dialogue also resulted in the highest agreement values, we suspect that this may not have a significant impact on agreement.

Table 4.6: Min, max, mean, and standard deviation of annotators mean utterance annotation time (seconds) in the order dialogues were completed.

Dialogue	Practice	1	2	3	4
Min	37.75	21.02	10.55	9.9	10.71
Max	117.87	89.06	95.76	46.17	30.20
μ	77.89	43.85	37.42	23.62	19.81
σ	27.52	20.48	24.85	11.08	6.03

4.2.5.8 CAMS Evaluation Summary

Our findings indicate that inter-annotator agreement is significantly higher for the biased Beta coefficient, than that of unbiased Alpha, and this is principally caused by the differences in annotator label distributions increasing the Beta values. We therefore advise caution when comparing the two coefficients using the standard scales of interpretation (Geiß, 2021), particularly when biased measures diverge from unbiased ones. Nevertheless, if we assess agreement values of each dialogue set, using the somewhat arbitrary scale of Landis and Koch (1977), we find that for Beta DA and AP-type agreement can be considered 'substantial', while AP fall into the 'moderate' agreement category. However, agreement for the Alpha coefficient is less convincing. DA show a 'moderate' level of agreement, while AP and AP-types only achieve 'slight' and 'fair' respectively. If we use the more stringent range [.67, .8], often used in Computational Linguistics to allow for 'tentative conclusions to be drawn' (Krippendorff, 2004; Carletta, 1996), we find that only DA and AP-type labels are able to reach this threshold for the Beta coefficient. These results seem to concur with Poesio and Vieira (1998), and Hearst (1997), that reaching the .67 threshold is difficult for discourse annotation tasks. Indeed, Pareti and Lando (2019) showed that for a similar 'dialogue act linking' task even expert annotators mis-

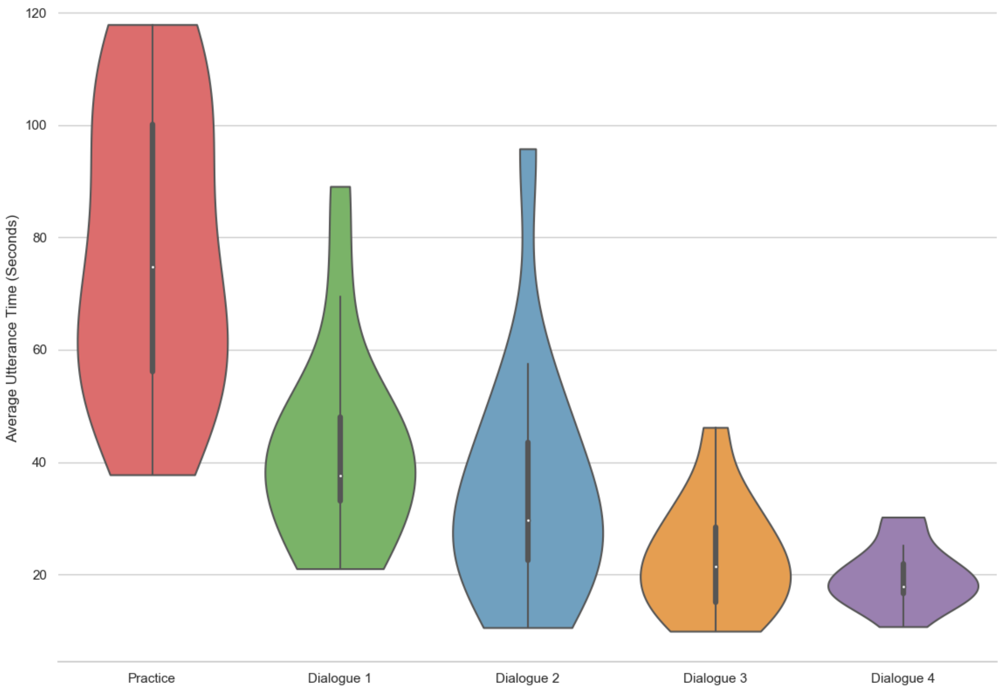


Figure 4.8: Distribution of annotators mean utterance annotation time (seconds) in the order dialogues were completed.

understood the guidelines, resulting in poor F1 scores. In this case, it may be due to our use of non-expert annotators, who have been shown to misunderstand the proper use of AP, and therefore more intense training should be provided, or expert annotators used. It may also be due to differences in individual annotator interpretations of the dialogues and appropriate AP labels. However, our participants agreement values can be considered an indication of moderate reliability.

Regarding task-oriented and non-task-oriented dialogues, both annotator agreement and self-reported annotator confidence scores are higher for task-oriented dialogues than non-task. However, when considered from the perspective of the individual corpora this distinction is not as clear. With the (task-oriented) KVRET corpus resulting in higher agreement and confidence scores than the other 3. We therefore conclude that, while CAMS is indeed applicable to both task and non-task-oriented dialogues, our results show that it is more intuitively applied to task-oriented dialogues. The determining factor, however, is not the division between task and non-task, but rather the content of the dialogue itself.

Notably, we observed that utterances where the DA label is ambiguous, or multidimensional, can lead to different interpretations of the dialogue and result in a high number of disagreements for both DA *and* AP. Regarding the constituent label types within the schema, we found that DA labels consistently resulted in higher agreement and confidence scores than AP. This is perhaps not surprising, given that DA labels need only apply to one utterance at a time and generally use more intuitive names. AP on the other hand, require more specialised knowledge, and annotators must also consider relationships between utterances in order to apply them correctly. We found that many annotators misunderstood, and incorrectly applied the FPP and SPP labels, potentially caused by the similarity between the two. Perhaps changing the labels to, for example, ‘first-part’ and ‘second-part’, would help mitigate the problem of assigning these in the wrong order.

We confirmed the above observations by measuring the agreement of two expert annotators for a single set of dialogue. Expert agreement was consistently higher in all cases, reaching 'perfect' agreement for the KVRET dialogues, and indicated lower bias and reduced the disparity between DA and AP, with only small differences between Alpha and Beta. Therefore, if labelling accuracy is required for the creation of an annotated corpus, this task may be better suited to experts, or novice annotators who have received more training than ours. In order to produce accurate agreement scores the annotation tool intentionally placed no restrictions on label assignments; In future iterations this could be altered, to prevent, for example, the invalid creation of a new AP before a prior pair is completed. However, measuring the average time taken to annotate each utterance shows a clear pattern of learning, with annotation time decreasing for all annotators the longer they spent on the task. This indicates that the schema is inherently learnable and becomes more intuitive to apply with practice. Promisingly, both novice and expert annotators consistently produced high agreement scores for the task-oriented KVRET corpus, indicating that it is well suited to our purposes.

4.3 CAMS-KVRET Annotation

In this section we provide details of the annotation process that was used to develop CAMS-KVRET, a task-oriented corpus annotated with our CAMS, and intended to facilitate the development of an automatic method of identifying the CA structure of dialogue (**O2**). As discussed in section 3.1.3, CAMS-KVRET is derived from the KVRET corpus (Eric and Manning, 2017b), and was developed as a multi-turn, multi-domain dataset which contains 3,031 dialogues in three distinct domains appropriate for an in-car assistant: calendar scheduling, weather information retrieval, and point-of-interest navigation. KVRET was collected using a Wizard-of-Oz scheme in which users had two potential modes they could play: Driver and Car Assistant. In the *Driver* mode, users were presented with a task that listed certain information they were trying to extract from the Car Assistant as well as the dialogue history exchanged between Driver and Car Assistant up to that point. For example, "*You want to find what the temperature is like in San Mateo over the next two days.*" In the *Car Assistant* mode, users were presented with the dialogue history exchanged up to that point and a private knowledge base, known only to the Car Assistant, with information that could be useful for satisfying the Driver query. Knowledge bases could include a calendar of event information, a collection of weekly forecasts for nearby cities, or nearby points-of-interest with relevant information. The Car Assistant was then responsible for using this private information to provide a single utterance that progressed the user-directed dialogues and also asked to fill in dialogue state information for mentioned slots and values in the dialogue history up to that point. For our purposes KVRET is therefore, i) task-oriented, with three distinct tasks, ii) large enough to meaningfully train a ML model to identify the annotated AP-types, yet small enough to annotate within a reasonable amount of time, and iii) contains slot and task information, along with an accompanying knowledge base, for developing a dialogue system, and hence is applicable to future applications of our work.

We annotated KVRET using specially developed annotation software, with a similar interface to that shown in figure 4.1. Throughout the annotation process each dialogue is shown in turn, and for each utterance a single DA and AP are selected to form an AP-type. With the exception of fixing several typos, and removing two dialogues with only a single utterance from the test set, we make no changes to the underlying data. We also maintain the original authors training, evaluation, and test sets. The tool saves dialogues in JSON format which preserves the original alignment of slots and values for each system (Car Assistant) turn, and the alignment of 'scenario' data: task type, id, and items.

4.3.1 CAMS-KVRET Overview

Here we provide a brief overview of the resulting CAMS-KVRET corpus, consisting of 3029 dialogues annotated with the CAMS schema.[5] The following is an example of a fully annotated dialogue. Tables 4.7, and 4.8, respectively show a general overview of the corpus stats and the training, test, and evaluation sets. Including, number of utterances, length of dialogues, vocabulary size, and number of each label type.

USR:	What time is dinner and who is it with?	***FPP-base - setQuestion***
SYS:	Dinner is at 7 pm with Jon.	***SPP-base - answer***
USR:	Okay, perfect!	***Post - autoPositive***
USR:	Thanks.	***FPP-base - thanking***
SYS:	you're welcome.	***SPP-base - acceptThanking***

Table 4.7: Overview of the CAMS-KVRET Corpus.

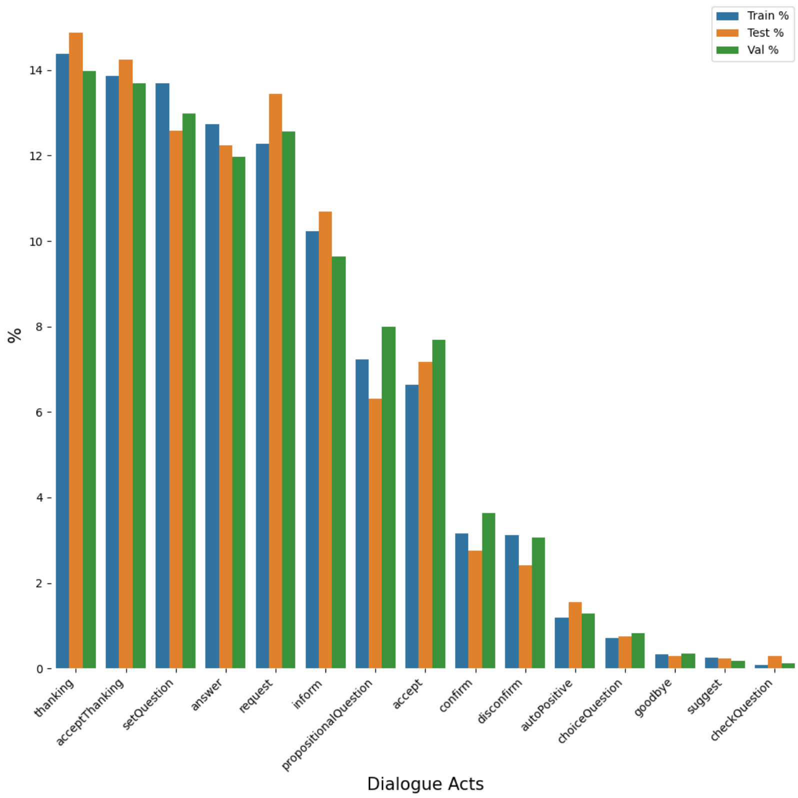
	Num
Total Utterances	17307
Max Utterance Len	95
Mean Utterance Len	8.65
Total Dialogues	3029
Max Dialogue Len	13
Mean Dialogue Len	5.71
Vocabulary Size	1912
DA Labels	23
AP Labels	9
AP-type Labels	104

Table 4.8: Overview of the CAMS-KVRET train, test and validation sets.

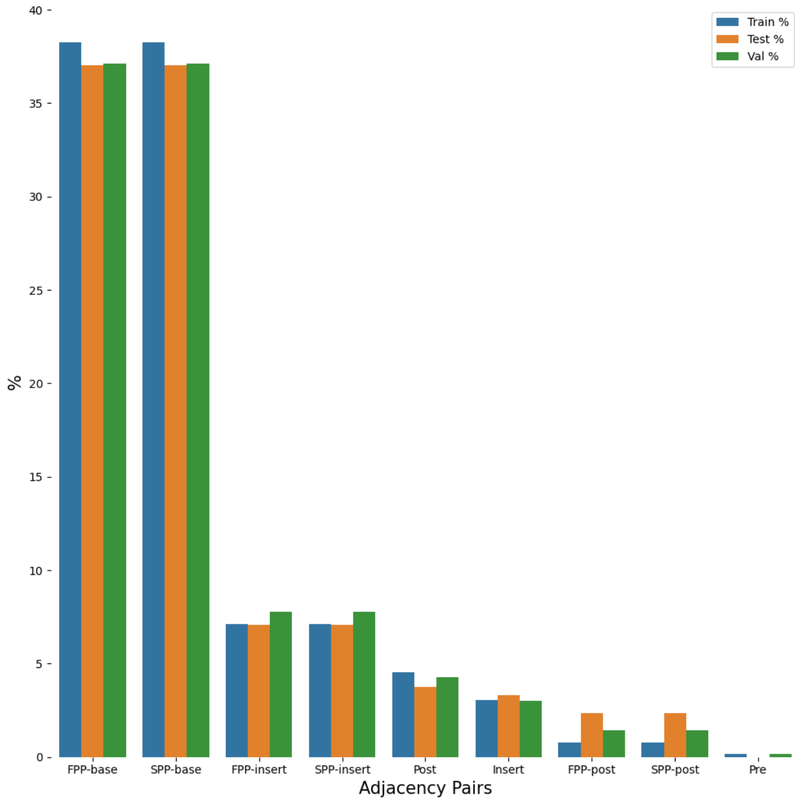
Set	**Num Dialogues**	**Max Len**	**Mean Len**	**Num Utterances**
Train	2423	13	5.72	13863
Test	304	13	5.73	1741
Validation	302	13	5.64	1703

Figure 4.9 shows the proportions of DA and AP labels within the corpus, and their distribution between each set.[6] In both cases, the proportion of labels is relatively balanced between the training, test, and evaluation sets. For DA, the two most common labels are *thanking* and *acceptThanking*, because most dialogues within the corpus end with reciprocal thanking, as in the example above. Equally, several information-seeking and information-providing DA, such as *setQuestion* and *answer*, occur frequently. Unsurprisingly, the FPP-base and SPP-base labels are the two most common AP. It should also be noted that the FPP-pre and SPP-pre expansion were not assigned in any case. This is because the KVRET dialogues are always initiated by the user (Driver), who, during data collection, had been given a specific task to complete. Thus there are no instances of preparatory, or preliminary utterances within the corpus.

[5]The corpus is available at: github.com/NathanDuran/CAMS-KVRET.
[6]Full label distribution data is available in appendix D.



(a) DA distribution of the most frequent 15 labels.



(b) AP distribution.

Figure 4.9: Distribution of DA and AP labels within the CAMS-KVRET corpus.

Chapter 5

Dialogue Act Identification

In this chapter we present experimental results for the first two phases of our dialogue classification system, as discussed in 3.2 (**O3**, **O4**, and **Q2**). We begin with *sentence encoding* in Section 5.1, where we focus on the various text pre-processing considerations and single-sentence classification models, without any additional contextual information. Then, in Section 5.2, we explore different forms of contextual input (speakers, labels, and utterances), contextual multi-sentence architectures and sequence classification models.

5.1 Sentence Encoding for Dialogue Act Classification

Here we present the results and analysis for each of our sentence encoding experiments.[1] For each of the input sequence and word embedding experiments we kept all parameters fixed at a default value, and only changed the parameter relevant to the given experiment. For example, when testing different vocabulary sizes, only the parameter that determined the number of words to keep in the vocabulary during text pre-processing was changed, all other parameters (letter case, use of punctuation, maximum sequence length, and word embeddings) remained fixed. By default, we lower cased all words, kept all punctuation marks, and used 50 dimensional GloVe embeddings. Additionally, for all supervised models, word embeddings were fine-tuned alongside the model during training. These default values were chosen so as not to restrict the amount of information available to the model while testing other parameters. For example, having an arbitrarily small sequence length while testing different vocabulary sizes, and vice versa. Further, these values represent the upper-bound of values to be tested, and are at, or near, the maximum possible value for their respective corpora.

5.1.1 Dialogue Act Corpora

The following provides an overview of each corpus, such as DA label categories, selection of training and test data, and a description of some corpus-specific pre-processing steps that were performed. Table 5.1 summarises key features of these two corpora, such as number of DA labels, vocabulary size, and so on, once pre-processing is completed.

SwDA During pre-processing, in some cases, it makes sense to remove or collapse several of the DA label categories. We remove all utterances marked as *Non-verbal*, for example, *[laughter]* or *[throat-clearing]*, as these do not contain any relevant lexical information (Ribeiro, Ribeiro, and De Matos, 2019; Stolcke et al., 2000). The *Abandoned* and *Uninterpretable* labels are also merged since these both represent disruptions to the conversation

[1] All code, data, and accompanying analysis for the sentence encoding experiments is available at: github.com/NathanDuran/Sentence-Encoding-for-DA-Classification.

flow and consist of incomplete or fragmented utterances (Ribeiro, Ribeiro, and De Matos, 2019; Kalchbrenner and Blunsom, 2013). Some utterances are also marked as *Interrupted*, indicating that the utterance was interrupted but continued later in the conversation. All interrupted utterances are concatenated with their finishing segment and assigned its corresponding DA label, effectively creating full uninterrupted utterances (Ribeiro, Ribeiro, and De Matos, 2019; Webb and Hepple, 2005). The resulting set therefore contains a total of 41 DA labels, with the removal of Non-verbal labels reducing the number of utterances by ~2%. Finally, all disfluency and other annotation symbols are removed from the text.

The 1155 conversations are split into 1115 for the training set and 19 for the test set, as suggested by Stolcke et al. (2000), and widely used throughout the literature (Cerisara, Král, and Lenc, 2017; Papalampidi, Iosif, and Potamianos, 2017; Kalchbrenner and Blunsom, 2013). The remaining 21 conversations are used as the validation set. It should be noted that this training and test split results in a large imbalance between two of the most common labels within the corpus, Statement-non-opinion (sd) and Statement-opinion (sv). However, to enable comparison with much of the previous work that uses this corpus we retain this imbalanced split.

For SwDA the default vocabulary size was set at 10,000 words with a maximum sequence length of 128 tokens. This vocabulary size is less than half that of the full vocabulary, however, as discussed in 2.2.3, the typical range used for this corpus is 10,000 to 20,000 words, and Cerisara, Král, and Lenc (2017) achieved their best results with a much smaller vocabulary size.

Maptask The HCRC Maptask corpus (Thompson et al., 1991), transcribed utterances were annotated with 13 DA labels. However, this is reduced to 12 DA labels by removing utterances tagged with *Uncodable*, as these are not part of the Maptask coding scheme. As with the SwDA corpus all disfluency symbols are removed, including incomplete words, for example *'th–'*. However, unlike the SwDA corpus, Maptask contains no punctuation, aside from a few exceptions, for example, *"sort of 's' shape"* to describe shapes on the map, and it also contains no capital letters.

The authors do not define any training and test data split for the Maptask corpus; we randomly split the 128 dialogues into 3 parts. The training set comprises 80% of the dialogues (102), and the test and validation sets 10% each (13), which is similar to proportions used in previous studies (Tran, Haffari, and Zukerman, 2017; Tran, Zukerman, and Haffari, 2017). Finally, for Maptask we use a default vocabulary size of 1,700 words and a maximum sequence length of 115.

Table 5.1: Overview of the SwDA and Maptask corpora used throughout this study.

Corpus	Num DA	Vocabulary Size	Utt Length Max (Mean)	Total Utts	Train	Val	Test
SwDA	41	10,000	128 (9.6)	199,740	192,390	3,272	4,078
Maptask	12	1,700	115 (6.2)	26,743	21,052	2,929	2,762

5.1.2 Sentence-encoders

The sentence encoder models can be separated into two categories: those trained in a fully supervised fashion, and those that use transfer learning via a pre-trained language model to generate utterance representations. In both cases the default classification component is a two-layer FFNN where the number of nodes in the final layer is equal to the number of labels in the training corpus. The final layer uses softmax activation to calculate the probability distribution over all possible labels and we use categorical cross entropy for the loss function. In the following, model hyperparameters such as, number of filters, kernel size, recurrent units, pool size and type (max or average), are selected based on results of a Bayes search algorithm exploring a maximum of 100 parameter combinations, with each run consisting of 5 epochs. However, in cases where we use existing published models (that is, excluding CNN, LSTM and GRU), we keep all parameters consistent with those reported in the original publications where possible.[2]

For our supervised models we use a selection of six based on convolutional and recurrent architectures, with a further set of bi-directional, multi-layer, and attentional variants.[3] The first layer of each model is an embedding layer, and the final layer performs dimensionality reduction; either a pooling operation over the entire output sequence, or outputs are simply flattened to a single dimensional sequence representation.

CNN The Convolutional Neural Network is intended as a simple baseline for convolutional architectures. It consists of two convolution layers with a max pooling operation after each. We use 64 filters with a kernel size of 5 for each layer and a pool size of 8.

TextCNN An implementation of the CNN for text classification proposed by Kim (2014). It is comprised of 5 parallel convolution layers with a max pooling operation after each. Convolutional layers use the same number of filters, 128, but with different kernel sizes in the range [1, 5]. The use of different kernel sizes is intended to capture the relationships between words at different positions within the input sentence. For dimensionality reduction the output of each pooling operation is concatenated before flattening into a single sequence vector.

DCNN The Dynamic Convolutional Neural Network implements the model proposed by Kalchbrenner, Grefenstette, and Blunsom (2014). The DCNN uses a sequence of 3 convolutional layers, each with 64 filters, the first layer uses a kernel size of 7 and the following layers a kernel size of 5. In contrast to the previous convolutional models the DCNN uses a *dynamic K-max pooling* operation after each convolution, which aims to capture a variable (per-layer) number of the most relevant features. Finally, dimensionality reduction is simply the flattened output of the last K-max pooling layer.

LSTM and GRU The Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997b), and Gated Recurrent Unit (GRU) (Cho et al., 2014b) are simple baselines for recurrent architectures. Both models follow the standard implementation and consist of one LSTM, or GRU, layer with 256 hidden units. We take the output at each timestep and apply average (LSTM), or max (GRU) pooling for dimensionality reduction.

RCNN The Recurrent Convolutional Neural Network is effectively a 'hybrid' of recurrent and convolutional paradigms. Our implementation is based on the model proposed by Lai et al. (2015), and has previously been applied to DA classification by Ribeiro, Ribeiro, and De Matos (2019). The RCNN consists of two recurrent layers, each with a dimensionality of 256. One processes the sequence forwards and the other in reverse. The output of these two layers is then concatenated with the original input embedding matrix, in the format *forwards-embeddings-backwards*. This concatenation 'sandwich' is then passed as input to a convolutional layer with 64 filters and a kernel size of 1. Finally, a max pooling operation is performed for dimensionality reduction.

[2] Appendix E contains a full summary of sentence encoder hyperparameters.

[3] In all cases we found these variants did not result in statistically significant improvements and thus we report those results in appendix F.1.

Language Models In addition to the supervised models, we test a selection of 10 pre-trained LMs as sentence encoders. Due to the variety of model architectures, training objectives and training data that was used to generate these LMs we omit them from the input sequence experiments. Differences in training data, for example, use of punctuation, the vocabulary, and so on, would make fair comparison between the models difficult. Further, and as previously stated, the input to these models is typically a tokenised sentence, where each token is mapped to an integer representation, and does not require the further step of mapping tokens to word embeddings. Therefore, we also do not include the LMs in our word embedding experiments. Instead, we use the standard input format and model parameters, defined by the original authors. The following provides a brief overview of the ten LMs, four of which are based on recurrent, or FFNN, and the remaining six on transformer architectures (Vaswani et al., 2017).

NNLM The Neural Network Language Model (Bengio et al., 2003).

mLSTM Character based multiplicative long short-term memory (mLSTM) language model proposed by Krause et al. (2016b), and applied to DA classification by Bothe et al. (2018a,b).

ELMo Embeddings from Language Models (Peters et al., 2018).

USE The Universal Sentence Encoder (Cer et al., 2018).

BERT Bi-directional Encoder Representations from Transformers (Devlin et al., 2019). We use the BERT-base version, we also tested the BERT-Large model but found it did not result in any significant improvements. This also allows us to maintain a similar number of layers and parameters as other transformer models we tested, for example RoBERTa.

RoBERTa A Robustly Optimised BERT Pretraining Approach (Liu et al., 2019). We use the RoBERTa-base version.

ConveRT Conversational Representations from Transformers (Henderson et al., 2020).

XLNET A generalised autoregressive pretraining method (Yang et al., 2019), that integrates ideas from Transformer-XL (Dai et al., 2019).

GPT-2 Generative Pretrained Transformer 2 (Radford et al., 2019).

DialoGPT The Dialogue Generative Pre-trained Transformer (Zhang et al., 2020a).

5.1.3 Input Sequence Representations

In the following sections we report our findings for each of the sequence representation experiments, that is, letter case and punctuation, vocabulary size and maximum sequence lengths. Due to differences in pre-training data, vocabulary, and so on, these were only carried out with the selection of supervised models, and not the LMs.

5.1.3.1 Letter Case and Punctuation

Here we present the results from both the letter case and punctuation experiments, that is, during pre-processing of the text, whether to convert all mixed-case letters to lower-case, and whether to keep, or remove, all punctuation marks. As mentioned in section 5.1.1 the Maptask corpus does not contain any words with capital letters or punctuation marks (apart from rare non-grammatical cases), and it was therefore not included in the

Table 5.2: Validation accuracy for the letter case and punctuation experiments.

	Punct		No Punct		Mixed-case		Lower-case	
Model	μ	σ	μ	σ	μ	σ	μ	σ
CNN	**74.46**	**0.23**	73.49	0.24	74.45	0.23	**74.73**	**0.19**
TextCNN	**75.61**	**0.25**	74.64	0.16	74.77	0.20	**75.33**	**0.23**
DCNN	**75.02**	**0.12**	73.96	0.18	74.13	0.20	**74.54**	**0.19**
RCNN	**74.06**	**0.45**	73.37	0.26	74.13	0.32	**74.46**	**0.29**
LSTM	**75.25**	**0.18**	74.34	0.21	74.84	0.15	**75.24**	**0.20**
GRU	**73.70**	**0.24**	72.91	0.33	**74.28**	**0.40**	74.26	0.17

letter case and punctuation experiments. Results for these two parameters, obtained on the SwDA corpus, are shown in table 5.2.

Regarding the use of punctuation, it can be seen that keeping punctuation marks results in an improvement in accuracy for all models, with a mean increase of 0.9%. BSR tests comparing the punctuation and no-punctuation groups for each model confirms that this difference is statistically significant in all cases ($P(Punct > NoPunct) = .99$).

Similarly, with the exception of the GRU model, lower-casing all letters also improves accuracy, though to a lesser degree, with a mean increase of 0.3%. However, this is not statistically significant when comparing the mixed-case and lower-case groups ($P(rope) \geq .92$), with the exception of TextCNN where $P(Lower > Mixed) = .66$, and therefore this parameter appears inconsequential for these models.

These results confirm some of the assumptions discussed in 2.2.3.1, and the results of Ortega et al. (2019). Firstly, that lowercasing words reduces unnecessary repetition in the vocabulary, which in turn may improve learned associations between word occurrence and DA label. Secondly, that certain punctuation marks may serve as strong indicators for the utterances DA, for example an interrogation mark indicating a question. Table 5.3 shows averaged, per-label, F1 scores for the best performing model (TextCNN), on the SwDA test set. We can see that, when punctuation is retained, F1 scores for all question-type DA labels is improved, apart from Declarative Wh-Question (qwˆd), which appears only once, and was not predicted. Though, collectively, the question-type labels only constitute 5.5% of all labels, and as such, this represents a minimal overall improvement. For the 3 most common DA labels, *Statement-non-opinion* (sd), *Acknowledge/backchannel* (b), and *Statement-opinion* (sv), which collectively make up 68.64% of all DA labels, the F1 score differs by +0.59%, +1.02%, and −2.81% respectively. This pattern is also repeated for most of the remaining labels, where small improvements are mitigated by negative changes elsewhere, resulting in the small overall accuracy increase that we have observed. This indicates that, (i) punctuation marks are beneficial in more circumstances than simply a question-type DA and interrogation mark relationship, and (ii) including punctuation can also reduce accuracy for specific label types.

5.1.3.2 Vocabulary Size

For each of the vocabulary size experiments only the most frequently occurring words, up to the current vocabulary size, were kept within the text. Less frequent words were considered OOV and replaced with the <unk> token. We test 16 different values in the range [500, 8000] with increments of 500, and [100, 1600] with increments of 100, for SwDA and Maptask respectively.

Table 5.3: TextCNN averaged F1 scores for the three most frequent labels (sd, b, sv), and all question-type labels in the SwDA test set (Tag-Question does not appear).

Dialogue Act	**Label**	**Count (%)**	**Punct**	**No Punct**
Statement-non-opinion	sd	1317 (32.3%)	**79.93**	79.34
Acknowledge (Backchannel)	b	764 (18.73%)	**83.55**	82.53
Statement-opinion	sv	718 (17.61%)	62.43	**65.24**
Yes-No-Question	qy	84 (2.06%)	**73.49**	71.43
Wh-Question	qw	55 (1.35%)	**71.92**	67.24
Declarative Yes-No-Question	qy^d	36 (0.88%)	**23.94**	21.62
Backchannel in Question Form	bh	21 (0.51%)	**64.64**	50.21
Open-Question	qo	16 (0.39%)	**73.21**	70.05
Rhetorical-Question	qh	21 (0.29%)	**35.06**	32.75
Declarative Wh-Question	qw^d	1 (0.02%)	0.0	0.0

Table 5.4: Vocabulary size which produced the best validation accuracy for each model on the SwDA and Maptask data.

	SwDA			**Maptask**		
Model	**Vocab Size**	μ	σ	**Vocab Size**	μ	σ
CNN	2500	74.50	0.24	500	57.85	0.18
TextCNN	5500	75.61	0.25	200	56.60	0.33
DCNN	7500	75.15	0.13	600	56.08	0.32
RCNN	7500	74.36	0.49	1100	58.28	0.20
LSTM	7000	75.25	0.18	800	55.88	0.38
GRU	8000	73.89	0.36	1200	58.87	0.24

As shown in table 5.4, with the exception of the GRU applied to the SwDA, the best performance was consistently achieved using a smaller vocabulary than the largest value tested. Figure 5.1 displays Maptask results for the full range of vocabulary sizes and models. Vertical lines indicate the average frequency of word occurrence for a given range, for example, the 200-300 most frequent words appear ~71 times within the Maptask training data. For both SwDA and Maptask, increasing vocabulary sizes steadily improves accuracy up to ~4k, or ~400, words respectively, beyond which further increases yield little to no improvement.

This observation is supported by BSR analysis comparing all vocabulary size combinations, which shows that, once a threshold is reached, further increase of vocabulary size does not result in a statistically significant difference in performance. For Maptask, the threshold is 400 words or less for all models, and for SwDA 2.5k words or less; except for the RCNN, where a clear threshold is higher, at 4k words. If we explore these thresholds in terms of frequency of word occurrences, the most frequent 2.5k, and 400, words account for 95.9% and 94.7% of all words in the respective SwDA and Maptask training data. The remaining less-frequent words appear, at most, 22.5, or 28.3 times, within the training data, typically much less.

These results suggest that words which appear below a certain frequency within the data do not contribute to overall performance, and that word frequency is correlated with

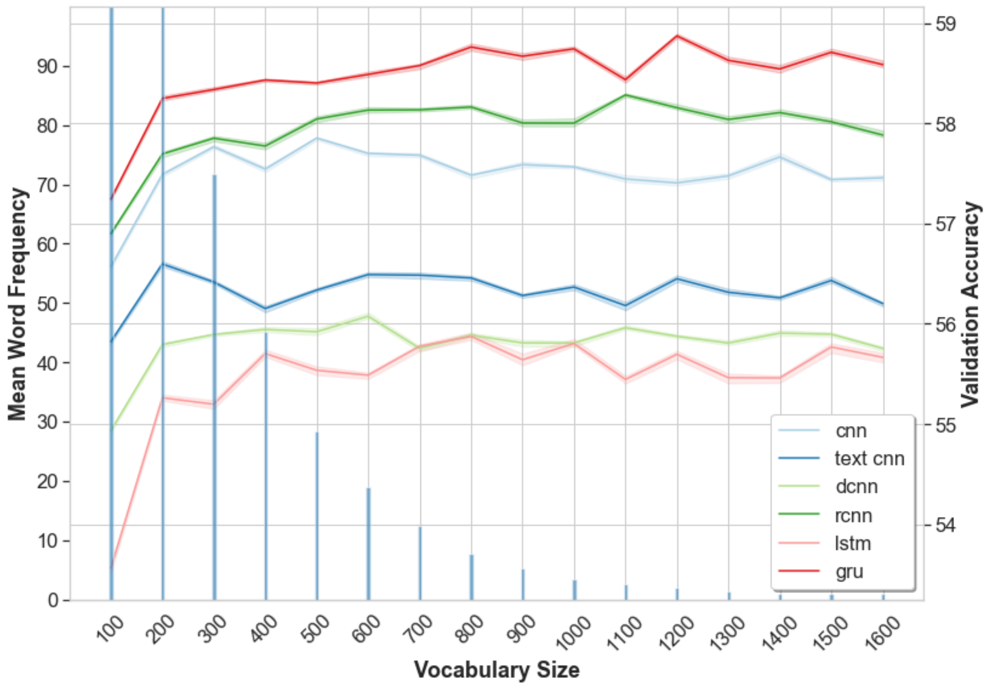


Figure 5.1: Maptask validation accuracy for all supervised models with different vocabulary sizes. Vertical lines are the mean word occurrence, per-vocabulary range (up to 100 words the mean frequency = 1268, and for 100 to 200 words the mean frequency = 162).

the observed performance thresholds. Either because of their sparsity within the data, or because they are not meaningfully related to any DA. On the SwDA data, the 2.5k threshold coincides with the optimal 1-2k word vocabulary reported by Cerisara, Král, and Lenc (2017). Though, apart from the CNN, we did not observe any degradation in performance from increasing vocabulary size further. Certainly, it does not appear that using large vocabularies, typically 10k or 20k words for SwDA (Li et al., 2019b; Raheja and Tetreault, 2019; Chen et al., 2018; Kumar et al., 2017; Ji, Haffari, and Eisenstein, 2016; Lee and Dernoncourt, 2016), is necessary or beneficial for the DA classification task. While larger vocabularies do not create significant additional storage or computational requirements, it may be more efficient to remove very infrequently occurring words. Thus, removing a large number of words from the vocabulary which do not contribute to model performance.

5.1.3.3 Sequence Length

To explore the effect of varying the input sequence lengths, all utterances were truncated, or padded, to a fixed number of word tokens before training. Sequences are padded with a <pad> token up to the current maximum sequence length. For both SwDA and Maptask we test values in the range [5, 50], in increments of 5. Table 5.5, shows the sequence length which produced the best performance for each model. Notably, in all cases, the best validation accuracy was obtained using a sequence length that is shorter than the largest value tested; which in turn, is less than half of the longest utterances in both corpora, 133 and 115 words, for SwDA and Maptask respectively.

Figure 5.2 shows SwDA results for the full range of sequence lengths and models. Vertical lines indicate the cumulative sum of utterances, up to a given length, within the

Table 5.5: Input sequence length which produced the best validation accuracy for each model on the SwDA and Maptask data.

	SwDA			**Maptask**		
Model	**Seq Length**	μ	σ	**Seq Length**	μ	σ
CNN	45	74.43	0.17	25	57.49	0.26
TextCNN	25	75.63	0.24	40	56.40	0.33
DCNN	30	75.10	0.23	25	56.14	0.24
RCNN	40	74.50	0.28	25	58.13	0.26
LSTM	25	75.35	0.16	10	57.98	0.26
GRU	25	73.94	0.27	30	58.68	0.25

training data. It can be observed that, increasing the number of tokens steadily improves performance up to a point, beyond which we see no further improvement. On both SwDA and Maptask performance levels off at sequence lengths of ~20-25 tokens.

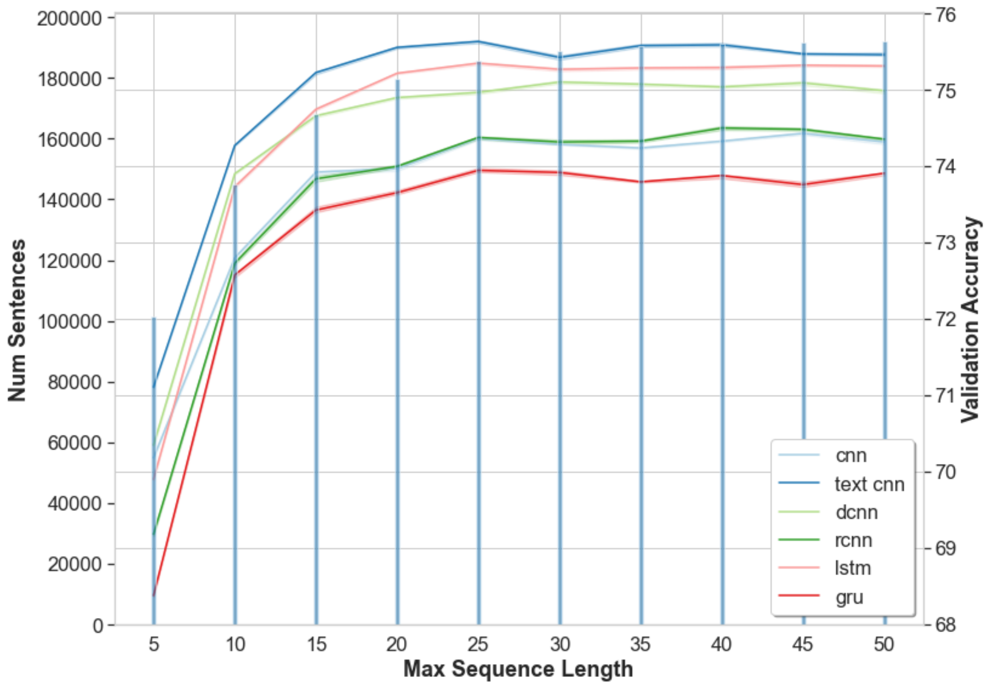


Figure 5.2: SwDA validation accuracy for all supervised models with different sequence lengths. Vertical lines are the cumulative sum of utterances up to a given length.

Again, these observations are supported by BSR analysis comparing all sequence length combinations, which shows that, for SwDA there is no significant difference in performance for sequence lengths greater than 25 tokens, and for Maptask the threshold is 15 tokens; with the exception of the LSTM where performance steadily *decreases*, and the TextCNN where there is no clear threshold. Examining these thresholds in terms of the frequency of utterances within the training data, 96.5% of all utterances in SwDA are $\leq$ 25 words, while for Maptask 92.6% are $\leq$ 15 words. This is also clearly reflected in the cumulative sum of utterance lengths shown in figure 5.2. The values closely match the shape of the accuracy curves, steadily increasing before starting to level off at the 20-25 token threshold.

Our results, and the stated thresholds, for both datasets strongly support the work of Cerisara, Král, and Lenc (2017), who found that 15-20 tokens was optimal on the SwDA data. Wan et al. (2018), also reported their best result was achieved using sequence lengths of 40, which coincides with the sequence lengths that produced the best (though not statistically significant) results for some of our models. Additionally, our thresholds for both datasets, and the results reported by Cerisara, Král, and Lenc (2017), can be considered in terms of the average number of words in an English sentence. According to Cutts (2013), and Dubay (2004), the average number of words is 15-20 per sentence. While Deveci (2019), in a survey of research articles, found the average to be 24.2 words. Thus, it should perhaps not be surprising to find that a significant proportion of utterances in our datasets are of similar, or smaller, lengths.

Certainly, it seems that, similar to word occurrences, utterances above a certain length appear so infrequently that they do not contribute to overall performance. For example, in the SwDA training data, the number of utterances longer than 50 tokens is 342 (0.18%), and for Maptask it is just 11 (0.05%). Therefore, padding sequences up to the maximum utterance length does not produce any benefit, and in some cases, it may actually reduce performance (Cho et al., 2014a). Additionally, padding sequences results in a significant increase in storage and computational effort. Instead, appropriate values should be chosen based on the distribution of utterance lengths within the data, and where possible, padding mini-batches according to the longest utterance within the batch.

5.1.3.4 Input Sequences Comparison

The vocabulary size and sequence length experiments were conducted while keeping all other parameters fixed at their default values. This leads to the possibility that using both smaller vocabularies and shorter sequence lengths, in combination, may result in too much information loss and harm performance. To explore this hypothesis, we conducted further experiments with 3 combinations of, 'small', 'medium', and 'large', vocabulary sizes and sequence lengths. For SwDA we used vocabularies of 2.5k, 5k, and 10k words, and for Maptask 400, 800, and 1.7k words. Each of these was combined with a respective sequence length of 25, 50, and 128 (SwDA), or 115 (Maptask). We can see from Table 5.6 that in most cases models achieved higher accuracy with small vocabularies and sequence lengths. BSR analysis reveals that, for the SwDA corpus, there is no significant difference between the groups for any model ($P(rope) \geq .99$). Indeed, for the two models which achieved higher accuracy with the large group, RCNN and LSTM, the difference between the small and large groups mean accuracies is just 0.26% and 0.21% respectively. For Maptask, analysis only shows statistically significant results for the LSTM, which obtained higher accuracy with the small and medium groups ($P(Small/Medium > Large) = 1$). Again, for the two models which favoured the large group, TextCNN and RCNN, the difference between the small and large groups mean accuracies is 0.09% and 0.25% respectively. Thus, we can conclude that reducing both vocabulary size and sequence length does not negatively impact performance.

5.1.4 Word Embeddings

Throughout our word embeddings experiments we test 5 different pre-trained word embeddings; Word2Vec, GloVe, FastText, Dependency, and Numberbatch. Each of these is tested at 5 different dimensions in the range [100, 300], at increments of 50. Table 5.7 shows the combination of embedding type and dimension which produced the best accuracy for each model.

Table 5.6: Vocabulary size and sequence length group which produced the best validation accuracy for each model on the SwDA and Maptask data.

	SwDA				Maptask			
Model	**Vocab**	**Seq Len**	μ	σ	**Vocab**	**Seq Len**	μ	σ
CNN	5000	50	74.86	0.24	400	25	57.72	0.23
TextCNN	2500	25	75.41	0.09	1700	115	56.33	0.28
DCNN	5000	50	74.68	0.24	800	50	56.11	0.26
RCNN	10000	128	74.51	0.17	1700	115	58.05	0.23
LSTM	10000	128	75.35	0.15	400	25	57.47	0.25
GRU	5000	50	74.37	0.17	800	50	58.59	0.31

Table 5.7: Embedding type and dimension which produced the best validation accuracy for each model on the SwDA and Maptask data.

	SwDA				Maptask			
Model	**Embedding**	**Dim**	μ	σ	**Embedding**	**Dim**	μ	σ
CNN	Numberbatch	100	74.59	0.16	FastText	300	57.88	0.20
TextCNN	Numberbatch	300	76.01	0.12	FastText	300	58.97	0.24
DCNN	FastText	200	75.66	0.15	FastText	250	57.37	0.31
RCNN	FastText	200	75.06	0.27	Dependency	100	59.45	0.24
LSTM	GloVe	300	75.57	0.21	GloVe	300	57.93	0.28
GRU	FastText	100	74.87	0.28	Dependency	200	59.46	0.23

It can be seen that there is no clearly optimal embedding type and dimension combination. Instead, it seems to be dependent on a particular task, or model, in most cases. Though, FastText does more consistently – in 50% of cases – improve performance. It is also worth noting that Word2Vec frequently resulted in poorer accuracy and therefore does not appear in table 5.7 at all.

We analyse these results further by conducting multiple BSR tests comparing different dimensions for each embedding type, and the different embedding types to one another. For SwDA, in most cases we see no significant difference in embedding dimension, only the LSTM with Word2Vec 100 vs 250 dimensions ($P(100 > 200) = .82$). On Maptask the DCNN shows significant difference in dimensions for Numberbatch, Word2Vec, and Dependency, and the LSTM for all five embedding types. In both cases, larger dimensions correlate with improved performance. When comparing embedding types, for SwDA only the DCNN with FastText shows any significant results ($P(Fasttext > Dependency/Numberbatch/Word2Vec) = 1$). To illustrate this observation figure 5.3 shows the results obtained on SwDA with the DCNN model. We can see that FastText and GloVe resulted in a clear improvement in performance over the remaining embedding types. This is also true for the DCNN and LSTM applied to the Maptask data, where GloVe and FastText both show statistically significant improvements over the other embedding types. Interestingly, the optimal embedding type and dimension for these two models is consistent across the two datasets, FastText 200-250 for the DCNN, and GloVe 300 for the LSTM. For the remaining models the picture is less clear, with GloVe consistently outperformed by other embbeddings. For the TextCNN $P(FastText/Numberbatch > GloVe) \geq .83$;

the RCNN $P(Dependency/Numberbatch > GloVe) \geq .89$; the GRU $P(Dependency > GloVe) = .86$; and for the CNN we see no significant results. As we observed in Section 2.2.3.2, in most cases the differences between embedding type and dimension is very small, and in our experiments, it is often not statistically significant. However, for some models determining an optimal embedding type is more impactful than simply testing different dimensionalities of a single arbitrarily chosen embedding type. Additionally, Word2vec consistently underperformed on all models, and both datasets, which suggests it is not suitable for this task, a conclusion that was also reached by Cerisara, Král, and Lenc (2017). Instead, we suggest using FastText in the first instance as this embedding most often resulted in a significant performance increase in our experiments.

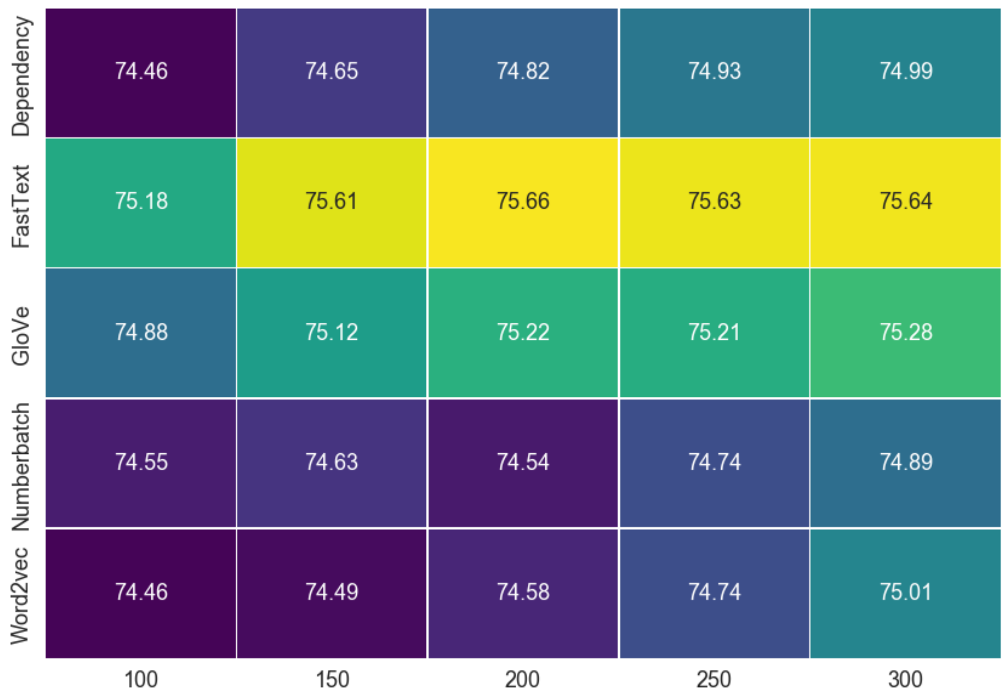


Figure 5.3: The DCNN model's SwDA validation accuracies for all embedding type and dimension combinations.

5.1.5 Supervised Models

Here we present the final test set results for all of our supervised models. In each case the model was trained and tested using the parameters (Vocabulary size, Sequence length, etc) determined by our previous experiments. Table 5.8 shows results for both the SwDA and Maptask data. The TextCNN performs well on both datasets, outperforming the other convolutional models. For the recurrent models the results are a little more variable. The LSTM achieves higher test accuracy and F1 score than the GRU on the SwDA, while for Maptask the reverse is true and by a larger margin.

Rigorous comparison with previous work is challenging due to differences in text preprocessing and other parameters. Additionally, most recent studies do not report results for *single-sentence* classification, that is, DA classification without context/discourse information. Nevertheless, it seems these results are on the high end of what might be expected for single-sentence DA classification. Papalampidi, Iosif, and Potamianos (2017), included salient key words as extra features in their experiment and report test set accuracy of

73.8% (though it is not clear if this result is the single best run, or an average of several), and Bothe et al. (2018b) reported 73.96% using the mLSTM language model. Yet, our TextCNN and LSTM attain similar accuracies with only pre-trained word embeddings. Our best models also outperform all of the other single-sentence results we were able to find within the literature Cerisara, Král, and Lenc (2017) and Shen and Lee (2016), and Lee and Dernoncourt (2016), who report 70.4%, 69.3%, and 67% respectively. They are also competative with, or higher than, several studies which also include context or discourse information (Cerisara, Král, and Lenc, 2017; Ortega and Vu, 2017; Lee and Dernoncourt, 2016; Shen and Lee, 2016; Kalchbrenner and Blunsom, 2013), though they are far from the best contemporary approaches in that regard. However, these results do indicate that we may be at, or near, the limit of what these kinds of standard model architectures can attain for single-sentence classification. The difference between the best and worst performing model on the SwDA test set is just 1.22%, and for Maptask it is 1.49%. Though, small differences in accuracy are perhaps more noteworthy on the DA classification task than other classification problems. Of the studies directly comparable to ours (models that do not consider surrounding sentences, and that use the same training and test datasets), the difference between the lowest, 67% (Lee and Dernoncourt, 2016), and highest, 73.96% (Bothe et al., 2018b), is just 6.96%. Further, the results of Bothe et al. (2018b), represent only a 2.96% increase over those reported by Stolcke et al. (2000), nearly two decades earlier. Thus, while some reported increases are small, parameters that produce consistent improvements, such as keeping punctuation, are meaningful for this problem. This supports the need for more sophisticated methods of sentence encoding, such as that of contextual LMs.

Table 5.8: Test set accuracy for each of the supervised models on the SwDA and Maptask data.

SwDA			Maptask		
Model	μ	σ	**Model**	μ	σ
CNN	71.16	0.64	CNN	59.22	1.01
TextCNN	**73.36**	**0.34**	TextCNN	60.29	0.26
DCNN	72.87	0.53	DCNN	59.96	0.58
RCNN	72.44	0.41	RCNN	60.43	0.62
LSTM	73.06	0.37	LSTM	59.86	0.62
GRU	72.27	0.74	GRU	**61.12**	**0.64**

5.1.6 Language Models

All 10 LMs were trained with the same parameters and default values for vocabulary size, sequence length, letter case, and punctuation. Results for the pre-trained LMs applied to the SwDa and Maptask data are shown in Table 5.9.

Starting with the SwDA corpus, the models based on transformer architectures all resulted in an improvement in accuracy over our best performing supervised model, TextCNN. Ranging from +0.95% with ConveRT, to +2.86% for RoBERTA, and in all cases this is statistically significant ($P(LM > TextCNN) \geq .98$). The remaining models show either negligible improvements or, for ELMo and NNLM, lower but statistically equivalent test set accuracy ($P(rope) \geq .91$). Both BERT and RoBERTa reach test set accuracies that outperform many of the contextual models from previous studies, for example, Papalampidi, Iosif, and Potamianos (2017), and Tran, Haffari, and Zukerman (2017). They also begin to approach some of the current best contextual models, such as those reported by Bothe et al. (2018a), 77.42%, Li et al. (2019b), 78.3% and Ribeiro, Ribeiro, and De Matos (2019), 79.11%.[4]

On the Maptask corpus the LMs fared much worse. Only three managed to improve upon our best supervised model, GRU, and in most cases were only marginally better than the 2nd and 3rd best. Again though, BERT and RoBERTa improve upon the supervised model by +1.74%, and +1.46%, respectively ($P(LM > GRU) \geq .99$). Here comparison with previous work is more difficult as the Maptask corpus is less studied. Still, both models are comparable with the 63.3% accuracy, that is also achieved with a contextual model (Tran, Zukerman, and Haffari, 2017). The relatively poor results for the Maptask data is somewhat surprising. For some of the transformer-based models this may be due to the smaller dataset, and the comparatively larger gains in performance on the SwDA corpus would seem to support that assumption. However, it does contradict the 'few-shot-learning', task-specific fine-tuning, paradigm that has led to much of the success of these models (Wang et al., 2020). Nevertheless, both BERT and RoBERTa achieve a significant and consistent improvement on both corpora. Thus we can conclude that – despite an increase in computational effort, training time, and storage requirements – the contextual sentence representations are superior to those of supervised models and pre-trained word embeddings (Fiok et al., 2020).

Table 5.9: Validation set accuracy, and test set accuracy for each of the pre-trained LMs on the SwDA and Maptask data.

	SwDA				**Maptask**			
	Validation		**Test**		**Validation**		**Test**	
Model	μ	σ	μ	σ	μ	σ	μ	σ
BERT	76.87	0.24	76.07	0.42	61.12	0.44	**62.91**	**0.32**
RoBERTa	**78.17**	**0.33**	**76.22**	**0.56**	**61.18**	**0.40**	62.63	0.24
GPT2	77.47	0.44	75.16	0.62	60.18	0.28	61.04	0.98
DialoGPT	77.82	0.44	75.30	0.37	57.04	1.83	56.70	1.85
XLNet	78.15	0.46	75.88	0.45	61.21	0.51	61.61	0.78
ConveRT	76.54	0.22	74.31	0.34	58.16	0.21	60.94	0.63
ELMo	76.00	0.20	73.19	0.53	58.34	0.21	60.44	0.35
USE	76.20	0.15	73.51	0.38	59.35	0.22	60.67	0.56
mLSTM	75.78	0.25	73.48	0.61	58.50	0.27	60.79	0.63
NNLM	73.44	0.07	70.12	0.26	52.44	0.18	56.65	0.24

5.1.7 Sentence Encoding Summary

Throughout this section we have explored numerous factors which may affect the task of sentence encoding for the purpose of DA classification. We first considered various aspects of text pre-processing and representation, which are often overlooked or underreported within the literature, such as whether to keep, or remove punctuation, selecting vocabulary size, input sequence length and word embeddings. Each of these was assessed on the SwDA and Maptask corpora, using a selection of 6 supervised models, that are

[4] Several studies have reported higher accuracies than these (Ribeiro, Ribeiro, and De Matos, 2019; Chen et al., 2018), however they also include *future* utterances, or 'gold-standard' labels as context information and we have therefore omitted them.

intended to be representative of the common architectures applied to DA classification task. Finally, we also applied a selection of 10 pre-trained LMs, including transformer based contextual models, such as BERT and XLNET, and draw comparisons between the supervised approaches. To the best of our knowledge this is also the first time most of these comparatively new LMs have been applied to the DA classification problem.

Our results show that the text pre-processing parameters we investigated should not be arbitrarily chosen, because they can produce a notable effect on model performance. Firstly, keeping punctuation always improves accuracy when compared to the alternative options, while converting all words to lower-case is less impactful. Interestingly, keeping punctuation appears beneficial for several of the most common DA labels within the SwDA corpus, even those that are not a type of question, where intuitively one might expect an interrogation mark to strongly correlate with a question type DA label.

Considering the selection of vocabulary size, we found that using smaller vocabularies was beneficial in most cases. Certainly, our results show that the number of words, for the best performing models, was $1/4$ to $3/4$ of the largest vocabulary size tested, which equates to around $1/10^{th}$ of the corpora's full vocabulary. Additionally, increasing vocabulary sizes results in diminishing, or detrimental, returns in performance. These values are much lower than those typically used in most DA classification studies, for example, 10k or 20k words for SwDA (Li et al., 2019a; Chen et al., 2018; Kumar et al., 2017; Ji, Haffari, and Eisenstein, 2016; Lee and Dernoncourt, 2016). Instead, using smaller vocabularies could prune out highly infrequent words which are unlikely to be relevant to the DA classification task, and reduce noise within the data.

Similarly, for input sequence lengths, we showed that beyond a certain threshold using longer sequences has no significant impact on classification accuracy. For SwDA the threshold is 25 words, and for Maptask 15. These thresholds, and the optimal sequence lengths for all models, on both datasets, were shorter than the maximum sequence length we tested (50 tokens), which in turn is <50% of the longest utterances in either corpora. Thus, we conclude that padding sequences to lengths nearer that of the longest utterances in the data is a waste of computational effort and storage. We also found that calculating the cumulative frequency of utterance lengths within the data produced a reasonable approximation of the resulting accuracy curves within our experiments. When the cumulative frequency began to level off, so too did the model's accuracy. This technique could be used to select a viable sequence length which minimises both information loss (through truncation), and the number of unnecessary padding tokens. It should also be noted that when using smaller vocabularies and sequence lengths in combination we observed no significant difference when compared to larger valued combinations. Certainly, in most cases, including the best performing models, higher accuracies were achieved when using a combination of smaller values.

Results for our word embedding experiments were perhaps less conclusive. Of the pre-trained embeddings we tested, none was shown to be clearly optimal across both datasets and models. It seems that the selection of embedding is highly dependent on both model and data, though the overall impact of this choice is often negligible. This is supported by our statistical analysis which showed that, when comparing embedding type and dimension combinations, we mostly observe a statistically significant difference in performance when comparing different embedding *types*. Thus, while choice of embedding may result in a small (and likely statistically non-significant) effect on performance, the selection of embedding type tends to be more impactful than the dimension.

Regarding the selection of models, we found that performance was often inconsistent when applying the same, or similar, architecture to different datasets. Most notably, the use of bi-directional or multi-layered recurrent architecture, or the addition of attention layers – which are so frequently applied to DA classification – often did not yield any improvement over their simpler baseline version. These inconsistencies suggest that these architectural additions should be accompanied by appropriate ablation experiments, to determine their true impact on performance. And further, applying a single model, or small variations thereof, to a single dataset is not enough to draw firm conclusions on its generalisable performance. This is similarly true for the selection of LMs we tested. Where, even amongst the transformer-based models, on the smaller, sparser, Maptask data some models failed to improve upon our best performing supervised model. However, on both datasets, BERT and RoBERTa represent a significant improvement in sentence encoding for DA classification and are therefore the current best choice for applying to the sentence encoding task.

5.2 Context Encoding for Dialogue Act Classification

In this section we present the results and analysis for each of our context encoding experiments.[5] We continue to utilise the SwDA and Maptask corpora, with appropriate selection of sequence representation and word embedding parameters, as determined by our sentence encoding experiments. Specifically, we retain punctuation and lower-case all words; for supervised models we use a sequence length of 50, and for LMs 100; since larger vocabularies were not shown to reduce performance the vocabulary size is maintained at 10k words and 1.7k words for SwDA and Maptask respectively. Because annotation of the CAMS-KVRET corpus was conducted in parallel to our sentence encoding experiments we are now able to include that corpus within our context encoding experiments. For CAMS-KVRET we use the same parameters discussed above, with a vocabulary size of 1.9k words.

5.2.1 Context-encoders

Here we discuss the core dialogue context encoder and the speaker or label encoders as separate components, since the latter two are optional and work on different forms of input. As with the sentence encoders, to tune hyperparameters we use a Bayes search algorithm with a maximum of 100 parameter combinations, over 5 epochs.[6] We also investigated different encoding architectures, such as recurrent layers (GRU or LSTM), bi-directionality, and attention mechanisms. However, again, in all cases we found these variants did not result in statistically significant improvements and thus we report those results in appendix F.2.

Dialogue Context For our dialogue context encoder we experiment with two architectures, one based on recurrent layers, the other convolutional. The former consists of a single LSTM layer with either 256 or 512 hidden units, dependant on the size of the sentence encoders' output dimension. We consider the hidden state at the last timestep as the dialogue segment encoding. Our convolutional encoder consists of three layers with a max pooling operation after each. Convolutional layers use the same number of filters, either 32 or 128 depending on the sentence encoders' output dimension, but with different kernel sizes of 6, 4, and 2. Finally, for dimensionality reduction we apply a max pooling operation to produce the encoded segment vector $\boldsymbol{d}$.

Context Speakers and Labels For both the speaker and label encoders we experiment with three different architectures: Recurrent, FFNN, and 'Flat'. Since speaker and label

[5] All code, data, and accompanying analysis for the context encoding experiments is available at: github.com/NathanDuran/Context-Encoding-for-DA-Classification.

[6] Appendix E contains a full summary of context encoder hyperparameters.

inputs are functionally equivalent (one-hot vectors), and either can be included independently of the other, there are no architectural differences between our speaker and label encoding components. Both the recurrent and FFNN methods process each input one-hot vector in turn (m speakers or $m-1$ labels), and in the latter case we apply the same FFNN to each timestep. For the flat approach we simply reshape the input matrix to a single vector of length $\boldsymbol{m} \times |\boldsymbol{S}|$ for speakers and $\boldsymbol{m\text{-}1} \times |\boldsymbol{L}|$ for labels.

We further provide the option for the dialogue segments' speaker and label encodings to be concatenated with sentence encodings prior to being processed by the dialogue context encoder, or concatenated with the dialogue segment encoding itself. In other words, either the speakers and labels are processed *sequentially* by the dialogue context encoder, at the same time as the sentences, or they are joined with the encoded dialogue segment ($\boldsymbol{d}$) as a *summary* of its speakers and labels. In the sequential case, for recurrent and FFNN encoders we concatenate the output at each timestep with their respective sentence encodings. For the summary, we simply flatten the speaker or label encodings for each timestep into a single vector, before concatenating with $\boldsymbol{d}$.

Sequence Classifiers As discussed in 2.2.3.5, while FFNN are the most prevalent classification approach, several recent studies have shown the utility of sequential classifiers. That is, a classification architecture that is capable of producing label predictions for each sentence in the input segment m, such as a CRF (Li et al., 2019b; Ortega et al., 2019; Raheja and Tetreault, 2019; Chen et al., 2018; Kumar et al., 2017), or Seq2seq model (Colombo et al., 2020). Thus, to produce our final contextual classifier model, we also experiment with the addition of a CRF layer and Seq2seq architectures similar to that of Colombo et al. (2020). Here we additionally report the sequence accuracy – the predictions for each label in the input dialogue segment – along with the current classification target.

5.2.2 Context Utterances

Throughout our context utterance experiments we use two supervised models (TextCNN and LSTM), and two LMs (BERT and RoBERTa), as sentence encoders. We selected the former because they consistently performed well throughout our sentence encoding experiments, and represent two distinct paradigms, convolutional and recurrent. The latter we selected because they were the best performing LMs of those tested.

For reference, and to determine a set of non-contextual single-sentence baselines for CAMS-KVRET, we re-applied our selection of four sentence encoders to that corpus. Baseline results for validation and test set accuracy are shown in table 5.10. Promisingly, all models achieve a strong baseline of ~87-89% without any contextual information. This indicates that our corpus is well annotated, and suited for our purpose, such that there is a strong correlation between the semantic content of utterances and their associated DA.

Table 5.10: Baseline validation and test set accuracy for each of the sentence encoder models on the CAMS-KVRET corpus.

	Validation		**Test**	
Model	μ	σ	μ	σ
BERT	**89.0**	**0.43**	**88.66**	**0.47**
RoBERTa	88.97	0.54	88.46	0.52
LSTM	87.74	0.38	87.09	0.43
TextCNN	86.85	0.57	86.85	0.49

In the following we examine varying numbers of utterances that make up the input dialogue segment m. We begin by looking only at previous, or historical, utterances (5.2.2.1). That is, the $m-1$ utterance that preceded the current target for classification within the dialogue. We then explore combinations of previous *and* future utterances, to estimate an upper limit to the benefit that such contextual information can provide (5.2.2.2). Evidently including future utterances is not appropriate for certain tasks, such as dialogue management systems, where future information is not available. However, there may be other applications for which the full dialogue is available, such as post-hoc analysis of the dialogue flow or structure (Bifis et al., 2021; Lee et al., 2019). Moreover, several DA classification studies have included such future information (Ribeiro, Ribeiro, and De Matos, 2019; Chen et al., 2018), and therefore we consider it worthy of further investigation. It should be noted that, due to GPU memory constraints, for the transformer based models, BERT and RoBERTa, we were unable to process sequences of >6 utterances. Therefore, we limit the number of contextual utterances to 5, giving a maximum dialogue segment size of 6. Similarly, due to immense processing times, particularly for the SwDA corpus where each run may take upwards of 10 hours, in places we limit the number of models or corpora included within our experiments.

5.2.2.1 Previous Context

Investigating the influence of previous context, we tested values in the range [1, 5] for the number of utterances to include. Results for Maptask and CAMS-KVRET are shown in table 5.11, and results for SwDA are shown in table 5.12. Due to the significant processing times mentioned earlier, for the SwDA corpus we only test four contextual models, two using BERT as a sentence encoder and two using an LSTM.

Table 5.11: Mean validation set accuracy for each context encoder model, using previous context utterances in the range [1, 5], on the Maptask and CAMS-KVRET corpora.

	Maptask					**KVRET**				
Model	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
BERT-CNN	66.17	67.2	**67.21**	67.11	66.84	92.56	92.55	92.92	93.18	**93.35**
BERT-LSTM	66.59	66.89	67.06	67.13	**67.36**	93.0	92.95	93.49	93.26	**93.64**
RoBERTa-CNN	66.86	**67.74**	67.44	67.59	66.75	92.74	92.57	**93.22**	93.19	92.91
RoBERTa-LSTM	66.77	67.27	67.18	67.54	**67.93**	93.21	93.41	93.66	93.4	**93.83**
LSTM-CNN	65.38	66.17	**66.58**	66.38	66.52	91.22	91.37	92.05	92.13	**92.32**
LSTM-LSTM	65.73	66.98	**67.12**	67.0	67.03	91.46	91.41	92.11	**92.45**	92.3
TextCNN-CNN	63.46	63.53	63.25	**63.67**	63.63	90.76	90.74	91.0	90.87	**91.05**
TextCNN-LSTM	64.16	65.51	65.67	65.21	**65.78**	91.1	91.63	92.3	**92.48**	92.43

Tables 5.11 and 5.12 show that, in all cases, including just a single context utterance results in significant improvements over the single-sentence baselines ($P(1\text{-}context > No\text{-}context) = 1$). For Maptask this represents an average increase of ~5.9%, for CAMS-KVRET ~4.65%, and for SwDA ~2.7%. If we instead consider the number of context utterances that resulted in the best performance, this is increased by ~1% for Maptask and CAMS-KVRET. Additionally, we can see that in most cases the optimal number of utterances is ≥ 2, and more often 4 to 5, particularly for CAMS-KVRET. However, these optimal numbers are typically not a substantial improvement over similar values. To explore this observation we conduct multiple BSR tests comparing all combinations of context utterances:

- For Maptask, we see a threshold at ~2 utterances beyond which there is no statistically significant improvement; with the exception of both BERT-LSTM and

TextCNN-CNN where we see no significant results at all, and RoBERTa-LSTM where we only see significant results when comparing 1 to 5 utterances ($P(5 > 1) = .97$).

- On CAMS-KVRET the threshold is higher, at ~3 utterances for both LSTM models and TextCNN-LSTM. However, for all transformer based models, and the TextCNN-CNN, we see no statistically significant results.
- Finally, for SwDA in most cases results are either statistically equivalent or inconclusive, with two marginal exceptions, for BERT-LSTM $P(5 > 1) = .79$, and for LSTM-LSTM $P(2 > 5) = .76$.

Therefore, while including a single context utterance always improves upon single-sentence methods, and larger context windows tend to result in higher accuracy, in general, including more that 2 context utterances does not result in statistically significant improvements. These results also concur with those of previous studies which have shown that between 2 and 4 context utterances yield the best performance (He et al., 2021; Ahmadvand, Choi, and Agichtein, 2019; Ortega et al., 2019; Bothe et al., 2018b; Ortega and Vu, 2017; Papalampidi, Iosif, and Potamianos, 2017; Lee and Dernoncourt, 2016).

Table 5.12: Mean SwDA validation set accuracy for each context encoder model, using previous context utterances in the range [1, 5].

	SwDA				
Model	**1**	**2**	**3**	**4**	**5**
BERT-CNN	80.2	80.43	80.27	80.47	**80.57**
BERT-LSTM	80.09	80.19	80.09	79.89	**80.61**
LSTM-CNN	**77.62**	77.44	77.49	77.26	77.28
LSTM-LSTM	78.07	**78.3**	77.9	78.15	77.57

We also conducted BSR tests comparing each model to one another, using results from the number of context utterances that resulted in the best performance:

- Comparing the four transformer based models to the two LSTM, and two TextCNN models; on SwDA both BERT models show a statistically significant improvement over the LSTMs ($P(BERT > LSTM) \geq .99$); for CAMS-KVRET all four show significant improvement over the LSTMs and TextCNNs ($P(BERT/RoBERTa > LSTM/TextCNN) \geq .88$); on Maptask both BERT models are equivalent to the LSTM-LSTM ($P(rope) \geq .83$), and with the exception of RoBERTa-CNN compared to LSTM-LSTM, all four tranformer based models show significant improvement over the remaining LSTM and TextCNN models ($P(BERT/RoBERTa > LSTM/TextCNN) \geq .81$).
- If we now compare the CNN and LSTM variants of both transformer models, for SwDA and Maptask they are equivalent ($P(rope) \geq .81$), though on CAMS-KVRET this is inconclusive ($P(rope) \geq .71$). Similarly, when comparing all BERT and RoBERTa models, we see no statistically significant results in any case.

Thus we can conclude that, unsurprisingly our transformer based models outperformed the LSTM and TextCNN in most cases. Additionally, we see no noteworthy differences between either BERT or RoBERTa and their contextual LSTM or CNN models (RoBERTa models tended to outperform BERT, but only by a small margin), though in all cases the LSTM did perform slightly better.

5.2.2.2 Full Context

Following our investigation of previous context utterances we explore the impact of including future utterances as well. We tested all combinations of previous and future utterances that sum to 5, and due to the long processing times previously mentioned, only gather results for the Maptask corpus. Table 5.13 shows which combinations achieved the best validation set accuracy for each model.

Table 5.13: Best combination of previous and future context utterances, and validation set accuracy, for each context encoder model on the Maptask corpus.

Model	**Previous**	**Future**	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
BERT-CNN	2	2	68.39	0.47
BERT-LSTM	2	1	67.59	0.53
RoBERTa-CNN	1	1	**68.47**	**0.77**
RoBERTa-LSTM	5	0	**67.93**	**0.57**
LSTM-CNN	3	1	67.82	0.4
LSTM-LSTM	3	1	67.49	0.68
TextCNN-CNN	1	1	64.49	0.73
TextCNN-LSTM	5	0	65.78	0.45

Firstly, we can see that for two models, RoBERTa-LSTM and TextCNN-LSTM, including future utterances made no impact, and their best combination remains 5 previous utterances. For the remaining models including future utterances does slightly improve performance in all cases. However, this is more pronounced for the CNN contextual models, with an average accuracy increase of ~0.99%, while for the LSTM models it is just ~0.15%. This is evidenced in figure 5.4, which shows both RoBERTa context models results over all combinations. It can be seen that RoBERTa-CNN tends to favour a *combination*, with the best results clustered around 1-3 previous utterances and 1-2 future. On the other hand RoBERTa-LSTM favours longer *sequences*, and including more than 1 future utterance actually diminishes performance. A similar pattern is observed when comparing the remaining LSTM and CNN context models. This is likely due to the nature of the two context encoding approaches. A CNN, performing convolutions over the entire dialogue segment, is better able to capture the relationship between the current target, and any previous or future utterances. However, the LSTM, updating its hidden state as each utterance in the sequence is processed, is less able to identify the association between future utterances, the previous context, and target. To explore these observations further, we conduct multiple BSR tests examining the influence of future utterances, comparing results for the best combination to using only previous context, and each model to one another:

- Beginning with future utterances; for both transformer based CNN context models including up to 3 future utterances always improves upon the single-sentence baseline, beyond which we see no statistically significant improvement ($P(1{-}3\text{-}context > No\text{-}context) \geq .8$); this threshold is 2 for both transformer based LSTM models ($P(1-2\text{-}context > No\text{-}context) \geq .91$); and for both LSTM context models we see a significant difference in all cases ($P(1-5\text{-}context > No\text{-}context) \geq .91$); for the TextCNN models we see no significant result in any case. However, as with previous context, once these thresholds are reached, we see no significant improvements with increasing numbers.

- Comparing results for the best context combination with those obtained using only previous context, in most cases we see no statistically significant results. Naturally, RoBERTa-LSTM and TextCNN-LSTM are equivalent, and for the remaining LSTM context models results are non-significant. Only for BERT-CNN ($P(\mathit{Full\text{-}best} > \mathit{Prev\text{-}best}) = .99$), and LSTM-CNN ($P(\mathit{Full\text{-}best} > \mathit{Prev\text{-}best}) = .98$), do we see statistically significant results.

- If we again compare models using their best context combination; for CNN and LSTM context models we only see a statistically significant difference for BERT ($P(BERT\text{-}CNN > BERT\text{-}LSTM) = .85$); RoBERTa-CNN shows a significant difference over BERT-LSTM ($P(RoBERTa\text{-}CNN > BERT\text{-}LSTM) = .96$), and in all other cases transformer models are equivalent; equally both transformer based CNN context models improve upon the LSTM-LSTM $P(BERT/RoBERTa\text{-}CNN > LSTM\text{-}LSTM) \geq .9$.

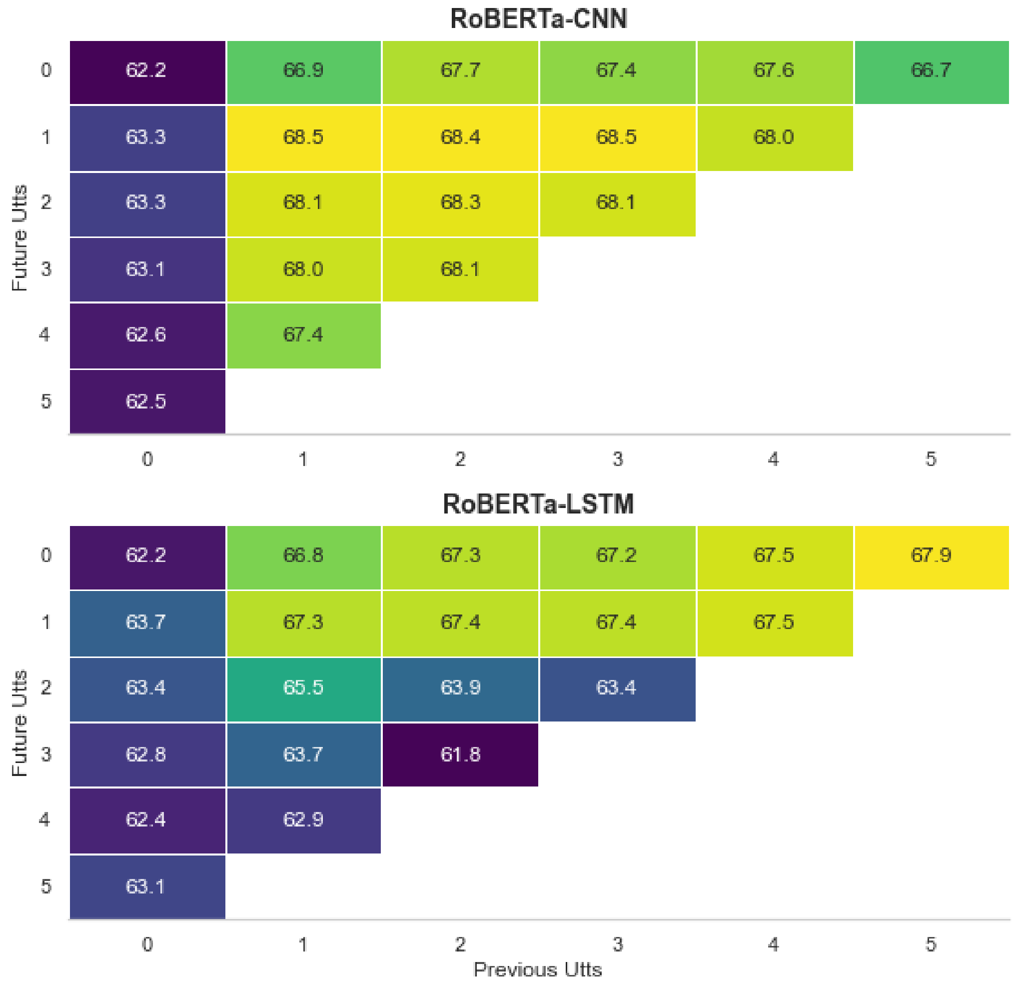


Figure 5.4: Maptask validation set accuracy with each previous and future context combination for the RoBERTa-CNN, and LSTM context models.

In summary, including one or more future utterances does improve upon the single-sentence baseline, but to a lesser degree than previous context. When using CNN context encoders a combination of previous and future utterances tends to result in ~1% improvement upon previous context alone, while for LSTM encoders we see no statistical difference, and this is likely due to the manner in which these models process the input dialogue segment. However, that including one or more future utterances can be beneficial is intuitive if we consider that DA can have both a forward and backward looking function. Indeed, several studies have shown that including several future dialogue utterances (Ribeiro, Ribeiro, and De Matos, 2019), or even the entire conversation as input (Chen et al., 2018; Kumar et al., 2017), can improve upon previous context alone. Finally, throughout our previous and full context experiments we have observed that the transformer based models have regularly outperformed those with LSTM or TextCNN sentence encoders. Also, in most cases the RoBERTa based models tended to result in a slight improvement over BERT based models, though in some cases this was not statistically significant. We therefore abandon the BERT, LSTM, and TextCNN context models and continue only with RoBERTa as our base sentence encoder. Specifically RoBERTa-LSTM, which favours previous context inputs, because we wish to focus primarily on information that would be available to a dialogue management system and results for previous and full context were equivalent.

5.2.3 Speakers and Labels

In this section we investigate several candidate speaker and label encoder models and evaluate their benefit across all three corpora. Throughout these experiments we use the RoBERTa-LSTM as a base context model, with 5 previous utterances as input. Thus, we also include 6 speakers, 5 labels, or both, as additional inputs. We begin by determining suitable speaker and label context encoder architecture in 5.2.3.1 and 5.2.3.2 respectively. Within these two sections we only gather results for the Maptask corpus, and as discussed in 5.2.1, we experiment with three different approaches: Recurrent, FFNN, and 'Flat'. We also explore whether encoded speakers or labels are joined with dialogue segment *sequentially*, or as a *summary*. Finally, in 5.2.3.3, for each corpus we evaluate the impact of speaker and label combinations, including predicted and 'gold-standard' labels.

5.2.3.1 Speakers

For our GRU and FFNN speaker encoders we test an increasing number of units in the range [16, 512], which represents the size of the hidden state, or number of nodes, for GRU and FFNN respectively. We begin with the *summary* join method and then apply the two best performing models, in terms of number of units, to the *sequence* join method. Results for both are shown in table 5.14.

Beginning with results for the summary join method, we can see that including speaker information consistently improves performance over the utterance-only context model. Using the simple flatten approach acheives an accuracy of 69.9%, which represents a +1.97% increase in validation accuracy, though in most cases this is outperformed by the GRU and FFNN. Comparing the best performing model, a FFNN with 512 units, we see an increase of +4.14%, and an improvement over the best performing GRU model of +0.81%. A BSR test confirms that this is also statistically significant ($P(FFNN\text{-}512 > GRU\text{-}256) = .81$). Though, while the FFNN outperforms the GRU in all cases it appears that increasing the number of units is less impactful. For the GRU, there appears to be a threshold at ~128 units, which results in a statistically significant improvement over smaller unit sizes ($P(GRU\text{-}128/256/512 > GRU\text{-}16/32/64) \geq .84$). Yet, beyond this threshold we see no significant difference between 128 and 512 units. On the other hand, for the FFNN in most cases results are inconclusive, or in a few cases statistically equivalent, for example, between FFNN-512 and FFNN-256 ($P(rope) = .84$).

Considering the results for the sequence join method, with the exception of FFNN-256, we see an improvement over the summary join method. However, BSR tests comparing the

Table 5.14: Results for the GRU and FFNN speaker encoder models, using either summary or sequence join method, on the Maptask validation set. The number of units is the size of the hidden state, or number of nodes, for GRU and FFNN respectively.

	GRU		**FFNN**	
Units	μ	σ	μ	σ
		Summary		
16	69.12	0.61	71.32	0.56
32	70.23	0.88	71.42	0.42
64	70.16	0.62	71.94	0.64
128	71.15	0.71	71.73	0.42
256	71.26	0.57	71.98	0.48
512	70.99	0.5	**72.07**	**0.56**
		Sequence		
256	71.38	0.48	71.8	0.77
512	71.71	0.56	**72.26**	**0.37**

number of units reveals that again there is no statistically significant difference. Similarly, when comparing the best summary and sequence models results are inconclusive, only the FFNN-512-summary is shown to be statistically equivalent to the sequence approach ($P(rope) = .8$).

Thus we conclude that, for the Maptask corpus including speaker information always improves performance. Of the three approaches, the FFNN consistently resulted in higher accuracy over the GRU, and both are preferable to simply concatenating the one-hot speaker representations. Further, using larger number of units and the sequence join method lead to marginal improvements, though in most cases this is not statistically significant. Nevertheless, we adopt the FFNN with 512 units, using the sequence join method, as our speaker context encoder for all remaining experiments.

5.2.3.2 Labels

For our label encoder experiments we conduct a scaled-down version of the speaker encoding experiments. We maintain the GRU and FFNN for label encoding, but with only 256 or 512 units because these sizes were previously shown to improve performance, and input labels will consist of larger one-hot vectors than speakers. Results for these two models, using both summary and sequence join methods, are shown in table 5.15. It should be noted that, throughout these experiments we use the 'gold-standard' labels from the corpus itself. As with the inclusion of future utterances, using the gold-standard labels may not be appropriate for certain tasks, such as dialogue management systems, where the true labels are not available. Nevertheless, for other applications, and in line with other studies (Ribeiro, Ribeiro, and De Matos, 2019), we wish to determine the upper bounds of what is feasible when such contextual information is available, and leave comparison with predicted labels for the following section (5.2.3.3).

As with the inclusion of speakers, incorporating contextual label information results in consistent improvements over the utterance-only context model, though to a lesser degree. The worst performing model, GRU-256-sequence, only achieves a +0.34% increase, while the best model, FFNN-512-summary, results in an increase of +3.51%. Again, the FFNN

Table 5.15: Results for the GRU and FFNN label encoder models, using either summary or sequence join method, on the Maptask validation set. The number of units is the size of the hidden state, or number of nodes, for GRU and FFNN respectively.

	GRU		**FFNN**	
Units	μ	σ	μ	σ
Summary				
256	70.01	0.61	70.89	0.33
512	70.31	0.34	**71.44**	**0.45**
Sequence				
256	68.27	0.45	69.97	0.57
512	68.72	0.4	**70.36**	**0.42**

outperforms the GRU in all cases, and BSR test confirm that, with the exception of FFNN-256 compared to GRU-512 using the summary join method, this is statistically significant ($P(FFNN > GRU) \geq .89$). Similarly, a larger unit size consistently results in improved performance, but for both models, and join methods, we see no statistically significant results.

Results for label encoders are similar to those for speaker encoders, where the FFNN and a larger unit size, reliably outperforms the GRU or a smaller unit size. However, in contrast, our results show that the label encoders benefit from the summary join method, and BSR test show this is statistically significant in all cases ($P(Summary > Sequence) \geq .93$). Thus, for our label encoder we adopt the same architecture as the speaker encoder (FFNN with 512 units), except we use the summary join method instead of sequence.

5.2.3.3 Speaker and Label Combinations

Here we present our results for including speakers *and* labels on the SwDA and CAMS-KVRET corpora. We also investigate using combinations of both speakers and labels, as well as predicted labels. To generate label predictions we process each utterance within the dialogue segment in turn, concatenating with all previous utterances, to produce a prediction at each timestep. For example, first a prediction is generated for the $m - 5^{th}$ utterance, then the utterance encoding and its predicted label are concatenated with the $m - 4^{th}$ encoding, to produce a prediction for that utterance, and so on up to the mth utterance. Results for all speaker, label, and predicted labels are shown in table 5.16.

Beginning with speaker and gold-standard label combinations, we can see that for all corpora including any such context information always improves upon the utterance-only context model, particularly for speakers+labels. Notably, for Maptask the combination improves upon speakers or labels alone, and represents a +7.1% increase over utterances-only. To the best of our knowledge this represents a state-of-the-art result for this corpus. For SwDA, it appears that including labels is more impactful than speakers, resulting in a +2.03% improvement over utterances-only, while for speakers this is just +0.22%. However, for CAMS-KVRET, in each case we see only minimal improvements, with speakers+labels achieving +0.3% over utterances-only.

Considering predicted labels, for both Maptask and CAMS-KVRET these resulted in a slight decrease in performance when compared to utterances-only, and for Maptask the improvement of speakers+predicted labels appears to be the contribution of speakers alone. However, for SwDA, predicting labels does result in a small increase of +0.4%. We

Table 5.16: Results for speakers, labels, and predicted labels context information on the validation set for each corpora.

	SwDA		**Maptask**		**KVRET**	
Context Info	μ	σ	μ	σ	μ	σ
Speakers	80.96	0.44	72.26	0.37	93.93	0.31
Labels	**82.77**	**0.4**	71.44	0.45	94.12	0.24
Speakers+Labels	82.7	0.54	**75.03**	**0.41**	**94.13**	**0.26**
Pred-labels	81.14	0.32	67.72	0.56	93.78	0.19
Speakers+Pred-labels	80.96	0.33	72.17	0.6	93.86	0.26
Utterances-only	80.74	0.47	67.93	0.57	93.83	0.11

examine these results further by conducting BSR tests to compare each combination to one another across all corpora:

- For SwDA, we find that including labels results in statistically significant improvement over both speakers and utterances-only ($P(Labels > Speakers/Utts\text{-}only) = 1$). Yet the speakers+labels combination is equivalent to just labels ($P(rope) = .81$), thus labels are the only extra contextual information that result in significant improvements. Both predicted labels, and speakers+predicted labels, are statistically equivalent to speakers alone ($P(rope) = .83$), and in all other cases we see no significant differences.

- On Maptask, all speaker and gold-standard label combinations are clearly a significant improvement over the utterances-only baseline, and speakers represent a statistically significant improvement over labels ($P(Speakers > Labels) = .84$). That speakers+predicted labels is statistically equivalent to only speakers ($P(rope) = .88$), confirms our previous observation, and in all other cases we see no significant differences.

- Finally, for CAMS-KVRET, in *all* cases we see statistical equivalence ($P(rope) \geq .81$), confirming that neither of these result in significant benefits for this corpus.

While we have observed that including speaker or label information is generally beneficial, that we see some differences between these corpora is more intuitive if we consider each of their domains. Within the non-task-oriented, general talk, of SwDA each speaker is equally likely to produce any DA. This implies there is less correlation between sequences of speakers and specific DA labels. Sequences of labels however, do suggest the next likely DA, and therefore we see improvements in performance when including those. Our results coincide with those of previous work, which indicates label information is more impactful than speaker for this corpus (Ribeiro, Ribeiro, and De Matos, 2019; Bothe et al., 2018b). On the other hand, Maptask is task-oriented, with specifically defined roles for both speakers and a strong correlation between a given speaker and certain DAs. Thus, we see a significant improvements, particularly when including both kinds of contextual information. Results for CAMS-KVRET are perhaps more surprising, because the corpus is also task-oriented with clearly defined roles for both speakers. However, while some DA are more likely to be associated with the driver role (such as *requests*), both speakers are equally capable of, for example, asking or responding to a clarifying question, and initiating or concluding a goodbye. Therefore, we see less correlation between speaker and DA label. It may also be the case, that with high classification accuracy from context utterances alone, adding further contextual information simply does not benefit the model.

5.2.4 Classifiers

In our final set of DA classification experiments, we compare three sequential classifier models with the FFNN used throughout our previous experiments. We test a CRF layer, and two Seq2seq models, one using additive attention (Bahdanau, Cho, and Bengio, 2015) and the other 'hard' attention. The latter two are an attempt to re-create the experiments of Colombo et al. (2020), who use Seq2seq models with various different attention mechanisms to achieve competitive results on the SwDA corpus.[7] Throughout these experiments we use the RoBERTa-LSTM context model with 5 previous utterances and no speaker or label information. Results for accuracy and sequence accuracy – the predictions for each utterance in the input dialogue segment – are shown in table 5.17.

Table 5.17: Classifier accuracy and sequence accuracy for the validation set of each corpora.

	SwDA				**Maptask**				**KVRET**			
Classifier	μ	σ	*seq*-μ	*seq*-σ	μ	σ	*seq*-μ	*seq*-σ	μ	σ	*seq*-μ	*seq*-σ
Seq2seq-Add	79.08	0.29	77.82	0.37	62.6	0.25	54.8	2.87	91.62	0.3	86.89	1.0
Seq2seq-Hard	79.29	0.4	78.12	0.34	62.74	0.34	57.44	2.26	91.48	0.4	87.4	0.89
CRF	80.61	0.7	80.19	0.82	67.26	0.32	65.95	0.3	93.56	0.22	92.44	0.18
FFNN	**80.74**	**0.74**	-	-	**67.93**	**0.57**	-	-	**93.83**	**0.11**	-	-

Our results comprehensively show that in all cases the sequential classifiers do not improve over the FFNN. In fact, both Seq2seq models resulted in significantly diminished performance over the CRF and FFNN layers. Though, Colombo et al. (2020) provided no code for their implementation, so it may be that our Seq2seq models were missing a key detail that resulted in such poor performance. However, results for the CRF closely matched that of a FFNN; on SwDA and CAMS-KVRET they are found to be statistically equivalent ($P(rope) \geq .8$), but on Maptask we see no significant results. Thus we conclude that, despite its widespread use within the DA classification literature, a CRF does not produce higher classification accuracy than FFNN approaches, and we therefore maintain the latter as our classifier component.

5.2.5 Context Encoding Summary

Throughout this section we have explored numerous context encoding architectures for the purpose of DA classification on the SwDA, Maptask, and CAMS-KVRET corpora. First, we considered the inclusion of previous, and future utterances, and evaluated both convolutional and recurrent context encoder models. Our results show that, including just a single context utterance (previous or future) always improves performance over the single-sentence baseline. Increasing the size of the input dialogue segment also tends to improve performance, but only up to a point, and typically beyond 2-3 utterances we see no statistically significant difference in performance. Additionally, the convolutional context encoder was shown to benefit from dialogue segments which include both future *and* previous utterances, while the recurrent models performed better with either previous *or* future context. This is likely due to the manner in which these models process the input dialogue segment, either performing convolutions over the entire segment, or processing each utterance in turn.

Next, we investigated the inclusion of contextual information, in the form of speakers and labels. For both speakers and labels the FFNN with the largest number of units tested (512) was shown to outperform the flatten and GRU methods. However, our results show that joining the speaker encodings using the *sequential* method resulted in better

[7] A key difference is that for our models we did not include beam search. However, our results indicate that the improvement the authors claim beam search made are inconsequential to our model.

performance, while for labels using the *summary* vector was preferable. Interestingly, we found that while including speaker or gold-standard label information always improved performance over an utterance-only context model, results varied significantly between our three corpora. On Maptask we see statistically significant improvements when including both speakers and labels, even more so in combination. We speculate that this is likely due to the nature of the corpus being task-oriented, with clearly defined roles for both speakers. Yet, for SwDA only the inclusion of labels (predicted or gold-standard) resulted in significant improvements, and on CAMS-KVRET we saw no statistically significant results.

Finally, we compared the use of sequential classifiers to a standard FFNN. In all cases we found the FFNN superior to the Seq2seq and CRF models, though performance of the CRF was generally equivalent to a FFNN, and therefore we maintain the latter for our classification layers.

Table 5.18: Final RoBERTa-LSTM context models validation and test set accuracy for each corpus. Full context results are obtained using 4 previous and 1 future utterances, speakers, and labels. For partial context we use 5 previous utterances and speakers.

	Validation		**Test**	
Corpus	μ	σ	μ	σ
Full Context				
SwDA	83.61	0.23	81.47	0.25
Maptask	75.64	0.57	77.40	0.58
KVRET	93.93	0.30	94.33	0.38
Partial Context				
SwDA	80.96	0.44	78.85	0.39
Maptask	72.26	0.37	73.04	054
KVRET	93.93	0.31	94.53	0.42

On the basis of our context encoding experiments we select the RoBERTa-LSTM with a FFNN classifier as our base context model. Where speaker or label context is included we use a FFNN with 512 units, speakers are join with the sequence method, and labels with a summary. Using this model, final results for the validation and test set of each corpus are shown in table 5.18. Here, we present results for 'full context', which uses all available context information, and 'partial context', which only uses context information that would be available to a dialogue system. Specifically, full context uses 4 previous and 1 future utterance, the speakers, and gold-standard labels, whereas partial context uses the 5 previous utterances and speakers.

For both SwDA and Maptask the full context inputs result in a statistically significant increase over partial ($P(Full > Partial) = 1$), with a ~+2.6%, and ~+3.9% improvement respectively. These SwDA results are equivalent to, or competitive with, results reported within the literature (He et al., 2021; Raheja and Tetreault, 2019), while for Maptask, to the best of our knowledge, partial context alone represents an ~5% improvement over the best previously reported results (Wan et al., 2018). On the other hand, for KVRET there is no notable difference between the two context inputs. We explore this observation further, along with the automatic identification of AP, in the next chapter.

Chapter 6

Adjacency Pair Identification

In this chapter we present experimental results for the final phase of our dialogue classification system, AP classification and the development of a multi-label classifier, as discussed in 3.2 (**O3**, **04**, **Q2**, and **Q3**). We begin with CAMS *single-label* classification in Section 6.1, where we obtain results for each label type within CAMS, using the classification model developed in the previous chapter. Then, in Section 6.2, we present results for the different CAMS *multi-label* classifier architectures that are capable of simultaneously identifying both the DA and AP of a given utterance, and hence its AP-type, as discussed in 3.2.3.[1]

6.1 CAMS Single-labels

For our CAMS *single-label* classification experiments we utilise the RoBERTa-LSTM context model to replicate the context utterance, speakers, and labels experiments for AP, to determine if our findings for DA also apply to AP. For reference, and to determine a set of non-contextual single-sentence baselines for CAMS-KVRET, we re-applied the RoBERTa sentence encoder to each label type within the corpus. Baseline results for validation and test set accuracy are shown in table 6.1, along with two further baseline classifiers 'Prior' and 'Most Frequent'. The former represents a classifier that generates label predictions according to the actual distribution of labels within the training data, while the latter is the accuracy obtained by simply choosing the most common label for all predictions. The single-sentence baseline results show AP labels are able to reach a similar accuracy as DA. Though, we must consider that for AP there is significant class imbalance, with the two most frequent labels, *FPP-base* and *SPP-base*, constituting 74.22% of all labels. However, recall that there are 104 AP-type labels within the corpus, and yet accuracy also remains high for AP-types, despite the significant increase in the number of labels. This suggests that, even without context, the classifier is able to learn associations between the semantic content of an utterance and appropriate AP, or AP-type labels.

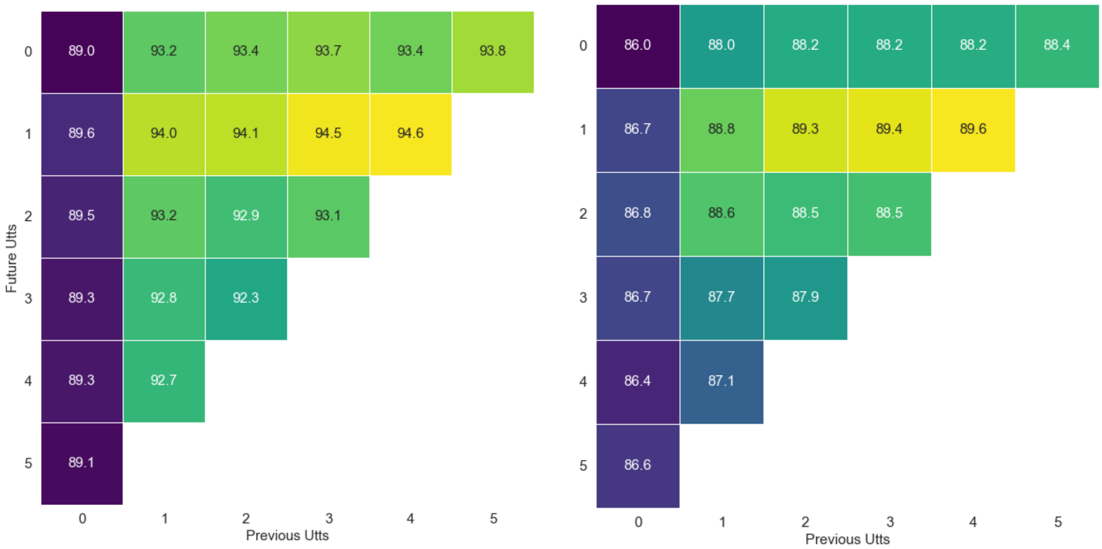
6.1.1 The Benefit of Context Utterances for CAMS

In this section we re-create our context utterance experiments from section 5.2.2. We tested all combinations of previous and future utterances that sum to 5 for both DA and AP – which were not previously included in the full context utterance experiments – to determine if the optimal number, or combination, of context utterances is similar for both label types. Results for both label types across all context combinations are shown in figure 6.1.

[1]All code, data, and accompanying analysis for the CAMS classification experiments is available at:github.com/NathanDuran/CAMS-Dialogue-Classification.

Table 6.1: CAMS-KVRET Single-sentence baselines for validation and test set. *Prior* represents a classifier that generates label predictions according to the actual label distribution within the training data, and *Most Frequent* is the accuracy obtained by simply choosing the most common label.

	RoBERTa				**Prior**		**Most Frequent**	
	Validation		**Test**		**Validation**	**Test**	**Validation**	**Test**
Label Type	μ	σ	μ	σ				
DA	88.97	0.54	88.46	0.52	10.98	12.23	13.98	14.88
AP	86.01	0.55	86.08	0.58	29.6	29.7	37.11	37.05
AP-type	79.0	0.82	79.22	0.5	6.64	8.90	13.15	13.96



(a) DA

(b) AP

Figure 6.1: Mean validation set accuracy for DA and AP labels, using previous and future context utterance combinations that sum to 5.

From these results we can make two observations: i) across each of the context utterance combinations we see the same pattern for both DA and AP, and ii) context appears to be slightly less impactful for AP than it does for DA. We discuss our first observation, and examine these results further, by conducting BSR tests to compare each context combination for both label types:

- For both label types the optimal combination of context utterances is the same, 4 previous and 1 future, and this represents an increase of +5.61% and +3.56% for DA and AP respectively. This generally coincides with our full context results for the Maptask corpus discussed in Section 5.2.2.2, where we observed that including 1 future, and larger numbers of previous utterances, typically improved performance. Also, that adding additional future utterances yielded diminishing returns. Though, on Maptask the above is true for the RoBERTa-CNN context model, while here it is also the case with RoBERTa-LSTM. Nevertheless, for both label types using the optimal combination of previous and future utterances represents a statistically significant improvement over using either alone ($P(\textit{Full-best} > \textit{Prev/Future-best}) \geq .99$).

- Considering only previous utterances, again we see that including just 1 results in a significant improvement over the single-sentence baseline of +4.24% and +1.94% for DA and AP respectively, and in all cases this is statistically significant ($P(Context > No\text{-}context) \geq .99$). Additionally, for both label types, increasing the size of the context window tends to improve performance, with 5 previous utterances resulting in the highest accuracy. Though, in most cases we see minimal difference between 5 previous utterances or fewer. In only one case do we see statistically significant results, that is for DA when comparing 5 previous utterances to 1 ($P(5\text{-}context > 1\text{-}context) = .82$), and all other cases are either statistically equivalent ($P(rope) \geq .82$), or inconclusive.

- Similarly, for future utterances including just 1 results in an improvement over the single-sentence baseline, though overall the benefit is negligible at $\leq 0.67\%$ and $\leq 0.82\%$, for DA and AP respectively. Indeed, only for AP with 2 context utterances do we see a significant difference over the baseline ($P(2\text{-}context > No\text{-}context) = .88$). Again, we see minimal difference when comparing different size context windows, and for both label types we see statistical equivalence ($P(rope) \geq .85$), or inconclusive results.

Our second observation – that including context utterances results in a smaller accuracy increase for AP than for DA – is surprising, because AP might naturally take place over longer sequences of utterances, so intuitively one might expect larger context windows to benefit AP more. However, a further complication is that, in certain situations it may be difficult to select between one of several labels. For example, if a *SPP-base* has just concluded a pair, the following utterance could be the beginning of a new base-pair, post-pair, or minimal post expansion. To select between these options it must be decided whether the following utterance(s) are recognisably associated with the preceding sequence, and there may only be a subtle variation within the semantic content. Similarly, a singular *Insert* or *Post* label could be mistaken for the completion of a base or insert-pair. This problem is closely related to the 'semantic ambiguity' faced by our human annotators, and discussed in 4.2.5.2. This notion is illustrated in figure 6.2, which shows confusion matrices for a contextual and non-contextual model, and table 6.2 that displays the per-label change in F1 score between the two approaches.[2]

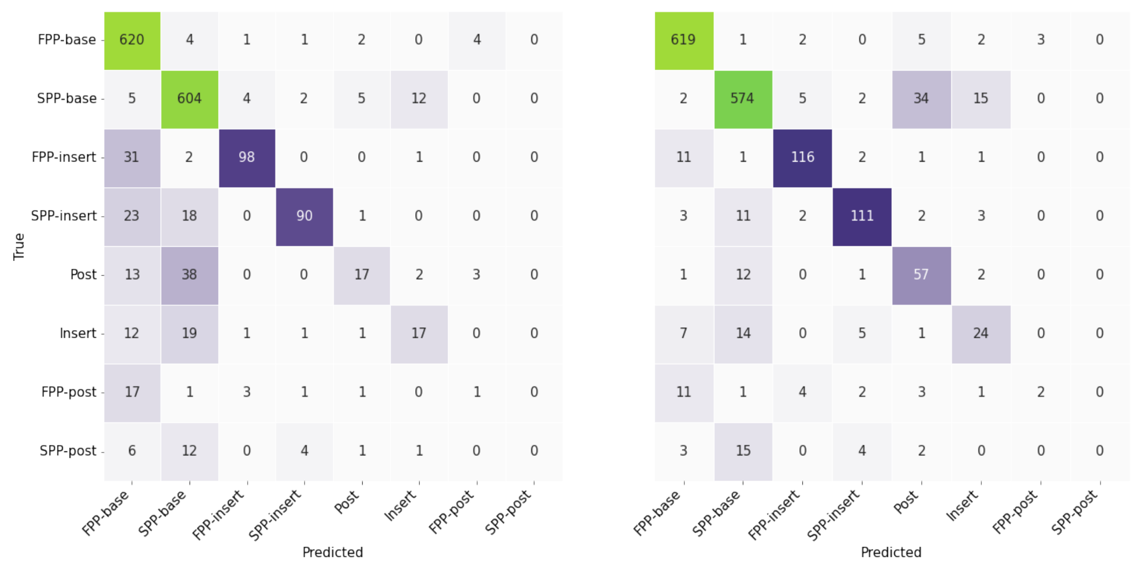
Table 6.2: Per-label AP F1 score comparison between contextual and non-contextual model predictions on the validation set.

Label	**Count**	**No-Context**	**Context**	**Change**
FPP-base	632	0.91	0.96	+.05
SPP-base	632	0.91	0.91	±.0
FPP-insert	132	0.82	0.89	+.07
SPP-insert	132	0.78	0.86	+.08
Post	73	0.34	0.63	+.29
Insert	51	0.4	0.48	+.08
FPP-post	24	0.06	0.14	+.08
SPP-post	24	0.0	0.0	±.0
Pre	3	0.0	0.0	±.0

Firstly, we can see that the non-contextual model more often predicts the two most common labels, and this is where the majority of incorrect predictions arise. With the

[2] A similar figure for DA labels can be found in appendix G.1.

contextual model we see a significant reduction in this behaviour and it more often correctly predicts insert pairs, and *Insert* or *Post* expansions. For the latter we see an increase in F1 score of +.29 between the two models. However, there is still a considerable amount of confusion around the *SPP-base*, and *Insert* or *Post* labels. Indeed, the contextual model so frequently predicts *Post*, when the true label is *SPP-base*, that the overall correct predictions of *SPP-base* decreased, and the F1 score remains static. This alone equates to a 1.76% loss in accuracy, which accounts for most of the ~2% disparity between DA and AP observed previously. Interestingly, because a *Post* expansion must take place *after* the completion of a base-pair, this also indicates the model is not correctly finishing base pairs. Again, this mistake is not dissimilar to those of our human annotators discussed in 4.2.5.2. Nevertheless, that the model is predicting *Insert* and *Post* for *SPP-base*, and not for *FPP-base*, along with the high accuracy and F1 scores for base and insert pairs, suggest the model is able to capture some aspects of the sequential relationships between AP. It should also be noted that neither contextual nor non-contextual models predict complete post pairs. In fact, no *SPP-post* labels are predicted in either case. Though this is likely due to an imbalance between the training set compared to the validation and test sets, which contain 0.65% to 1.6% fewer post pairs respectively. Similarly, the *Pre* expansion label only appears 3 times within the validation set, and is not predicted at all.



(a) No-context (b) Context

Figure 6.2: Confusion matrices for AP predictions on the validation set using non-contextual (a), and contextual (b), classification models.

Briefly considering the impact of context on DA, table 6.3 displays the per-label change in F1 score between the two approaches. Notably the *answer* and *inform* labels show the most significant increases in F1 score. The non-contextual model was frequently confusing these two labels, which is unsurprising given the communicative function and semantic content of their associated utterances is likely very similar. Indeed, the non-contextual model miss-classified *inform* as *answer* 38 times, and vice versa 39 times. The contextual model is able to differentiate between these two, and the incorrect predictions are reduced to 8 (+1.76%), and 14 (+1.47%) respectively, which represents an accuracy increase of +3.23%, and makes up a significant proportion of the overall improvement (+5.61%). Curiously, the contextual model actually performs worse on four label types, *autoPositive*, *choiceQuestion*, *goodbye*, and *suggest*. However, these labels combined only appear 45 times, or make up 2.64% of the validation set, so the overall impact is negligible.

Table 6.3: Per-label DA F1 score comparison between contextual and non-contextual model predictions on the validation set.

Label	Count	No-Context	Context	Change
thanking	238	0.98	0.98	±.0
acceptThanking	233	0.97	0.98	+.01
setQuestion	221	0.98	0.98	±.0
request	214	0.94	0.95	+.01
answer	204	0.74	0.88	+.14
inform	164	0.66	0.85	+.19
propositionalQuestion	136	0.96	0.96	±.0
accept	131	0.91	0.91	±.0
confirm	62	0.85	0.90	+.05
disconfirm	52	0.91	0.91	±.0
autoPositive	22	0.79	0.73	−.06
choiceQuestion	14	0.97	0.72	−.25
goodbye	6	0.92	0.83	−.09
suggest	3	0.8	0.0	−.8
checkQuestion	2	0.0	0.0	±.0
apology	1	0.0	0.0	±.0

6.1.2 The Influence of Speakers and Labels

In this section we re-create our speaker and label context experiments from section 5.2.3. The previous approach is adapted, such that we may begin to explore the association between the two label types, and whether the contextual information provided by one may provide useful information when predicting the other. Thus, we test the inclusion of speakers, DA and AP labels, or a combination of both label types, and evaluate the impact on AP and DA classification. Results for both label types across all context combinations are shown in table 6.4. This data was gathered using 5 previous utterances as input, and for the Speakers+Labels combination, the label type included is the one being predicted.

Table 6.4: Results for speakers, DA and AP label context information on the validation set. Note that for Speakers+Labels, the label type included is the one being predicted.

	AP		DA	
Context Info	μ	σ	μ	σ
DA Labels	88.81	0.43	94.12	0.24
AP Labels	93.1	0.35	94.33	0.28
AP+DA Labels	**93.55**	**0.17**	**94.57**	**0.28**
Speakers	88.57	0.43	93.93	0.31
Speakers+Labels	93.34	0.42	94.13	0.26
Utterances-only	88.36	0.35	93.83	0.11

Beginning with DA, we can see from table 6.4 that, in agreement with our previous experiment (5.2.3.3), the inclusion of speaker and label information has minimal impact on DA classification. However, the best results are achieved when including both AP and DA labels and this represents a +0.74% increase compared to the inclusion of speakers or utterances-only. BSR tests also confirm that this is statistically significant ($P(AP + DALabels > Speakers/Utts\text{-}only) = .8$), and in all other cases results are statistically equivalent ($P(rope) \geq .89$) or inconclusive.

For AP we see quite a different picture. In each case where AP labels are included there is a significant improvement over other context types. Again, the best results are achieved when including AP+DA Labels, which represents a +5.19% increase compared to utterances-only and is statistically significant in all cases ($P(AP/AP + DALabels/Speakers + Labels > Speakers/Utts\text{-}only) = 1$). However, speakers and DA labels seem to have a similarly minimal impact on AP classification. Comparing the inclusion of speakers and DA labels to utterances-only, for the former we find them to be statistically equivalent ($P(rope) = .85$), the latter marginal but inconclusive ($P(rope) = .79$).

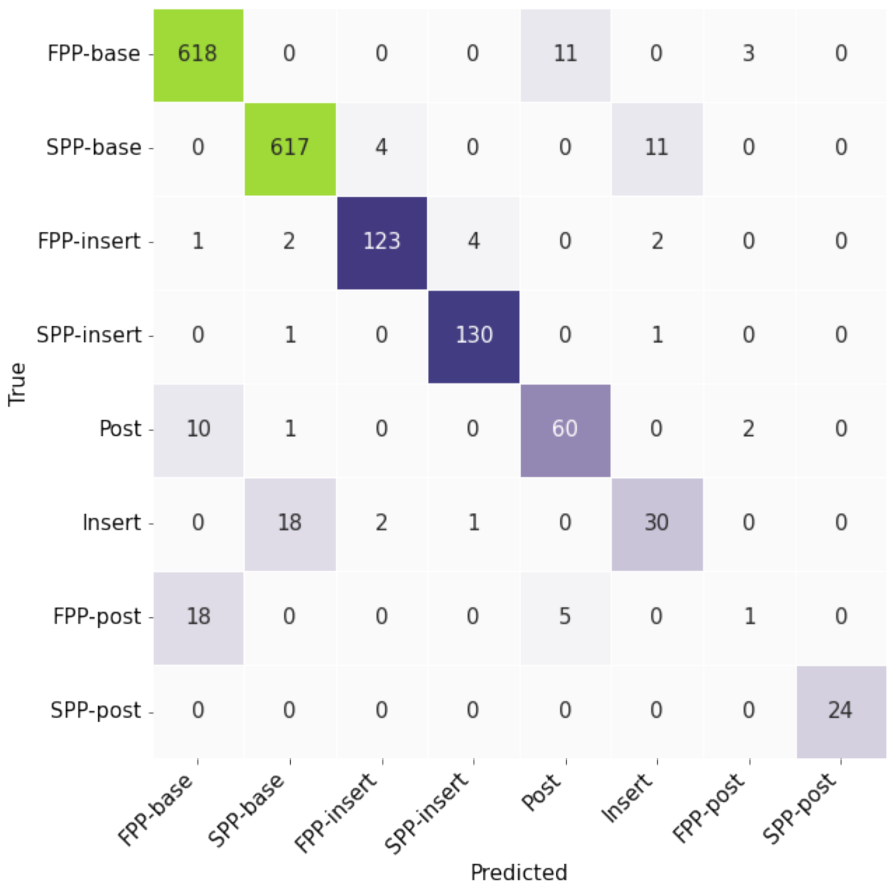


Figure 6.3: Confusion matrix for AP predictions on the validation set using AP and DA label context information.

The benefit of including contextual label information for predicting AP is illustrated in fig 6.3, and table 6.5, which shows the per-label change in F1 score between an utterance-only context model and one that also includes DA and AP label information. Comparing the confusion matrices in figure 6.3 and 6.2b, we see a reduction in the number of incorrectly classified *SPP-base*, and insert-pair labels. Additionally, there is a significant increase in the number of correct classifications for *Post* and *Insert*, while *SPP-post* is now being predicted with 100% accuracy. These results suggest that the model is not only better able to disambiguate between appropriate *Post* and *Insert* utterances, but has also learnt a more correct representation of AP sequences; because SPP base, insert, and post pairs are more often correctly completed. Instead, we now predominantly see confusion around *FPP-base*, and either *Post* or *FPP-post* labels, or *SPP-base* and *Insert* labels. In both cases these mistakes are intuitive, because the model was given no future context information. Thus, upon completion of a prior pair, the next utterance could reasonably be the beginning of a new base or post pair, or a post minimal expansion. Similarly, without a further utterance for context, it is more correct to complete the current base pair than to assign a single insert expansion.

The considerable increase in performance when prior labels are provided implies that AP, unlike DA, are more representative of the overall dialogue structure and sequential order of utterances. Previous AP labels are therefore more predictive of appropriate future AP, while in contrast sequences of DA are less indicative of suitable future labels. This observation supports the core principle of our proposed approach, that DA labels can be considered descriptions of the *intra-utterance* features of a dialogue, while AP represent the *inter-utterance* features. However, it appears that including DA when predicting AP, or vice versa, is not overly beneficial in either case. This is somewhat surprising given that one might expect some DA to be highly correlated with certain AP, and conversely, some AP may be more often associated with certain DA. For example, FPP are more often associated with requests and questions, and SPP are more frequently answers, informs, and accepts. Yet, it appears the model is unable to make use of such information, possibly because there are so many potential combinations due to the lack of strict association within CAMS. This observation is explored further with our CAMS multi-label experiments in section 6.2.

Table 6.5: Per-label AP F1 score comparison between utterance-only and AP+DA labels contextual model predictions on the validation set.

Label	**Count**	**No-Labels**	**Labels**	**Change**
FPP-base	632	0.96	0.96	±.0
SPP-base	632	0.91	0.97	+.06
FPP-insert	132	0.89	0.94	+.05
SPP-insert	132	0.86	0.97	+.11
Post	73	0.63	0.81	+.18
Insert	51	0.48	0.63	+.15
FPP-post	24	0.14	0.07	−.07
SPP-post	24	0.0	1.0	+1.0
Pre	3	0.0	0.0	±.0

6.1.3 Evaluating CAMS Single-label Classification

In this section we present our final results for single-label CAMS dialogue classification. For each label type we gather accuracy and top-k accuracy on the CAMS-KVRET validation and test sets, and for the latter we additionally report the macro averaged F1, and weighted F1 scores. Again, we obtain results for 'full context', which uses all available context information, and 'partial context', which only uses context information that would be available to a dialogue system. Specifically, full context uses 4 previous and 1 future utterance, as well as speaker, DA, and AP labels as input, and for partial context, we use

5 previous utterances and speakers only. Results for each of these measures are shown in table 6.6.

Table 6.6: Full and partial context models validation and test set accuracy and top-3 accuracy, as well as macro and weighted F1 scores, for each label type. Note that, AP results for partial context differ slightly from those previously reported because experiments were repeated in order to gather top-k accuracy measures.

	Validation				**Test**					
Label Type	μ	σ	*top3-*μ	*top3-*σ	μ	σ	*top3-*μ	*top3-*σ	***macro-F1***	***weighted-F1***
					Full Context					
DA	94.54	0.27	99.27	0.19	94.52	0.31	98.91	0.15	0.75	0.94
AP	99.38	0.11	99.97	0.03	99.18	0.18	99.95	0.06	0.94	0.99
AP-types	91.61	0.58	97.22	0.15	90.43	0.38	96.09	0.21	0.54	0.89
					Partial Context					
DA	93.99	0.38	99.34	0.07	94.34	0.33	99.07	0.12	0.7	0.94
AP	88.29	0.42	97.29	0.23	88.4	0.44	97.06	0.74	0.64	0.86
AP-types	84.86	0.47	94.48	0.46	84.61	0.4	93.16	0.29	0.31	0.81

Beginning with DA results, we can see that in agreement with our previous findings, including future utterances, speakers and labels has a minimal impact, with an improvement of just +0.55%, and +0.18% on the validation and test accuracy respectively, though the macro-F1 does improve by +.05%. BSR tests also confirm that this is not significant ($P(Full > Partial) = .59$). However, regardless of context information, top-3 accuracy is $\geq$ 98.9%, which indicates that the correct label is consistently in the top-3 predictions.

Both AP and AP-types show significant improvements when full context information is included. Remarkably, AP achieve $\geq$ 99.1% accuracy with full context, which represents an average increase of +10.9%. Macro and weighted F1 also increased by +.24% and +.13% respectively. On the other hand AP-types reach $\geq$ 90.4% accuracy with full context, an average increase of +6.3%, and macro and weighted F1 scores improve by +.23% and +.08% respectively. For both label types this is also statistically significant ($P(Full > Partial) = 1$). Again, regardless of context information, top-3 accuracy is significantly higher than base accuracy for AP and AP-types (with the exception of APs and full context, where they are both > 99%), though, for partial context in particular it is ~9% higher. High top-3 accuracy in all instances suggests that the model is often selecting from a small subset of possible labels. Thus, even where final predictions are incorrect, the correct label lies within the next two most probable labels. We consider the implications of this observation at the end of this section.

Label Cardinality and Distribution Our results agree with previous observations, that both AP and AP-types benefit more from the sequential, or relational, information provided by the context of surrounding utterances and their labels. In contrast, for DA this kind of contextual information seems largely inconsequential, which suggests the semantic content of the utterances themselves is more important. However, we must also consider these results from another perspective, and that is the difference between the number and distributions of the different label types. Recall the baseline results reported in table 6.1, which shows comparatively high accuracy for AP when using a classifier that assigns labels according to the prior distribution, or simply selecting the most frequent label. This is caused by the high prevalence of base pairs, which combined make up ~74% of all labels in the corpus. One possibility is that the high prevalence of just a few label types, and that there are simply fewer AP labels to choose from, skews the observed improvements for AP and makes comparison between AP and DA difficult. Particularly regarding top-k accuracy where fewer label types would naturally lead to higher values. Therefore, it may be beneficial to consider the *difference*, or *improvement*, between our contextual models and the baseline classifiers. The difference in validation set accuracy, between full and partial context models and the baseline classifiers, is shown in figure 6.4.[3]

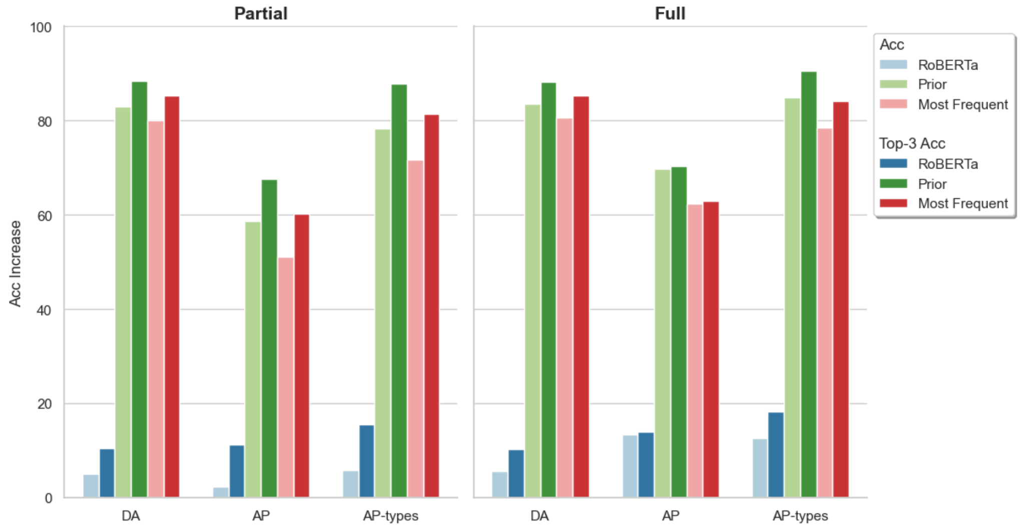


Figure 6.4: Difference in validation set accuracy and top-3 accuracy, between full and partial context models and the baseline classifiers (RoBERTa, prior, and most frequent label).

As expected, for DA we see fairly consistent improvements over the baselines for both context models, while both AP and AP-types show greater improvement over the single-sentence baseline when using full context information. For AP the smaller improvement over the prior and most frequent baselines is due to the high prevalence of base-pair labels mentioned previously. Notably, with partial context AP improve by only +2.28% over the single-sentence baseline. On the other hand, for AP-types using either context model results in a larger improvement over the single-sentence baseline than that of DA. Additionally, improvements over prior and most frequent baselines are comparable to those of DA, despite the baseline accuracies being significantly lower in most cases. These findings seem to discount the notion that the high frequency of a few AP labels was contributing to large gains in accuracy, and top-3 accuracy, when compared to DA labels. We may also consider the number of labels in each of these groups; there are 104 AP-type labels within the corpus, which is more than 11 times the number of AP (9), and more than four times the number of DA (23). With increasing numbers of labels one might typically expect lower predictive accuracy, and if we compare DA or AP to AP-types this is indeed the case. However, that we observe larger improvements over each of the baseline classifiers for AP-types, particularly for top-3 accuracy, suggests that the observed improvements for AP is also not due to the number of labels.

Components of AP-types We have observed that AP-types, like AP, benefit more from the context of surrounding utterances than DA. However, even with partial context AP-types show greater improvement over the single-sentence baseline than either DA or AP. These findings seem to suggest that when predicting AP-types, the contextual models are simultaneously learning the intra-utterance, or semantic content, aspects of DAs, and also the inter-utterance, or structural, features provided by AP. To explore this possibility

[3]Exact figures can be found in appendix G.1.

we can consider the number of incorrect predictions for each label type, and further, determine how many occasions incorrect DA or AP predictions overlap with those of AP-types. Examining the number of incorrect validation set predictions from three different classifiers, each using partial context to predict a different label type, we find that:

- The DA classifier made 108 incorrect predictions, AP 195, and AP-types 246. Thus, predicting AP-types, despite significantly more labels, results in fewer inaccuracies than predicting the DA and AP components separately.
- If we concatenate the individual DA and AP predictions to form AP-types, we find 34 instances where *both* labels were incorrect, and of these, AP-types were *also* incorrect on 33 occasions. In other words, there are only 33 instances (~1.9% of the validation set) where predictions for all three label types were incorrect. Of these there are also 23 instances where predictions were the same.
- We may also consider the number of instances where the individual components of AP-types (DA or AP) were incorrect, and compare these to the instances where the separately predicted label type was also incorrect. We find that the AP-types DA component was incorrect on 105 instances, and the AP component on 191. Of the 108 incorrect DA predictions, the AP-types DA component was also incorrect on 76 occasions (~70%), and of these, 70 predictions were the same. For the 195 incorrect AP predictions, the AP component was also incorrect on 153 (~78%) of instances, and of these, 122 predictions were the same.

These figures show that there is a significant amount of overlap, not only for the instances of incorrect predictions, but the predicted labels themselves. Thus, whether comparing concatenated DA and AP labels to AP-types, or the individual components, all three label types tend to make similar mistakes. In conjunction with the improvements over baseline classifiers previously discussed, this demonstrates that when predicting AP-types the model does indeed learn the semantic and structural aspects of both the DA and AP components. We can explore this further by scrutinising some examples. Table 6.7 shows the actual and predicted labels for the validation set dialogue *Val 302*. This dialogue was chosen because it contains 2 of the 33 utterances where all label types predictions were incorrect, yet agreed.

Table 6.7: Actual and predicted labels for the validation set dialogue *Val 302*.

	Utterances	**Actual**	**Predicted**
USR 1:	I need the time and parties for taking medicine please.	FPP-base request	FPP-base request
SYS 1:	For which one?	Insert choiceQuestion	Insert choiceQuestion
SYS 2:	I have three, one with Jeff at 11 am.	SPP-base inform	SPP-base inform
SYS 3:	One with Alex at 10am, and one with your Mother at 2pm.	Post inform	**SPP-base answer**
USR 2:	That will do just fine, goodbye.	FPP-base goodbye	**Post autoPositive**
SYS 4:	you are welcome	SPP-base acceptThanking	SPP-base acceptThanking

Beginning with turn *SYS 3*, we can see that the prediction of *SPP-base* is invalid because the prior *FPP-base* has already been concluded. This turn should in fact be assigned a *Post* label, because it is a continuation of turn *SYS 2*. For similar reasons *answer* is also incorrect, this turn is a continuation, and additionally the *inform* DA is responsive to the original request for information. The former mistake is certainly reminiscent of the incorrect usage of FPP and SPP we observed amongst our human annotators in section 4.2.5.2. The latter, is likely due to the difficulty of disambiguating these two semantically similar labels, as discussed in 6.1.1. Interestingly, the second most probable label for the AP model is *Post*, for the DA model *inform*, and for the AP-type model *SPP-base inform*.

In other words, the DA and AP models second most likely predictions are correct, and the DA component of the AP-type models is correct.

Without knowledge of the following turn, the incorrect *Post* label for turn *USR 2* would actually be correct, *if* the previous turn was a SPP as predicted. Unfortunately, this also creates a second invalid SPP for turn *SYS 4*. However, for the DA component of turn *USR 2* we find an interesting result. The 'correct' label is *goodbye*, while the predicted label is *autoPositive*. In fact, this utterance is *both*, a signal of positive understanding or feedback, *and* a goodbye. This is a clear example of a multidimensional utterance that comprises several communicative functions (Petukhova, 2011; Bunt, 2006); which several DA annotation schema attempt to model, including DiAML, and SWBD-DAMSL. It is therefore interesting to note that the second most probable label for the DA model was *goodbye*. Further, the probabilities for both these labels are very close, at 32.6% for *autoPositive*, and 29.5% for *goodbye*. Put another way, the model was able to recognise the multidimensional nature of this utterance and 'had difficulty choosing' between the two possibilities.

Utility of Top-k Throughout this section we have highlighted several instances where examining the top-k predictions provided insight into the behaviour of our models. In our view, these reveal several reasons why considering top-k predictions may be useful if such a model is used as a component of a dialogue system – to determine user intents or dialogue state. If the classifier 'confidence' for the predicted label is below a certain threshold, it may provide additional information for the dialogue manager to select an appropriate action. As we have previously shown, on occasion our model assigns invalid SPP labels where there is no previous FPP to conclude. This could be solved algorithmically, to prevent incorrect pair usage. However, if the two or three most probable labels are SPP, this strongly indicates that the utterances in question is *responsive* to a previous utterance, rather than the *initiation* of a new sequence. Similarly, without knowledge of any future utterances, the presence of post-type labels indicates the utterance is associated with the preceding sequence. At the same time, for DA we have shown that the top-k predictions may be used to identify multidimensional utterances. Using turn *USR 2* from table 6.7 as an example, *autoPositive* signifies the user is satisfied with, and understands, the previous response, while *goodbye* expresses their desire to end the conversation, and the system can respond appropriately. Further, there may be several valid response types for particular utterances (Feng et al., 2021; Zhang, Ou, and Yu, 2020), and therefore knowing the 3 most likely label types the model has selected, along with their probabilities, may yield valuable information for other components of a dialogue manager.

6.2 CAMS Multi-labels

For our *multi-label* experiments we explore three different architectures, DA First, AP First, and Parallel, shown in figure 6.5. Each of these use the dialogue segment vector $\boldsymbol{d}$ produced by the context encoder model, and then apply different arrangements of classifier layers. In the case of DA First, classifier layers first generate predictions for the current DA label. The predicted probabilities over all labels, are then concatenated with $\boldsymbol{d}$ and input into the AP classifier layers. Finally, the DA and AP predictions are concatenated with $\boldsymbol{d}$ and passed to the AP-type classifier layers, producing a model that is capable of outputting DA, AP, and AP-type predictions for a given input dialogue segment. AP First performs the same operation, except predictions are first generated for AP and then passed to the DA classifier. In the parallel arrangement, we simply produce DA and AP predictions independently of each other, before concatenation with $\boldsymbol{d}$ for generating AP-type label predictions.

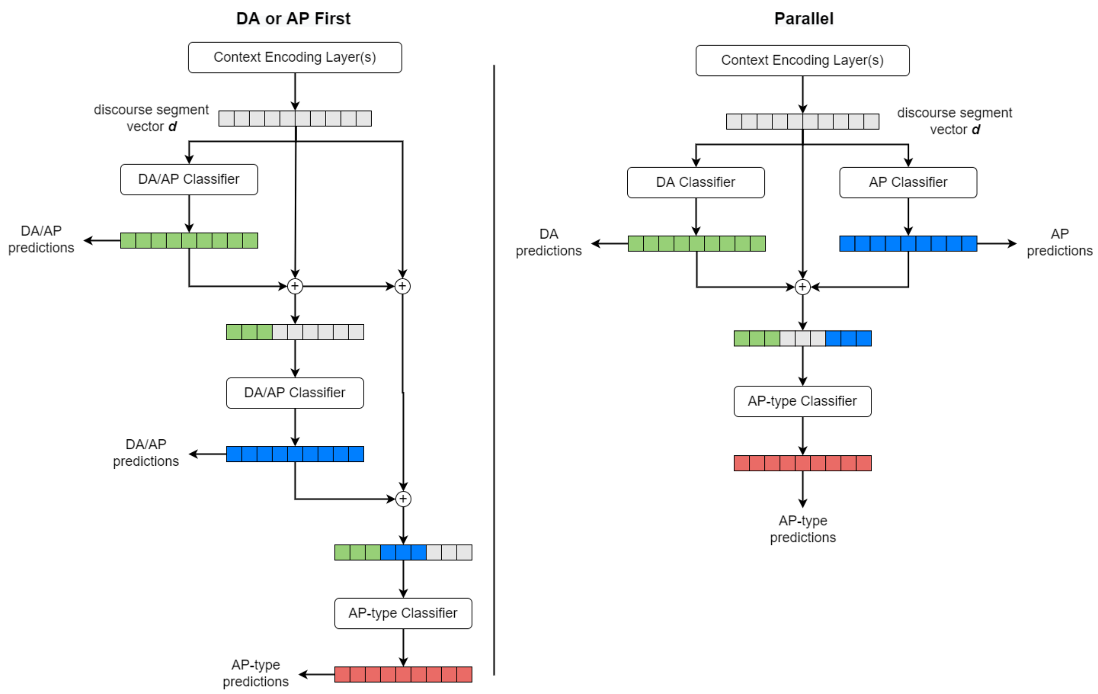


Figure 6.5: Three proposed CAMS Multi-label classification architectures, DA First, AP First, and Parallel. Each arrangement of classifier layers produces a DA, AP, and AP-type prediction for the current target utterance. Intermediate predictions, or top-k predictions, are concatenated with the dialogue segment encoding $\boldsymbol{d}$, before being passed to subsequent classifier layers.

Our intention is to explore various combinations of dialogue segment encodings and current label predictions, to determine whether the addition of probabilities for a given label type are beneficial when predicting *other* label types. In other words, does information about the probable DA labels improve the classification of AP labels, or vice versa, and similarly, do the probabilities for the component labels improve AP-type prediction. Further, because each of these approaches simultaneously produce predictions for the individual DA and AP labels, as well as AP-types, we are able to classify dialogue segments with CAMS labels using only a single model.

For each configuration we may optionally concatenate only the top-k predictions, instead of the predicted probabilities over all labels, allowing the following classifier layers to focus only on the k most likely set of labels. For example, we may pass only the top 3 DA predictions to the AP classifier, thus removing potentially redundant probabilities from the input to the later classification layers. Given that we have consistently found high top-k accuracies, the purpose of the latter approach was to allow the model to focus only on the most probable labels. However, in practice there were only minor differences in performance between the two, and BSR test confirm that they were either statistically equivalent, or inconclusive ($P(rope) \geq .71$). Nevertheless, we found that using the top-3 probabilities resulted in marginally higher performance in most cases, and we report those results here. Table 6.8 shows validation and test set results for each label type and the three different multi-label architectures, using partial context information.[4]

[4]Results obtained using *all* probabilities and only top-3 can be found in appendix G.1.

Table 6.8: Validation and test set accuracy and top-3 accuracy, as well as macro and weighted F1 scores, for each label type. *DA First*, *AP First*, and *Parallel* are the three different multi-label architectures tested, and results shown are obtained using the top-3 probabilities.

	Validation				Test					
Label Type	μ	σ	*top3-μ*	*top3-σ*	μ	σ	*top3-μ*	*top3-σ*	*macro-F1*	*weighted-F1*
					DA First					
DA	93.58	0.31	99.1	0.08	94.46	0.31	99.03	0.1	0.7	0.94
AP	88.88	0.34	97.65	0.29	88.58	0.18	97.83	0.41	0.75	0.87
AP-types	85.43	0.31	93.52	0.56	85.05	0.32	92.47	0.53	0.28	0.81
					AP First					
DA	93.48	0.35	99.11	0.07	94.47	0.48	99.06	0.13	0.7	0.94
AP	88.58	0.3	97.51	0.23	88.37	0.44	97.9	0.47	0.73	0.86
AP-types	85.05	0.39	93.63	0.52	84.7	0.48	92.53	0.51	0.28	0.81
					Parallel					
DA	93.68	0.24	99.09	0.11	94.37	0.31	99.04	0.12	0.7	0.94
AP	88.62	0.18	97.57	0.17	88.34	0.33	97.94	0.53	0.72	0.87
AP-types	85.03	0.16	93.81	0.51	84.76	0.36	92.63	0.39	0.28	0.81

From these results we can see there is only minor performance differences between the three configurations. Both AP and AP-types performed slightly better using *DA First*, and for DA it was the *Parallel* configuration. However, BSR test comparing each configuration confirm that for DA and AP labels results are statistically equivalent in all cases ($P(rope) \geq .85$), and for AP-types equivalent or inconclusive ($P(rope) \geq .74$). With no clear improvement from any of these configurations it is also worth determining if there is any advantages, or disadvantages, to using the multi-label approach as apposed to single-label.

Comparing Single-label and Multi-label Approaches If we compare our single-label partial context results (Table 6.6) to those of our multi-label architectures, we find they are largely the same in most cases. BSR tests confirm that for DA the best multi-label arrangement (Parallel) and single-label validation accuracies are statistically equivalent ($P(rope) = .85$), while for both AP and AP-types, using the DA first arrangement, we see no significant results. For AP we do see a higher macro-F1 score of +.08 to +.11, though considering the static weighted-F1 score this indicates the average per-label F1 for some less frequent labels has *increased*, while for some more common labels it has *decreased*. However, we can explore the difference between these two approaches using two further measures, the number of incorrectly labelled instances and the overall confidence of the models predictions. Beginning with incorrect instances we compare results for the single-label and DA first multi-label approaches:[5]

- The multi-label DA classifier made 108 incorrect predictions, the same as single-label. For AP 187, and for AP-types 251 were incorrect, which is 8 less and 5 more than single-label respectively.
- Again, concatenating the individual DA and AP predictions to form AP-types, we find 41 instances were were incorrect, the same for AP-types, and of these, 26 instances where the single-label model was also incorrect. Thus, we see slightly more overlap of incorrect instances when using a multi-label approach.

[5]We chose DA first because it resulted in the best performance for AP and AP-types, while for DA it resulted in only a single instance difference.

- Comparing the number of instances where the individual components of AP-types (DA or AP) were incorrect, to instances where that separately predicted label type was also incorrect: we find that the AP-types DA component was incorrect on 114 instances, and the AP component on 187, compared to the single-label model, at 105 and 191 respectively. Of the 114 incorrect DA predictions, the the AP-types DA component was also incorrect on 99 occasions (~91%), and of these, 68 predictions were the same as single-label. For the 187 incorrect AP predictions, the AP component was also incorrect on 170 (~91%) of instances, and of these, 138 predictions were the same as single-label.

Overall, as well as a slight improvement for AP labels, we see much more consistency of incorrect predictions between the individual labels and their respective AP-type components. This suggest the model is indeed using the DA probabilities when predicting AP labels, but they are not making a large enough impact to be reflected in our metrics. This is further illustrated when we examine the average probability, or 'confidence', of the chosen label. Table 6.9 shows the overall mean probability for each label type, as well as the average probability for incorrect and correct labels, when using the single-label and multi-label models.

Table 6.9: Mean prediction probabilities for each label type using the single-label and multi-label approaches.

Label Type	**Single-label**	**DA First**	**AP First**	**Parallel**
		All		
DA	94.72	94.35	94.93	93.18
AP	92.08	93.3	92.82	91.57
AP-types	88.27	88.57	88.72	86.12
		Correct		
DA	96.16	95.95	96.42	94.83
AP	94.99	95.75	95.98	94.63
AP-types	92.41	92.57	92.99	90.84
		Incorrect		
DA	73.46	70.68	74.23	68.51
AP	69.58	73.4	70.58	66.98
AP-types	63.72	65.4	64.5	58.84

Firstly, we can see that in each case confidence is lower when using the Parallel configuration and this is likely because the model must consider more information than the specialised single-label models, whilst being unable to make use of the DA or AP probabilities for those label types. Generally there are fairly minor differences between *all* labels and the *correct* predictions. It is noticeable, however, that for AP and AP-types confidence increases over the single-label model when using the DA First configuration, and this is particularly apparent for incorrect predictions. For DA the opposite is true, and confidence increases when using the AP First configuration, though to a lesser degree. Therefore, it appears the multi-label model is making use of the probability information, and in fact the DA First or AP First configurations seem to influence the model in the manor we had surmised. Unfortunately, the effect is so slight it has no noteworthy impact on the resulting classification accuracy. From one perspective this conclusion is surprising, because certain types of DA are naturally associated with specific AP, for example questions and offers with FPP, and answers, accepts, or declines with SPP, and so on. Indeed, this may be why we see a slight performance increase for AP and AP-types when using the *DA First* configuration. Yet, considering our results from section 6.1.2, these findings are perhaps less surprising. If including gold-standard DA labels did not improve AP results, and vice versa, then including their probabilities is also unlikely to improve performance. It may be that these label associations are simply not strong enough, or that the model is focusing on more valuable information within the encoded dialogue segment, and we therefore see little impact on results. Nevertheless, these results do show that our multi-label architecture is capable of predicting all three label types without any loss in performance when compared to single-label models.

6.3 CAMS Classification Summary

Throughout the first part of this chapter we applied our single-label contextual classifier model to the task of AP classification and also investigated the impact that different context information has on the classification of both DA and AP labels. We found that the optimal number of context utterances was the same for both label types, and that these contextual models are better able to disambiguate between certain labels, for example, *inform* and *answer*, or *Insert* and *Post* expansions. However, overall the effect is less impactful for AP, and this is likely because certain labels, such as *Insert* or *Post*, can be mistaken for the completion of a base or insert-pair, because they frequently appear in the same positions within a dialogue.

As with our previous experiments (5.2), the inclusion of speaker and label context information was shown to have negligible impact for DA classification. On the other hand, for AP, including context labels resulted in statistically significant improvements, and the model was better able to differentiate between *SPP-base*, and *insert-pairs*, as well as *Insert* and *Post* expansions. The considerable performance increase for AP suggests that they are more predictive of future AP and thus representative of the overall dialogue structure. In contrast, sequences of DA labels are less indicative of suitable future labels and the semantic content of the utterances themselves is more important for classification.

To conclude our single-label classification experiments, for each CAMS label type, we also compared the impact of using all available context information (including gold-standard labels and future utterances) to using only historical utterances and speakers. For both AP and AP-types this was shown to result in statistically significant improvements, whereas for DA we saw no significant difference. This further supports the observation that AP and AP-types benefit more from the sequential, or relational, information provided by the context of surrounding utterances and their labels.

Investigating the influence of label cardinality and distribution, for AP-types we found large improvements over each of the baseline classifiers, which suggests that the observed improvements for AP are also not due to the number of labels. Indeed, even with partial context inputs, AP-types show greater improvement over the single-sentence baseline than either DA or AP, despite significantly more labels within the set. Comparing the individual utterance predictions across the three label types we were additionally able to show that there is a significant amount of overlap, not only for the instances which resulted in incorrect predictions, but the predicted labels themselves. Thus, classifiers for each CAMS label type tend to make similar mistakes. Most importantly, when predicting AP-types, the model frequently selects the same DA and AP component as a classifier for those individual labels would. When considered with the improvements over baseline classifiers previously discussed, this provides some evidence that an AP-type model is able learn the semantic and structural aspects of both the respective DA and AP components.

In order to determine if predictions for a given label type are beneficial for predicting *other* label types we experimented with various multi-label architectures. However, these multi-label configurations produced statistically equivalent results to those of our single-label classifiers. Once again, we compared individual utterance predictions, and were able to show that our DA First or AP First configurations were influencing the model in the manor we had surmised. Unfortunately, the effect is so slight it has no noteworthy impact on the resulting classification accuracy. Nevertheless, the multi-label architectures are able to simultaneously predict all CAMS label types without any loss in performance.

Chapter 7

Dialogue Structure Evaluation and Analysis

In this chapter we present a collection of experiments intended to evaluate, and analyse, our classification model and proposed method of dialogue representation against several dialogue modelling related tasks, as discussed in 3.3 (**O5**, **Q3**, and **Q4**). In Section 7.1, we begin by applying the single-label models, to the task of *next-label prediction* as a means to evaluate our models ability to predict the next likely DA, AP, and AP-types given the current dialogue state. Then, in Section 7.2, we use a cutting edge generative LM, GPT-3 (Brown et al., 2020), to produce appropriate responses given a segment of dialogue and the next label (DA, AP, or AP-type) as a prompt. Finally, we evaluate the representation of dialogue structure facilitated by CAMS via dialogue graph representations produced with χ^2 analysis of the gold-standard annotations within CAMS-KVRET.

7.1 Next-Label Prediction

For our *next-label prediction* experiments we reuse the contextual single-label models discussed in the previous chapter.[1] Here we simply alter the training objective such that current target for prediction is the *next* label (DA, AP, or AP-type) given the current segment of dialogue, as discussed in 3.3.1. In practical, dialogue management system terms, if we consider selecting the next label(s) as a representation of possible system actions, then naturally we would only need to make predictions for the system turns. Our training objective is therefore considerably more demanding, increasing the number of overall predictions, whilst removing the valuable contextual information of user turns. However, we wish to compare and contrast the performance of each component label type within CAMS when given minimal additional information, and thus our approach is suitable.

Again, we obtain results for 'full context', which uses all available context information, and 'partial context', which only uses context information that would be available to a dialogue system. However, since it is not appropriate to include future utterances, here full context refers to the 5 previous utterances, as well as speaker, DA, and AP labels as input, and for partial context, we use 5 previous utterances and speakers only. Results for each of these measures are shown in table 7.1, and figure 7.1 illustrates the difference between our next-label results and those from our standard classification objective (the three baseline models and contextual RoBERTa-LSTM).[2]

[1] We applied our multi-label model to the task in the same manner, and found that in agreement with our previous findings there was no statistical difference in results, regardless of model architecture or label type. Therefore, here we only discuss single-label results and report multi-label in appendix G.2.

[2] Exact figures can be found in appendix G.2.

Table 7.1: Full and partial context models next-label validation and test set accuracy and top-3 accuracy, as well as macro and weighted F1 scores, for each label type.

	Validation				Test					
Label Type	μ	σ	*top3-*μ	*top3-*σ	μ	σ	*top3-*μ	*top3-*σ	*macro-F1*	*weighted-F1*
					Full Context					
DA	66.03	0.49	91.12	0.39	68.12	0.55	91.73	0.31	0.39	0.65
AP	88.11	0.35	99.34	0.11	86.62	0.67	99.78	0.13	0.62	0.82
AP-types	63.3	0.29	86.97	0.26	64.95	0.5	86.45	0.38	0.18	0.58
					Partial Context					
DA	64.52	1.0	90.23	0.13	66.56	0.68	90.54	0.28	0.37	0.63
AP	81.97	0.24	95.68	0.3	81.11	0.37	95.21	0.28	0.46	0.75
AP-types	60.47	0.85	84.37	0.26	61.92	0.87	84.02	0.43	0.14	0.55

Unsurprisingly these results show a fairly substantial drop in accuracy compared to the standard classification objective. The difference between full and partial context accuracy is much larger, and is now statistically significant for all label types ($P(Full > Partial) \geq .93$), rather than AP and AP-types only. This suggests that previous label information is much more valuable for next-label classification. It is likely that the model becomes less reliant on the semantic content of the current utterance, as it is instead forced to learn the sequential relationships between them. Additionally, the larger difference between full and partial context for AP agrees with our previous findings, that AP benefit more from the context of surrounding utterances than DA. Thus, AP fair better when the inter-utterance features become more important during the next-label prediction task.

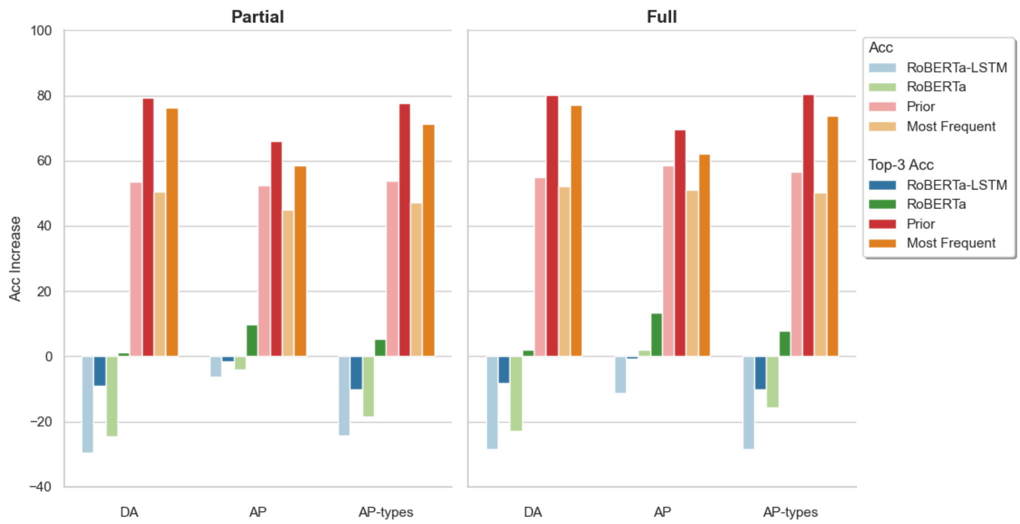


Figure 7.1: Difference in validation set accuracy and top-3 accuracy, between full and partial context next-label models, the baseline classifiers (RoBERTa, prior, and most frequent label), and the contextual RoBERTa-LSTM.

Indeed, as shown in Figure 7.1, when compared to our standard classification results AP exhibit a much smaller reduction in accuracy than DA and AP-types, regardless of context information. With partial context AP remain almost equivalent to the contextual and non-contextual RoBERTa models, with a drop of only ~6% and ~4% respectively. In contrast, DA and AP-types show a $\geq$ 18% reduction in accuracy. That accuracy is reduced further for these two label types, with their greater number of labels, suggests that unlike our analysis of cardinality in section 6.1.3, here the number of labels may be having an effect. However, the reduction for both DA and AP-types is similar, and for partial context it is in fact ~6% higher for DA. Thus, considering there are more than four times the number of AP-types, we must conclude these differences are not solely due to the number of labels. This also suggest that the AP component of AP-types is providing valuable sequential information and improving the accuracy (or reducing the detriment) on the next-label prediction task. Whereas for DA, which benefit far more from the semantic content of a given utterance, rather than the inter-utterance relationships, we see a greater reduction, despite fewer labels. This is further illustrated in figure 7.2 which shows confusion matrices for both DA and AP next-label predictions on the validation set, using full and partial context models. For AP with partial context the behaviour is similar to that of the non-contextual model and is frequently predicting the two most common base-type labels. However, it is still frequently able to distinguish between FPP and SPP for both base and insert pairs. Using full context results in a much clearer picture, and we are left primarily with confusion between *FPP-base*, and *FPP-post* or *Post* labels, and *SPP-base* and *FPP-insert*. In both cases, with no information about the next utterance, these alternatives are perfectly reasonable options and further supports our conclusion that the model is able to learn the sequential relationships between AP. It is also why top-3 accuracy remains considerably higher for AP. On the other hand, for DA, regardless of context, and the model is struggling to learn any form of pattern, or relationship, between the previous utterances and the next likely label.

Finally, top-3 accuracy remains high in all instances, particularly for AP which maintains 99.3% and 95.7% accuracy for full and partial context respectively. This shows that, for all label types, the model is correctly identifying candidates for the next label, but without any further information is simply making the wrong prediction more frequently. As discussed in 6.1.3, this is clearly a useful property for dialogue management systems; especially where only the next system turn would have to be selected in this manner, and where further task-related information would be available, such as slot-value pairs.

7.2 Response Generation

For our *response generation* experiments, our intention is to evaluate the extent to which our proposed dialogue representation method affects the generation of appropriate natural language system responses. Together with the work in the previous section (7.1), these two tasks can be considered key components of a dialogue management system. The former is analogous to learning dialogue policy – the selection of next system actions based on the current dialogue state. While response generation is the NLG component, which given a system action – represented by a DA or AP-type label generated by the dialogue policy – maps the action to a natural language utterance (Dai et al., 2020; Zhang et al., 2020b; Chen et al., 2017). Note that in a complete dialogue system both the dialogue policy and NLG components would likely also include further task relevant information, such as slot-value pairs, whereas here we are only concerned with the 'action', or selected response type, and the content of the generated text.

To generate responses we fine-tune GPT-3 to generate only the system utterances within the CAMS-KVRET test set, with one prompt (dialogue state and response type) per turn. An example prompt is shown in Listing 1. Each "*prompt*" begins with the dialogue type, in this case *scheduling*, followed by the complete dialogue history up to the current system turn, for which the response type is specified (*FPP-insert setQuestion*). The "*completion*" is the ground truth target utterance that should be generated. The generated responses are then evaluated with a range of well-known language modelling and generation metrics (Yeh, Eskenazi, and Mehri, 2021; Finch and Choi, 2020; Sharma

(a) AP Partial Context

(b) AP Full Context

(c) DA Partial Context

(d) DA Full Context

Figure 7.2: Confusion matrices for DA and AP next-label predictions on the validation set using partial context (a and c), and full context (b and d) classification models. Note that here we are only showing the 10 most frequent DA labels.

et al., 2017; Liu et al., 2016), as discussed in 3.3.2.

Word Overlap The *BLEU* (Papineni et al., 2002), and *METEOR* (Banerjee and Lavie, 2005) metrics, were developed to automatically evaluate the results of machine translation given some reference sentences, and *ROUGE* (Lin, 2004) was proposed for evaluation of summaries. Each of these metrics produces a value between 0 and 1, where 1 represents perfect alignment with the reference sentence.[3]

Overall we see only slight differences between each of the label types, and across all metrics. Indeed, for BLEU there is $< .001$ difference in any case. For METEOR and

[3]Full results are shown in appendix H.

```
{
"prompt":"schedule dialogue.
          FPP-base request USR: remind me to take my pills
          FPP-insert setQuestion SYS:  ->",
"completion":" What time do you need to take your pills?"
}
```

Listing 1: Example prompt for GPT-3 response generation.

ROUGE-L, including *any* label does improve the score over no labels in the prompt, with a more substantial increase for DA and AP-types. This seems to suggest that the semantic content, or communicative function, represented by DA provides useful information that the model is able to exploit when generating responses. In contrast, the structural, or sequential, information provided by AP is less useful. It also appears the model is able to take advantage of the DA component of AP-types, because we observe minimal differences between those and DA. Thus, according to these metrics, DA are more beneficial than AP, and yet, AP-types are similarly useful on account of their DA component.

Sentence Similarity In contrast to word-overlap metrics, we can also measure how semantically similar our generated utterances are to the reference utterance. We calculate cosine distance for each sentence pair using the contextual embeddings generated by the RoBERTa-large model, as well as several metrics that use pre-trained GloVe embeddings, average, extrema, and greedy. Results for each of these cosine similarity scores, averaged over all generated responses, are shown in table 7.2, where different label types, or no labels, were included in the prompt.

Table 7.2: RoBERTa contextual embeddings and GloVe Average, Extrema, and Greedy cosine similarity scores, averaged over all generated responses, with different label types included in the prompt.

Label Type	RoBERTa	Average	Extrema	Greedy
No Label	0.996	0.85	0.594	0.74
DA	**0.997**	**0.866**	**0.621**	**0.762**
AP	0.996	0.861	0.612	0.755
AP-types	0.996	**0.866**	**0.621**	0.76

Firstly, we can see that for the contextual RoBERTa sentence embeddings similarity scores are extremely high, and there is minimal difference between any label type. This indicates that, while there may be some difference between individual words, or their arrangement, the semantic content of the generated responses is effectively identical to that of the reference utterance. For the GloVe word embedding similarity scores, particularly for the most commonly used *Average* metric, we again see high similarity for all label types. We also find the pattern is similar to that of our word-overlap metrics. Though there are only slight differences between each of the label types, across all metrics, including *any* label results in an improvement over no labels in the prompt; with a more substantial increase for DA and AP-types. Again, this indicates the semantic representations of DA provide more useful information for the generative model than AP, and yet the model is able to make use of the DA component of AP-types to produce the same effect.

It may therefore be useful to examine some instances where one or more of these label types prompted the generation of utterances that are semantically *dissimilar* to the

Table 7.3: Examples of generated utterances for two dialogues, Test 95 and 203. Speaker and reference utterances are shown in white, and generated responses, for each label type included in the prompt, are shown in grey.

<table>
<tr><th></th><th>DA</th><th>AP</th><th>AP-type</th></tr>
<tr><td colspan="4">Test 95</td></tr>
<tr><td>USR</td><td colspan="3">take me to the parking garage closest to my destination</td></tr>
<tr><td>SYS</td><td colspan="3">The nearest parking garage is Palo Alto Garage R, it's 1 miles away.</td></tr>
<tr><td></td><td>The closest parking garage is Dish Parking at 550 Alester Ave.</td><td>The closest parking garage is Palo Alto Garage R, which is 2 miles away.</td><td>The closest parking garage is Palo Alto Garage R which is 2 miles away.</td></tr>
<tr><td>SYS</td><td colspan="3">Navigate you there?</td></tr>
<tr><td></td><td>Would you like directions?</td><td>Would you like directions there?</td><td>Would you like directions there?</td></tr>
<tr><td>USR</td><td colspan="3">Sure, thank you, just pick the quickest route and send me the address on screen</td></tr>
<tr><td>SYS</td><td colspan="3">I sent the info and the quickest info on your screen, you're welcome!</td></tr>
<tr><td></td><td>You're welcome, I sent the info on your screen, drive carefully!</td><td>Palo Alto Garage R is located at 481 Amaranta Ave.</td><td>The address is 481 Amaranta Ave, I sent it on your screen</td></tr>
<tr><td colspan="4">Test 203</td></tr>
<tr><td>USR</td><td colspan="3">give me the location for the nearest starbucks</td></tr>
<tr><td>SYS</td><td colspan="3">There is no Starbucks listed locally but I have another coffee shop listed would you like the address to it?</td></tr>
<tr><td></td><td>There is a Starbucks nearby, would you like the address?</td><td>There are no Starbucks nearby but we could go to Palo Alto Cafe, it's 2 miles away</td><td>There is no Starbucks nearby, would you like the closest one?</td></tr>
<tr><td>USR</td><td colspan="3">Yes please.</td></tr>
<tr><td>SYS</td><td colspan="3">Then let's go at Cafe Venetia, 269 Alger Dr</td></tr>
<tr><td></td><td>Coupa is located at 394 Van Ness Ave.</td><td>Cafe Venetia is located at 269 Alger Dr.</td><td>The name is Coupa and it is located at 394 Van Ness Ave.</td></tr>
<tr><td>USR</td><td colspan="3">Okay thank you.</td></tr>
<tr><td>SYS</td><td colspan="3">The destination is 4 miles away in no traffic.</td></tr>
<tr><td></td><td>You're welcome and I've set the GPS.</td><td>You're welcome!</td><td>Setting navigation now.</td></tr>
<tr><td>USR</td><td colspan="3">Please set the GPS to avoid all heavy traffic and let's go there, thank you!</td></tr>
<tr><td>SYS</td><td colspan="3">You are welcome!</td></tr>
<tr><td></td><td>You're welcome!</td><td>GPS set to Cafe Venetia</td><td>You're welcome, I sent the info on your screen.</td></tr>
</table>

reference sentence (Miltenburg et al., 2021). Two example dialogues in Table 7.3 show utterances generated with each label type included in the prompt, as well as the speaker and reference utterances. These examples were chosen because they contain utterances among the 10 lowest Average cosine similarity scores for one or more of the label types.

Beginning with Test 95 dialogue, for AP and AP-types the first generated utterance is semantically more similar to the reference utterance ($\geq .97$), than DA (.81). However, It should be noted that locations, times, dates, and other such entities were not removed from the prompts during fine-tuning. Without additional mechanisms, such as a knowledge-base, the model simply generates a 'random' address, hence the difference in similarity scores. It is still interesting to see that AP and AP-types produced the correct address, though it is 1 mile further away! The second utterance is a good example of why we observed relatively low word-overlap scores, responses are generally of the same DA *type*, and semantically quite similar ($\geq .88$), yet with little to no n-gram alignment. Therefore, in agreement with Deriu et al. (2020) and Liu et al. (2016), we would advise *against* using word-overlap measures for evaluation of NLG for dialogue systems. In the final utterance we begin to see some evidence that AP-type labels are prompting the generation of utterances that contain elements similar to those of the individual DA and AP responses. The DA utterance is semantically more similar (.98), while AP simply states the address, which receives a low similarity score (.42). The AP-type utterance (.89) is effectively a combination of these two, with the address information and statement that it was sent to the screen.

In the first utterance of Test 203, both AP (.9) and AP-type (.94) responses receive lower similarity scores than DA (.96), despite correctly stating that there is *no* Starbucks nearby. The second and third utterances are further examples of responses that would produce low word-overlap scores, and despite the incorrect addresses, all six are perfectly valid. Finally, we again see that the AP-type response contains the "You are welcome" of the reference and DA response, as well as a statement "I sent the info on your screen", which is semantically similar to the AP response.

These examples are too few to draw firm conclusions, however, it does appear that AP-type labels may induce the generation of utterances that contain more detail, or information, than those of DA or AP alone. Further, in some instances they may result in an approximate amalgamation of the individual label types. We leave further study of this observation for future work, and in the following section explore the impact of these label types on the generative model itself.

Perplexity We can consider perplexity as a measure of a LMs uncertainty when predicting a sequence, and thus lower perplexity represents lower uncertainty. Table 7.4 shows the average utterance and overall corpus perplexity scores for GPT-3 when generating responses with the different label types, or no labels, included in the prompt.

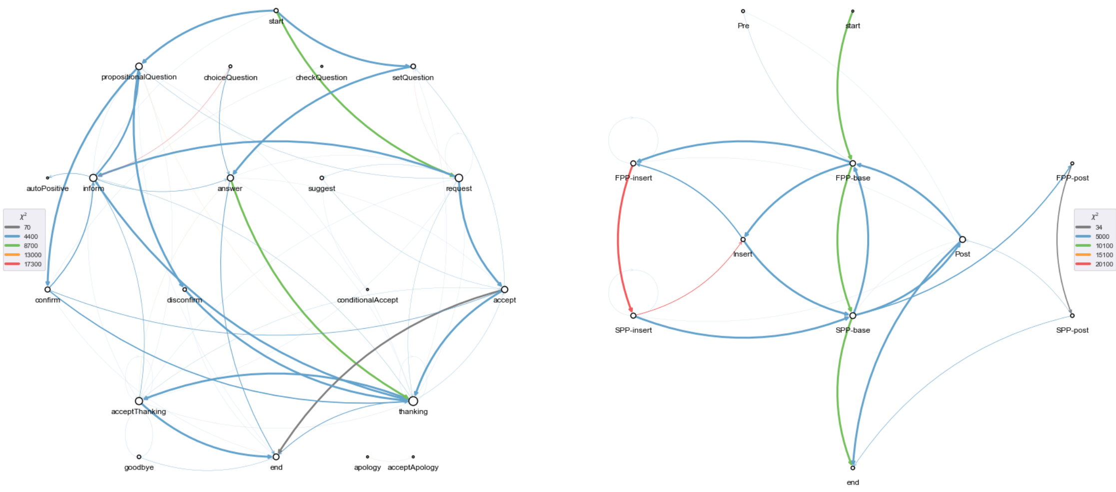
Table 7.4: Mean utterance and full corpus Perplexity scores for GPT-3 response generation with different label types included in the prompt.

Label Type	**P Average**	**P Corpus**
No Label	1.33	73.17
DA	1.33	60.57
AP	**1.28**	**40.04**
AP-types	1.29	45.95

We can see that including any label type reduces perplexity in comparison to no labels. Clearly utterance labels provide valuable information which increases the models certainty whilst generating the sequence of words. However, the perplexity score for the entire corpus is significantly lower for AP and AP-types compared to DA labels, which suggests that AP are even more beneficial to the generative model. This is perhaps surprising, given that DA represent the communicative function of an utterance, one might expect such information to be more useful when generating a single utterance of dialogue. On the other hand, the cardinality of each label type may play a part, with fewer AP resulting in less complexity and lower perplexity. Yet, AP-types still result in an improvement even when their larger number is considered and we see a perplexity score that lies slightly above that of AP, and still well below DA. We speculate that, while *all* label types provide useful information to the generative model, AP provide enough information to stimulate the correct response whilst decreasing uncertainty. With AP-types, as we have observed with our previous classification studies, the model is able to learn the meaning of both components (DA and AP). Thus, despite the increased number of labels, it is able to use the AP component to reduce uncertainty further than DA, and hence we see a perplexity score that lies in between DA and AP. Unfortunately, because the perplexity score is calculated using the $-log$ probabilities for each token, it is difficult to interrogate the model further to determine if this may be the case. Nevertheless, these results are a positive indication that AP and AP-types are more beneficial than DA for reducing the uncertainty of a LM when generating dialogue system responses.

7.3 Analysis of Dialogue Structure

In this section we deviate from the ML considerations of classification and NLG, in order to analyse the structural representation of dialogue provided by CAMS. To do so, we produce graph representations of CAMS-KVRET dialogues using the gold-standard annotations with the modified χ^2 technique presented by Midgley, Harrison, and Macnish (2006), and discussed in Section 3.3.3. Figure 7.3 shows the directed dialogue structure graphs, produced using *all* dialogues within CAMS-KVRET, with the χ^2 data for DA and AP respectively.[4]



(a) DA graph with 20 nodes and 63 edges.

(b) AP graph with 11 nodes and 25 edges.

Figure 7.3: Directed dialogue structure graphs. Nodes are DA or AP labels and edges are transitions from one label to another. Edges are coloured according to the χ^2 value, and widths represent the frequency of occurrences.

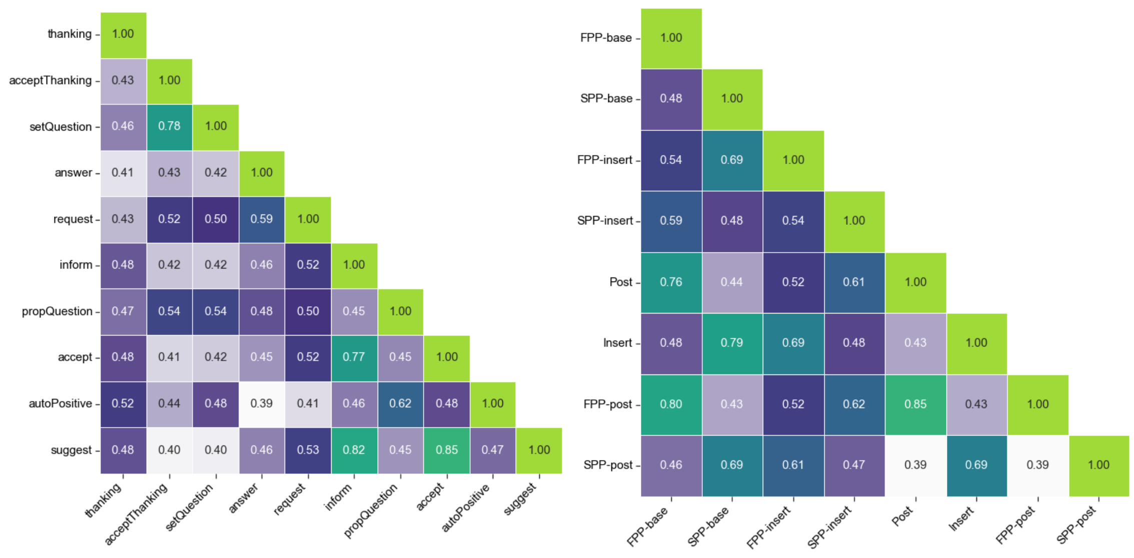
Examining figure 7.3 we can see that, naturally, the DA graph is much more complex than the AP graph, containing almost twice the number of nodes and more than twice the number of edges, though a large number of these occur relatively infrequently and are therefore very faint. However, in both cases looking at edges with the most significant χ^2 values we can begin to see a representation of the overall sequential structure, or flow, of dialogue produced by these two label types. With DA, for example, conversations are predominantly initiated by a *setQuestion*, *propositionalQuestion*, or a *request*. A *setQuestion* is typically followed by an *answer*, a *propositionalQuestion* by a *confirm* or *disconfirm*, a

[4]The AP-type graph, comprising multiple edges for each node, is naturally larger and more complex than our individual label graphs. Thus for conciseness we present the AP-type graph in appendix H.

request followed by an *inform* or *accept*, and so on. For AP, we of course see that most conversations are initiated by a *FPP-base*. On the left hand side the insert pair, or minimal inserts, take place between *FPP-base* and *SPP-base*, while on the right hand side post pairs and minimal posts occur after a *SPP-base*. If we consider a path up to the current node as a representation of dialogue state, we can easily identify all currently adjacent nodes as the next most probable labels. Thus, by reducing the range of all possible labels to a subset of just a few, such a graph could be used to inform a dialogue management system when selecting the next appropriate system action, or interpreting a users current utterance.

We can further explore these dialogue graph representations by considering the similarity between adjacent nodes, or the graphs themselves, and also measuring how efficiently they exchange information.

SimRank Similarity When examining the DA and AP graphs we notice that the nodes of the DA graph are often adjacent to nodes with a very different communicative function. On the other hand, nodes in the AP graph are more 'modular', and are often adjacent to AP of a similar type. For example, the *insert* group of nodes are only adjacent to each other and the base pair, and similarly for the *post* group of nodes. This effect can be measured by means of SimRank similarity (Jeh and Widom, 2002), a measure of structural-context similarity which states that *"two objects are similar if they are related to similar objects"*. SimRank produces a score in the range [0, 1] for any pair of nodes in a graph, where 0 represents the nodes are completely dissimilar and 1 identical. We calculate the SimRank similarity scores between all node (label) pairs within the DA and AP graph, with results shown in figure 7.4.



(a) DA SimRank (b) AP SimRank

Figure 7.4: SimRank similarity values for all node pairs within the DA (a) and AP (b) graphs. Note that here we are only showing the 10 most frequent DA nodes.

Of course, that a given DA is frequently followed by, or adjacent to, a DA with a very distinct communicative function is not surprising. An *information-seeking* question is often followed by an *information-providing* answer, for example. Indeed, this property is partially demonstrated by the similarity scores shown in 7.4a. Examining scores $\geq .75$, we see that *accept* is closely related to *suggest* and *inform*, and therefore these DA frequently reference each other, or are commonly referenced by another DA. Somewhat less intuitively, we also see a high score for *setQuestion* and *acceptThanking*, which implies these two DA are frequently linked by a common subset of DA labels that is not immediately obvious from our graph representation. However, most similarity scores are in the middle of the possible range (.4 to .6), and we cannot say these DA are frequently referenced by, or similar to, any particular DA. Thus, we can infer from the SimRank scores, and the overall higher degree of most DA labels within the graph, that for any given DA there is a wide range of *next* most probable DA, and therefore the current DA is somewhat less informative about possible future utterance types.

For AP, in general we see higher SimRank scores between the label pairs. Perhaps more interestingly, examining scores $\geq .75$ we see a similar pattern to the confusion matrices presented in figures 6.3 and 7.2, for the contextual speaker and next-label prediction experiments respectively. Specifically, *FPP-base* shows a high similarity to *Post* and *FPP-post*, and *Insert* a high similarity to *SPP-base*. This demonstrates that our classifier models were correctly identifying patterns that exist within the actual data. For example, that *Insert* frequently occurs in a similar position within the dialogue as *SPP-base*, and so on. Further, for AP, and unlike DA, node pairs with a high SimRank similarity may also represent viable alternative dialogue paths, or next-label options.

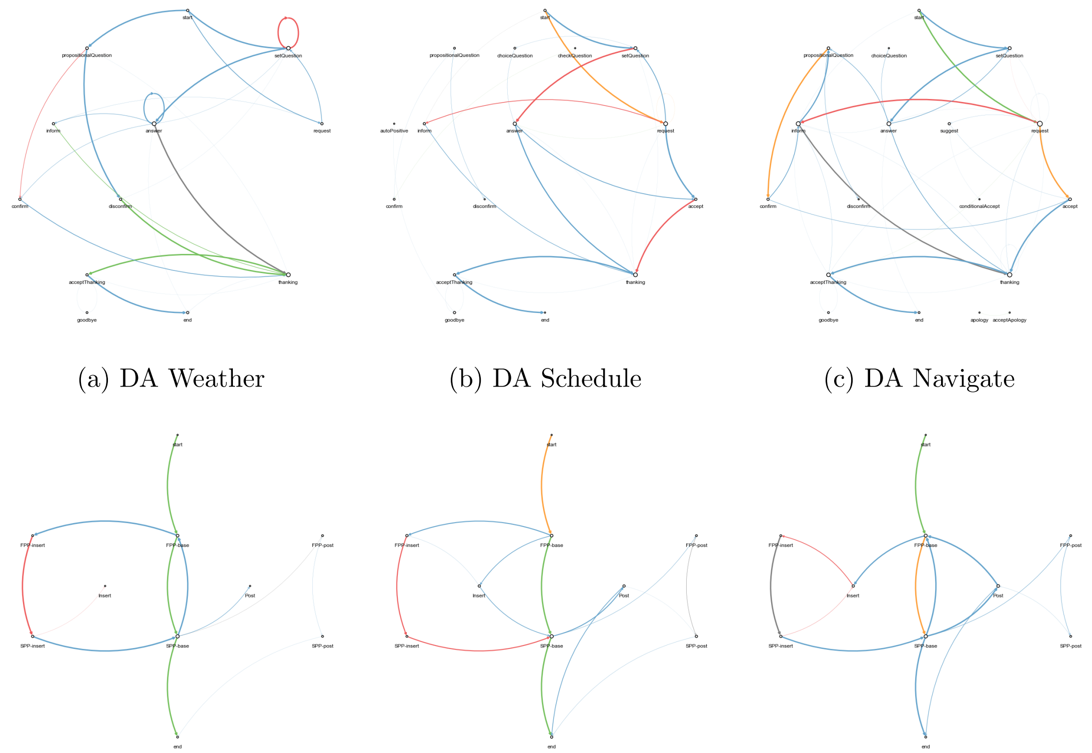
These observations further support our previous findings for contextual speaker and label information (6.1.2), and next-label prediction experiments (7.1), where AP performed better than DA because the model was able to make use of historical context information and learn the sequential relationships between AP labels. In contrast, for DA we see less well defined relationships, or sequential patterns, and thus the models were less able to utilise contextual information, particularly for the much more difficult next-label prediction task.

Similarity Edit Distance In addition to node context similarity, we can also measure the similarity between any two graphs using the Exact Graph Edit Distance algorithm proposed by Abu-Aisheh et al. (2015). The edit distance algorithm determines the minimum sequence of edit operations on nodes or edges (insertion, deletion, or substitution), in order to transform one graph into another. Of course, conducting such an analysis on graphs generated from different label types, with different label-nodes and edge relations, is of little value. Therefore, using the same χ^2 analysis approach, for each label type we generate graphs for the different dialogue tasks within CAMS-KVRET, *Navigate*, *Schedule*, and *Weather*. The resulting graphs for DA and AP labels are shown in figure 7.5, and provide us with view of the differences, or similarities, in dialogue structure representation for each label type over the three distinct tasks.[5]

When comparing the three tasks, we can see that the DA graphs are generally more distinct than those of AP, though some patterns are present in both sets. For AP, with the exception of *Pre*, all nodes are present within each of the three graphs, and these are connected with a similar pattern and weighting of edges. On the other hand, for each of the DA graphs a different set of nodes is present and there is a marked difference in the pattern of weighted edges connecting them. This observation is confirmed by calculating the Similarity Edit Distance between each combination of task graphs. The results in table 7.5 clearly show that the AP graphs are much more similar, requiring 5 to 7 edits to transform one task graph into another, while DA require 23 to 34. In $^2/_3$ comparisons the AP-type graphs are also more similar than those of DA.

In terms of dialogue management, in our view, highly similar or distinct graphs can fulfil two different purposes. Dialogue structure representations that produce very simi-

[5]The AP-type task graphs are available in appendix H.



(a) DA Weather (b) DA Schedule (c) DA Navigate

(d) AP Weather (e) AP Schedule (f) AP Navigate

Figure 7.5: DA and AP directed dialogue structure graphs for each dialogue task type within CAMS-KVRET. Nodes are DA or AP labels and edges are transitions from one label to another. Edges are coloured according to the χ^2 value, and edge widths represent the frequency of occurrences.

Table 7.5: Similarity Edit Distance for the minimum sequence of node and edge edit operations needed to make the dialogue task graphs isomorphic for each label type.

Label Type	**Navigate Schedule**	**Navigate Weather**	**Weather Schedule**
DA	29	34	23
AP	6	5	7
AP-types	19	33	26

lar graphs across different tasks provide a more uniform and generalisable representation which would require less hand crafting or fine tuning in order to adapt to new domains. In contrast, highly distinct graph representations could aid in determining the long-term user intents – the dialogue type or topic – by calculating the similarity between the current dialogue and those of different types within the training data. This kind of topic identification could also be used to further reduce the number of viable next-label types, by reducing the probability, or even removing, labels that are infrequently used within that dialogue type. As much of our previous work has shown, it appears that AP-type graph representations are to some extent a fusion of these two properties. That 2/3 comparisons of the AP-type graphs are more similar to each other than the DA graphs, despite the presence of significantly more edges, suggests a more generalised structure provided by fewer nodes and more uniform AP relations. At the same time, edges representing DA transitions maintain the task specific benefits discussed above, without dramatically increasing the complexity or diversity between different dialogue types.

Finally, these results further support the premise that underpins our proposed approach, that DA labels are descriptions of the *intra-utterance* features of a dialogue, while AP represent the *inter-utterance* features. The diversity amongst the DA graphs of different dialogue types indicates that they are more sensitive to small variations within the flow of dialogue at the local level. On the other hand, the similarity amongst the AP graphs indicates they are less sensitive to small local variations, and therefore better suited for a global representation of dialogue.

Efficiency Expanding on the notion of global and local properties for these different label types, we can also measure the efficiency of the graph representations using metrics proposed by Latora and Marchiori (2001). The *efficiency* of a graph is a measure of how efficiently information is exchanged, and we calculate both global and local efficiency, as discussed in 3.3. Results for each label type, over all task and full dialogue graphs, are shown in table 7.6.

Table 7.6: Global and Local efficiency of each label type, for all task and full dialogue graphs.

	Navigate	**Schedule**	**Weather**	**All**
		Global		
DA	.397	.280	.548	.5
AP	.641	.66	.579	.711
AP-types	.428	.594	.561	.693
		Local		
DA	.333	.468	.567	.554
AP	.407	.459	.0	.663
AP-types	.527	.576	.0	.544

Firstly, we can see that AP and AP-types are globally more efficient than DA across all graph types. This is intuitive if efficiency is inversely proportional to the shortest distance between two arbitrary nodes, then graphs with fewer nodes will tend to be more efficient. However, again the AP-type graphs have not been adversely affected by the presence of significantly more edges, with global efficiency values that lie in between those of DA and AP. For local efficiency the picture is rather mixed, with no label type resulting in consistently higher efficiency scores across all graph types. Though, it is worth noting that for AP and AP-types the local efficiency for the *Weather* task is 0, likely due to a significant number of nodes with fewer than two neighbours. What we can say however, is that for the remaining two task graphs AP-types are more locally efficient than DA or AP individually.

Again we find evidence that AP, and by extension AP-types, provide a more robust global, or *inter-utterance*, representation of dialogue. Regarding global efficiency, the AP-type values further supports our previous observations that AP-types successfully combine the properties of their constituent label types. If we consider local efficiency to be a measure of fault tolerance – the efficiency of communication between the neighbours of i when i is removed (Latora and Marchiori, 2001) – then these results suggest that

DA are locally more robust between small subsets of labels, purely on the basis of non-zero values for each graph type. However, we do not consider this property relevant for our purposes, as disconnected or sparsely connected nodes would not adversely affect the graphical dialogue representation applied to dialogue management.

Chapter 8

Conclusion

This chapter presents a summary of our work by reflecting on the objectives, research questions, and evaluating the hypotheses established in chapter 1. Subsequently, an outline of the limitations, possible applications of our findings, and future directions of this research is provided.

8.1 Evaluation of Objectives and Research Hypotheses

The findings and results discussed in the previous chapters demonstrate that we have successfully fulfilled the research objectives. Accordingly, with reference to these objectives we may now evaluate the research hypotheses established in chapter 1.

8.1.1 Hypothesis 1

> *CA theories on the structure of dialogue can be incorporated with DA as a method of effectively representing task-oriented dialogue for computational modelling purposes.*

Through our review of previous work on CA and DA theory we were able to establish a suitable approach to unify these two fields into a unique method of dialogue representation. In chapter 4 we outlined our approach and discussed the components, and annotation guidelines, of the schema we developed, the CAMS. The schema was then evaluated by means of an annotation study, that was conducted with novice annotators, assessed via two IAA coefficients, and several other quantitative and qualitative measures. Results of this study enabled us to examine the *reproducibility* of the schema and determine that it is inherently learnable, even for novice annotators. Though, we also found some aspects were misunderstood by our participants, which highlighted some refinements that should be made. With the development of CAMS, and subsequent evaluation study, we have therefore met **O1**. This study additionally allowed us to address **Q1**; Both annotator agreement and self-reported annotator confidence scores were higher for task-oriented dialogues than non-task. Although, when considering the individual corpora we found the KVRET corpus resulted in higher agreement and confidence scores than the other 3. Therefore, while CAMS may be applied to both task and non-task dialogues, we consider that it is more intuitively applied to task-oriented dialogues.

In order to meet **O2** we annotated a suitable task-oriented corpus with CAMS, using specially developed software which enables the assignment of a single DA and AP, to form an AP-type for each utterance of dialogue. We selected the KVRET corpus because it possesses several features that were advantageous for our purposes, including, three distinct tasks within the in-car personal assistant domain, and slot and task information for developing task-oriented dialogue systems. The resulting corpus, CAMS-KVRET, is a first-of-its-kind DA and AP annotated corpus which supported our work in meeting the remaining objectives.

With the completion of the above objectives we are therefore able to confirm **H1**. We have successfully incorporated the CA concept of AP with DiAML into a single annotation schema. Our work has shown that this schema is well suited to task-oriented dialogue, and our annotated corpus is compatible with computational dialogue modelling.

8.1.2 Hypothesis 2

> *Existing text classification methods can be adapted to automatically label task-oriented dialogues with DA and AP structure.*

After reviewing previous work on ML methods for DA classification in chapter 5, we explored the impact of text pre-processing and representation parameters, such as sequence length, vocabulary size, and embeddings, for several supervised sentence encoding models. Comparing the supervised approach to that of pre-trained language models, we found that several of the latter resulted in superior sentence encodings. Building on this work we explored *contextual* DA classification models, and considered different forms of context input, such as previous and future utterances, speakers, and labels. These experiments enabled us to refine and optimise our ML architecture, and method of input representation, for the task of DA classification.

In chapter 6 we applied our contextual RoBERTa-LSTM model to the task of AP and AP-type classification, to the best of our knowledge the first study of its kind. As we had surmised, we found that the influence of context utterances is consistent for both DA and AP. However, we were also able to show that for AP, unlike DA, the inclusion of context labels resulted in significant accuracy improvements. Thus, we found some confirmation that AP are able to represent the structural, *inter-utterance*, features of dialogue. In contrast, the surrounding labels are less indicative of the current DA, and instead the semantic, *intra-utterance*, features are more important.

Finally, using our contextual model as a basis, we explored several novel multi-label architectures that are capable of simultaneously classifying segments of dialogue with DA, AP, and AP-types. We were able to show our architectural motivations were correct – that including predictions for a given label type are beneficial when predicting *other* label types. Unfortunately, the effect was minimal, and classification accuracy for these models was equivalent to our single-sentence models. Nevertheless, both our single-sentence and multi-label models were able to achieve good classification results across all label types within CAMS. This work successfully meets our requirements for **O3**, and in doing so we have also addressed **Q2**.

Regarding **O4**, throughout section 5.2 and chapter 6 we evaluate our classification models against CAMS-KVRET. We achieved high test set classification accuracy of ~94% on the DA component of the corpus. This is significantly higher than the other corpora we tested, SwDA and Maptask, where our results are comparable with, or exceed, those of previous work. For AP we are unable to make direct comparisons with any previous studies or corpora. However, we were again able to report high test set classification accuracy for both AP (~88%) and AP-types (~84%). The latter result is particularly impressive considering there are significantly more AP-type labels than DA or AP. This indicates that the corpus is well annotated, with the semantic content of utterances highly correlated with appropriate DA labels, and their sequential relationships captured by AP. We also compared predictions of single-sentence models for each label type and found that there was significant overlap of the incorrectly predicted instances and the labels themselves. This demonstrates that, when predicting AP-types, the model learns the semantic and structural aspects of both the DA and AP components. In evaluating the CAMS-KVRET corpus using our contextual classifiers, we are able to show that the component labels within our schema can be successfully identified using ML methods, thus addressing **Q3**.

In meeting **O3** we were able to extend and refine existing DA classification techniques to produce single and multi-label architectures. Evaluating our models against CAMS-KVRET, to fulfil **O4**, confirms that they are suitable for our task of automatically labelling task-oriented dialogues with AP and DA. Therefore, we are able to confirm **H2**.

8.1.3 Hypothesis 3

Dialogues labelled with DA and AP provide a more syntactically and semantically rich method of dialogue representation than existing methods.

In chapter 7 we evaluate our classification model and proposed method of dialogue representation from the perspective of dialogue modelling. The first two tasks, next-label prediction and response generation, represent key components of a dialogue system – learning dialogue policy for response selection and NLG. Both AP and AP-types performed better than DA on the next-label prediction task. We were able to show that this was not solely due to fewer AP labels, but in agreement with our previous results, because AP are more predictive of *future* labels than DA. Additionally, a smaller reduction in accuracy for AP-types, despite the increased number of labels, provides further evidence that our model learns aspects of both component labels.

For response generation, utterances were generally similar to the ground truth utterances, although we found no significant differences between different CAMS label types and their ability to induce quality responses. However, including *any* label type resulted in improvements across all metrics. Notably, we found that AP and AP-types significantly reduced perplexity compared to DA, or including no labels in the prompt. We speculate that AP provide enough positional information to stimulate the correct response from the model, whilst also reducing complexity and hence lowering perplexity. This is a positive indication that AP and AP-types are more beneficial than DA for reducing the uncertainty of a language model when generating dialogue system responses.

Finally, we used χ^2 analysis to produce dialogue structure graphs and evaluated the representations produced by the different label types within CAMS. We evaluated these graphs using two measures of similarity: SimRank and Graph Edit Distance, and also measured their efficiency, in terms of how information is exchanged. In agreement with results from our classifier studies, the SimRank scores for DA showed few substantive relationships, and thus, for any given DA there is a large number of possible *next* DA, making them less informative of future utterance types. In contrast, for AP we observed a similar pattern to our classifiers predictions, and therefore our models were correctly identifying patterns that exist within the actual data. Considering the Graph Edit Distance, we found that DA graphs for different tasks were highly distinct, AP were very similar, and AP-types lie in between the two. We suggest that, graphs which are similar across different tasks would provide a more uniform and generalisable representation of dialogue, which would require less hand crafting or fine tuning in order to adapt to new domains. Whereas highly distinct representations could aid in determining the long-term user intents, the dialogue type, or topic. Of course, our CAMS representations are able to capture both of these features. In terms of efficiency, both AP and AP-types were found to be *globally* more efficient than DA across all task types, and further evidences their suitability as *inter-utterance* representations of dialogue. With the measure of *local* efficiency between small subsets of labels we found DA were more consistent, and robust.

The above summarises our evaluation of the proposed approach and completion of **O5** In so doing, we have discussed a series of characteristics that are potentially, or directly, beneficial for dialogue modelling tasks. Thus, we are also able to confirm **H3**.

8.2 Challenges and Limitations

Throughout this work we encountered difficult challenges that we discussed in the previous chapters. Some of these challenges are summarised in the following points, though it should be noted that others remain as work to be further studied and examined in the future.

Corpora and Model Generalisability Supervised text classification approaches such as ours require large quantities of labelled data. However, it is extremely costly, in terms of time and resources, to produce dialogue corpora and validate the accompanying annotations. The complex and diverse nature of dialogue adds further considerations which define the scope of the corpus, such as the number of participants or topics, and the interactional context ('spontaneous' or goal-oriented, human-human or human-machine, and so on). By selecting KVRET as the basis for our annotated corpus we were able to mitigate some of these potential limitations because it was created by human participants emulating a human-machine scenario, and it contains multiple topics. Nevertheless, the scope of CAMS-KVRET is limited, as all corpora are, and we therefore cannot say how generalisable our proposed method of dialogue representation is to other domains. For example, through our annotation study reported in 4.2 we were able to show that CAMS is more intuitively applied to task-oriented dialogues, yet agreement scores for KVRET were consistently higher than the other task-oriented corpus. By extension, we also cannot be certain how generlisable our ML classifiers are either. We attempted to mitigate the potential for overfitting the model to a single dataset by including the SwDA and Maptask corpus throughout much of our development process. Yet, as our work in chapter 5 has shown, not all input pre-processing and architectural decisions effect all corpora equally. For example, including context speakers resulted in significant accuracy improvements for the Maptask corpus, but not SwDA.

The issue of how comprehensively a dataset encapsulates a particular phenomenon, certainly regarding dialogue (Enayet and Sukthankar, 2022), and the generalisabilty of models trained on said data, is common to all supervised machine learning endeavours. However, we must acknowledge that the scope of our findings is limited by the issues discussed above.

Association of DA and AP As discussed in 2.1.1, the concept of *typed* AP already exists within the CA literature. The basis for our proposed representation attempts to codify this concept by unifying two well established, yet disparate, methods of dialogue representation into a single computationally compatible approach, which subsequently benefits from the strengths of its DA and AP components. We speculated that one characteristic of our approach would be clear associations between certain DA and AP, for example, questions and offers with FPP, and answers, accepts, or declines with SPP. Indeed, examining the AP-type graph (in appendix H) we can see some evidence for this, because certain DA (edges) are more frequently connected to particular AP (nodes). However, from a ML perspective this characteristic was not as evident. In section 6.1.2 we tried including AP labels as additional contextual input when predicting DA labels, and vice versa. Yet, in both cases this was shown to make no statistically significant difference to classification accuracy. Further, all of our multi-label architectures, discussed in 6.2, were largely predicated on the assumption that this association would provide information beneficial for classification. While we were able to show that the DA First and AP First configurations were likely utilising the input predictions, the effect was too slight to improve performance. The results of these two experiments seem to indicate that the association between opposing label types is simply not beneficial for ML classifiers. It could be that the dialogue segment encodings alone are of high enough quality – with clear separation between the classes – that prior label predictions are simply not needed; high classification accuracy for all label types supports this conclusion. Alternatively, our method of incorporating the information within the model may need refining. In any case, in terms of ML classification at least, our speculation that associations between label types would be advantageous was incorrect. Fortunately, this finding does not undermine our approach, as both components of AP-types are intended to represent different features of dialogue, and can be applied, identified, and interpreted independently.

8.3 Applications and Further Work

Regarding future research, there is potential for several significant extensions and refinements of the current study. We also outline some of our recommendations for the utilisation of our approach to task-oriented dialogue management, as well as possible applications outside of this domain, the analysis of dialogue itself.

Development of CAMS and Corpus Resources The findings of our schema evaluation study, in 4.2, highlighted some common mistakes made by our novice annotators. In part this may be due to the limited time and training they received. However, there are some refinements that could be made to the schema and annotation tool to aid decision making and mitigate common errors. Primarily we found annotators were often misusing FPP and SPP, for instance, beginning a sequence with a SPP. Although this label naming convention adheres to CA practices, changing these to 'First-part' and 'Second-part' (or simply 'First' and 'Second'), would signify the functional difference between the two. The interface of the annotation tool itself could also be updated to prevent the creation of incorrect sequences, in most cases. For example, prohibiting the assignment of a SPP if there is no previous FPP to complete, or ensuring pre and post labels occur prior to, or after, base pairs. The schema annotation guidelines should additionally be updated to include examples, and instruction for identifying, the core action or communicative goals of a dialogue, or deciding appropriate labels for utterances with ambiguous, or multiple valid meanings. Regarding the last point, in 2.4 we discussed prior work on the multidimensionality of DA (Petukhova, 2011; Bunt, 2006). Within this work we chose to limit the scope to only one 'dimension', or meaning, per DA. However, in 6.1.3 we were able to show that, for utterances with multiple meanings, the associated DAs frequently appeared within the top-k predictions of our model. Thus, future iterations of our schema and ML models could be adapted to incorporate multidimensional DA representations. Indeed, this functionality is already supported within DiAML via the 'dimension' categories, such as Task, Time Management, Turn Management, and so on. These facilitate the categorisation of a DA, or the functional segments of a DA, into various dimensions, according to their communicative function (British Standards Institution, 2012). For example, in the utterance "OK, let me check that for you", the segment "OK" relates to the Auto-feedback dimension, while "let me check that for you." concerns the Task dimension. An additional aspect of DiAML that could be incorporated into the schema, is the concept of functional and feedback dependence relations, which relate a given DA to previous utterances in the dialogue, and are therefore closely related to the concept of AP. For example, to indicate which question is being answered, or which utterance the speaker is providing feedback for (Bunt et al., 2012).

Because KVRET was included within our evaluation of CAMS, we also carried out a cursory evaluation of our annotated corpus. Throughout that work KVRET consistently reached higher IAA than the other corpora, for both novice and expert annotators. However, only a small selection of dialogues were annotated; although these are representative of the wider corpus. Therefore, a further IAA study should be conducted, preferably with expert annotators, to verify the reliability of the labelled data and produce an IAA score for the entire corpus.

Finally, as mentioned above (8.2), a single corpus annotated with CAMS labels does not allow us to draw firm conclusions about how generalisable our representations are. Annotating further corpora, of differing domains and contexts, would allow us to further explore the efficacy of our approach and ML classification models.

Task-oriented Dialogue Modelling Of course, one of the primary motivations for this research is facilitating dialogue management for CAI. However, developing such a system, with sufficient rigour, was beyond the scope of this work. Going forward, we would like to apply our findings to developing a task-oriented dialogue system that incorporates our CA inspired method of dialogue representation. Considering the different methods of implementing CAI discussed in chapter 1, a rule-based approach would not be appropriate. After all, their dialogue model is entirely defined by the developer. However, both hybrid and ML approaches are capable of integrating a computationally compatible dialogue model, and are therefore suitable candidates.

Beginning with hybrid, we can envision a developer constructing branching dialogue 'paths' that are reminiscent of our AP-type dialogue structure graph. In this scenario, AP *nodes* represent the current utterance position (and may also contain further information such as named entities), and DA *edges* represent types of transition. The current dialogue state could also be viewed as the current path through the graph (or sub-graphs) up to the current node. A developer could assign different behaviour paths, depending on the type of DA that is produced. For example, if the response to a *FPP-base setQuestion* is another question, instead of an answer, it may be a request for clarification or further information. Thus, a systems behaviour and response should change accordingly. What is more, this behaviour could be informed by our empirically derived probabilities of transitions between nodes, for instance, the transition from a *FPP-base* is much more likely to be a question than an answer.

Considering a ML approach, as we have already shown through our next-label prediction (akin to response selection) and response generation experiments, our proposed method of dialogue representation is beneficial for these two key components of any dialogue system, dialogue policy learning and NLG. In the first case, AP and AP-types were shown to be better predictors of the *next* label (or action), when the utterance itself is not known. Further work could improve performance of this component, by incorporating knowledge from our χ^2 analysis and associated graphs, to inform appropriate, or likely, next system actions. Regarding NLG, we showed that AP and AP-types reduced the perplexity of the generative model when compared to DA, without any detriment to the quality of the generated utterances. And further, that utterances generated with AP-type prompts often combine the characteristics of responses induced by the constituent label types; which often results in more informative or grammatically pleasing responses. A DST component could also make use of our representations, either by making use of the graphical form, as in our discussion of a hybrid application, or simply incorporating the sequence of AP-types into the representation of the current dialogue state. This may have the additional benefit of integrating an element of *explainability* into, otherwise often opaque, ML dialogue systems. It is easy to see how one, or all, of these components could be incorporated with other features of dialogue management, such as named entity recognition and a knowledge-base, to produce a task-oriented dialogue system that makes use of the natural conversational structures of human interaction.

Beyond the domain of CAI there are several other applications to which the proposed research may be relevant. Firstly, Speech-generating Devices (SGD), also known as augmentative and alternative communication devices, are computer-based systems that provide an electronic voice for individuals unable to speak (Cook, Polgar, and Encarnação, 2020). For example, the ACE-LP (Augmenting Communication using Environmental Data to drive Language Prediction) project, which aims to combine environmental data (gathered through cameras, microphones, and other sensors), with conversational language models, to automatically populate the SGD with appropriate conversational items (Black et al., 2016). For this application, a computational model of dialogue, based on CA, has some clear benefits. For instance, the language model would already be closely related to natural human-human conversational structures, appropriate next-responses could be selected via the AP-type relationships, and so on. Our work may also be more generally applicable to projects such as Multi3Generation (Barreiro et al., 2022), a network of researchers exploring topics related to NLG, including data and information representation and ML for structured prediction and representation learning.

Analysis of Dialogue In contrast to the text or speech producing applications of CAI and SGD, when paired with an automated method of identification, the CA model may also aid in the analysis of dialogue structure itself. Our directed dialogue structure graphs, inspired by the work of Midgley, Harrison, and Macnish (2006), illustrate frequent, or likely, conversational 'paths', and identifies pairs of DA that are often produced consecutively. Similar work by Pareti and Lando (2019), identified the intents of the conversation independently from the semantic representation. The proposed CA model, with its richer representations, could similarly be applied to dialogue intent-induction, sentiment analysis, or dialogue clustering (Maitreyee, 2020).

Thus far we have described applications within the CAI, NLG domains, but our approach could be equally valuable within fields of CA, or Discourse Analysis, for post hoc evaluation of dialogue. For example, Pilnick et al. (2018) used CA to identify interactional practices of particular patient groups in order to inform simulated patients for the purpose of communication skills training for healthcare professionals. And Meredith (2020) examined the differences between online and offline interactions using CA for sequential and discursive analysis. In relation to these kinds of studies, and as our work on dialogue structure graphs has shown, the proposed representations can be used to identify global structures common to all dialogues within a given domain, or to identify distinct sub-groups. Or alternatively, for a more general identification of conversational discourse types (Biber et al., 2021). Additionally, our methods of automatic annotation could reduce the labour required for such studies. For instance, Motozawa et al. (2021) manually annotated AP for the purpose of analysing children's intercultural communication, for those that speak low-resource languages (in terms of machine translation). Indeed, DA classification has already been applied to aid post hoc analysis. Lee et al. (2019) classified DA to identify therapist conversational actions and gain an understanding of conversational flow during therapy sessions. Thus, a generalised method of identifying DA *and* AP may be useful for a wide range of sociological and psychological studies.

The above points have highlighted some applications and future directions for our research. It is our hope that this approach to dialogue modelling may have applications, not only for dialogue systems and HCI, but also other fields that might benefit from the identification and representation of dialogue in a human-centric format.

Appendices

Appendix A

CAMS Label Definitions

CAMS is intended to combine concepts of DA and AP into a single annotation scheme that is able to capture the semantic and syntactic structure of a dialogue at the *inter* and *intra* utterance level. Additionally, AP and DA may be applied to any type of conversational interaction, independent of domain and topic, and as such, the schema is entirely domain agnostic and applicable both to task and non-task-oriented dialogues.

The schema defines two sets of labels, DA and AP, which are combined to form AP-type labels. When applying the schema, the intent is to assign each utterance of a dialogue one DA and one AP label, which together are considered the AP-type label for that utterance. The AP-type labels, for a fully annotated dialogue, can then be viewed as a representation of its semantic and syntactic structure. It should be noted that the concept of a *typed AP* is a key feature of AP present within the CA literature (Clift, 2016; Sidnell, 2010; Liddicoat, 2007; Schegloff, 2007). However, the standard annotation schemes for CA do not strictly require each utterance of dialogue to be labelled with an AP. Additionally, CA annotation often includes non-verbal sounds, pauses and other types of disfluencies. Gaps in annotations, where utterances are not labelled with AP, and other forms of non-verbal annotation, for example 'breathing', are generally undesirable for computational purposes. CAMS, therefore, is an attempt to define these concepts, and how they may be applied, into a computationally compatible format where each utterance is labelled with an AP-type.[1]

A.1 Adjacency Pairs

Adjacency pairs (AP) are the basic units on which sequences in conversation are built. Their core features are:

1. Consist of two turns (utterances) by different speakers.
2. Placed next to each other in their basic (unexpanded) form.
3. Are ordered, so that one always occurs after another. Initiation of a sequence is a 'First Pair Part' (FPP) and the response 'Second Pair Part' (SPP).
4. Differentiated into AP-types. The relationship between FPP and SPP is constrained by the type of FPP produced. For example, a 'question' followed by an 'answer'.

[1]Label definitions and annotation guidelines are also available at: nathanduran.github.io/Conversation-Analysis-Modelling-Schema.

A.1.1 Base Pairs

The basic sequence is composed of two ordered turns at talk, the FPP and SPP. Participants in conversation orient to this basic sequence structure in developing their talk and AP have a normative force in organising conversation, in that, AP set up expectations about how talk will proceed.

Labels: *FPP-base*, *SPP-base*

Example:

A:	What time is it?	***FPP-base***
B:	Three o' clock.	***SPP-base***

A.1.2 Expansions

Expansion allow talk which is made up of more than a single AP to be constructed and understood as performing the same basic action and the various additional elements are seen as doing interactional work related to the basic action under way (Liddicoat, 2007). Sequence expansion is constructed in relation to a base sequence of a FPP and SPP in which the core action under way is achieved. There are three types of expansion pairs:

Pre-expansions

Pre-expansions are designed to be preliminary to some projected base sequence and are hearable by participants as preludes to some other action.

Labels: *FPP-pre*, *SPP-pre*

Example:

A:	What you doing?	***FPP-pre***
B:	Not much.	***SPP-pre***
A:	Wanna drink?	***FPP-base***
B:	Sure.	***SPP-base***

Insert-expansions

Insert-expansions occur between base adjacency pairs and separates the FPP and SPP. Insert-expansions interrupt the activity previously underway but are still relevant to that action and allows the second speaker (who must produce the base SPP), to do interactional work relevant to the base SPP. Insert expansion is realised through a sequence of its own and is launched by a FPP from the second speaker which requires a SPP for completion. Once the sequence is completed the base SPP once again becomes relevant as the next action.

Labels: *FPP-insert*, *SPP-insert*

Example:

A:	Do you know the directions to the zoo?	***FPP-base***
B:	Are you driving or walking?	***FPP-insert***
A:	Walking.	***SPP-insert***
B:	Get on the subway…	***SPP-base***

Post-expansions

Sequences are also potentially expandable after the completion of the base SPP. Once an SPP has been completed, the sequence is potentially complete: the action launched by the FPP has run its course and a new action could appropriately be begun. However, it is also possible for talk to occur after the SPP which is recognisably associated with the preceding sequence. That is, it is possible for sequences to be expanded after their SPP.

Labels: *FPP-post*, *SPP-post*

Example:

A:	What is the weather like today?	***FPP-base***
B:	Forecast for cloudy skies today.	***SPP-base***
A:	Okay.	***FPP-post***
B:	No problem.	***SPP-post***

A.1.3 Minimal-expansions

Because dialogue does not always contain even numbers of utterances, there are also single-utterance *minimal-expansions*, for utterances that do not belong to conventional AP. CAMS defines three types of minimal-expansion *Pre*, *Post*, and *Insert*, which behave in a similar manner to their expansion counterparts. That is, they must be produced before, after, or inside a base sequence. These are closely related to the idea of minimal post-expansions (Schegloff, 2007), in that they are not designed to project any further sequences of talk, but rather open, close or add to sequences respectively. The primary role is to allow for additional turns that behave as expansions but consist only of one turn. There is no restriction on speaker order for minimal-expansions, which allows the same speaker to produce more than one utterance of different types in succession, or for a speaker to produce one utterance that does not belong to (initiate or conclude) an AP.

Labels: *Pre*, *Insert*, *Post*

Example:

A:	When is my dentist appointment?	***FPP-base***
B:	The appointment is at 11 am with your Aunt.	***SPP-base***
A:	Thanks.	***Post***

A.2 Dialogue Acts

An utterances dialogue act (DA) describes not just its meaning, but the speakers intentions in the wider context of the conversation, and therefore, facilitate the computational modelling of communicative behaviour in dialogue (Bunt et al., 2012). The DA within CAMS are aligned with DiAML (ISO 24617-2) (British Standards Institution, 2012; Bunt et al., 2012) and are arranged into eight categories according to their function. They are grouped by their communicative function (though this differs slightly from the original DiAML organisation): Information-seeking, information-providing, commissives, directives, feedback, time management, communication management, and social obligations management.

A.2.1 Information-seeking Functions

propositionalQuestion (Yes/No)

Communicative function of a dialogue act performed by the sender, S, in order to know whether the proposition, which forms the semantic content, is true. S assumes that A knows whether the proposition is true or not and puts pressure on A to provide this information.

A propositional question corresponds to what is commonly termed a YN-question in the linguistic literature. This standard prefers the term 'propositional question' because the term 'YN-Question' carries the suggestion that this kind of question can only be answered by 'yes' or 'no', which is not the case.

Example: "Does the meeting start at ten?"

setQuestion (Who/What/Where/How)

Communicative function of a dialogue act performed by the sender, S, in order to know which elements of a given set have a certain property specified by the semantic content; S puts pressure on the addressee, A, to provide this information, which S assumes that A possesses. S believes that at least one element of the set has that property.

A set question corresponds to what is commonly termed a WH-question in the linguistic literature. The term 'set question' is preferred because: (a) it clearly separates form from function by removing any oblique reference to syntactic criteria for the identification of such acts; and (b) it is not a language specific term (it may be further noted that even in English, not all questioning words begin with 'wh', for example, "How?").

Example: "What time does the meeting start?"; "How far is it to the station?"

choiceQuestion

Communicative function of a dialogue act performed by the sender, S, in order to know which one from a list of alternative propositions, specified by the semantic content, is true; S believes that exactly one element of that list is true; S assumes that the addressee, A, knows which of the alternative propositions is true, and S puts pressure on A to provide this information.

It is not very common in annotation schemes to specifically distinguish the concept of choice questions from that of set questions. However, whereas it is common for the concept set question to carry the expectation that all members of the set with a given property should be returned by the addressee, for a choice-question the expectation is that there will be exactly one. The different preconditions and effects indicate that these are semantically different concepts, and they have been treated here as such.

Example: "Should the telephone cable go in telephone line or in external line?"

checkQuestion

Communicative function of a dialogue act performed by the sender, S, in order to know whether a proposition, which forms the semantic content, is true, S holds the uncertain belief that it is true S. S assumes that A knows whether the proposition is true or not and puts pressure on A to provide this information.

Example: "The meeting starts at ten, right?"

A.2.2 Information-providing Functions

inform (Statement)

Communicative function of a dialogue act performed by the sender, S, in order to make the information contained in the semantic content known to the addressee, A; S assumes that the information is correct.

The inform function may also have more specific rhetorical functions such as: explain, elaborate, exemplify and justify; this is treated in this standard by means of rhetorical relations.

Example: "The 6.34 to Breda leaves from platform 2."

answer

Communicative function of a dialogue act performed by the sender, S, in order to make certain information available to the addressee, A, which S believes A wants to know; S assumes that this information is correct.

Example:

S: What does the display say?
A: Send error document ready.

agreement

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S assumes a given proposition to be true, which S believes that A also assumes to be true.

DAMSL and SWBD-DAMSL use "Agreement" to refer to various degrees in which some previous proposal, plan, opinion or statement is accepted; "accept" is one of these degrees; "reject" is another.

Example: "Exactly."

disagreement

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S assumes a given proposition to be false, which S believes that A assumes to be true.

DAMSL and SWBD-DAMSL use "Agreement" to refer to various degrees in which a speaker accepts some previous proposal, plan, opinion or statement; "accept" is one of these degrees; "reject" is another.

Example:

S: Do you know where to find the ink cartridge?
A: Oh I think to the left of the paper.
S: Uh... no.

correction

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that certain information which S has reason to believe that A assumes to be correct, is in fact incorrect and that instead the information that S provides is correct.

Example: "To Montreal, not to Ottawa."

confirm

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that certain information that A wants to know, and concerning which A holds an uncertain belief, is indeed correct.

Example: "Indeed."

disconfirm

Communicative function of a dialogue act performed by the sender, S, in order to let the addressee, A, know that certain information that A wants to know, and concerning which A holds an uncertain belief, is incorrect.

Example: "Nope."

A.2.3 Commissive Functions

offer

Communicative function of a dialogue act by which the sender, S, indicates their willingness and ability to perform the action, specified by the semantic content, conditional on the consent of addressee A that S do so.

Example: "I will look that up for you."

conditionalAccept (Consider/Address a Request/Suggestion/Offer)

Communicative function of a dialogue act by which the sender, S, indicates that they will consider the performance of an action, depending on certain conditions that they make explicit. The action may be one that they were requested to perform or was suggested that they perform. Or to indicate that they are considering the possibility that the addressee, A, performs the action that A has previously offered to perform.

The conditionalAccept function covers a range of possible responses to a request, suggestion or offer. If the condition specified is met the sender commits to the action, or accepts the offer, otherwise the sender in fact declines to perform the requested action or accept the offer.

Example:

A: Please give me the gun.
S: If you push the bag to me.

accept (Request/Suggestion/Offer)

Communicative function of a dialogue act by which the sender, S, commits them self to perform an action that they have been requested to perform or was suggested that they perform. Or to inform the addressee, A, that S would like A to perform the action that A has previously offered to perform.

Example:

A: Would you like help with that?
S: Sure.

decline (Request/Suggestion/Offer)

Communicative function of a dialogue act by which the sender, S, indicates that they refuse to perform an action that they have been requested to perform or was suggested that they perform. Or to inform the addressee, A, that S does not want A to perform the action that A has previously offered to perform.

Example:

A: Would you like help with that?
S: No thank you.

A.2.4 Directive Functions

request

Communicative function of a dialogue act performed by the sender, S, in order to create a commitment for the addressee, A, to perform a certain action in the manner or with the frequency described by the semantic content, conditional on A's consent to perform the action. S assumes that A is able to perform this action.

Example: "Please turn to page five"; "Please don't do this ever again"; "Please drive very carefully".

suggest

Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, consider the performance of a certain action, specified by the semantic content, S believes that this action is in A's interest, and assumes that A is able to perform the action.

Example: "Let's wait for the speaker to finish."

A.2.5 Feedback Functions

autoPositive (Positive Understanding/Feedback)

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S believes that S's processing of the previous utterance(s) was successful.

Feedback mostly concerns the processing of the last utterance from the addressee, but sometimes, especially in the case of positive feedback, it concerns a longer stretch of dialogue.

Example: "Uh-huh"; "Okay"; "Yes"

autoNegative (Negative Understanding/Feedback)

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S's processing of the previous utterance(s) encountered a problem.

Example: “Sorry?”; “What?”

A.2.6 Time Management Functions

stalling (Pausing)

Communicative function of a dialogue act performed by the sender, S, in order to have a little extra time to construct their contribution or to suspend the dialogue for a short while.

Pausing occurs either in preparation of continuing the dialogue, or because something else came up which is more urgent for the sender to attend to.

Example: “Let me see...”; “Ehm...”; “Just a moment”; “Umm...”

A.2.7 Own and Partner Communication Management Functions

retraction (Abandon)

Communicative function of a dialogue act performed by the sender, S, in order to withdraw or abandon something that they just said within the same turn.

Example: “Then we’re going to g– ”

A.2.8 Social Obligations Management Functions

greeting

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S is present and aware of A’s presence.

Greetings usually come in initiative-response pairs and are commonly used to open a dialogue.

Example: “Hello!”; “Good morning”

goodbye

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S intends the current utterance to be their final contribution to the dialogue.

Goodbyes usually come in initiative-response pairs and are commonly used to close a dialogue.

Example: “Bye bye, see you later.”

thanking

Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S is grateful for some action performed by A; S puts pressure on A to acknowledge this.

Utterances used for thanking often also indicate that the sender wants to end the dialogue.

Example: “Thanks a lot.”

acceptThanking

Communicative function of a dialogue act performed by the sender, S, in order to mitigate to the feelings of gratitude which the addressee, A, has expressed.

Example: "Don't mention it."

apology

Communicative function of a dialogue act performed by the sender, S, in order to signal that they want the addressee, A, to know that S regrets something; S puts pressure on A to acknowledge this.

Example: "Sorry about that."

acceptApology

Communicative function of a dialogue act performed by the sender, S, in order to mitigate, the feelings of regret that the addressee, A, has expressed.

Example: "No problem."

Appendix B

Inter-annotator Agreement

IAA measures can be used as a means of assessing the *reproducibility* of a coding scheme or determining the *reliability* of a produced 'gold standard' labelled dataset. Given that the focus of this study is the labelling schema itself, the purpose of measuring IAA refers to the former. That is, determining if the schema is inherently learnable, that the labels applied to utterances are not entirely dependent on the biases of an individual annotator, and that there is a common understanding of the meaning of labels and the utterances to which they are applicable (Craggs and Wood, 2005). It should be noted, that reproducibility is a natural prerequisite to demonstrating reliability of a coding scheme. If annotators produce similar results, they likely have a similar understanding of the annotation scheme and guidelines, and that these are able to represent the desired characteristics of the data (Artstein and Poesio, 2008). Within the literature chance-corrected coefficients, that is, accounting for the probability that annotators select the same label by chance, such as Cohen's Kappa (Cohen, 1960), or Scott's Pi (Scott, 1955), are the preferable measures of IAA (Craggs and Wood, 2005; Di Eugenio, 2000; Carletta, 1996). However, weighted coefficients, such as Krippendorff's Alpha (Krippendorff, 2004), are more suitable to annotation tasks such as this, which require an element of semantic interpretation.

B.1 Agreement Coefficients

Agreement can be measured as the percentage of cases in which different annotators agree on assigned labels. Percentage agreement, however, is distorted by the number of labels within the coding scheme (Scott, 1955); where fewer labels naturally results in higher agreement. Further, it does not correct for the distribution of labels, and as such, agreement may be skewed when some labels are much more common than others (Artstein and Poesio, 2008; Hsu and Field, 2003). Similarly, criticisms have been noted for measures of association χ^2 (Cohen, 1960), and correlation coefficients (Artstein and Poesio, 2005b). For these reasons, the consensus within the literature is that chance-corrected coefficients, that is, accounting for the probability that annotators select the same label by chance, such as Cohen's Kappa (Cohen, 1960), or Scott's Pi (Scott, 1955), are generally more preferable (Di Eugenio, 2000; Carletta, 1996). Though, as will be shown, weighted coefficients, such as Krippendorff's Alpha (Krippendorff, 2004), are more suitable to annotation tasks which require an element of semantic interpretation.

B.1.1 Unweighted Coefficients

Here the family of unweighted agreement coefficients is briefly described. Cohen's Kappa (Cohen, 1960), and Scott's Pi (Scott, 1955), calculate agreement between two annotators. However, in practice, two annotators are rarely enough to generate reliable agreement statistics. It is often more preferable to use generalised versions of the coefficients for multiple annotators. Fleiss (1971), proposed a generalisation of Scott's Pi, and a generalisation of Cohen's Kappa was suggested by Davies and Fleiss (1982), or Multi-Pi and Multi-kappa respectively. [1] All four of these coefficients can be expressed in the form:

$$\pi, \kappa = \frac{A_o - A_e}{1 - A_e} \tag{B.1}$$

Where A_o is the *observed* agreement, the proportion of items where annotators agree, and A_e is the *expected* agreement. These coefficients generate values that lie within $\frac{-A_e}{1-A_e}$ (no agreement) and 1 (perfect agreement), with 0 signifying agreement is the same as would be expected by chance, $A_o = A_e$. The coefficients all consider the probability that two annotators will assign labels independently, and therefore, the probability they assign the same label to a given item i, is the product of the probability that each annotator a, assigns the label l, to the item, $P(l|a_1) \times P(l|a_2)$. The expected agreement is therefore the sum of the product for all labels:

$$A_e = \sum_{l \in L} P(l|a_1) \times P(l|a_2) \tag{B.2}$$

Where Pi, Kappa and their multi-annotator generalisations differ, is in the assumptions around the distribution of assigned labels for an annotator operating only by chance, that is, how $P(l|a_k)$ is estimated. When calculating A_e, Pi assumes that each annotator assigns labels with the same distribution and therefore considered an *unbiased* coefficient, whereas Kappa is *biased*, in that it calculates A_e from the observed distribution of individual annotators.

B.1.1.1 Pi

Given that Pi assumes the same distribution is used by both annotators, the probability that either annotator assigns a label to the item is the same:

$$P(l|a_1) = P(l|a_2) = P(l) \tag{B.3}$$

$P(l)$ is the total number of times label l is assigned by both annotators $\mathbf{n}_l$, divided by the total number of assigned labels, that is, with two annotators, twice the number of items $\mathbf{i}$:

$$P(l) = \frac{\mathbf{n}_l}{2\mathbf{i}} \tag{B.4}$$

For Pi, expected agreement can then be defined as:

$$A_e^{\pi} = \sum_{l \in L} (\frac{\mathbf{n}_l}{2\mathbf{i}})^2 \tag{B.5}$$

B.1.1.2 Kappa

As previously stated, Kappa assumes the distribution of labels assigned to items is unique to each annotator, and as such, reflects the individual annotators bias. The probability that an annotator a_j assigns a label to an item is estimated from the proportion of items

[1] Artstein and Poesio (2005b), noted that Fleiss (1971), originally named the coefficient K, even though it assumes a single probability distribution for all annotators and is therefore a generalisation of Scott's Pi. Here, the naming convention used by Artstein and Poesio (2005b), is adopted, where Multi-pi and Multi-kappa refer to the generalisation of Scott's Pi and Cohens Kappa respectively.

actually assigned to l by the annotator, $\mathbf{n}_{a_j l}$. Thus, $P(l|a_j)$ is the number of times l is assigned by a_j divided by the number of items $\mathbf{i}$:

$$P(l|a_j) = \frac{\mathbf{n}_{a_j l}}{\mathbf{i}} \tag{B.6}$$

For Kappa, expected agreement can then be calculated as the joint probability for both annotators:

$$A_e^K = \sum_{l \in L} \frac{\mathbf{n}_{a_1 l}}{\mathbf{i}} \times \frac{\mathbf{n}_{a_2 l}}{\mathbf{i}} \tag{B.7}$$

B.1.1.3 Multi-Pi

With more than two annotators there will naturally be some items on which some annotators agree, and others do not, and therefore A_o cannot be defined as the percentage of items where there is agreement. Fleiss (1971), defined the amount of agreement on a particular item as the proportion of pairwise agreements out of the total number of annotator pairs for that item.
The number of times an item i is assigned label l is calculated, $\mathbf{n}_{il}$. The number of pairwise agreements for item i, and label l, is $\mathbf{n}_{il}(\mathbf{n}_{il} - 1)$. The amount of agreement for item i, is therefore, the sum of $\mathbf{n}_{il}$ for all labels ($\mathbf{L}$), divided by the total number of annotator pairs $\mathbf{a}(\mathbf{a} - 1)$.

$$agr_i = \frac{1}{\mathbf{a}(\mathbf{a} - 1)} \sum_{l \in L} \mathbf{n}_{il}(\mathbf{n}_{il} - 1) \tag{B.8}$$

Observed agreement is then the mean agreement for all items:

$$A_o = \frac{1}{\mathbf{i}} \sum_{i \in I} agr_i \tag{B.9}$$

Similarly, pairwise comparisons are used in the calculation for expected agreement as well. Fleiss (1971), also considered chance agreement on the basis of a single distribution for all annotators (hence his Kappa being more properly thought of as a generalisation of Scott's Pi). As with Pi, $P(l)$ is the number of items assigned label l by all coders $\mathbf{n}_l$, divided by the total number of assigned labels - the number of annotators ($\mathbf{a}$) multiplied by the number of items $\mathbf{i}$:

$$P(l) = \frac{\mathbf{n}_l}{\mathbf{ai}} \tag{B.10}$$

And, as with Pi, the probability that two annotators assign a label to a particular category is the joint probability that they assign the label independently. Expected agreement is therefore the sum of joint probabilities over all categories:

$$A_e^\pi = \sum_{l \in L} (\frac{\mathbf{n}_l}{\mathbf{ai}})^2 \tag{B.11}$$

B.1.1.4 Multi-Kappa

Because Multi-kappa uses a separate probability distribution for each annotator, the probability of a given annotator a_j, assigning a label to an item is the same as the two-annotator version of Kappa:

$$P(l|a_j) = \frac{\mathbf{n}_{a_j l}}{\mathbf{i}} \tag{B.12}$$

The probability of a given annotator pair, a_j and a_k, selecting a label for an item, is the joint probability for each annotator $P(l|a_j) \times P(l|a_k)$, divided by the total number of annotator pairs. The expected agreement is then the sum of all pairwise joint probabilities for each label:

$$A_e^K = \sum_{l \in L} \frac{1}{\mathbf{a}(\mathbf{a}-1)} \sum_{j=1}^{\mathbf{a}-1} \sum_{k=j+1}^{\mathbf{a}} P(l|a_j) \times P(l|a_k) \tag{B.13}$$

B.1.2 Weighted Coefficients

For some annotation tasks it does not make sense to treat all disagreements equally. For example, the DA *choiceQuestion* and *checkQuestion* are semantically more similar than *request* and *accept*. Both Pi and Kappa are limited in such circumstances, because they only consider identical labels for agreement. This can result in very poor agreement values and as such they are not considered an acceptable measure of agreement for DA labelling tasks (Geertzen and Bunt, 2006; Artstein and Poesio, 2005b). A solution to this problem is the use of weighted agreement coefficients, which consider the magnitude of disagreement between assigned labels. Cohen (1968), proposed a weighted variation of Kappa for two annotators. More frequently used however, and appropriate for this study, is Krippendorff's Alpha (Krippendorff, 2004), and the Beta statistic, proposed by Artstein and Poesio (2005b). Both Alpha and Beta are calculated from the observed and expected *disagreements*, rather than the agreement of the previously discussed coefficients. The ratio of observed and expected disagreement is then subtracted from 1 to produce the final agreement value:

$$\alpha, \beta = 1 - \frac{D_o}{D_e} \tag{B.14}$$

Further, weighted coefficients use a distance function (see section B.2), which returns a value between 0 and 1 representing the similarity between an arbitrary pair of labels. 0 indicates the two labels are identical and 1 indicates they are completely dissimilar. This value is then used to weight pairs of assigned labels, penalising those that are more dissimilar. The amount of disagreement for a given item is, therefore, the mean of the distances between all pairwise assignments for that item. The number of annotators who label item i, with label l, is $\mathbf{n}_{il}$. For every label pair l_j and l_k, there are $\mathbf{n}_{il_j}\mathbf{n}_{il_k}$ pairs of assigned labels for an item, and each has a distance ($\mathbf{d}$) of $\mathbf{d}_{l_j l_k}$, calculated by the distance function. The mean disagreement for an item is then the sum of all weighted label pairs, divided by the total number of annotator pairs, $\mathbf{a}(\mathbf{a}-1)$:

$$disagr_i = \frac{1}{\mathbf{a}(\mathbf{a}-1)} \sum_{j=1}^{l} \sum_{k=1}^{l} \mathbf{n}_{il_j}\mathbf{n}_{il_k}\mathbf{d}_{l_j l_k} \tag{B.15}$$

Observed disagreement is then the mean disagreement for all items:

$$D_o = \frac{1}{\mathbf{i}} \sum_{i \in I} disagr_i \tag{B.16}$$

Just as with Pi and Kappa, where Alpha and Beta differ is in their assumptions about the distribution of assigned labels for an annotator operating only by chance. Similar to Pi, Alpha assumes a single distribution for all annotators (unbiased), while Beta, like Kappa, considers the individual annotators distributions (biased).

B.1.2.1 Alpha

Given the single probability distribution for all annotators, the probability of assigning a label to an item is the number of assignments of the label by all annotators $\mathbf{n}_l$, divided by the total number of assignments - items $\mathbf{i}$ multiplied by the number of annotators $\mathbf{a}$.

$$P(l) = \frac{\mathbf{n}_l}{\mathbf{ai}} \tag{B.17}$$

Again, the probability that two annotators assign labels l_j and l_k, is the joint probability of each annotator assigning the label independently. The expected disagreement is, therefore, the sum of the weighted joint probabilities for all label pairs, divided by the total number of assignments:

$$D_e^{\alpha} = \frac{1}{\mathbf{ai}(\mathbf{ai}-1)} \sum_{j=1}^{l} \sum_{k=1}^{l} \mathbf{n}_{l_j} \mathbf{n}_{l_k} \mathbf{d}_{l_j l_k} \tag{B.18}$$

B.1.2.2 Beta

The Beta coefficient is, in essence, multi-annotator generalisation of Cohens weighted Kappa (Artstein and Poesio, 2005b); in that, it is a weighted coefficient which considers individual annotators label distributions (bias) and is applicable to more than two annotators. The probability that annotator a, assigns label l, to an item, is the total number of such assignments $\mathbf{n}_{al}$, divided by the total number of assignments for that annotator (the same as Kappa and Multi-kappa):

$$P(l|a_j) = \frac{\mathbf{n}_{a_j l}}{\mathbf{i}} \tag{B.19}$$

The probability that two annotators a_m and a_n, selecting different labels l_j and l_k, is $P(l_j|a_m) \times P(l_k|a_n) + P(l_j|a_m) \times P(l_k|a_n)$. The probability that a given pair of coders assigns labels l_m and l_n, is the mean of the probabilities for all annotator pairs:

$$P(l_j, l_k) = \frac{1}{\mathbf{ai}(\mathbf{ai}-1)} \sum_{m=1}^{\mathbf{a}-1} \sum_{n=1}^{\mathbf{a}} \mathbf{n}_{a_m l_j} \mathbf{n}_{a_n l_k} + \mathbf{n}_{a_m l_k} \mathbf{n}_{a_n l_j} \tag{B.20}$$

The expected agreement for Beta is then, the mean of the probabilities for each pair of labels weighted by the distances:

$$D_e^{\beta} = \sum_{j=1}^{L-1} \sum_{k=j+1}^{L} P(l_j, l_k) \mathbf{d}_{l_j l_k} \tag{B.21}$$

It is worth noting, that if all disagreements are considered equal, with distance 1, then Alpha and Beta produce the same result as their non-weighted equivalents Multi-pi and Multi-kappa. Similarly, if data from only two annotators is used, and the distances are equal, the results are the same as the non-weighted two annotator variants Pi and Kappa. Figure B.1 summarises some of the characteristics of each coefficient with respect to three different dimensions, bias and unbiased (Kappa and Pi), two or multiple coders (multi-Kappa and multi-Pi) and weighted (Alpha and Beta).

B.2 Weighted Coefficient Distance Functions

The calculation of Alpha and Beta requires a distance function $\mathbf{d}$, that returns a distance value between 0 and 1 for each possible label pair. The value indicates the amount of dissimilarity between the two labels, with 0 indicating they are identical and 1 indicating

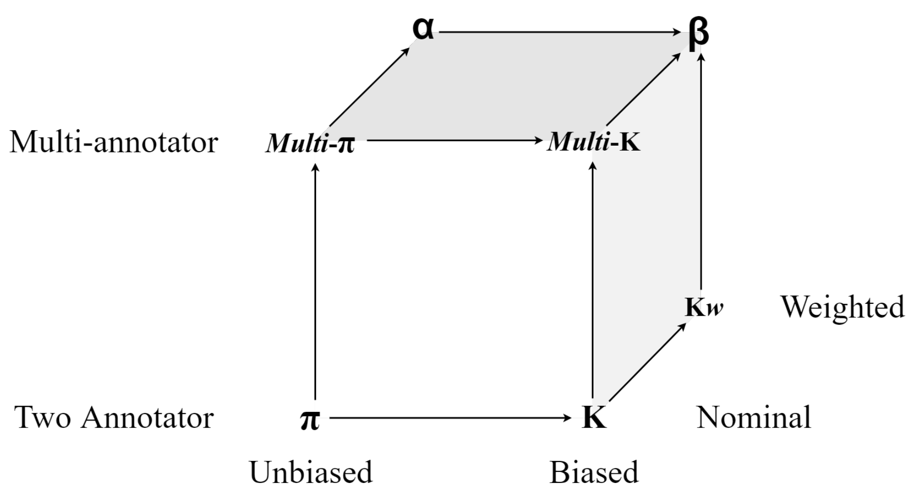


Figure B.1: Agreement coefficients in three dimensions, bias, number of coders, and weighted. Adapted from the 'Coefficient Cube', (Artstein and Poesio, 2005b).

they are completely dissimilar. In this section 3 distance functions are defined, one for each of the label types defined within CAMS. The constraints suggested by Artstein and Poesio (2005b), to which all distance metrics in (Krippendorff, 2004), and (Geertzen and Bunt, 2006) conform, are adopted here. That is, (1) the distance between a label and itself is 0, and (2), the distance between two labels is not dependent on their order. Because CAMS defines DA and AP, and they combine to form AP-types, it is necessary to define distance functions, such that, the distance of the combined DA and AP label still falls in the range 0 to 1, and conforms to the above constraints.

B.2.1 Dialogue Act Distance Function

Geertzen and Bunt (2006), proposed a distance function based on a hierarchical ancestor-offspring relationship between DA labels within the Dynamic Interpretation Theory (DIT^{++}) annotation scheme. Given that DIT^{++} shares many characteristics of the DAMSL scheme (Allen and Core, 1997), and that both of these are precursors to DiAML (British Standards Institution, 2012), a similar approach is employed here. However, their metric considered both the difference in depth and the minimal depth between two labels in the hierarchy, and these are each modified by two constants a and b. To avoid selecting two arbitrarily chosen constant values, which may affect the coefficient calculation, the DA distance function defined here only considers the distance between two labels within the relationship hierarchy.

The DA relationships are characterised in an undirected graph, where leaf nodes are DA labels and intermediate nodes represent the communicative function subcategories. All edges are considered to have an equal distance of 1. DA are arranged according to their communicative functions which closely match those defined in DiAML. However, in a number of cases DA have been separated into subcategories that more closely resemble their semantic intent. For example, within DiAML the information-providing functions include the DA *agreement* and *disagreement*, which clearly have opposing sentiments, positive and negative. In such cases, DA that are assigned to more appropriate subcategories, for example, positive and negative responses. Figure B.2 depicts the Information-transfer sub-tree of the DA relationship graph. [2]

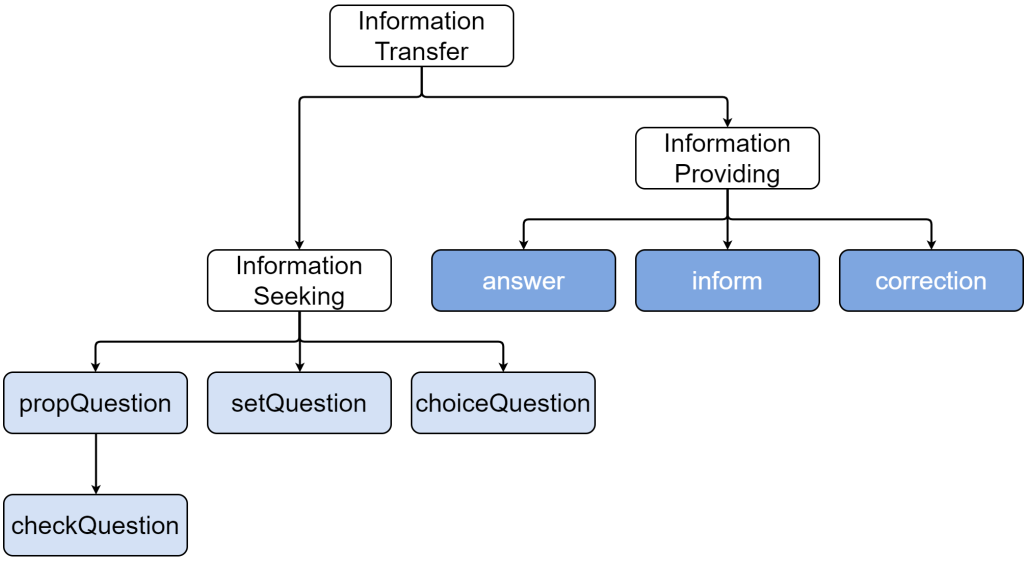


Figure B.2: The Information-transfer sub-tree of the DA relationship graph. Leaf nodes are DA, while intermediate nodes represent the communicative function subcategories.

For each pair of DA, $da_j, da_k \in \mathbf{DA}$, the distance value is calculated as follows. First the path distance ($\mathbf{p}$), between da_j and da_k, is calculated as the sum of the number ($\mathbf{N}$) of edges e, each with distance 1 for the shortest path between da_j and da_k:

$$\mathbf{p}_{da_j da_k} = \sum_{i=1}^{\mathbf{N}} e_i \tag{B.22}$$

The path distance $\mathbf{p}_{da_j da_k}$, is then normalised by the *minimum* and *maximum* path distances for all possible label pairs ($\mathbf{P}_{min}$ and $\mathbf{P}_{max}$), to yield the distance $\mathbf{d}(da_j, da_k)$, in the range 0 to 1:

$$\mathbf{d}(da_j, da_k) = \frac{\mathbf{p}_{da_j da_k} - \mathbf{P}_{min}}{\mathbf{P}_{max} - \mathbf{P}_{min}} \tag{B.23}$$

B.2.2 Adjacency Pair Distance Function

AP, like DA, can be organised into categories that represent their function: *base*, *pre*, *post* and *insert*. However, the paired nature of FPP and SPP, means representing their relationship in a graph-like structure is less appropriate. For example, FPP-pre and FPP-post could be considered similar, in that they both initiate a sequence. Yet functionally, the *pre* and *post* expansion types have opposing meanings, pre expansions should take place *before* a base pair and post expansions *after*. Therefore, the distance function defined here considers the difference between the AP labels prefix and postfix, that is, whether they are part of an adjacency pair and initiating or responsive within a sequence (FPP or SPP), or a minimal expansion, and whether they belong to the same *base* sequence or *expansion type* (pre, post and insert).

For each pair of AP, $ap_j, ap_k \in \mathbf{AP}$, the distance value is calculated as follows. First set the distance between ap_j and ap_k to 0, ($\mathbf{d}_{ap_j ap_k} = 0$). Then, separately compare the

[2] The full DA relationship graph can be found in figure B.3.

prefix and postfix of the two labels. If they do not match, increase the distance by 0.5:

$$\mathbf{d}(ap_j, ap_k) = \sum 0.5(1 - \delta(ap_j^{pre}, ap_k^{pre})) + 0.5(1 - \delta(ap_j^{post}, ap_k^{post})) \tag{B.24}$$

Thus, two identical AP labels will have a distance of 0, and two completely different labels will have the maximum distance of 1, and two FPP labels will have a distance of 0.5, as in the previous example with FPP-pre and FPP-post. Similarly, a minimal expansion will have a distance of 0.5 to the FPP and SPP expansions within the same functional category.

B.2.3 AP-type Distance Function

Within CAMS, an AP-type label is considered the combination of the DA and AP labels assigned to that utterance, and a similar approach is taken for the AP-type distance calculation. The distance between two AP-type labels is considered the sum of the distances for the individual components, $\mathbf{d}(da_j, da_k) + \mathbf{d}(ap_j, ap_k)$, normalised by the minimum and maximum distances for all possible label pairs ($\mathbf{D}_{min}$ and $\mathbf{D}_{max}$). Thus, for each pair of AP-type labels, $apt_j, apt_k \in (\mathbf{DA} \cup \mathbf{AP})$, the 'raw' distances, $\mathbf{d}_{apt_j apt_k}$, are calculated as:

$$\mathbf{d}_{apt_j apt_k} = \mathbf{d}(da_j, da_k) + \mathbf{d}(ap_j, ap_k) \tag{B.25}$$

The distance function is then:

$$\mathbf{d}(apt_j, apt_k) = \frac{\mathbf{d}_{apt_j apt_k} - \mathbf{D}_{min}}{\mathbf{D}_{max} - \mathbf{D}_{min}} \tag{B.26}$$

This simple formulation has the advantage of maintaining consistency with the DA and AP distance functions, allowing for comparison of coefficient values between the component label types. Additionally, the large number of possible combinations of DA and AP (297, though not all combinations are valid), would make defining a distinct AP-type distance function laborious and prone to errors and inconsistencies. It should also be noted that this distance function effectively results in the normalised mean of the DA and AP distances.

B.3 Coefficient Selection

The following section discusses considerations around the selection of agreement coefficients for calculating IAA. Given that annotators assign the CAMS DA and AP labels independently, and that each label type has a distinct distance function, it is also possible to calculate independent IAA values for each label type.

The DA within the schema can be grouped into semantically similar communicative functions (Bunt, 2011), such as, information seeking and information providing. Further, some utterances can be thought of as *multidimensional* (Bunt, 2006), that is, they could be assigned two equally valid DA labels (or arguably both). Consider the following example:

A1: What is the weather going to be today and tomorrow?
B1: What city would you like to know the weather about?
A2: I want to know if it will drizzle in Durham.

Utterance A2 could be considered an answer to the previous question B1, the location they want to know the weather for, or a question in its own right, "*will it drizzle in Durham.*" Clearly, even with well-defined label definitions, there is a certain amount of subjectivity in assigning a single label to certain utterances. A similar semantic grouping is also true for AP, where, for example, FPP-insert and SPP-insert are more closely related to an insert expansion than AP from the Pre and Post groups. It seems reasonable to treat assignments that belong to different expansion types more seriously than those from the same group. The above, and the use of weighted agreement for DA annotation by (Geertzen and Bunt, 2006), indicates the use of weighted agreement measures, such as Alpha and Beta, are the appropriate choice for DA and AP annotation because the labels are not equally distinct from each other.

What is less clear, however, is the choice between these two coefficients. There has been much debate on this matter (Artstein, 2018; Craggs and Wood, 2005; Di Eugenio and Glass, 2004; Krippendorff, 2004; Hsu and Field, 2003; Byrt, Bishop, and Carlin, 1993; Zwick, 1988). Of course, Krippendorff built the notion of a single distribution into his Alpha coefficient, and Craggs and Wood (2005), argued strongly against the use of coefficients with bias, stating that, "*the purpose of assessing the reliability of coding schemes is not to judge the performance of the small number of individuals participating in the trial, but rather to predict the performance of the schemes in general.*" Yet, Artstein and Poesio (2005b), in their proposal of the Beta statistic believe that, "*assuming that coders act in accordance with the same probability distribution is too strong of an assumption, hence 'biased' measures are more appropriate.*"

The argument against the use of biased coefficients, illustrated by Krippendorff (2004), and others (Di Eugenio and Glass, 2004; Byrt, Bishop, and Carlin, 1993; Zwick, 1988), lies in its calculation of expected agreement. Though biased measures, such as Kappa and Beta, estimate expected agreement on the basis of individual annotator label distributions, they fail to account for unequal distributions *between* annotators. In so doing, biased coefficients effectively discount some of the disagreement resulting from different annotator distributions by incorporating it into expected agreement (Artstein and Poesio, 2008). Thus, for a fixed observed agreement, when annotators produce unequal distributions for the available categories – when bias is present – the values of biased coefficients will *exceed* those of non-biased coefficients. The objection, then, is the 'paradox' that as annotators become less similar, biased measures can *increase* (Di Eugenio and Glass, 2004), and begin to diverge from their non-biased counterparts. However, Artstein and Poesio (2005b) point out that in practice the difference between biased and non-biased measures often doesn't amount to much, and that bias is a source of disagreement in its own right. To this latter point, Banerjee et al. (1999), in reference to Zwick (1988), suggested that, "*rather than straightway ignoring marginal disagreement or attempting to correct for it, researchers should be studying it to determine whether it reflects important rater differences or merely random error.*" For example, Hsu and Field (2003) demonstrated how Kappa can give useful information even when the individual annotators distributions are very different, and Wiebe, Bruce, and O'Hara (1999), exploited bias to improve the annotation process. In any case, what does seem to be agreed upon, is that as the number of annotators is increased the difference between biased and non-biased measures becomes less significant (Artstein and Poesio, 2008, 2005a; Craggs and Wood, 2005). Further, as stated by Di Eugenio and Glass (2004), the biased and non-biased paradigms reflect distinct conceptualizations of the problem, and in agreement with Artstein and Poesio (2008), the choice should depend on the desired interpretation of chance agreement. However, Di Eugenio and Glass (2004), also believed the bias coefficient (Kappa) is more appropriate for discourse and DA tagging, because "*it is questionable whether the assumption of equal distributions underlying Pi is appropriate for coding in discourse and dialogue work*". Yet, they also suggested reporting Kappa and Pi together, to account for the 'bias problem' we have just described. Here a similar approach is taken, and both Alpha and Beta will be reported.

B.4 Coefficient Evaluation

To reiterate, the purpose of measuring agreement for this study is to assess the *reproducibility* of the schema for annotating dialogues with DA, AP and ultimately AP-types. If multiple annotators can be shown to *reliably* assign similar labels to a set of data, it can be inferred that they have a similar understanding of the meaning of the labels, the data items to which they are applicable and that the observed agreement (or disagreement) is not purely a product of chance or an individual's interpretation of the scheme. Unfortunately, the question of what constitutes reliable agreement when interpreting agreement coefficients seems to be an unanswered question (Artstein and Poesio, 2008; Craggs and Wood, 2005; Krippendorff, 2004).

The principal approach is based on a range of values proposed by Landis and Koch (1977). Values below zero are considered 'Poor' agreement, and values between 0 and 1 are separated into five ranges: *Slight* (.0 - .2), *Fair* (.21 - .4), *Moderate* (.41 - .6), *Substantial* (.61 - .8), and *Perfect* (>.81). Though they themselves concede that the divisions are arbitrary and only provide a useful benchmark. In Computational Linguistics, it is generally accepted that values of >0.8 can be considered "good reliability", and values in the range [0.67, 0.8] allow for "tentative conclusions to be drawn" (Krippendorff, 2004; Carletta, 1996). Though it is acknowledged that, as with the original Landis and Koch (1977) values, these ranges are somewhat arbitrary and are not suitable in all cases (Di Eugenio and Glass, 2004; Krippendorff, 2004; Carletta, 1996). This is especially true for annotation tasks such as this, where there is a degree of subjectivity in choosing an appropriate label, where some prior subject-specific knowledge is required, and notably for AP, prefect agreement will generally require annotators to agree on two (or more) labels, rather than one for DA. Indeed, it has been shown that achieving even the minimum 0.67 value is extremely difficult for discourse annotation (Poesio and Vieira, 1998; Hearst, 1997). This problem is further compounded when using weighted agreement coefficients, because the choice of distance function greatly impacts the calculated coefficient value, as shown by Artstein and Poesio (2005b). Furthermore, regarding the bias problem discussed in the previous section, differences in annotator distributions (bias) will *increase* biased coefficient values, causing them to diverge from non-biased measures. Thus, in the presence of bias, a biased coefficient will always be larger than a non-biased one, and for this reason Geiß (2021) suggests that applying the same range of values is not appropriate, because they warrant different interpretations. Unfortunately, to the best of our knowledge no alternative scale for interpreting biased coefficients has been proposed within the literature, though some have made attempts to 'correct' for bias when there are only two categories (Byrt, Bishop, and Carlin, 1993). We therefore choose to evaluate both coefficients, Alpha and Beta, with respect to the ranges typically adopted throughout the literature; with the caveat that, for Beta it is necessary to be cautious when drawing conclusions if there is a significant difference between the two coefficients.

Ultimately, choosing an arbitrary agreement threshold should not be the sole measure upon which an annotation schema, or labelled corpus, should be considered valid or reliable (Artstein and Poesio, 2008; Craggs and Wood, 2005). Instead, the methodology for collecting and calculating reliability should be thoroughly communicated, so that conclusions can be drawn based on the characteristics and motivations of the particular study (Artstein and Poesio, 2008). Thus we follow the 8 recommendations outlined by Bayerl and Paul (2011),for information that should be included when reporting annotation reliability:

1. Number of annotators.
2. Type and amount of material annotated.

3. Number of categories in the scheme.
4. Criteria for selecting annotators.
5. Annotators' expert status (novices, domain experts, schema developers, native speakers).
6. Type and intensity of training.
7. Type and computation of the agreement index.
8. Purpose for calculating the agreement index (including whether the goal was to reach a certain threshold or achieve "highest-possible" agreement).

The following additional annotation methodology considerations as suggested by Krippendorff (2004, ch. 11), and reiterated by (Artstein, 2018):

1. Annotators must work independently, so agreements come from a shared understanding not through discussion.
2. Annotators should come from a well-defined population, so that researchers are aware of previous knowledge or assumptions they bring to the annotation process.
3. Annotation instructions should be exhaustively formulated, clear and contain step-by-step instructions on how to use it.

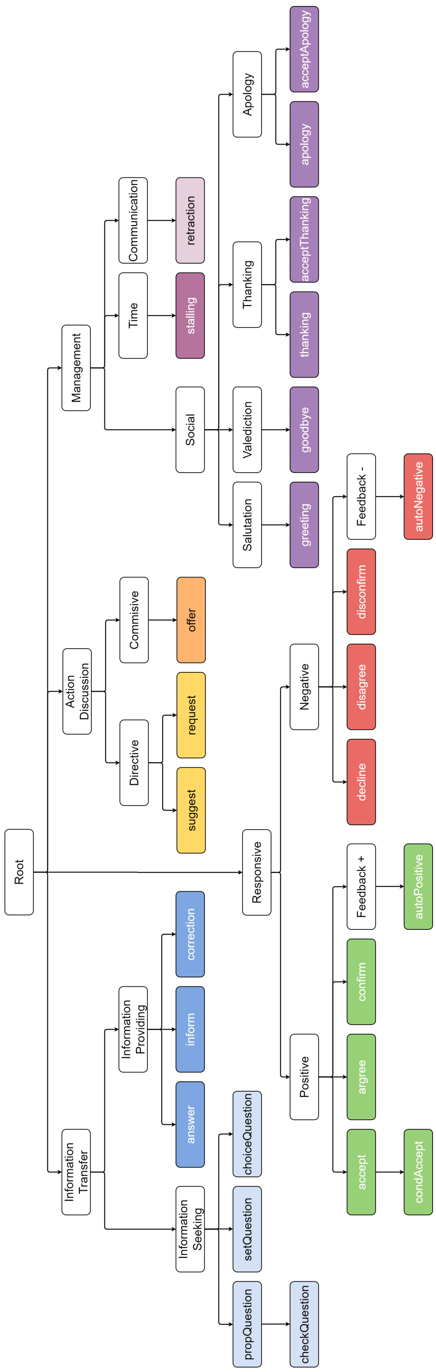


Figure B.3: The full DA relationship graph represents distances between DA for calculating weighted agreement coefficients. Leaf nodes are DA, while intermediate nodes represent the communicative function subcategories.

Appendix C

Alpha vs Beta

Previous results have shown that in all cases the Beta coefficient results in significantly higher agreement values than Alpha, and that this is principally caused by the differences in annotator label distributions increasing the Beta values. As discussed in section 4.2.3, the difference between these two coefficients lies only in their calculation of expected disagreement. That is, Alpha estimates disagreement on the basis that all annotators assign labels with the same probability distribution, while Beta considers the individual annotators distributions. Here, these different estimations are tested, using the actual annotator label distributions from this study, to determine the extent to which annotators use similar, or different distributions.

Jensen-Shannon Divergence The difference, or similarity, between probability distributions can be calculated using the Jensen-Shannon divergence (JSD) method. Here, the generalisation of JSD is adopted, which calculates a distance value between two or more probability distributions. The distance value is bounded in the range $0 \leq JSD \leq log_2(n)$, where n is the number of input distributions; the lower bound represents identical distributions and the upper bound maximally different distributions. For each dialogue set the JSD distance was calculated for the probability distributions of all annotators that labelled that set. Thus, in each case $n = 3$ and the range is $0 \leq JSD \leq 1.58$. Table C.1 shows the JSD distances for the DA and AP label distributions over each dialogue set. We can see that both DA and AP have low distance values, within ~$1/6^{th}$ of the lower range, and therefore, overall differences between annotator distributions is relatively small using this measure. AP labels show a lower average distance than DA over all dialogue sets, with a mean of 0.22 and 0.25 respectively, which is likely due to the fewer number of AP labels. However, AP also show a higher standard deviation than DA and this may reflect the higher disagreement and bias for AP labels that was previously observed.

Table C.1: JSD distance for DA and AP labels of each dialogue set.

Group	**DA**	**AP**
set 1	0.272	0.15
set 2	0.305	0.177
set 3	0.183	0.307
set 4	0.232	0.17
set 5	0.26	0.296
$\boldsymbol{\mu}$	0.251	0.22
$\boldsymbol{\sigma}$	0.041	0.067

Pearsons Chi-squared In addition to calculating the distance between groups of annotator probability distributions, we can also examine the extent to which label distributions are dependent on the individual annotators that assigned them. For this purpose, an χ^2 test was conducted using the *cumulative* annotator label distributions. For each dialogue set a separate χ^2 test was performed for all pairwise annotator combinations, results are shown in table C.2.

Table C.2: χ^2 analysis of annotator label distributions.

			DA			AP		
Group	**User 1**	**User 2**	χ^2_{crit}	χ^2	**p-value**	χ^2_{crit}	χ^2	**p-value**
Set 1	usr1-1	usr11-1	31.41	19.953	.461	15.507	11.5	.175
	usr1-1	usr6-1	28.869	21.281	.265	16.919	14.496	.106
	usr11-1	usr6-1	31.41	20.61	.42	16.919	12.353	.194
Set 2	usr12-2	usr2-2	28.869	24.878	.128	18.307	14.181	.165
	usr12-2	usr7-2	30.144	24.91	.164	18.307	16.687	.082
	usr2-2	usr7-2	30.144	19.882	.402	15.507	11.973	.152
Set 3	usr13-3	usr3-3	30.144	13.084	.834	18.307	17.263	.069
	usr13-3	usr8-3	28.869	18.934	.396	18.307	40.194	<.001
	usr3-3	usr8-3	27.587	9.853	.91	18.307	23.635	.009
Set 4	usr14-4	usr4-4	30.144	17.509	.555	18.307	9.667	.47
	usr14-4	usr9-4	27.587	12.955	.739	18.307	15.892	.103
	usr4-4	usr9-4	24.996	19.519	.191	18.307	17.311	.068
Set 5	usr10-5	usr15-5	28.869	26.42	.091	18.307	20.48	.025
	usr10-5	usr5-5	28.869	17.459	.492	18.307	18.725	.044
	usr15-5	usr5-5	30.144	17.333	.567	16.919	26.192	.002

1. For DA, in none of the pairwise comparison between annotators are the observed label frequencies significantly different. In other words, regardless of which annotator assigned the labels, the distribution would still be largely the same - although individual assignments could still be very different.

2. For AP, in 1/3 of cases (2 in set 3 and all of set 5), we see significant results when comparing the critical value to the test statistic, and also significant p-values. As such, we must reject the null hypothesis and concluded that the label distributions (in 1/3 of cases) were dependent on the annotator that assigned them. Therefore, certain annotators were producing label distributions that were quite distinct from each other.

These two conclusions seem to support the results from the JSD comparison. Firstly, there seems to be less variance in the annotator's DA label assignments, likely contributing to the observed higher agreement values. Secondly, AP seem to be more dependent on the individual annotator which assigned them (overall p-values are lower, indicating a higher degree of idiosyncratic interpretation). As such, agreement for AP was lower, while bias was higher, and this may also be indicative of the misunderstanding surrounding the use of FPP and SPP that was discussed in section 4.2.5.4, and the differences in interpretation observed in 4.2.5.2 and 4.2.5.3. These results also suggest that both the JSD and χ^2 tests could serve as additional measures for the homogeneity of annotators interpretation, and understanding, of the material and coding scheme.

From these measures, and regarding Alpha and Beta, it seems that annotators do, in fact, use more similar distributions for DA labels. In most cases this also appears true for AP, though there is a greater variance (in part due to misunderstanding FPP and SPP) between some groups of annotators. However, as we have seen, these small differences can result in drastically different values between the two coefficients. Given that there is a certain amount of semantic interpretation when assigning both DA and AP labels, the assumption that annotators will use the same distribution is, as Artstein and Poesio (2005b) stated, too strong. Consequently, Alpha may be too harsh in its estimation of annotator distributions and punish individual interpretation too severely. Yet, as shown in our AP label agreement results (4.2.5.4), when using the suffix-only distance function, the Beta coefficient exhibited smaller changes in agreement values. Further, as shown throughout our results, in the presence of bias – which is itself a form of disagreement – the Beta coefficient is consistently higher than Alpha. Therefore, it may be a less sensitive measure of agreement, even hiding some causes of disagreement, which makes drawing conclusions of reliability problematic, using the Beta coefficient alone. However, that Alpha and Beta diverge, and the extent to which they do, can provide useful information in its own right. In our case it has clearly signified the higher degree of idiosyncratic interpretation between annotators when assigning AP labels, and also highlighted differences between task and non-task-oriented, or dialogue corpora, groups. This information would not have been apparent from the calculation of either coefficient alone, and so in agreement with Di Eugenio and Glass (2004), for annotation that require a high degree of semantic interpretation, it seems more helpful to report both biased and unbiased values. Though, if the goal is to reach high agreement values, and hence reliability of labelled data, the more stringent unbiased coefficient should be used.

Appendix D

CAMS-KVRET Label Distributions

Table D.1: Count and proportion of DA labels within the CAMS-KVRET corpus.

DA	Count	%	Train Count	Train %	Test Count	Test %	Val Count	Val %
thanking	2491	14.39	1994	14.38	259	14.88	238	13.98
acceptThanking	2403	13.88	1922	13.86	248	14.24	233	13.68
setQuestion	2337	13.50	1897	13.68	219	12.58	221	12.98
answer	2182	12.61	1765	12.73	213	12.23	204	11.98
request	2150	12.42	1702	12.28	234	13.44	214	12.57
inform	1767	10.21	1417	10.22	186	10.68	164	9.63
propositionalQuestion	1247	7.21	1001	7.22	110	6.32	136	7.99
accept	1177	6.80	921	6.64	125	7.18	131	7.69
confirm	549	3.17	439	3.17	48	2.76	62	3.64
disconfirm	527	3.05	433	3.12	42	2.41	52	3.05
autoPositive	215	1.24	166	1.20	27	1.55	22	1.29
choiceQuestion	126	0.73	99	0.71	13	0.75	14	0.82
goodbye	57	0.33	46	0.33	5	0.29	6	0.35
suggest	41	0.24	34	0.25	4	0.23	3	0.18
checkQuestion	19	0.11	12	0.09	5	0.29	2	0.12
apology	8	0.05	5	0.04	2	0.11	1	0.06
conditionalAccept	4	0.02	4	0.03	0	0.00	0	0.00
acceptApology	2	0.01	1	0.01	1	0.06	0	0.00
agreement	1	0.01	1	0.01	0	0.00	0	0.00
autoNegative	1	0.01	1	0.01	0	0.00	0	0.00
disagreement	1	0.01	1	0.01	0	0.00	0	0.00
correction	1	0.01	1	0.01	0	0.00	0	0.00
greeting	1	0.01	1	0.01	0	0.00	0	0.00

Table D.2: Count and proportion of AP labels within the CAMS-KVRET corpus.

AP	Count	%	Train Count	Train %	Test Count	Test %	Val Count	Val %
FPP-base	6579	38.01	5302	38.25	645	37.05	632	37.11
SPP-base	6579	38.01	5302	38.25	645	37.05	632	37.11
FPP-insert	1243	7.18	988	7.13	123	7.06	132	7.75
SPP-insert	1243	7.18	988	7.13	123	7.06	132	7.75
Post	766	4.43	628	4.53	65	3.73	73	4.29
Insert	529	3.06	420	3.03	58	3.33	51	2.99
FPP-post	170	0.98	105	0.76	41	2.35	24	1.41
SPP-post	170	0.98	105	0.76	41	2.35	24	1.41
Pre	28	0.16	25	0.18	0	0.00	3	0.18

Appendix E

Model Hyperparameters

Table E.1: Summary of Hyperparameters for the supervised sentence encoders. We use Comet.ml to tune each model and results can be viewed at: comet.ml/nathanduran/sentence-encoding-for-da-model-optimisation.

Model	**Encoder**	**Dim Reduction**	**Classifier**	**Optimiser (α)**	**Trainable Params**
CNN	Filters: 64 Kernel Size: 5	Max Pool Size: 8	Nodes: 224 Dropout: 0.27	Adam (0.002)	294,905
TextCNN	Filters: 128 Kernel Size: [1, 2, 3, 4, 5]	Max Pool Size: 8	Nodes: 224 Dropout: 0.1	Adagrad (0.02)	2,357,625
DCNN	Filters: 64 Kernel Size: [7, 5]	K-max	Nodes: 128 Dropout: 0.1	Adagrad (0.02)	1,677,129
RCNN	Units: 256 Filters: 64	Max	Nodes: 128 Dropout: 0.02	RMSprop (0.001)	2,497,225
LSTM	Units: 256 Dropout: 0.2	Average	Nodes: 128 Dropout: 0.02	RMSprop (0.001)	2,709,577
GRU	Units: 256 Dropout: 0.2	Max	Nodes: 128 Dropout: 0.02	RMSprop (0.001)	1,113,129
Bi-LSTM	Units: 256 Dropout: 0.2	Average	Nodes: 128 Dropout: 0.02	RMSprop (0.001)	3,313,737
Bi-GRU	Units: 256 Dropout: 0.2	Max	Nodes: 128 Dropout: 0.02	RMSprop (0.001)	1,420,841
LSTM 2-lyr	Units: 256 Dropout: 0.2	Average	Nodes: 128 Dropout: 0.02	RMSprop (0.00075)	3,235,913
LSTM 3-lyr	Units: 256 Dropout: 0.2	Average	Nodes: 128 Dropout: 0.02	RMSprop (0.00075)	3,762,249
GRU 2-lyr	Units: 256 Dropout: 0.2	Average	Nodes: 128 Dropout: 0.02	RMSprop (0.00075)	1,507,881
GRU 3-lyr	Units: 256 Dropout: 0.2	Average	Nodes: 128 Dropout: 0.02	RMSprop (0.00075)	1,902,633

Table E.2: Summary of Hyperparameters for the language model sentence encoders.

Model	**Encoder**	**Dim Reduction**	**Classifier**	**Optimiser** (α)	**Trainable Params**
BERT	Units: 768 Layers: 12	Average	Nodes: 256 Dropout: 0.05	Adagrad (0.0015)	85,261,865
RoBERTa	Units: 768 Layers: 12	Average	Nodes: 256 Dropout: 0.05	Adam ($2e-5$)	124,853,033
GPT2	Units: 768 Layers: 12	Average	Nodes: 256 Dropout: 0.02	Adam ($2e-5$)	124,647,209
DialoGPT	Units: 768 Layers: 12	Average	Nodes: 256 Dropout: 0.02	Adam ($2e-5$)	124,853,033
XLNet	Units: 768 Layers: 12	Average	Nodes: 256 Dropout: 0.02	Adam ($2e-5$)	116,925,737
ConveRT	Units: 512 Layers: 2	N/A	Nodes: 256 Dropout: 0.02	Adam (0.001	272,937
ELMo	Units: 1024	Average	Nodes: 256 Dropout: 0.01	Adagrad (0.04)	272,941
USE	Units: 512	N/A	Nodes: 256 Dropout: 0.02	Adam (0.001	141,865
mLSTM	Units: 1024 Chars: 64	Average	Nodes: 128 Dropout: 0.02	Adam (0.001)	529,705
NNLM	Units: 128	N/A	Nodes: 256 Dropout: 0.02	Adam (0.0001	124,686,249

Table E.3: Summary of Hyperparameters for the base context encoders. We use Comet.ml to tune each model and results can be viewed at: comet.ml/nathanduran/context-encoding-for-da-model-optimisation.

Sentence Model	**Context Encoder**		**Dim Reduction**	**Classifier**	**Optimiser** (α)	**Trainable Params**
BERT	CNN	Filters: 32 Kernel Size: [6, 4, 2]	Max Pool Size: 2	Nodes: 256 Dropout: 0.1	Adam ($2e-5$)	109,901,420
	LSTM	Units: 512 Dropout: 0.2	Average	Nodes: 512/256 Dropout: 0.1		113,821,708
RoBERTa	CNN	Filters: 32 Kernel Size: [6, 4, 2]	Max Pool Size: 2	Nodes: 256 Dropout: 0.1	Adam ($2e-5$)	125,064,812
	LSTM	Units: 512 Dropout: 0.2	Average	Nodes: 512/256 Dropout: 0.1		128,985,100
LSTM	CNN	Filters: 128 Kernel Size: [6, 4, 2]	Average Pool Size: 2	Nodes: 128/64 Dropout: 0.1	RMSprop (0.001)	2,495,740
	LSTM	Units: 256 Dropout: 0.1	Average	Nodes: 128 Dropout: 0.1		3,594,940
TextCNN	CNN	Filters: 128 Kernel Size: [6, 4, 2]	Average Pool Size: 2	Nodes: 256 Dropout: 0.1	RMSprop (0.001)	1,167,036
	LSTM	Units: 256 Dropout: 0.1	N/A	Nodes: 256 Dropout: 0.1		2,036,796

Appendix F

Model Variants Results

F.1 Supervised Model Variants

F.1.1 Multi-layer and Bi-directional Models

In addition to our baseline recurrent models (LSTM and GRU) we also test their bi-directional and multi-layer variants, both of which have previously been explored within DA classification studies (Ribeiro, Ribeiro, and De Matos, 2019; Bothe et al., 2018a; Chen et al., 2018; Kumar et al., 2017). The bi-directional models (Bi-LSTM and Bi-GRU) process the input sequence in the *forwards* and then *backwards* directions. Each pass generates a 256 dimensional vector (equivalent to the number of hidden units) per timestep, which are then concatenated to form a single 512 dimensional vector. As with the baseline recurrent models, we take the output at each timestep and apply max pooling for dimensionality reduction. The multi-layer models (Deep-LSTM and Deep-GRU), simply stack multiple recurrent layers on top of each other, with the output for a given layer, at each timestep, becoming the input for the following layer. We use the same number of hidden units and apply the same max pooling operation as the other recurrent models.

Table F.1 shows our results for 1, 2, and 3-layer LSTM and GRU models on both corpora. Starting with the LSTM models, we can see that for the SwDA data the single layer LSTM outperforms the 2 and 3-layer variants, and on Maptask the 2-layer LSTM yields a small improvement, though in all cases this is non-significant ($P(rope) \geq .97$). For the GRU models results are inverted, with the 2-layer, and 1-layer GRU resulting in better performance on SwDA and Maptask respectively. However, in both cases the differences between the GRU models is not statistically significant, with $P(2\text{-}lyr > 1\text{-}lyr) = .38$, and $P(1\text{-}lyr > 2\text{-}lyr) = .61$, for SwDA and Maptask respectively. Our results for the LSTM models support those reported by Kumar et al. (2017), and others (Ribeiro, Ribeiro, and De Matos, 2019; Papalampidi, Iosif, and Potamianos, 2017), who also found that increasing the number of layers did not lead to an improvement in performance. Regarding the difference we observed between the LSTM and GRU models, we speculate that this is likely due to the difference in the number of parameters between them. The single layer LSTM has ~2.7 million parameters, while the GRU has ~1.1 million. Thus, the GRU benefited more from an increased number of parameters when applied to the larger SwDA dataset, while the same factor may have led to overfitting on the smaller Maptask data.

Comparing the Bi-LSTM and Bi-GRU to their uni-directional equivalents: for SwDA both uni-directional models outperformed the bi-directional ($P(Uni > Bi) \geq .99$), and for Maptask they were equivalent ($P(rope) \geq .99$). Bi-directional models have been employed at both the context/discourse level (Chen et al., 2018; Kumar et al., 2017), and for sentence encoding (Li et al., 2019b; Bothe et al., 2018a). However, for the latter task it seems bi-directionality, at least in isolation, has no benefit.

Table F.1: Validation accuracy for 1, 2, and 3-layer recurrent models on the SwDA and Maptask data.

	SwDA		Maptask	
Model	μ	σ	μ	σ
LSTM 1-lyr	**75.76**	**0.16**	58.15	0.15
LSTM 2-lyr	75.40	0.14	**58.30**	**0.17**
LSTM 3-lyr	75.37	0.20	58.10	0.24
GRU 1-lyr	74.80	0.16	**58.49**	**0.36**
GRU 2-lyr	**75.32**	**0.18**	57.96	0.35
GRU 3-lyr	75.09	0.19	57.48	0.28

F.1.2 Attentional Models

Throughout the DA classification literature different attention mechanisms have been applied, in various contexts (Bothe et al., 2018a; Chen et al., 2018; Ortega and Vu, 2017; Tran, Haffari, and Zukerman, 2017; Shen and Lee, 2016). We investigate the effect of adding a simple attention mechanism to each of our supervised models. During parameter tuning we tested both additive attention (Bahdanau, Cho, and Bengio, 2015), and multiplicative attention (Luong, Pham, and Manning, 2015), and found that in all cases additive resulted in the best performance. We incorporated the attention mechanism into our models by inserting an attentional layer between the utterance encoder layer and the dimensionality reduction layer. The attention layer takes as input the encoded utterance, and its output is later concatenated with the *original* utterance encoding, *before* being passed to the classification layers.

Our experimental results indicate that no attentional models show an improvement, and in most cases attention was detrimental to performance. On SwDA the only statistically significant result is for the DCNN model over its attentional variant ($P(DCNN > DCNN\text{-}Attn) = .99$), with an increase of just 0.77%, and for Maptask all models are considered equivalent to their attentional variants($P(rope) \geq .81$). As with bi-directional recurrent models, attention mechanisms are frequently combined at both the context/discourse, and sentence encoding, level (Li et al., 2019b; Bothe et al., 2018a; Tran, Haffari, and Zukerman, 2017; Tran, Zukerman, and Haffari, 2017; Shen and Lee, 2016). While most of these attention mechanisms are unique implementations, and therefore not directly comparable to our experiments, it does seem that the benefit of using standard attention mechanisms for sentence encoding should be accompanied by appropriate testing to establish its true impact on performance.

F.2 Context Model Variants

As with our sentence encoders (F.1), we explore several variations of recurrent context encoder models. Including, a GRU, a bi-directional GRU and LSTM, and additive, or multiplicative attention with a GRU as the base context encoder. Each uses RoBERTa as a sentence encoder and we use 5 previous context utterances. We only apply these models to the Maptask corpus, with results shown in table F.2.

It can be seen that there are minimal differences across all model variants, with at most a +0.61% difference between Bi-GRU and Attn-Add. Unsurprisingly, in no case do we see statistically significant differences. Instead, we only see statistically significant equivalence; for Bi-GRU and LSTM ($P(rope) = .82$), and conversely Bi-LSTM and

Table F.2: Context encoder bi-directional and attentional variant result for the Maptask validation set.

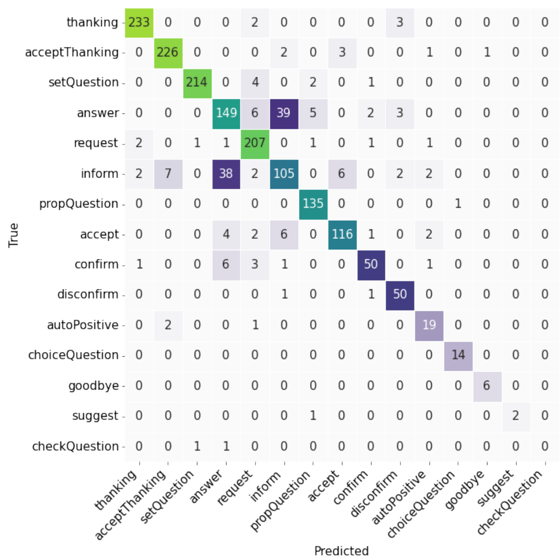
Model	μ	σ
RoBERTa-LSTM	67.93	0.57
RoBERTa-Bi-LSTM	67.76	0.54
RoBERTa-GRU	67.84	0.45
RoBERTa-Bi-GRU	67.46	0.50
RoBERTa-Attn-Add	**68.07**	**0.47**
RoBERTa-Attn-Dot	68.06	0.46

GRU ($P(rope) = .84$). For the attentional models Attn-Add is equivalent to the GRU ($P(rope) = .82$), and also Attn-Dot ($P(rope) = .85$). Thus, similar to our results for sentence encoder variants, it appears the addition of bi-directional and attentional layers do not result in significant improvements. Such models are used extensively throughout the DA classification literature (Li et al., 2019b; Raheja and Tetreault, 2019; Chen et al., 2018; Kumar et al., 2017; Ortega and Vu, 2017), yet are often not accompanied by appropriate ablation studies to determine if their effect on performance is indeed significant. We, therefore, continue using only RoBERTa-LSTM as our base context model. In the following section we extend this model, by investigating various speaker and label encoders, and establish the impact of incorporating such contextual information.

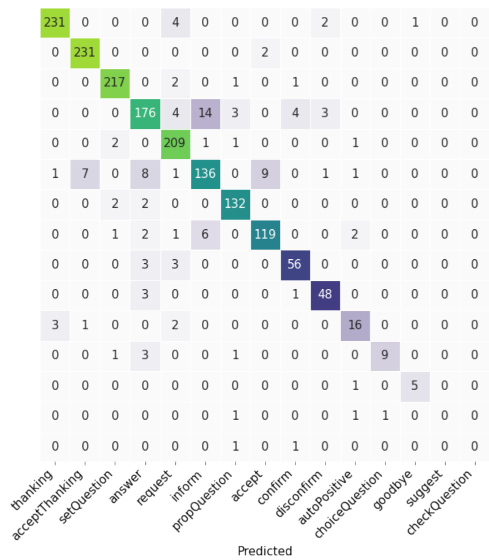
Appendix G

CAMS Classification Results

G.1 AP Identification Results



(a) No-context



(b) Context

Figure G.1: Confusion matrices for DA predictions on the validation set using non-contextual (a), and contextual (b), classification models. Note that here we are only showing the 15 most frequent labels.

Table G.1: Difference in validation set accuracy and top-3 accuracy, between full and partial context models and the baseline classifiers (RoBERTa, prior, and most frequent label).

	RoBERTa		**Prior**		**Most Frequent**	
Label Type	**Acc**	**Top-3 Acc**	**Acc**	**Top-3 Acc**	**Acc**	**Top-3 Acc**
			Full Context			
DA	5.57	10.3	83.56	88.29	80.56	85.29
AP	13.37	13.97	69.78	70.37	62.27	62.86
AP-type	12.61	17.23	84.97	90.58	78.46	84.07
			Partial Context			
DA	5.02	10.37	83.01	88.36	80.01	85.36
AP	2.28	11.29	58.69	67.69	51.18	60.18
AP-type	5.86	14.49	78.22	87.84	71.71	81.33

Table G.2: Validation and test set accuracy and top-3 accuracy, as well as macro and weighted F1 scores, for each label type. *DA First*, *AP First*, and *Parallel* are the three different multi-label architectures tested, and k denotes the number of probabilities that were concatenated with the dialogue segment encoding.

	Validation				**Test**					
Label Type	μ	σ	*top3-μ*	*top3-σ*	μ	σ	*top3-μ*	*top3-σ*	*macro-F1*	*weighted-F1*
					DA First					
DA	93.73	0.46	99.15	0.12	94.35	0.28	99.11	0.13	0.7	0.94
AP	88.56	0.45	97.56	0.28	88.25	0.48	97.62	0.4	0.75	0.86
AP-types	85.23	0.45	93.55	0.18	84.78	0.4	92.72	0.4	0.28	0.81
					AP First					
DA	93.5	0.43	99.12	0.22	94.44	0.33	99.05	0.16	0.7	0.94
AP	88.51	0.4	97.4	0.32	88.48	0.29	97.79	0.49	0.71	0.86
AP-types	85.19	0.49	93.62	0.42	85.01	0.2	92.49	0.42	0.27	0.81
					Parallel					
DA	93.61	0.38	99.14	0.14	94.38	0.3	99.11	0.12	0.7	0.94
AP	88.87	0.32	97.66	0.33	88.22	0.28	97.81	0.27	0.74	0.86
AP-types	85.31	0.27	93.57	0.39	84.67	0.28	92.44	0.35	0.28	0.81
					DA First k=3					
DA	93.58	0.31	99.1	0.08	94.46	0.31	99.03	0.1	0.7	0.94
AP	88.88	0.34	97.65	0.29	88.58	0.18	97.83	0.41	0.75	0.87
AP-types	85.43	0.31	93.52	0.56	85.05	0.32	92.47	0.53	0.28	0.81
					AP First k=3					
DA	93.48	0.35	99.11	0.07	94.47	0.48	99.06	0.13	0.7	0.94
AP	88.58	0.3	97.51	0.23	88.37	0.44	97.9	0.47	0.73	0.86
AP-types	85.05	0.39	93.63	0.52	84.7	0.48	92.53	0.51	0.28	0.81
					Parallel k=3					
DA	93.68	0.24	99.09	0.11	94.37	0.31	99.04	0.12	0.7	0.94
AP	88.62	0.18	97.57	0.17	88.34	0.33	97.94	0.53	0.72	0.87
AP-types	85.03	0.16	93.81	0.51	84.76	0.36	92.63	0.39	0.28	0.81

G.2 Next-label Results

Table G.3: Difference in validation set accuracy and top-3 accuracy, between full and partial context next-label models, the baseline classifiers (RoBERTa, prior, and most frequent label), and the contextual RoBERTa-LSTM.

	RoBERTa-LSTM		**RoBERTa**		**Prior**		**Most Frequent**	
Label Type	**Acc**	**Top-3 Acc**	**Acc**	**Top-3 Acc**	**Acc**	**Top-3 Acc**	**Acc**	**Top-3 Acc**
				Full Context				
DA	-11.26	-0.63	2.11	13.33	58.52	69.75	51.0	62.23
AP	-28.31	-10.24	-15.69	7.98	56.67	80.34	50.15	73.82
AP-type	-28.51	-8.14	-22.94	2.15	55.05	80.14	52.05	77.15
				Partial Context				
DA	-6.32	-1.62	-4.04	9.67	52.37	66.08	44.86	58.56
AP	-24.39	-10.11	-18.53	5.37	53.84	77.73	47.32	71.21
AP-type	-29.47	-9.11	-24.45	1.26	53.54	79.25	50.55	76.25

Table G.4: Next-label validation and test set accuracy and top-3 accuracy, as well as macro and weighted F1 scores, for each label type. *DA First*, *AP First*, and *Parallel* are the three different multi-label architectures tested, and k denotes the number of probabilities that were concatenated with the dialogue segment encoding.

	Validation				**Test**					
Label Type	μ	σ	*top3-μ*	*top3-σ*	μ	σ	*top3-μ*	*top3-σ*	***macro-F1***	***weighted-F1***
					DA First					
DA	64.48	0.54	89.96	0.52	66.57	0.64	90.76	0.32	0.36	0.63
AP	82.15	0.27	96.19	0.34	81.54	0.32	95.59	0.24	0.44	0.76
AP-types	60.31	0.61	84.13	0.37	62.39	0.58	83.52	0.22	0.14	0.55
					AP First					
DA	64.59	0.6	90.04	0.35	66.36	0.64	90.64	0.39	0.36	0.63
AP	82.15	0.4	96.18	0.16	81.39	0.6	95.61	0.27	0.48	0.76
AP-types	60.46	0.46	83.87	0.46	62.18	0.71	83.71	0.44	0.13	0.55
					Parallel					
DA	64.14	0.78	89.99	0.29	66.33	0.94	90.58	0.31	0.36	0.63
AP	82.27	0.41	96.06	0.25	81.35	0.43	95.57	0.28	0.48	0.76
AP-types	60.39	0.56	83.91	0.28	62.26	0.58	83.56	0.31	0.14	0.55
					DA First k=3					
DA	64.5	0.74	90.18	0.3	66.71	1.04	90.74	0.32	0.36	0.63
AP	82.24	0.37	96.11	0.33	81.47	0.38	95.6	0.18	0.46	0.76
AP-types	60.34	0.92	84.1	0.4	62.39	0.8	83.69	0.39	0.14	0.55
					AP First k=3					
DA	64.25	0.41	89.95	0.3	66.7	0.62	90.6	0.41	0.37	0.63
AP	82.2	0.63	96.08	0.22	81.38	0.4	95.51	0.16	0.46	0.76
AP-types	60.03	0.84	83.88	0.35	62.24	0.59	83.59	0.37	0.14	0.55
					Parallel k=3					
DA	64.63	0.56	90.02	0.52	66.49	0.85	90.51	0.41	0.36	0.62
AP	82.21	0.47	96.13	0.19	81.16	0.64	95.53	0.23	0.48	0.76
AP-types	60.83	0.46	83.86	0.39	62.29	0.82	83.4	0.45	0.14	0.54

Appendix H

Dialogue Structure Evaluation and Analysis

Table H.1: BLEU, METEOR, and ROUGE-L scores for GPT-3 response generation with different label types included in the prompt.

Label Type	BLEU	METEOR	ROUGE-L
No Label	0.078	0.412	0.413
DA	0.078	**0.452**	0.449
AP	0.078	0.436	0.434
AP-types	0.078	0.451	**0.452**

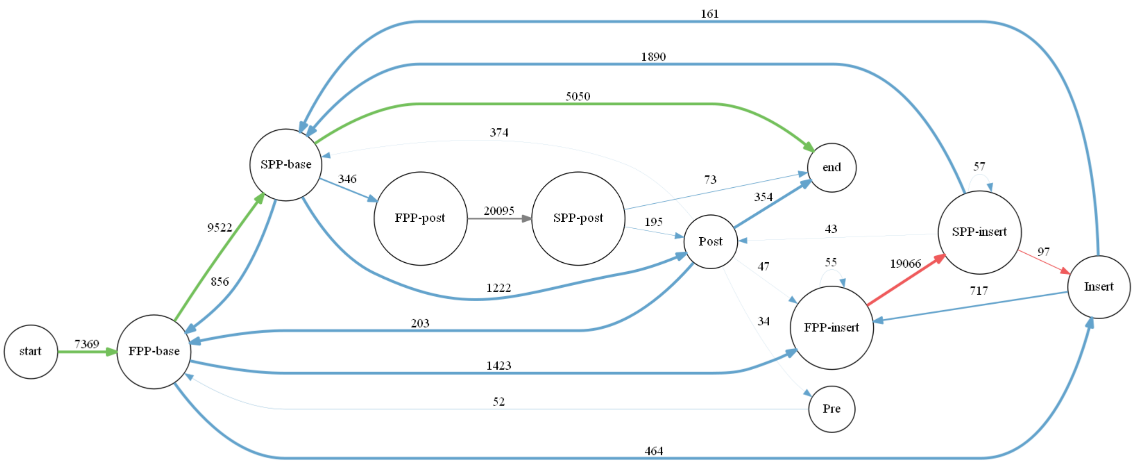


Figure H.1: A directed dialogue structure graph with 11 nodes and 25 edges. Created using the χ^2 data for all AP within CAMS-KVRET. Nodes are AP and edges are transitions from one AP to another. Edges are coloured according to the χ^2 value, and edge widths represent the frequency of occurrences.

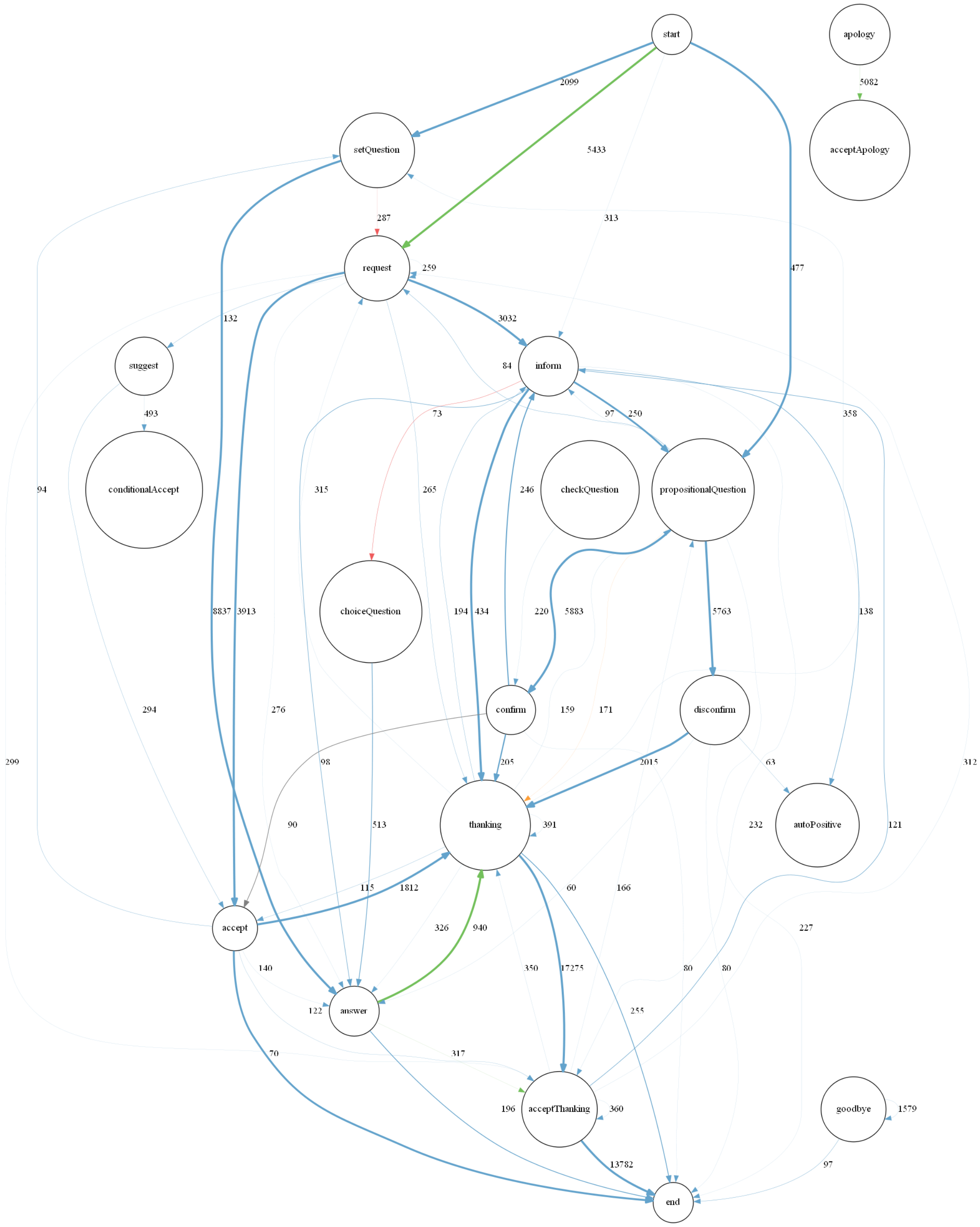


Figure H.2: A directed dialogue structure graph with 20 nodes and 63 edges. Created using the χ^2 data for all DA within CAMS-KVRET. Nodes are DA and edges are transitions from one DA to another. Edges are coloured and labelled according to the χ^2 value, and edge widths represent the frequency of occurrences.

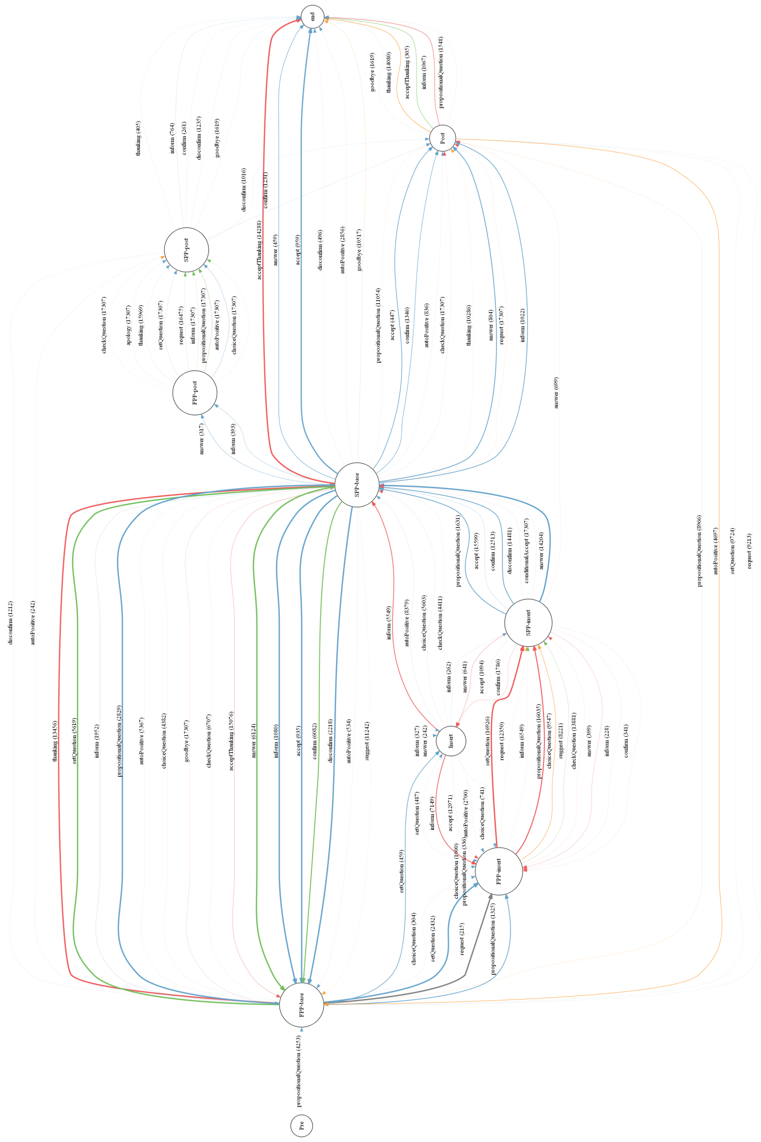
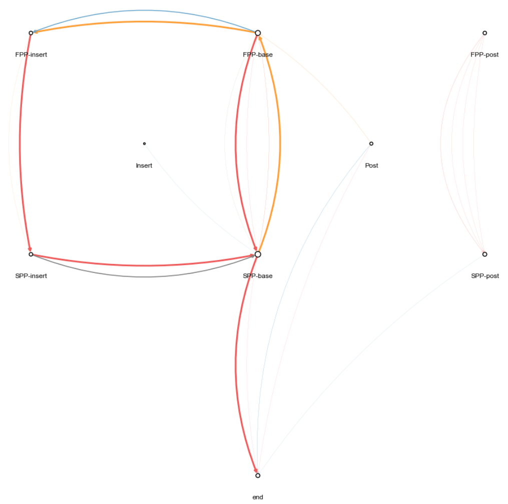
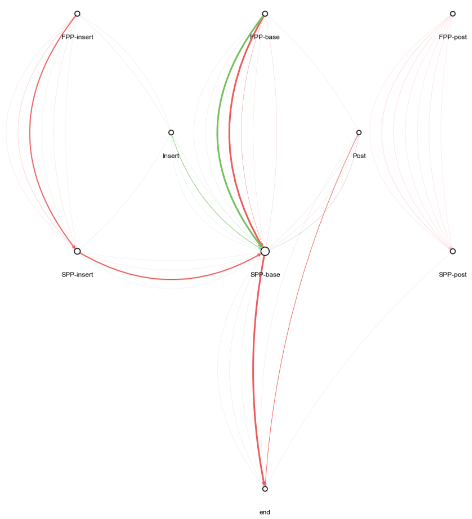


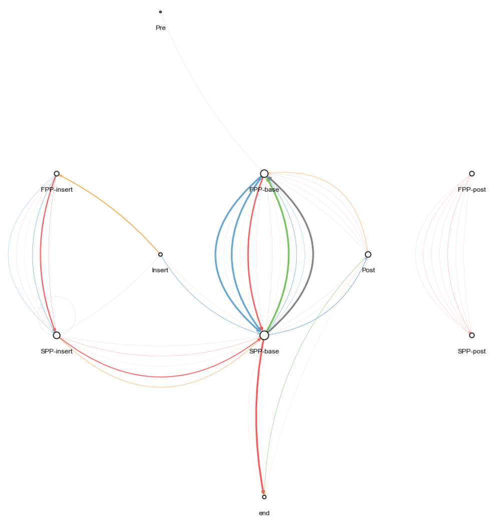
Figure H.3: A directed dialogue structure graph with 10 nodes and 100 edges. Created using the χ^2 data for all AP-types within CAMS-KVRET. Nodes are AP and edges are DA transitions from one AP to another. Edges are coloured and labelled according to the χ^2 value, and edge widths represent the frequency of occurrences.



(a) DA Weather



(b) DA Schedule



(c) DA Navigate

Figure H.4: AP-type directed dialogue structure graphs for each dialogue task type within CAMS-KVRET. Nodes are AP and edges are DA transitions from one AP to another. Edges are coloured according to the χ^2 value, and edge widths represent the frequency of occurrences.

Bibliography

Abu-Aisheh, Z., R. Raveaux, J. Y. Ramel, and P. Martineau (2015). "An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems". In: *ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings*. Vol. 1. Lisbon, Portugal, pp. 271–278. ISBN: 9789897580765. DOI: 10.5220/0005209202710278.

Ahmadvand, A., J. I. Choi, and E. Agichtein (2019). "Contextual Dialogue Act Classification for Open-Domain Conversational Agents". In: *SIGIR'19 Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Paris, France: ACM, pp. 1273–1276. ISBN: 9781450361729. DOI: 10.1145/3331184.

Albert, S., L. E. de Ruiter, and J. de Ruiter (2015). *CABNC: The Jeffersonian Transcription of the Spoken British National Corpus*.

Alexandersson, J., B. Buschbeck-Wolf, T. Fujinami, E. Maier, N. Reithinger, B. Schmitz, and M. Siegel (1997). *Dialog Acts in VERBMOBIL-2*. Tech. rep.

Allen, J. F., B. W. Miller, E. K. Ringger, and T. Sikorski (1996). "A Robust System for Natural Spoken Dialogue". In: *Proceedings of the 34th annual meeting of the Association of Computational Linguistics*. June, pp. 62–70. DOI: 10.3115/981863.981872.

Allen, J. and M. Core (1997). *Draft of DAMSL: Dialog Act Markup in Several Layers*. Tech. rep.

Alloatti, F., L. D. Caro, and A. Bosca (2020). "Conversation Analysis, Repair Sequences and Human Computer Interaction A Theoretical Framework and an Empirical Proposal of Action". In: *The Fourth Workshop on Reasoning and Learning for Human-Machine Dialogues at the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Allwood, J. (1976). "Linguistic Communication as Action and Cooperation". PhD thesis. G¨oteborg University, p. 170. DOI: 10.1016/0378-2166(78)90006-1.

Anderson, A. H., M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson, and R. Weinert (1991). "The HCRC Map Task Corpus". In: *Language and Speech* 34.4, pp. 351–366. ISSN: 00238309. DOI: 10.1177/002383099103400404.

Anikina, T. and I. Kruijff-Korbayova (2019). "Dialogue Act Classification in Team Communication for Robot Assisted Disaster Response". In: *Proceedings of the SIGDial 2019*. Stockholm, Sweden: Association for Computational Linguistics, pp. 399–410. DOI: 10.18653/v1/w19-5946.

Artstein, R. (2018). "Inter-annotator Agreement". In: *Handbook of Linguistic Annotation*. Ed. by N. Ide and J. Pustejovsky. Springer. Chap. 10, pp. 297–314. ISBN: 9789402408799.

Artstein, R. and M. Poesio (2005a). "Bias Decreases in Proportion to the Number of Annotators". In: *Proceedings of the Conference on Formal Grammar and Mathematics of Language (FG-MoL)*, pp. 141–150.

Artstein, R. and M. Poesio (2005b). *Kappa 3 = Alpha (or Beta)*. Tech. rep. September, pp. 1–40.

Artstein, R. and M. Poesio (2008). "Inter-Coder Agreement for Computational Linguistics". In: *Computational Linguistics* 34.4, pp. 555–596.

Asri, L. E., H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman (2017). "Frames: A Corpus for Adding Memory to Goal-Oriented Dialogue Systems". In: *Proceedings of the SIGDIAL 2017 Conference*. Saarbrücken, Germany: Association for Computational Linguistics, pp. 207–219. DOI: 10.18653/v1/W17-5526. arXiv: 1704.00057.

Aulamo, M., M. Creutz, and E. Sjöblom (2019). "Annotation of Subtitle Paraphrases Using a New Web Tool". In: *Proceedings of 4th Conference of The Association Digital Humanities in the Nordic Countries*. CEUR-WS.org.

Austin, J. L. (1962). *How To Do Things With Words*. London: Oxford University Press, p. 168.

Bahdanau, D., K. Cho, and Y. Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *ICLR 2015*. ISBN: 0147-006X (Print). DOI: 10.1146/annurev.neuro.26.041002.131047. arXiv: 1409.0473.

Banerjee, M., M. Capozzoli, L. McSweeney, and D. Sinha (1999). "Beyond kappa: A Review of Interrater Agreement Measures". In: *Canadian Journal of Statistics* 27.1, pp. 3–23. ISSN: 03195724. DOI: 10.2307/3315487.

Banerjee, S. and A. Lavie (2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Transla- tion and/or Summarization,* Ann Arbor, Michigan, pp. 65–72.

Bangalore, S., G. Di Fabbrizio, and A. Stent (2008). "Learning the Structure of Task-Driven Human–Human Dialogs". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.7, pp. 1249–1259. ISSN: 1558-7916. DOI: 10.1109/TASL.2008.2001102.

Barreiro, A., J. G. C. de Souza, A. Gatt, M. Bhatt, E. Lloret, A. Erdem, D. Gkatzia, H. Moniz, I. Russo, F. Kepler, I. Calixto, M. Paprzycki, F. Portet, I. Augenstein, and M. Alhasani (2022). "Multi3Generation: Multitask, Multilingual, Multimodal Language Generation". In: *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pp. 347–348.

Bayerl, P. S. and K. I. Paul (2011). "What Determines Inter-coder Agreement in Manual Annotations? A meta-analytic Investigation". In: *Computational Linguistics* 37.4, pp. 699–725. ISSN: 15309312. DOI: 10.1162/COLI_a_00074.

Beltagy, I., M. E. Peters, and A. Cohan (2020). "Longformer: The Long-Document Transformer". In: *arXiv*. arXiv: 2004.05150.

Benavoli, A., G. Corani, J. Demšar, and M. Zaffalon (2017). "Time for a Change: A Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis". In: *Journal of Machine Learning Research* 18, pp. 1–36. ISSN: 15337928. arXiv: 1606.04316.

Bengio, Y., R. Ducharme, P. Vincent, C. Jauvin, J. Kandola, T. Hofmann, T. Poggio, and J. Shawe-Taylor (2003). "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3, pp. 1137–1155.

Biber, D., J. Egbert, D. Keller, and S. Wizner (2021). "Towards a Taxonomy of Conversational Discourse Types: An Empirical Corpus-based Analysis". In: *Journal of Pragmatics* 171, pp. 20–35. ISSN: 03782166. DOI: 10.1016/j.pragma.2020.09.018.

Bifis, A., M. Trigka, S. Dedegkika, P. Goula, C. Constantinopoulos, and D. Kosmopoulos (2021). "A Hierarchical Ontology for Dialogue Acts in Psychiatric Interviews". In: *PETRA 2021: The 14th PErvasive Technologies Related to Assistive Environments Conference*. ACM, pp. 330–337. ISBN: 9781450387927. DOI: 10.1145/3453892.3461349.

Black, R., P. O. Kristensson, J. Zhang, A. Waller, S. Bano, Z. Rashid, and C. Norrie (2016). "ACELP - Augmenting Communication using Environmental Data to drive Language

Prediction". In: *Poster session presented at Communication Matters - CM2016 National Conference*. Leeds, UK.

Bothe, C., S. Magg, C. Weber, and S. Wermter (2018a). "Conversational Analysis Using Utterance-level Attention-based Bidirectional Recurrent Neural Networks". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 2018-Sept. Hydrabad, pp. 996–1000. DOI: 10.21437/Interspeech.2018-2527. arXiv: 1805.06242.

Bothe, C., C. Weber, S. Magg, and S. Wermter (2018b). "A Context-based Approach for Dialogue Act Recognition using Simple Recurrent Neural Networks". In: *Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan. arXiv: 1805.06280.

Bouckaert, R. R. (2003). "Choosing Between Two Learning Algorithms Based on Calibrated Tests". In: *Proceedings, Twentieth International Conference on Machine Learning*. Vol. 1, pp. 51–58. ISBN: 1577351894.

Boxman-Shabtai, L. (2020). "Meaning Multiplicity Across Communication Subfields: Bridging the Gaps". In: *Journal of Communication* 70.3, pp. 401–423. ISSN: 14602466. DOI: 10.1093/joc/jqaa008.

Boyer, K. E., E. Y. Ha, R. Phillips, M. D. Wallis, M. A. Vouk, and J. Lester (2009a). "Inferring Tutorial Dialogue Structure with Hidden Markov Modeling". In: *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications - EdAppsNLP '09*. June, pp. 19–26. ISBN: 9781932432374. DOI: 10.3115/1609843.1609846.

Boyer, K. E., E. Y. Ha, R. Phillips, M. D. Wallis, M. A. Vouk, and J. Lester (2010a). "Dialogue Act Modeling in a Complex Task-Oriented Domain". In: *Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group in Discourse and Dialogue*, pp. 297–305. ISBN: 978-1-932432-85-5.

Boyer, K. E., R. Phillips, E. Y. Ha, M. D. Wallis, M. A. Vouk, and J. Lester (2009b). "Modeling Dialogue Structure with Adjacency Pair Analysis and Hidden Markov Models". In: *NAACL-HLT 2009*. June. Association for Computational Linguistics, pp. 49–52. DOI: 10.3115/1620853.1620869.

Boyer, K. E., R. Phillips, E. Y. Ha, M. D. Wallis, M. A. Vouk, and J. Lester (2010b). "Leveraging Hidden Dialogue State to Select Tutorial Moves". In: *NAACL-HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*. June, pp. 66–73.

Boyer, K. E., R. Phillips, A. Ingram, E. Y. Ha, M. D. Wallis, M. A. Vouk, and J. Lester (2011). "Investigating the Relationship Between Dialogue Structure and Tutoring Effectiveness: A Hidden Markov Modeling Approach". In: *International Journal of Artificial Intelligence in Education* 21.1, pp. 65–81.

British Standards Institution (2012). *ISO 24617-2: Language Resource Management - Semantic Annotation Framework (SemAF) Part 2: Dialogue acts.*

Brown, P. E., V. J. D. Pietra, R. L. Mercer, S. a. D. Pietra, and J. C. Lai (1992). "An Estimate of an Upper Bound for the Entropy of English". In: *Computational Linguistics* 18.1, pp. 31–40. ISSN: 0891-2017.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020). "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* NeurIPS. ISSN: 10495258. arXiv: 2005.14165.

Budzinski, O., V. Noskova, and X. Zhang (2019). "The Brave New World of Digital Personal Assistants: Benefits and Challenges from an Economic Perspective". In: *NET-*

NOMICS: Economic Research and Electronic Networking 20.2-3, pp. 177–194. ISSN: 15737071. DOI: 10.1007/s11066-019-09133-4.

Bunt, H. (1978). *Conversational Principles in Question-answer Dialogues*. Essen: Tubingen, pp. 119–142. ISBN: 3-87808-652-0.

Bunt, H. (1990). *DIT: Dynamic Interpretation in Text and Dialogue*. Tech. rep. Tilburg School of Economics and Management, p. 45.

Bunt, H. (2000). "Dialogue Pragmatics and Context Specification". In: *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*. Ed. by H. Bunt and W. Black. January 2000. Amsterdam: John Benjamins, pp. 81–149. DOI: 10.1075/nlp.1.03bun.

Bunt, H. (2006). "Dimensions in Dialogue Act Annotation". In: *Proceeding of LREC 2006*, pp. 919–924.

Bunt, H. (2009). "The DIT++ Taxonomy for Functional Dialogue Markup". In: *Proceedings of the AAMAS 2009 Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts" (EDAML 2009)*, pp. 13–24. arXiv: 1205.0675 [cond-mat.mes-hall].

Bunt, H. (2011). "The Semantics of Dialogue Acts". In: *International Conference on Computational Semantics IWCS '11*. Oxford, England: Association for Computational Linguistics, pp. 1–13.

Bunt, H. (2012). *Data Categories for Dialogue Acts*. Tech. rep. Tilburg University.

Bunt, H. (2017). *Guidelines for using ISO standard 24617-2*. Tech. rep. Tilburg Center for Cognition and Communication.

Bunt, H., J. Alexandersson, J.-W. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-belis, and D. Traum (2012). "ISO 24617-2 : A Semantically-based Standard for Dialogue Annotation". In: *Proceedings of LREC 2012*, pp. 430–437. ISBN: 978-2-9517408-7-7.

Bunt, H., A. C. Fang, and X. Liu (2013). "Issues in the Addition of ISO Standard Annotations to the Switchboard Corpus". In: *Proceedings of the 9th Joint ISO ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9)*. Stroudsburg, Pennsylvania: Association for Computational Linguistics, pp. 67–78.

Burke, P. (1993). *The Art of Conversation*. 1st. Polity Press. ISBN: 9780745665818.

Byrt, T., J. Bishop, and J. B. Carlin (1993). "Bias, Prevalence and Kappa". In: *Journal of Clinical Epidemiology* 46.5, pp. 423–429. ISSN: 08954356. DOI: 10.1016/0895-4356(93)90018-V.

Carletta, J. (1996). "Assessing Agreement on Classification Tasks: The Kappa Statistic". In: *Computational linguistics* 22.2, pp. 249–254. ISSN: 0891-2017.

Carletta, J., A. Isard, S. Isard, J. C. Kowtko, G. Doherty-Sneddon, and A. H. Anderson (1997). "The Reliability of a Dialogue Structure Coding Scheme". In: *Computational Linguistics* 23.1, pp. 13–31. ISSN: 0891-2017.

Carletta, J., A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon, and A. Anderson (1996). *HCRC Dialogue Structure Coding Manual*. Tech. rep. Human Communication Research Centre.

Cer, D., Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil (2018). "Universal Sentence Encoder". In: *arXiv*. arXiv: 1803.11175.

Cerisara, C., P. Král, and L. Lenc (2017). "On the Effects of Using Word2vec Representations in Neural Networks for Dialogue Act Recognition". In: *Computer Speech and Language* 47.July, pp. 175–193. ISSN: 10958363. DOI: 10.1016/j.csl.2017.07.009.

Chen, H., X. Liu, D. Yin, and J. Tang (2017). "A Survey on Dialogue Systems: Recent Advances and New Frontiers". In: *ACM SIGKDD Explorations Newsletter* 19.2, pp. 25–35. DOI: https://doi.org/10.1145/3166054.3166058. arXiv: 1711.01731.

Chen, Z., R. Yang, Z. Zhao, D. Cai, and X. He (2018). "Dialogue Act Recognition via CRF-Attentive Structured Network". In: *SIGIR '18 The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*. Ann Arbor, USA, pp. 225–234. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3209997. arXiv: 1711.05568.

Cho, K., B. van Merrienboer, D. Bahdanau, and Y. Bengio (2014a). "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111. DOI: 10.3115/v1/w14-4012. arXiv: 1409.1259.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014b). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings ofthe 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Quatar: Association for Computational Linguistics, pp. 1724–1734.

Chotimongkol, A. (2008). "Learning the Structure of Task-Oriented Conversations from the Corpus of In-Domain Dialogs". PhD thesis. Carnegie Mellon University, p. 290.

Clift, R. (2016). *Conversation Analysis*. Cambridge University Press. ISBN: 978-0-521-19850-9.

Cohen, J. (1960). "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1, pp. 37–46. DOI: 10.1177/001316446002000104.

Cohen, J. (1968). "Weighted kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit". In: *Psychological Bulletin* 70.4, pp. 213–220. ISSN: 00332909. DOI: 10.1037/h0026256.

Collins, H., W. Leonard-Clarke, and H. O'Mahoney (2019). "'Um, er': How Meaning Varies Between Speech and its Typed Transcript". In: *Qualitative Research* 19.6, pp. 653–668. ISSN: 17413109. DOI: 10.1177/1468794118816615.

Colombo, P., E. Chapuis, M. Manica, E. Vignon, G. Varni, and C. Clavel (2020). "Guiding Attention in Sequence-to-sequence Models for Dialogue Act Prediction". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. DOI: 10.1609/aaai.v34i05.6259. arXiv: 2002.08801.

Cook, A. M., J. M. Polgar, and P. Encarnação, eds. (2020). *Assistive Technologies*. 5th, pp. 393–439. ISBN: 9780323523387.

Coulter, J. (1995). "The Sacks Lectures". In: *Human Studies* 18, pp. 327–336.

Craggs, R. and M. M. Wood (2005). "Evaluating Discourse and Dialogue Coding Schemes". In: *Computational Linguistics* 31.3, pp. 289–295. ISSN: 08912017. DOI: 10.1162/089120105774321109.

Cuayáhuitl, H., S. Yu, A. Williamson, and J. Carse (2016). "Deep Reinforcement Learning for Multi-Domain Dialogue Systems". In: *NIPS Workshop on Deep Reinforcement Learning*. Barcelona, Spain, pp. 1–9. arXiv: arXiv:1611.08675v1.

Cutts, M. (2013). *Oxford Guid to Plain English*. 4th. New York, NY: Oxford University Press. ISBN: 9780199669172.

Dai, Y., H. Yu, Y. Jiang, C. Tang, Y. Li, and J. Sun (2020). "A Survey on Dialog Management: Recent Advances and Challenges". In: *arXiv*. ISSN: 2331-8422. arXiv: 2005.02233.

Dai, Z., Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov (2019). "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context". In: *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. Association for Computational Linguistics, pp. 2978–2988. ISBN: 9781950737482. DOI: 10.18653/v1/p19-1285. arXiv: 1901.02860.

Davies, M. and J. L. Fleiss (1982). "Measuring Agreement for Multinomial Data". In: *Biometrics* 38.4, p. 1047. ISSN: 0006341X. DOI: 10.2307/2529886.

Demšar, J. (2006). "Statistical Comparisons of Classifiers Over Multiple Data Sets". In: *Journal of Machine Learning Research* 7, pp. 1–30. ISSN: 15337928.

Demšar, J. (2008). "On the Appropriateness of Statistical Tests in Machine Learning". In: *3rd Workshop on Evaluation Methods for Machine Learning*, p. 65.

Deriu, J., A. Rodrigo, A. Otegi, G. Echegoyen, S. Rosset, E. Agirre, and M. Cieliebak (2020). "Survey on Evaluation Methods for Dialogue Systems". In: *Artificial Intelligence Review*. ISSN: 15737462. DOI: 10.1007/s10462-020-09866-x. arXiv: 1905.04071.

Deveci, T. (2019). "Sentence Length in Education Research Articles: A Comparison Between Anglophone and Turkish Authors". In: *Linguistics Journal* 13.1, pp. 73–100. ISSN: 17182301.

Devlin, J., M. W. Chang, K. Lee, and K. Toutanova (2019). "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. Vol. 1, pp. 4171–4186. ISBN: 9781950737130. arXiv: 1810.04805.

Dhillon, R., S. Bhagat, H. Carvey, and E. Shriberg (2004). *Meeting Recorder Project: Dialog Act Labeling Guide*. Tech. rep., p. 129.

Di Eugenio, B. (2000). "On the Usage of Kappa to Evaluate Agreement on Coding Tasks". In: *2nd International Conference on Language Resources and Evaluation, LREC 2000*, pp. 441–444.

Di Eugenio, B. and M. Glass (2004). "The Kappa Statistic: A Second Look". In: *Computational Linguistics* 30.1, pp. 95–101. DOI: https://doi.org/10.1162/089120104773633402.

Dietterich, T. G. (1998). "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". In: *Neural Computation* 10.7, pp. 1895–1923. ISSN: 08997667. DOI: 10.1162/089976698300017197.

Dubay, W. H. (2004). *The Principles of Readability*. Tech. rep.

Eisenstein, J. (2018). *Natural Language Processing*. ISBN: 9781614993797. DOI: 10.1126/science.253.5025.1242.

Ekman, P. and K. Scherer (1984). *Structures of Social Action - Studies in Conversation Analysis*. Ed. by J. Atkinson and J. Heritage. Cambridge University Press. ISBN: 0521248159. DOI: 10.1017/CBO9780511665868.

Enayet, A. and G. Sukthankar (2022). "An Analysis of Dialogue Act Sequence Similarity Across Multiple Domains". In: *Proceedings of the Language Resources and Evaluation Conference*. June. Marseille, France: European Language Resources Association (ELRA), pp. 3122–3130.

Eric, M. and C. D. Manning (2017a). "Key-Value Retrieval Networks for Task-Oriented Dialogue". In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 37–49. arXiv: 1705.05414.

Eric, M. and C. D. Manning (2017b). "Key-Value Retrieval Networks for Task-Oriented Dialogue". In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Saarbrucken, Germany: Association for Computational Linguistics, pp. 37–49. arXiv: 1705.05414.

Ezen-Can, A. and K. E. Boyer (2015). "Understanding Student Language: An Unsupervised Dialogue Act Classification Approach". In: *Journal of Educational Data Mining* 7.1, pp. 51–78. ISSN: 2157-2100.

Fang, A., H. Bunt, and J. Cao (2012). "Collaborative Annotation of Dialogue Acts: Application of a New ISO Standard to the Switchboard Corpus". In: *Eacl 2012*. Avignon, France: Association for Computational Linguistics, pp. 61–68.

Feng, S., N. Lubis, C. Geishauser, H.-c. Lin, M. Heck, C. van Niekerk, and M. Gašić (2021). "EmoWOZ: A Large-Scale Corpus and Labelling Scheme for Emotion Recognition in Task-Oriented Dialogue Systems". In: arXiv: 2109.04919.

Finch, S. E. and J. D. Choi (2020). "Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols". In: *SIGDIAL 2020 - 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue, Proceedings of the Conference*. Association for Computational Linguistics, pp. 236–245. ISBN: 9781952148026. arXiv: 2006.06110.

Fiok, K., W. Karwowski, E. Gutierrez, and M. Reza-Davahli (2020). "Comparing the Quality and Speed of Sentence Classification with Modern Language Models". In: *Applied Sciences* 10.10. ISSN: 20763417. DOI: 10.3390/APP10103386.

Firdaus, M., H. Golchha, A. Ekbal, and P. Bhattacharyya (2020). "A Deep Multi-task Model for Dialogue Act Classification, Intent Detection and Slot Filling". In: *Cognitive Computation*. ISSN: 18669964. DOI: 10.1007/s12559-020-09718-4.

Fleiss, J. L. (1971). "Measuring Nominal Scale Agreement Among Many Raters". In: *Psychological Bulletin* 76.5, pp. 378–382.

Foltz, P. W., W. Kintsch, and T. K. Landauer (1998). "The Measurement of Textual Coherence with Latent Semantic Analysis". In: *Discourse Processes* 25.2-3, pp. 285–307. ISSN: 0163-853X. DOI: 10.1080/01638539809545029.

Forgues, G., J. Pineau, J.-M. Larcheveque, and R. Tremblay (2014). "Bootstrapping Dialog Systems with Word Embeddings". In: *NIPS, Modern Machine Learning and Natural Language Processing Workshop*. Vol. 2, pp. 1–5.

Garfinkel, H. (1967). *Studies in Ethnomethodology*. Prentice Hall.

Gašić, M. (2011). "Statistical Dialogue Modelling". PhD thesis, p. 240.

Ge, W. and B. Xu (2015). "Dialogue Management Based on Multi-domain Corpus". In: *AnnualMeeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 364–373.

Geertzen, J. and H. Bunt (2006). "Measuring Annotator Agreement in a Complex Hierarchical Dialogue Act Annotation Scheme". In: *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. Sydney, Australia: Association for Computational Linguistics, pp. 126–133. ISBN: 193243271X. DOI: 10.3115/1654595.1654619.

Geertzen, J., V. Petukhova, and H. Bunt (2008). "Evaluating Dialogue Act Tagging with Naive and Expert Annotators". In: *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*. Marrakech, Morocco: European Language Resources Association (ELRA), pp. 1076–1082. ISBN: 2951740840.

Geis, M. L. (1995). *Speech Acts and Conversational Interaction*. Cambridge University Press. ISBN: 9780521464994.

Geiß, S. (2021). "Statistical Power in Content Analysis Designs: How Effect Size, Sample Size and Coding Accuracy Jointly Affect Hypothesis Testing – A Monte Carlo Simulation Approach." In: *Computational Communication Research* 3.1, pp. 61–89. ISSN: 2665-9085. DOI: 10.5117/ccr2021.1.003.geis.

Godfrey, J. J., E. C. Holliman, and J. McDaniel (1992). "SWITCHBOARD Telephone Speech Corpus for Research and Development". In: *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-92)*. Vol. 1, pp. 517–520. ISBN: 1520-6149 VO - 1. DOI: 10.1109/ICASSP.1992.225858.

Goodwin, C. (1981). *Conversational Organization: Interaction Between Speakers and Hearers*. Academic Press. ISBN: 9780122897801. DOI: 10.1007/BF01871829.

Grau, S., E. Sanchis, M. Castro, and D. Vilar (2004). "Dialogue Act Classification Using a Bayesian Approach". In: *9th Conference Speech and Computer*. St. Petersberg, Russia.

Green, J., M. Franquiz, and C. Dixon (1997). "The Myth of the Objective Transcript: Transcribing as a Situated Act". In: *TESOL Quarterly* 31.1, p. 172. ISSN: 00398322. DOI: 10.2307/3587984.

Grice, H. P. (1975). "Logic and Conversation". In: *Syntax and Semantics* 3, pp. 45–47.

Griol, D., Z. Callejas, R. López-Cózar, and G. Riccardi (2014). "A Domain-independent Statistical Methodology for Dialog Management in Spoken Dialog Systems". In: *Computer Speech Language* 28, pp. 743–768. DOI: 10.1016/j.csl.2013.09.002.

Griol, D., L. Hurtado, E. Segarra, and E. Sanchis (2008). "A Statistical Approach to Spoken Dialog Systems Design and Evaluation". In: *Speech Communication* 50.8-9, pp. 666–682. ISSN: 01676393. DOI: 10.1016/j.specom.2008.04.001.

Grosz, B. J. (2018). "Smart Enough to Talk With Us? Foundations and Challenges for Dialogue Capable AI Systems". In: *Computational Linguistics* 44.1. DOI: 10.1162/COLI.

He, Z., L. Tavabi, K. Lerman, and M. Soleymani (2021). "Speaker Turn Modeling for Dialogue Act Classification". In: *Findings ofthe Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, pp. 2150–2157. arXiv: 2109.05056.

Hearst, M. A. (1997). "TextTiling: Segmenting Text Into Multi-paragraph Subtopic Passages". In: *Computational Linguistics* 23.1, pp. 33–64. ISSN: 0891-2017.

Henderson, M., I. Casanueva, N. Mrkšić, P. H. Su, T. H. Wen, and I. Vulić (2020). "ConveRT: Efficient and Accurate Conversational Representations from Transformers". In: *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, pp. 2161–2174. DOI: 10.18653/v1/2020.findings-emnlp.196. arXiv: 1911.03688.

Henderson, M., B. Thomson, and J. D. Williams (2014). "The Second Dialog State Tracking Challenge". In: *Proceedings of the SIGDIAL 2014 Conference*. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 263–272. ISBN: 9781479971299. DOI: 10.1109/SLT.2014.7078595. arXiv: 1702.06199.

Hochreiter, S. and J. Schmidhuber (1997a). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Hochreiter, S. and J. Schmidhuber (1997b). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Hoxha, J., P. Chandar, Z. He, J. Cimino, D. Hanhauer, and C. Weng (2016). "DREAM: Classification Scheme for Dialog Acts in Clinical Research Query Mediation". In: *Journal of Biomedical Informatics* 59, pp. 89–101. DOI: 10.1016/j.jbi.2015.11.011..

Hsu, L. M. and R. Field (2003). *Interrater Agreement Measures: Comments on Kappa n , Cohen's Kappa, Scott's π, and Aickin's α*. DOI: 10.1207/s15328031us0203_03.

Iseki, Y. (2019). "Characteristics of Everyday Conversation Derived from the Analysis of Dialog Act Annotation". In: *2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*. IEEE, pp. 1–6. ISBN: 9781728124490.

Jamison, E. and I. Gurevych (2014). "Adjacency Pair Recognition in Wikipedia Discussions using Lexical Pairs". In: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing*, pp. 479–488. ISBN: 9786165518871.

Janin, A., J. Ang, S. Bhagat, R. Dhillon, J. Edwards, J. Mac, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede (2004). "The ICSI Meeting Project: Resources and Research". In: *Proceedings of the 2004 ICASSP NIST Meeting Recognition Workshop*.

Jeh, G. and J. Widom (2002). "SimRank: A Measure of Structural-context Similarity". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 538–543.

Jeong, M., C. Y. Lin, and G. G. Lee (2009). "Semi-supervised Speech Act Recognition in Emails and Forums". In: *EMNLP 2009 - Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: A Meeting of SIGDAT, a Special Interest Group of ACL, Held in Conjunction with ACL-IJCNLP 2009* August, pp. 1250–1259. DOI: 10.3115/1699648.1699671.

Ji, Y., G. Haffari, and J. Eisenstein (2016). "A Latent Variable Recurrent Neural Network for Discourse Relation Language Models". In: *NAACL-HLT 2016*. San Diego, California: Association for Computational Linguistics, pp. 332–342. ISBN: 9781941643914. arXiv: 1603.01913.

Joulin, A., E. Grave, P. Bojanowski, and T. Mikolov (2017). "Bag of Tricks for Efficient Text Classification". In: *the Association for Computational Linguistics*. Vol. 2. Valencia, Spain: ACL, pp. 427–431.

Julia, F. N. and K. M. Iftekharuddin (2008). "Dialog Act Classification Using Acoustic and Discourse Information of Maptask Data". In: *2008 International Joint Conference on Neural Networks (IJCNN 2008)*. Hong Kong, China: IEEE, pp. 1472–1479. DOI: 10.1109/IJCNN.2008.4633991.

Jurafsky, D. and J. H. Martin (2017). *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd.

Jurafsky, D., E. Shriberg, and D. Biasca (1997). *Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual*. Tech. rep., pp. 97–102.

Kalchbrenner, N. and P. Blunsom (2013). "Recurrent Convolutional Neural Networks for Discourse Compositionality". In: *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 119–126. ISBN: 9781937284671. DOI: 10.1109/ICCV.2015.221. arXiv: 1306.3584.

Kalchbrenner, N., E. Grefenstette, and P. Blunsom (2014). "A Convolutional Neural Network for Modelling Sentences". In: *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*. Vol. 1. Baltimore, Maryland: Association for Computational Linguistics, pp. 655–665. ISBN: 9781937284725. DOI: 10.3115/v1/p14-1062. arXiv: 1404.2188.

Kazai, G. (2011). "In Search of Quality in Crowdsourcing for Search Engine Evaluation". In: *Proceedings ofthe 33rd European Conference on Information Retrieval (ECIR)*. Vol. 6611 LNCS. Berlin, Heidelberg, pp. 165–176. ISBN: 9783642201608. DOI: 10.1007/978-3-642-20161-5_17.

Keizer, S. (2001). "A Bayesian Approach to Dialogue Act Classification". In: *BI-DIALOG 2001: Proc. of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue*, pp. 210–218.

Keizer, S. and V. Rieser (2017). "Towards Learning Transferable Conversational Skills using Multi-dimensional Dialogue Modelling". In: *SEMDIAL 2017*. Saarbrücken, Germany.

Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Quatar: Association for Computational Linguistics, pp. 1746–1751. ISBN: 9781937284961. DOI: 10.3115/v1/D14-1181. arXiv: 1408.5882.

Král, P., T. Pavelka, and C. Cerisara (2008). "Evaluation of Dialogue Act Recognition Approaches". In: *Proceedings of the 2008 IEEE Workshop on Machine Learning for Signal Processing, MLSP 2008* November 2008, pp. 492–497. DOI: 10.1109/MLSP.2008.4685529.

Krause, B., L. Lu, I. Murray, and S. Renals (2016a). "Multiplicative LSTM for Sequence Modelling". In: *ICLR 2017*, pp. 1–11. arXiv: 1609.07959.

Krause, B., L. Lu, I. Murray, and S. Renals (2016b). “Multiplicative LSTM for Sequence Modelling”. In: *ICLR 2017*, pp. 1–11. arXiv: 1609.07959.

Krippendorff, K. (2004). *Content Analysis: An Introduction to its Methodology*. 2nd. Sage Publications. ISBN: 0761915443. DOI: 10.1103/PhysRevB.31.3460.

Kühnlein, P. and P. Piwek (2007). “Dialogue Modelling and Generation”. In: *Discourse Processes* 44.3, pp. 141–144. ISSN: 0163853X. DOI: 10.1080/01638530701600862.

Kumar, H., A. Agarwal, R. Dasgupta, S. Joshi, and A. Kumar (2017). “Dialogue Act Sequence Labeling using Hierarchical Encoder with CRF”. In: *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. AAAI, pp. 3440–3447. ISBN: 0631179623. arXiv: 1709.04250.

Kumar, V., R. Sridhar, S. Narayanan, and S. Bangalore (2008). “Enriching Spoken Language Translation with Dialog Acts”. In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Short Papers - HLT ’08* June, p. 225. DOI: 10.3115/1557690.1557755.

Lai, S., L. Xu, K. Liu, and J. Zhao (2015). “Recurrent Convolutional Neural Networks for Text Classification”. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI’15)*. AAAI, pp. 2267–2273. ISBN: 9781577357018. DOI: 10.1145/2808719.2808746. arXiv: 15334406.

Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut (2019). “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *ICLR 2020*. arXiv: 1909.11942.

Landis, J. R. and G. G. Koch (1977). “The Measurement of Observer Agreement for Categorical Data”. In: *Biometrics* 33.1, pp. 159–174. ISSN: 0006341X. DOI: 10.2307/2529310.

Latora, V. and M. Marchiori (2001). “Efficient Behavior of Small-world Networks”. In: *Physical Review Letters* 87.19. ISSN: 10797114. DOI: 10.1103/PhysRevLett.87.198701. arXiv: 0101396 [cond-mat].

Lee, F.-T., D. Hull, J. Levine, B. Ray, and K. McKeown (2019). “Identifying Therapist Conversational Actions Across Diverse Psychotherapeutic Approaches”. In: *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pp. 12–23. DOI: 10.18653/v1/w19-3002.

Lee, J. Y. and F. Dernoncourt (2016). “Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks”. In: *NAACL 2016*. ISBN: 9781941643914. arXiv: 1603.03827.

Levy, O. and Y. Goldberg (2014). “Dependency-Based Word Embeddings”. In: *Proceedings ofthe 52nd Annual Meeting ofthe Association for Computational Linguistics*. Baltimore, Maryland: ACL, pp. 302–308.

Li, R., C. Lin, M. Collinson, X. Li, and G. Chen (2019a). “A Dual-Attention Hierarchical Recurrent Neural Network for Dialogue Act Classification”. In: *Proceedings ofthe 23rd Conference on Computational Natural Language Learning*. Hong Kong, China: Association for Computational Linguistics, pp. 383–392. arXiv: 1810.09154.

Li, R., C. Lin, M. Collinson, X. Li, and G. Chen (2019b). “A Dual-attention Hierarchical Recurrent Neural Network for Dialogue Act Classification”. In: *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*, pp. 383–392. ISBN: 9781950737727. DOI: 10.18653/v1/k19-1036. arXiv: 1810.09154.

Li, W. and Y. Wu (2016). “Multi-level Gated Recurrent Neural Network for Dialog Act Classification”. In: *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*, pp. 1970–1979. arXiv: 1910.01822.

Li, X., Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz (2017). “End-to-End Task-Completion Neural Dialogue Systems”. In: *Proceedings of the The 8th International Joint Conference on Natural Language Processing*. Taipei, Taiwan: AFNLP, pp. 733–743. arXiv: 1703.01008.

Li, Z., J. Zhang, Z. Fei, Y. Feng, and J. Zhou (2021). “Conversations Are Not Flat: Modeling the Dynamic Information Flow Across Dialogue Utterances”. In: *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. Association for Computational Linguistics, pp. 128–138. ISBN: 9781954085527. DOI: 10.18653/v1/2021.acl-long.11. arXiv: 2106.02227.

Liddicoat, A. J. (2007). *An Introduction to Conversation Analysis*. London: Continuum, p. 319. ISBN: 0826491146.

Lin, C. Y. (2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *In Proceedings of the Workshop on Text Summarization Branches Out*. Barcelona, Spain: ACL, pp. 56–60. DOI: 10.1253/jcj.34.1213.

Liu, B. and I. Lane (2017). “Dialog Context Language Modeling with Recurrent Neural Networks”. In: *Acoustics, Speech and Signal Processing (ICASSP)*. 2. New Orleans, Louisiana: IEEE, pp. 5715–5719. ISBN: 9781509041176. DOI: 10.1109/ICASSP.2017.7953251.

Liu, B., G. Tur, D. Hakkani-Tur, P. Shah, and L. Heck (2018). “Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems”. In: *Proceedings of NAACL-HLT 2018*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2060–2069. arXiv: 1804.06512.

Liu, C.-W., R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau (2016). “How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation”. In: *Proceedings ofthe 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, TX: ACL, pp. 2122–2132. ISBN: 978-1-945626-25-8. arXiv: 1603.08023.

Liu, Y., K. Han, Z. Tan, and Y. Lei (2017). “Using Context Information for Dialog Act Classification in DNN Framework”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2160–2168.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, and P. G. Allen (2019). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv*. arXiv: 1907.11692.

Louwerse, M. and S. Crossley (2006). “Dialog Act Classification Using N-Gram Algorithms”. In: *FLAIRS Conference 2006*. Melbourne Beach, Australia, pp. 758–763.

Luan, Y., Y. Ji, and M. Ostendorf (2016). “LSTM based Conversation Models”. In: *arXiv*. arXiv: 1603.09457.

Luff, P., N. Gilbert, and D. Frolich, eds. (1990). *Computers and Conversation*. 1st ed. London: Academic Press, p. 284. ISBN: 9780124595606.

Luong, M. T., H. Pham, and C. D. Manning (2015). “Effective Approaches to Attention-based Neural Machine Translation”. In: *EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1412–1421. ISBN: 9781941643327. DOI: 10.18653/v1/d15-1166. arXiv: 1508.04025.

Macagno, F. and S. Bigi (2018). “Types of Dialogue and Pragmatic Ambiguity”. In: *Argumentation and Language-Linguistic, Cognitive and Discursive Explorations*. Vol. 32. March. Springer, pp. 191–218. ISBN: 9783319739724. DOI: 10.1007/978-3-319-73972-4_9.

Maitreyee, M. (2020). “Beyond Adjacency Pairs: Hierarchical Clustering of Long Sequences for Human-Machine Dialogues”. In: *Proceedings of the First Workshop on Computational Approaches to Discourse*. Association for Computational Linguistics, pp. 11–19. DOI: 10.18653/v1/2020.codi-1.2.

Malhotra, G., A. Waheed, A. Srivastava, M. S. Akhtar, and T. Chakraborty (2021). “Speaker and Time-aware Joint Contextual Learning for Dialogue-act Classification in Counselling Conversations”. In: *arXiv*. arXiv: 2111.06647.

Matějů, L., D. Griol, Z. Callejas, J. M. Molina, and A. Sanchis (2021). “An Empirical Assessment of Deep Learning Approaches to Task-oriented Dialog Management”. In: *Neurocomputing* 439, pp. 327–339. ISSN: 18728286. DOI: 10.1016/j.neucom.2020.01.126.

Mccowan, I., J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner (2005). “The AMI Meeting Corpus”. In: *Int'l. Conf. on Methods and Techniques in Behavioral Research.*

McTear, M., Z. Callejas, and D. Griol (2016). *The Conversational Interface - Talking to Smart Devices*. Springer Publishing Company, Incorporated. ISBN: 978-3-319-32965-9. DOI: 10.1007/978-3-319-32967-3_7.

Meredith, J. (2020). “Conversation Analysis, Cyberpsychology and Online Interaction”. In: *Social and Personality Psychology Compass*, pp. 1–10. ISSN: 17519004. DOI: 10.1111/spc3.12529.

Mezza, S., A. Cervone, G. Tortoreto, E. A. Stepanov, and G. Riccardi (2018). “ISO-Standard Domain-Independent Dialogue Act Tagging for Conversational Agents”. In: *COLING 2018*. June. Santa Fe, New Mexico, pp. 3539–3551. arXiv: 1806.04327.

Midgley, T. D., S. Harrison, and C. Macnish (2006). “Empirical Verification of Adjacency Pairs Using Dialogue Segmentation”. In: *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. Sydney, Australia: Association for Computational Linguistics, pp. 104–108.

Mikolov, T., G. Corrado, K. Chen, and J. Dean (2013). “Efficient Estimation of Word Representations in Vector Space”. In: *International Conference on Learning Representations (ICLR 2013)*. ISBN: 1532-4435. DOI: 10.1162/153244303322533223. arXiv: arXiv:1301.3781v3.

Milajevs, D. and M. Purver (2014). “Investigating the Contribution of Distributional Semantic Information for Dialogue Act Classification”. In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. May. Gothenburg: Association for Computational Linguistics, pp. 40–47. DOI: 10.3115/v1/W14-1505.

Miltenburg, E. van, M. Clinciu, O. Dušek, D. Gkatzia, S. Inglis, L. Leppänen, S. Mahamood, E. Manning, S. Schoch, C. Thomson, and L. Wen (2021). “Underreporting of Errors in NLG Output, and What To Do About It”. In: *INLG 2021 - 14th International Conference on Natural Language Generation, Proceedings* September, pp. 140–153. arXiv: 2108.01182.

Mitchell, J. and M. Lapata (2008). “Vector-based Models of Semantic Composition”. In: *Proceedings of ACL-08: HLT*. June. Association for Computational Linguistics, pp. 236–244. ISBN: 9781932432046.

Morelli, R. A., J. D. Brozino, and J. W. Goethe (1991). “A Computational Speech-Act Model of Human Computer Conversations”. In: *Proceedings of the 1991 IEEE Seventeenth Annual Northeast Bioengineering Conference*. IEEE, pp. 148–162. DOI: 10.1109/NEBC.1991.154675.

Motozawa, M., Y. Murakami, M. Pituxcoosuvarn, T. Takasaki, and Y. Mori (2021). “Conversation Analysis for Facilitation in Children's Intercultural Collaboration”. In: *IDC*

'21: Interaction Design and Children. ACM, pp. 62–68. ISBN: 9781450384520. DOI: 10.1145/3459990.3460721.

Norman, M. and P. Thomas (1990). "The Very Idea: Informing HCI Design from Conversation Analysis". In: *Computers and Conversation*. Ed. by P. Luff, N. Gilbert, and D. Frohlich. London Academic Press. Chap. 3, pp. 51–65.

Norrick, N. (2004). *Saarbrücken Corpus of Spoken English (SCoSE)*. DOI: 10.21415/T5DC77.

Novielli, N. and C. Strapparava (2009). "Towards Unsupervised Recognition of Dialogue Acts". In: *Proceedings of the NAACL-HLT Student Research Workshop and Doctoral Consortium*. Boulder, Colorado: Association for Computational Linguistics, pp. 84–89.

Nowak, S. and S. Rüger (2010). "How Reliable are Annotations via Crowdsourcing? - A Study about Inter-annotator Agreement for Multi-label Image Annotation". In: *MIR '10 Proceedings of the international conference on Multimedia information retrieval*. Philadelphia, Pennsylvania: Association for Computing Machinery, p. 557. ISBN: 9781605588155. DOI: 10.1145/1743384.1743478.

Ortega, D., C.-Y. Li, G. Vallejo, D. Pavel, and N. T. Vu (2019). "Context-aware Neural-based Dialogue Act Classification on Automatically Generated Transcriptions". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, pp. 7265–7269. ISBN: 9781538646588. DOI: 0.1109/ICASSP.2019.8682881.

Ortega, D. and N. T. Vu (2017). "Neural-based Context Representation Learning for Dialog Act Classification". In: *SIGDIAL 2017*. Saarbrücken, Germany: Association for Computational Linguistics, pp. 247–252. arXiv: 1708.02561.

Oyama, S., Y. Baba, Y. Sakurai, and H. Kashima (2013). "Accurate Integration of Crowdsourced Labels using Workers' Self-reported Confidence Scores". In: *IJCAI International Joint Conference on Artificial Intelligence*. Beijing, China: AAAI Press, pp. 2554–2560. ISBN: 9781577356332.

Papalampidi, P., E. Iosif, and A. Potamianos (2017). "Dialogue Act Semantic Representation and Classification Using Recurrent Neural Networks". In: *SEMDIAL 2017 (SaarDial) Workshop on the Semantics and Pragmatics of Dialogue*. August, pp. 77–86. DOI: 10.21437/SemDial.2017-9.

Papineni, K., S. Roukos, T. Ward, W. Zhu, and Z. Wei-Jing (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. July. Philadelphia, PA, pp. 311–318. ISBN: 1-55860-883-4. DOI: 10.3917/chev.030.0107. arXiv: 1702.00764.

Pareti, S. and T. Lando (2019). "Dialog Intent Structure: A Hierarchical Schema of Linked Dialog Acts". In: *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pp. 2907–2914.

Patten, T., M. L. Geis, and B. D. Becker (1992). "Toward a Theory of Compilation for Natural Language Generation". In: *Computational Intelligence* 8.1, pp. 77–101. ISSN: 00978507. DOI: 10.2307/414915.

Pennington, J., R. Socher, and C. D. Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. ISBN: 9781937284961. DOI: 10.3115/v1/D14-1162. arXiv: 1504.06654.

Peters, M. E., M. Neumann, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). "Deep Contextualized Word Representations". In: *NAACL 2018*. arXiv: arXiv:1802.05365v2.

Petukhova, V. (2011). "Multidimensional Dialogue Modelling". PhD thesis. Tilburg University, pp. 1–234. ISBN: 9789491211881.

Petukhova, V., A. Malchanau, and H. Bunt (2014). "Interoperability of Dialogue Corpora through ISO 24617-2-based Querying". In: *Lrec 2014*, pp. 4407–4414. ISBN: 978-2-9517408-8-4.

Pilnick, A., D. Trusson, S. Beeke, R. O'Brien, S. Goldberg, and R. H. Harwood (2018). "Using Conversation Analysis to Inform Role Play and Simulated Interaction in Communications Skills Training for Healthcare Professionals: Identifying Avenues for Further Development Through a Scoping Review". In: *BMC Medical Education* 18.1, pp. 1–11. ISSN: 14726920. DOI: 10.1186/s12909-018-1381-1.

Poesio, M. and D. Traum (1998). "Towards an Axiomatization of Dialogue Acts". In: *Proceedings of the Twentieth Workshop on the Formal Semantics and Pragmatics of Dialogues*. March 2013, pp. 207–222.

Poesio, M. and R. Vieira (1998). "A Corpus-based Investigation of Definite Description Use". In: *Computational Linguistics* 24.2, pp. 183–216. ISSN: 08912017. arXiv: 9710007 [cmp-lg].

Psathas, G. (1995). *Conversation Analysis - The Study of Talk-in-Interaction*. Sage Publications. ISBN: 0803957467. DOI: 10.1017/cbo9780511519888.004.

Qiu, L., Y. Zhao, W. Shi, Y. Liang, F. Shi, T. Yuan, Z. Yu, and S. C. Zhu (2020). "Structured Attention for Unsupervised Dialogue Structure Induction". In: *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. Association for Computational Linguistics, pp. 1889–1899. ISBN: 9781952148606. DOI: 10.18653/v1/2020.emnlp-main.148. arXiv: 2009.08552.

Quarteroni, S., A. V. Ivanov, and G. Riccardi (2011). "Simultaneous Dialog Act Segmentation and Classification from Human-human Spoken Conversations". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. May 2014, pp. 5596–5599. ISBN: 9781457705397. DOI: 10.1109/ICASSP.2011.5947628.

Radford, A., R. Jozefowicz, and I. Sutskever (2017). "Learning to Generate Reviews and Discovering Sentiment". In: *arXiv*. arXiv: 1704.01444.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In: *arXiv*.

Raheja, V. and J. Tetreault (2019). "Dialogue Act Classification with Context-Aware Self-Attention". In: *NAACL-HLT 2019*. ASCD, pp. 3727–3733. arXiv: 1904.02594v2.

Reithinger, N., R. Engel, M. Kipp, and M. Klesen (1996). "Predicting Dialogue Acts for a Speech-to-speech Translation System". In: *International Conference on Spoken Language Processing, ICSLP, Proceedings*. Vol. 2, pp. 654–657. DOI: 10.1109/icslp.1996.607446.

Reithinger, N. and M. Klesen (1997). "Dialogue Act Classification Using Language Models". In: *EuroSpeech*.

Ribeiro, E., R. Ribeiro, and D. M. De Matos (2015). "The Influence of Context on Dialog Act Recognition". In: *arXiv*.

Ribeiro, E., R. Ribeiro, and D. M. De Matos (2018). "Deep Dialog Act Recognition using Multiple Token, Segment, and Context Information Representations". In: *arXiv*. DOI: arXiv:1807.08587v1. arXiv: 1807.08587.

Ribeiro, E., R. Ribeiro, and D. M. De Matos (2019). "Deep Dialog Act Recognition Using Multiple Token, Segment, and Context Information Representations". In: *Journal of Artificial Intelligence Research* 66, pp. 861–899. ISSN: 10769757. DOI: 10.1613/jair.1.11594. arXiv: 1807.08587.

Ries, K. (1999). "HMM and Neural Network Based Speech Act Detection". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99*. Phoenix, Arizona, 497–500 vol.1. ISBN: 0-7803-5041-3. DOI: 10.1109/ICASSP.1999.758171.

Robinson, H. (1990). "Towards a Sociology of Human-Computer Interaction: A Software Engineer's Perspective". In: *Computers and Conversation*. Ed. by P. Luff, N. Gilbert, and D. Frohlich. London Academic Press. Chap. 2, pp. 39–49.

Rojas-Barahona, L. M., M. Gasic, N. Mrkšić, P.-H. Su, S. Ultes, T.-H. Wen, and S. Young (2016). "Exploiting Sentence and Context Representations in Deep Neural Models for Spoken Language Understanding". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*. Osaka, Japan, pp. 258–267. arXiv: 1610.04120.

Rus, V. and M. Lintean (2012). "A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics Vasile". In: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Montreal, Canada: Association for Computational Linguistics, pp. 157–162.

Sacks, H. (1995). *Lectures on Conversation, Volume I, II*. Vol. I. Blackwell, pp. 2–580. ISBN: 9781557867056.

Sacks, H., E. A. Schegloff, and G. Jefferson (1974). "A Simplest Systematics for the Organization of Turn-Taking for Conversation". In: *Language* 50.1, pp. 696–735.

Salzberg, S. L. (1997). "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach". In: *Data Mining and Knowledge Discovery* 1.3, pp. 317–328. ISSN: 13845810. DOI: 10.1023/A:1009752403260.

Schegloff, E. A. (1968). "Sequencing in Conversational Openings". In: *American Anthropologist* 70.6, pp. 1075–1095. ISSN: 0002-7294. DOI: 10.1525/aa.1968.70.6.02a00030.

Schegloff, E. A. (1972). "Notes on a Conversational Practice : Formulating Place". In: *Studies in Social Interaction*. Macmillan Publishers Limited, pp. 79–119.

Schegloff, E. A. (1979). "Identification and Recognition in Telephone Conversation Openings". In: *Everyday Language: Studies in Ethnomethodology*. New York: Irvington.

Schegloff, E. A. (2007). *Sequence Organization in Interaction: A Primer in Conversation Analysis I*. Cambridge: Cambridge University Press. ISBN: 978-0-521-82572-6. DOI: 10.1017/CBO9780511791208.

Schegloff, E. A., G. Jefferson, and H. Sacks (1977). "The Preference for Self-Correction in the Organization of Repair in Conversation". In: *Language* 53.2, p. 361. ISSN: 00978507. DOI: 10.2307/413107.

Schegloff, E. A. and H. Sacks (1973). "Opening Up Closings.pdf". In: *Semiotica* 7, pp. 289–327.

Schlangen, D. (2005). "Modelling Dialogue: Challenges and Approaches". In: *Künstliche Intelligenz* 3.5, pp. 23–28.

Schutze, H., D. A. Hull, and J. O. Pedersen (1995). "A Comparison of Classifiers and Document Representations for the Routing Problem". In: *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 229–237. DOI: https://doi.org/10.1145/215206.215365.

Scott, W. A. . (1955). "Reliability of Content Analysis : The Case of Nominal Scale Coding". In: *The Public Opinion Quarterly* 19.3, pp. 321–325.

Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. London: Cambridge University Press, p. 203. ISBN: 9780521096263.

Searle, J. R. (1979). *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge University Press, p. 187. ISBN: 9780521229012.

Searle, J. R. (1992). "Conversation". In: *(On) Searle on Conversation*. Ed. by J. L. Mey, H. Parret, and J. Verschueren. John Benjamins Publishing Company. Chap. 1, pp. 7–29.

Serafin, R. and B. Di Eugenio (2004). "FLSA: Extending Latent Semantic Analysis with Features for Dialogue Act Classification". In: *Proceedings of the 42nd Annual Meeting* *of the Association for Computational Linguistics*, pp. 692–699. DOI: 10.3115/1218955.1219043.

Serafin, R., B. D. Eugenio, and M. Glass (2003). "Latent Semantic Analysis for Dialogue Act Classification". In: *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*. DOI: 10.3115/1073483.1073515.

Shang, G., A. J. Tixier, M. Vazirgiannis, and J. P. Lorré (2020). "Speaker-change Aware CRF for Dialogue Act Classification". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Online, pp. 450–464. DOI: 10.18653/v1/2020.coling-main.40. arXiv: 2004.02913.

Sharma, S., L. E. Asri, H. Schulz, and J. Zumer (2017). "Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation". In: *arXiv*. arXiv: 1706.09799.

Shen, S. S. and H. Y. Lee (2016). "Neural Attention Models for Sequence Classification: Analysis and Application to Key Term Extraction and Dialogue Act Detection". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Vol. 08-12-Sept. San Francisco, California, pp. 2716–2720. DOI: 10.21437/Interspeech.2016-1359. arXiv: 1604.00077.

Shi, W., T. Zhao, and Z. Yu (2019). "Unsupervised Dialog Structure Learning". In: *Proceedings of NAACL-HLT 2019*. Association for Computational Linguistics, pp. 1797–1807.

Shriberg, E., R. Dhillon, S. Bhagat, J. Ang, H. Carvey, and C. S. U. Hayward (2004). "The ICSI Meeting Recorder Dialog Act (MRDA) Corpus". In: *SIGDIAL 2004*, pp. 97–100.

Shriberg, E., A. Stolcke, D. Jurafsky, N. Coccaro, M. Meteer, R. Bates, P. Taylor, K. Ries, R. Martin, and C. Van Ess-Dykema (1998). "Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech?" In: *Language and Speech* 41.3-4, pp. 443–492. ISSN: 00238309. DOI: 10.1177/002383099804100410. arXiv: 0006024 [cs].

Shum, H.-Y., X.-D. He, and D. Li (2018). "From Eliza to XiaoIce: Challenges and Opportunities with Social Chatbots". In: *Frontiers of Information Technology & Electronic Engineering* 19.1, p. 10. DOI: 10.1631/FITEE.1700826.

Sidnell, J. (2010). *Conversation Analysis - An Introduction*. Whiley-Blackwell. ISBN: 978-1-4051-5900-5. DOI: 10.1093/acrefore/9780199384655.013.40.

Sidnell, J. and T. Stivers, eds. (2013). *The Handbook of Conversation Analysis*. Whiley-Blackwell, p. 447. ISBN: 9781444332087. DOI: 10.1002/9781118325001.ch1.

Sinclair, J. and M. Coulthard (1974). *Towards an Analysis of Discourse: English Used by Teachers and Pupils*. Oxford University Press.

Smith, B. (1990). "Towards a History of Speech Act Theory". In: *Speech Acts, Meaning and Intentions Critical Approaches to the Philosophy of John R. Searle*. Ed. by A. Burkhardt. De Gruyter, pp. 29–62. DOI: 10.1075/vh.2.12smi.

Snow, R., B. O. Connor, D. Jurafsky, A. Y. Ng, D. Labs, and C. St (2008). "Cheap and Fast - But is it Good ? Evaluating Non-Expert Annotations for Natural Language Tasks". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. October. Honolulu: Association for Computational Linguistics, pp. 254–263. ISBN: 9781450329224. DOI: 10.1.1.142.8286. arXiv: 1601.06610.

Son, Y. and H. A. Schwartz (2021). "Discourse Relation Embeddings: Representing the Relations between Discourse Segments in Social Media". In: *arXiv*. arXiv: 2105.01306.

Speer, R., J. Chin, and C. Havasi (2016). "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) ConceptNet*, pp. 4444–4451. arXiv: 1612.03975.

Sridhar, V. K. R., S. Narayanan, and S. Bangalore (2009). "Incorporating Discourse Context in Spoken Language Translation Through Dialog Acts". In: *2008 IEEE Workshop* *on Spoken Language Technology, SLT 2008 - Proceedings* January, pp. 269–272. DOI: 10.1109/SLT.2008.4777892.

Stolcke, A., K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer (2000). "Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech". In: *Computational Linguistics* 26.3, pp. 339–373. ISSN: 0891-2017. DOI: 10.1162/089120100561737. arXiv: 0006023 [cs].

Surendran, D. and G.-A. Levow (2006). "Dialog Act Tagging with Support Vector Machines and Hidden Markov Models". In: *Interspeech 2006 and 9th International Conference on Spoken Language Processing*. Pittsburgh, pp. 1950–1953. ISBN: 9781604234497.

Sutskever, I., O. Vinyals, and Q. V. Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in Neural Information Processing Systems*. Vol. 4. January, pp. 3104–3112. ISBN: 1409.3215. DOI: 10.1007/s10107-014-0839-0. arXiv: 1409.3215.

Tavabi, L., T. Tran, K. Stefanov, B. Borsari, J. Woolley, S. Scherer, and M. Soleymani (2021). "Analysis of Behavior Classification in Motivational Interviewing". In: *Proceedings ofthe Seventh Workshop on Computational Linguistics and Clinical Psychology*. Association for Computational Linguistics, pp. 110–115. DOI: 10.18653/v1/2021.clpsych-1.13.

Tewari, M. and B. Suna (2018). "Natural Language Communication with Social Robots for Assisted Living". In: *International Conference on Intelligent Robots and Systems (IROS 2018)*, pp. 1–4.

Thompson, H. S., A. Anderson, E. G. Bard, G. Doherty-Sneddon, A. Newlands, and C. Sotillo (1991). "The HCRC Map Task Corpus: Natural Dialogue for Speech Recognition". In: *Language and Speech* 34.4, pp. 25–30. DOI: http://dx.doi.org/10.3115/1075671.1075677.

Thornbury, S. and D. Slade (2006). *Conversation: From Description to Pedagogy*. Cambridge University Press, p. 377. ISBN: 9.78052E12.

Tran, Q. H., G. Haffari, and I. Zukerman (2017). "A Generative Attentional Neural Network Model for Dialogue Act Classification". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 524–529. ISBN: 9781945626760. DOI: 10.18653/v1/P17-2083.

Tran, Q. H., I. Zukerman, and G. Haffari (2017). "A Hierarchical Neural Model for Learning Sequences of Dialogue Acts". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Vol. 1. Association for Computational Linguistics, pp. 428–437. ISBN: 9781510838604.

Traum, D. R. (2000). "20 Questions on Dialogue Act Taxonomies". In: *Journal of Semantics* 17.1, pp. 7–30. ISSN: 14774593. DOI: 10.1093/jos/17.1.7.

Traum, D. R. and E. A. Hinkelman (1992). "Conversation Acts in Task-Oriented Spoken Dialogue". In: *Computational Intelligence* 8.3, pp. 575–599.

Turing, A. M. (1950). "Computing Machinery and Intelligence". In: *Computing Machinery and Intelligence. Mind* 49, pp. 433–460.

Vanderveken, D. and S. Kubo, eds. (2001). *Essays in Speech Act Theory*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). "Attention Is All You Need". In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA. ISBN: 9781577357384. DOI: 10.1017/S0140525X16001837. arXiv: 1706.03762.

Vinyals, O. and L. V. Quoc (2015). "A Neural Conversational Model". In: *ICML Deep Learning Workshop*. arXiv: arXiv:1506.05869v3.

Wan, Y., W. Yan, J. Gao, Z. Zhao, J. Wu, and P. S. Yu (2018). "Improved Dynamic Memory Network for Dialogue Act Classification with Adversarial Training". In: *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*. IEEE, pp. 841–850. ISBN: 9781538650356. DOI: 10.1109/BigData.2018.8622245.

Wang, L., M. Lui, S. Nam Kim, J. Nivre, and T. Baldwin (2011). "Predicting Thread Discourse Structure over Technical Web Forums". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 13–25.

Wang, Y., Q. Yao, J. T. Kwok, and L. M. Ni (2020). "Generalizing from a Few Examples: A Survey on Few-shot Learning". In: *ACM Computing Surveys* 53.3. ISSN: 15577341. DOI: 10.1145/3386252.

Warren, M. (2006). *Features of Naturalness in Conversation*. Ed. by H. A. Jucker. John Benjamins Publishing Company. ISBN: 90 272 5395 1.

Webb, N. and M. Hepple (2005). "Dialogue Act Classification Based on Intra-Utterance Features". In: *Proceedings of the AAAI Workshop on Spoken Language Understanding*.

Weigand, E. (1994). "Discourse, Conversation, Dialogue". In: *Concepts of Dialogue: Considered from the Perspective of Different Disciplines*. Ed. by E. Weigand. 6th ed. Walter de Gruyter GmbH. Chap. 4, pp. 46–76. ISBN: 9783484750067.

Weizenbaum, J. (1966). "ELIZA - A Computer Program for the Study of Natural Language Communication Between Man and Machine". In: *Communications of the ACM* 9.1. DOI: https://doi.org/10.1145/365153.365168.

Wen, T.-H., M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young (2016). "A Network-based End-to-End Trainable Task-oriented Dialogue System". In: *Proceedings of EACL 2017*. ISBN: 9781510838604. arXiv: 1604.04562.

Wen, T.-H., Y. Miao, P. Blunsom, and S. Young (2017). "Latent Intention Dialogue Models". In: *Proceedings of the 34th International Conference on Machine Learning*. Sydney, Australia. ISBN: 9781510855144. arXiv: 1705.10229.

Weston, J., A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov (2015). "Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks". In: *arXiv*. arXiv: 1502.05698.

Wiebe, J. M., R. F. Bruce, and T. P. O'Hara (1999). "Development and Use of a Gold-standard Data Set for Subjectivity Classifications". In: *ACL '99: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. College Park, Maryland: ACM, pp. 246–253. DOI: https://doi.org/10.3115/1034678.1034721.

Williams, J. D., A. Raux, and M. Henderson (2016). "The Dialog State Tracking Challenge Series: A Review". In: *Dialogue Discourse* 7.3, pp. 4–33. DOI: 10.5087/dad.2016.301.

Wittgenstein, L. (1953). *Philosophical Investigations*. 2nd. Basil Blackwell. ISBN: 0631119000.

Wooffitt, R. (1990). "On the Analysis of Interaction: An Introduction to Conversation Analysis". In: *Computers and Conversation*. Ed. by P. Luff, N. Gilbert, and D. Frolich. 1st ed. Academic Press. Chap. 1, pp. 7–38.

Wooffitt, R. (2005). *Conversation Analysis and Discourse Analysis: A Comparative and Critical Introduction*. London: Sage Publications, p. 245. ISBN: 076197425 3.

Yang, X., J. Liu, Z. Chen, and W. Wu (2015). "Semi-supervised Learning of Dialogue Acts Using Sentence Similarity Based on Word Embeddings". In: *ICALIP 2014 International Conference on Audio, Language and Image Processing, Proceedings*. Shanghai, China: IEEE, pp. 882–886. ISBN: 9781479939022. DOI: 10.1109/ICALIP.2014.7009921.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems*. Vol. 32. arXiv: 1906.08237.

Yeh, Y.-T., M. Eskenazi, and S. Mehri (2021). "A Comprehensive Assessment of Dialog Evaluation Metrics". In: *Proceedings ofthe First Workshop on Evaluations and Assessments ofNeural Conversation Systems*. ACL, pp. 15–33. DOI: 10.18653/v1/2021.eancs-1.3. arXiv: 2106.03706.

Żelasko, P., R. Pappagari, and N. Dehak (2021). "What Helps Transformers Recognize Conversational Structure? Importance of Context, Punctuation, and Labels in Dialog Act Recognition". In: *Transactions of the Association for Computational Linguistics* 9, pp. 1179–1195. ISSN: 2307387X. DOI: 10.1162/tacl_a_00420. arXiv: 2107.02294.

Zhai, K. and J. D. Williams (2014). "Discovering Latent Structure in Task-Oriented Dialogues". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 36–46. ISBN: 9781937284725.

Zhang, Y., Z. Ou, and Z. Yu (2020). "Task-oriented Dialog Systems that Consider Multiple Appropriate Responses Under the Same Context". In: *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*. AAAI, pp. 9604–9611. ISBN: 9781577358350. DOI: 10.1609/aaai.v34i05.6507. arXiv: 1911.10484.

Zhang, Y., S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan (2020a). "DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation". In: *Proceedings of the 58th Annual Meeting ofthe Association for Computational Linguistics*. Association for Computational Linguistics, pp. 270–278. DOI: 10.18653/v1/2020.acl-demos.30. arXiv: 1911.00536.

Zhang, Z., R. Takanobu, Q. Zhu, M. L. Huang, and X. Y. Zhu (2020b). "Recent Advances and Challenges in Task-oriented Dialog Systems". In: *Science China Technological Sciences* 63.10, pp. 2011–2027. ISSN: 1862281X. DOI: 10.1007/s11431-020-1692-3. arXiv: 2003.07490.

Zhao, T. and M. Eskenazi (2018). "Zero-shot dialog generation with cross-domain latent actions". In: *SIGDIAL 2018 - 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue - Proceedings of the Conference*. Melbourne, Australia: Association for Computational Linguistics, pp. 1–10. ISBN: 9781948087674. DOI: 10.18653/v1/w18-5001. arXiv: 1805.04803.

Zwick, R. (1988). "Another Look at Interrater Agreement". In: *Psychological Bulletin* 103.3, pp. 374–378. ISSN: 00332909. DOI: 10.1037/0033-2909.103.3.374.