Generating unambiguous URL clusters from Web search

G. Smith[1], T. Brailsford[2], C. Donner[1], D. Hooijmaijers[1], M. Truran[3], J. Goulding[2] and H. Ashman[1]

1 ACRC, Computer Science
University of South Australia
{gavin.smith | helen.ashman}@unisa.edu.au

2 Computer Science,
University of Nottingham
{tjb | jog}@cs.nott.ac.uk

3 Computer Science,
University of Teesside
m.a.truran@tees.ac.uk

ABSTRACT

This paper reports on the generation of unambiguous clusters of URLs from clickthrough data from the MSN search query log excerpt (the RFP 2006 dataset). Selections (clickthroughs) by a single user from a single query can be assumed to have some mutual semantic relevance, and the URLs coselected in this way can be aggregated to form single-sense clusters. When the graphs for a single term separate into distinct clusters, the semantics of the distinct clusters can be interpreted as disambiguated aggregations of URLs. This principle had been tested on smaller and more constrained datasets previously, and this paper reports on findings from applying a method based on the principle to the RFP 2006 dataset.

This paper evaluates the proposed coselection method for generating single-sense clusters against two other methods, with varying parameters. The evaluation is done both with a human evaluation to determine the quality of the clusters generated by the different methods, and by a simple "edit distance" analysis to determine the content difference of the methods.

The main questions addressed are i) whether it is feasible to generate single-sense / sense-coherent clusters, and ii) whether, in a closed world, it would be feasible to discover ambiguous terms. The experimentation showed that sense-coherent clusters were found and further indicated that ambiguous terms could be detected from observing small overlap between large clusters.

Categories and Subject Descriptors

H.3.1 [Content Analysis and Indexing] H.3.3 [Information Search and Retrieval]

General Terms

Algorithms, Experimentation

Keywords

disambiguation, Web search, clickthrough

1. INTRODUCTION

The majority of disambiguation mechanisms in use on the web rely on the manual identification of ambiguous words and their associated, differing, meanings. Such approaches are extremely effective, providing that sufficient human effort is available. In the case of large collaborative projects, such as Wikipedia or WordNet, with their vast numbers of contributors, this is not a problem. However, manual disambiguation of this sort necessarily focuses on a small amount of popular projects – explicit disambiguation on such a scale is far less feasible for both smaller, specialist or less fashionable purposes. Hence, any means of automating disambiguation implicitly would be an extremely useful adjunct to web-based information retrieval. This paper analyses just such an approach.

It is well established that click-through data provides potential classification information for Web resources, with a search term providing a possible classifier or label for selected resources from a corresponding search result page [2]. What does not seem to be exploited yet is the co-selection of results[1]. This is the principle that when a user selects more than one result from a set of search results, the selected resources are likely to be semantically singular singular (with co-selected resources being those that one user chooses within one session[2]). For example, a user looking for information about why New York is called the Big Apple would be less likely to select pages on Australia's tourist destination of the same name or the Perth-based Big Apple company. This is based upon the premise of co-active intelligence [9], which is a technique used to leverage human intellect for the purposes of separating the distinct meanings of ambiguous search terms by modelling mass consensus as measured by cluster analysis. The [9] study was a small-scale experiment in which users' were required to group images that were deliberately designed to be ambiguous. Although this demonstrated the feasibility of disambiguating search terms by using users' co-selections from search results pages, it was based upon an artificial set of searches undertaken by volunteers under artificial controls. We thus determined to test the use of this approach for disambiguation using live Web log data.

From March 2006 to the present, we have collected Web log data from a University School of Computer Science (the "Teesside data"). In early 2008 we began to extract click-through data from the raw logs finding around half a million queries and associated click-throughs. While numerous well-established clusters were identified, the coverage of topics was not broad and, due to the well-defined nature of student-search tasks, lacked ambiguity – university based web-log data is generated by a small number of highly goal-focused participants (students) searching for information to assist with common assignments. As such, this data proved to be of limited value for disambiguation research. Nonetheless, semantic relationships were found between clusters, such as synonyms and translations (for example, the

[1] What we call co-selection data is not quite the same as click-through data generally, as co-selection data includes information about whether the selected URLs were selected at the same time from the same search results, hence having the implied sense-similarity.

[2] We define a session as a short period of time where a single user searches for a single query.

Czechoslovakian "hrad Pernstejn" was automatically linked to the English "Castle Pernstejn") and IS-A relationships ("films King Kong" to "1933 King Kong"), based on overlapping between clusters.

In this paper we examine a far larger, unbiased dataset (the RFP 2006 data set provided by Microsoft). Prior to the experiment described in this paper, the data available from university web-logs had exhibited low coverage of those topics with enough data to reliably form clusters around. However, we are now able to extend previous work on query clustering and propose algorithms to enable the extraction of clusters of URLs that are semantically similar (i.e. single-sense clusters), and we demonstrate the use of the algorithm to provide a method for the automatic identification of potentially ambiguous phrases. Such ambiguous phrases may then be clarified for example by reference to the target documents, or may form the basis of a "suggestion algorithm" that alerts users to potentially ambiguous terms. We evaluate the semantic coherence of the discovered "unambiguous" clusters through human evaluation of clusters to ground-truth, comparing the method firstly to a traditional query-clustering algorithm that is unable separate word senses as a baseline, and secondly to a comparable proposal based on preserving session information in queries. Finally we briefly examine the ability of the method to identify potentially ambiguous terms using a list of terms mined from Wikipedia (as [8]), determining to what extent it can identify not just known ambiguities but others that have not yet arisen in the Wikipedia list, either because the ambiguity has not appeared in Wikipedia articles, or perhaps because the ambiguity is subtle.

2. RELATED WORK

The co-selection method discussed here is primarily aimed at disambiguation. There is however some similarity in aims and methods between disambiguation and query clustering, hence both are discussed here.

Disambiguation aims to separate out the distinct meanings a single query term may have. On the other hand, query clustering aims to work out when different query terms are seeking the same target information, for example to be able to deal with spelling errors or typographical errors. It can also be used to detect synonyms (of which it might be argued that spelling and typographical errors are a special case).

2.1 Disambiguation

Word sense disambiguation aims to determine the correct (distinct) meaning for a given word in a given context. Typically this task involves determining a set of candidate terms and then choosing the most appropriate one. Word sense disambiguation methods are generally categorized based into categories based on the major resource they use.

The most basic approaches are dictionary or knowledge-based methods that use a pre-created resource such as machine-readable dictionaries, thesauri or WordNet [5]. Unsupervised learning approaches take raw unannotated data (traditionally large corpora of text) as input. Finally supervised methods take annotated corpora as training input. The techniques used under these categories are vast in number, varied in their approaches and include many combinations. The most common approach has tended to be semi-supervised methods for which systems have proven to be quite effective in recent SemEval tasks [1].

However, although more compelling due to the removal of the training data requirement, completely unsupervised methods have not shown quite as much promise [5] [1]. In addition approaches such as [6], while unsupervised, still rely on a manually constructed resource such as WordNet. Related to unsupervised methods is the task of inducing possible senses automatically. In SemEval 2007 [1] this represented one of the tasks, with the best system obtaining a F-Score of 78.9 in an unsupervised evaluation and performing only 6.9 percentage points below the best supervised system when compared with a sample subtask from another SemEval 2007 task. Recent work by Pedersen and Kulkarni also follows this approach, looking at discovering identities in web contexts by examining the text snippets surrounding a set of specified names.

Using textual features (in this case shallow lexical features) they cluster the contexts in an unsupervised fashion [7]. The use of click-through data for disambiguation falls into this last category, but in contrast to these approaches the use of click-through data does not require complex analysis of text, rather relying on simple clustering and aggregated mass human judgments.

2.2 Query clustering

Query clustering based on click-through data was initially used as a means of discovering similar queries [2], by using bipartite graphs that consisted of distinct query nodes on one side and distinct URL nodes on the other. This technique consisted of an iterative agglomerative clustering algorithm that clustered both queries and URLs. The process was evaluated with regard to query suggestions using 500,000 click-through records from the Lycos search engine.

A similar approach to this was taken by Wen et al. [10] in the context of FAQ identification. This work was based upon click-through data and used a combination of keyword analysis and keyword clustering although the potential of co-selections for disambiguation was only considered peripherally. They proposed session-based query clustering whose aim is to cluster different query terms based on query terms submitted in the same session, and they propose a click-through-based method for achieving this. The evaluation was much smaller than that of [2], evaluating 20,000 queries over a document space of 41,942 documents from Encarta, which is not typical of a Web space, being editorially controlled with a far smaller range of topics – so it is by no means certain that users have the same search habits as do Web users. Even though they claim that disambiguation would be possible, they seem to have made no attempt to follow this up. It may be that they perceived the level of ambiguity not to be high enough to justify the effort, as they noted that "the ambiguity of keywords [query terms] will only bring in about 4% errors. This number is much lower than our expectation" and that "users usually are aware of word ambiguity and would like to use more precise queries". This 4% is consistent with Sanderson's more recent analysis [8], although he notes in a footnote that the level of ambiguity may turn out to be higher as the reference material used, the Wikipedia disambiguation page. However, even 4% of ambiguous queries amounts to a large quantity in a general search engine, and 4% of the 15 million queries in the one month's data from RFP 2006 would still be around 600,000 ambiguous queries represented. In any case, it appears that the authors did not pursue the possibility of disambiguating based on co-selections any further. The work in this paper follows this up.

It appears that the session-based query clustering method can create disambiguated clusters in a quite similar way as the co-selection method, with the primary distinctions being the form of input into the chosen clustering algorithm. There is a difference in the way the data is represented however, and the individual co-selection data is preserved longer in the Wen et al. method than in

the coselection method. Since Wen et al. preserve the specific session ID of each clickthrough, the input of the clustering method is *(query concatenated with sessionID, url)* pairs. In contrast the coselection method inputs are *(query concatenated with senseID, url)* pairs, where the preprocessing groups coselected URLs together under the assumption that they belong to the same sense.

The Beeferman and Berger algorithm differs from the coselection method and the Wen et al. proposal in that the distinct senses of any query term are not disambiguated but are aggregated in the same cluster. In contrast, the co-selection method and the Wen et al. method both preserve the sense-singularity implied in the distinct selections.

The information stored by Wen et al. corresponds exactly to the information extracted from raw Web logs in the co-selection method. The primary differences between the two are that:

i) the co-selection method aggregates the co-selections for each query term, storing them as a doubly-weighted graph. It generates a graph for each term (the "term graph") which shows distinct subgraphs corresponding to distinct (disambiguated) senses of the query term;

ii) the co-selection method then clusters the doubly-weighted graph to detect the distinct subgraphs;

iii) for the subsequent synonym and translation detection, based on (sub)graph overlap, occur when the term graphs for different terms are inspected to see whether there is a significant overlap, in which case there is potentially a synonym.

In contrast, the Wen et al. method never separates the co-selection data into different terms, but clusters the entire set of co-selections. Thus the input for the chosen clustering algorithm is different - SBQC input is the entire graph or session graph with multiple terms, while the co-selection method inputs only the term graph (on a single term).

3. EXPERIMENTAL ANALYSIS

The experiment aimed to evaluate the coselection clustering method as a disambiguation tool. The prior experiment on artificial data suggested that clickthroughs could satisfactorily disambiguate meaning [9]. The limitations of this experiment were primarily the artificial data and the small quantity of data. Subsequent data collection of over two years of complete Web logs from a UK university School of Computer Science (the Teesside data) showed strongly-forming clusters but with very little ambiguity, hence the data was not really useful for evaluating the potential of co-selections for disambiguation.

The RFP 2006 data provides an excellent opportunity to evaluate the disambiguation potential of co-selections, even with a relatively small proportion of ambiguous queries believed to be present [8]. The experiment was set up to assess the single-sense clustering potential of the coselection method as follows.

3.1 Calculating senseIDs with coselection

For each query session pair recorded in the logs, the documents clicked are considered to be semantically similar, even if query terms that generated the results page are known to hold ambiguity. Since sessions typically result in few clicks per query the sessions for each query are merged if they share one or more URLs, effectively aggregating information across multiple users and sessions while still retaining the co-selection information. This process iteratively merges (query, session) pairs that have recorded similar clicks until only a handful of clusters exist containing the (or at least the discoverable number of) distinct meanings of the query. Aggregation of this sort can be performed either explicitly at the query level or implicitly through use an appropriate clustering algorithm. When aggregating and merging at a query level more control may be exercised, as thresholds for merging queries can be controlled both on intra-query level as well as the inter-query level. It is the latter approach we take here.

In order to track the disambiguation of queries based on the underlying URLs we build bipartite graphs based on the click-through data using a method described initially in Beeferman and Berger [2], extended in Chan et al. [3] and used more recently in conjunction with click-through data with success for ranking of documents for a given query, including those not yet clicked in conjunction with that query [4]. As in [10] graphs are created during this process such that the two sets of nodes are unique pairs of (query, sessionIDs) and URLs. This is done in order to preserve the co-selection information that is lost when using a bipartite graph consisting of unique sets of queries and URLs with user information aggregated across sessions. Distinct from previous work, bipartite graphs are first created for each unique query term.

The first stage of clustering is then performed over the bipartite graphs on a query level. At this stage a simple connected component-clustering algorithm is used and a unique identifier assigned to each unconnected component. This clustering can be seen to be creating a set of candidate senses – each connected component represents the aggregate of a set of URLs that were selected in conjunction with each other in some session. Therefore the result of such clustering in a set of ((query, senseID), URL) pairs, which is again a set of bipartite graphs. Weights for each pair is then the aggregate of sessions that contributed to the (query, senseID), URL) pair. It is of note that this weighting is a trivial one, and that more complex strategies, including those that favour co-selected weights over weights built up from single clicks is expected to improve the quality of clusters.

In the second stage the set of discovered bipartite graphs for each query are merged with respect to their URLs to create one large, but not necessarily connected, bipartite graph. At this stage the candidate senses for each query are then clustered with other queries. It is of note that in some cases this reveals a transitive path amongst the underlying candidate senses URLs for a single query merging them and removing what would otherwise be incorrectly detected senses. The output of the algorithm is a set of clusters for both queries and URLs.

3.2 Data generated for comparative analyses

The clusters generated by the coselection algorithms are evaluated by comparing their conceptual coherence as denoted by human evaluation performed by evaluating the clusters as generated by the different algorithms. The algorithms implemented and compared in addition to the one proposed here were the agglomerative clustering algorithm [2] and the session-preserved algorithm of Wen et al. [10].

The agglomerative clustering algorithm was implemented as described in the original publication [2] except for a new similarity metrics as proposed in Chan et al. [3]. This was the baseline algorithm *Method 0*.

Method 1 is the implementation of the coselection method using the DBSCAN clustering algorithm[3], and the similarity metric described in Wen et al. [10].

Method 2 is the implementation of the Wen et al. session-preserving algorithm, with exactly the same clustering algorithm and similarity metric as for Method 1.

[3] See http://en.wikipedia.org/wiki/DBSCAN

Method 3 is the coselection method again with the DBSCAN clustering algorithm, but this time with the similarity metric from Chan et al. [3].

Method 4 is the coselection method yet again, but this time with the agglomerative clustering instead of DBSCAN.

All methods were implemented with the similarity threshold set to 0.9. In preliminary work we noted that the output of the agglomerative clustering algorithm with similarity thresholds set to 0.1 through to 0.8 were essentially the same, with only negligible differences.

The following table sums up the 5 different methods used to generate sets of clusters for analysis:

	clustering algorithm	**input**	**sim. metric**	**sim. threshold**	**min nodes per cluster**
0	agglomerative	(query url) pairs	from Chen et al. [3]	0 .9	3
1	DBSCAN	(query concatenated with sessionID url) pairs	from Wen et al. [10]	0.9	3
2	DBSCAN	(query concatenated with senseID url) pairs	from Wen et al. [10]	0.9	3
3	DBSCAN	(query concatenated with senseID url) pairs	from Chen et al. [3]	0.9	3
4	agglomerative	(query concatenated with senseID url) pairs	from Chen et al. [3]	0.9	3

Table 1: cluster generation methods for the experiment

These five methods were chosen so that certain variables could be isolated for analysis. Methods 1 and 2 allow us to compare the input type (sessionID pairs or senseID pairs), while methods 1 and 3 allow comparison between the two similarity metrics. Methods 3 and 4 allow us to compare agglomerative clustering versus DBSCAN.

A key observation about the methods is that the baseline method, method 0 does not find sense-singular clusters, while the remaining methods should.

4. RESULTS

4.1 Comparing with human clustering

To assess how accurate the formed clusters were, a human ground-truthing was performed[4].

A set of between 49 and 50 clusters were randomly selected from those found by each of the five methods described in 3.2 from the RFP 2006 data, totalling 248 clusters for evaluation. Clusters generated by methods 1, 2, 3 and 4 were expected to be single-sense clusters while method 0 was expected to be combined sense clusters.

A number of human evaluators were then asked to find the major sense-singular cluster among the results pages. In a preliminary study involving five participants evaluating 300 clusters each (150 for the baseline method (0) and 150 for the co-selection method (4)) these instructions were deemed to be to vague, with participants commenting they did not know where to draw the line with respect to the level of ambiguity. This particularly seemed to occur when hyponyms were present. This sentiment was reflected in the inter-evaluator variance, which prevented any statistically significant results being discovered.

In response to this the instructions for the second evaluation was made more specific, asking evaluators to "select the most specific term that is related to the greatest number of other terms in the list both at the same or more generic concept level". Therefore the evaluators were told what level of conceptual specialty to consider and then given a guide as to how to select related concepts. By specifying that only more generic concepts would be considered related it was hoped that a consistent view on cluster coherency could be obtained. Despite this, however, a large amount of variance between evaluators was recorded, again preventing any statistically significant results being discovered.

Therefore, while the human evaluators were not being asked to detect multiple clusters, they were asked to select URLs that belonged to the "majority" sense of the cluster, i.e. to find the biggest single-sense grouping of URLs in the cluster. The presentation to the human evaluator of clusters from both methods was randomly mixed up to avoid any preconceptions contaminating the data. A number of human evaluators completed the entire evaluation with partial evaluations from others[5].

The difference between this human ground-truthing and the human judgement in the coselection is that the ground-truthed selections are deliberately and explicitly made for the purpose of clustering, whereas the coselection-based clusters are byproducts of other activity and are implicit relevance judgments.

The results were interesting as they showed that the greatest semantic coherence was found in the method that had no mechanism for disambiguation. The means for cluster coherence for each of the methods is given in the following table:

Method	**Average coherence of clusters**
Method 0	0.94779547
Method 1	0.8614083628571428
Method 2	0.90143668
Method 3	0.9106194785714286
Method 4	0.9140189185714286

Table 2: Average cluster coherence for each method

This may be due to the low number of clusters selected from only chose around 50 from each method so that they are not entirely typical of the entire cluster set. Alternatively it may be that the random selection of cluster to human-evaluate was a bad decision as the level of ambiguity in queries is quite low, around 4% [8] [10]. Investigating the ambiguity of the terms selected by comparing to the Wikipedia list of ambiguous terms, we found that while 1034 (distinct) and 1567(non-distinct) ambiguous terms from the Wikipedia list were present in the clusters generated in all methods, only 72 (distinct) and 79 (non-distinct), so that only 7% (distinct) and 5% (non-distinct) of the ambiguous terms appeared in the human-evaluated clusters. Clearly a more specific human evaluation of clusters containing ambiguous terms needs to be done.

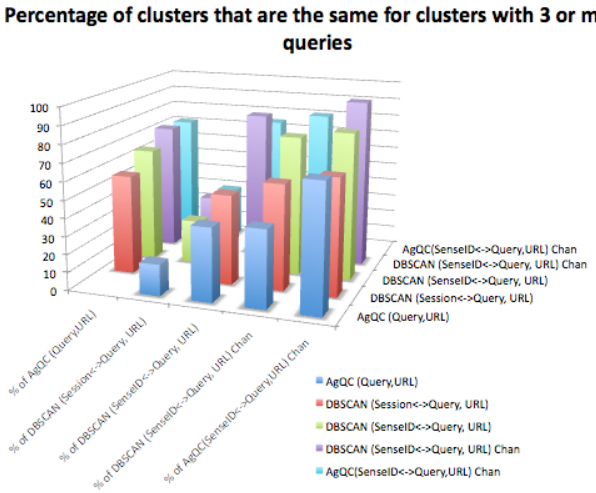
[4] See http://sl.cis.unisa.edu.au/~gavin/cevalre/ for the evaluation interface, including the example and instructions.

[5] At the time of writing, 7 people had completed all 248 evaluations from each method, with another 5 having done part of the evaluations. This data collection is ongoing.

When applying t-tests over the differences between the methods, there was no significant difference between methods when considering either the different similarity metric, clustering algorithm or input. However, the next analysis in section 4.2 did find differences for inputs and similarity metrics.

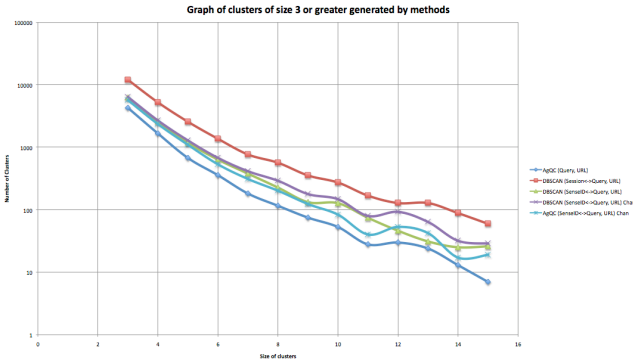
4.2 Edit distance based cluster content comparison

We also implemented a non-human comparison, based on how many clusters are the same between each method, combined with the edit distance between clusters. Since this analysis looked at all generated clusters it was not affected by the choice of randomly selected clusters as was the human evaluation. The comparison was based on a calculated a Levenshtein distance[6] between clusters, adapted in this case for the comparison of clusters as opposed to strings, i.e. we consider clusters to be one point different for each replacement, insertion or deletion required to make two clusters equal, when the operations are across queries, so that clusters that exactly match will have a Levenshtein distance of 0. These values can then be plotted into graphs and gives a syntactic view on cluster similarity and workings of the algorithms, as opposed to the semantic evaluation that will be obtained by the human evaluators.



The following figure plots the count of clusters of size 3 or more generated by each of the five methods. Beeferman and Berger's agglomerative clustering method unsurprisingly generates the smallest number of clusters as it does not separate out senses. The Wen et al. sessionID-preseving method created the largest number of clusters, while the three coselection-based methods appear in the middle ranges. While it requires further investigation, it seems that the sessionID-preserving method does not agglomerate all related clusters.

[6] See http://en.wikipedia.org/wiki/Levenshtein_distance



We also measured the average edit distance between most-like clusters generated by the different methods.

Methods	major distinction	average edit distance
Comparison of methods (1) & (2)	Input (SessionID|Query) vs (SenseID|Query)	2.569
Comparison of methods (0) & (4)	Input (SenseID|Query) vs (Query)	1.483
Comparison of methods (3) & (4)	DBSCAN vs Agglomerative Query Clustering	1.027
Comparison of methods (2) & (3)	Similarity Metric Wen et al. vs Chan	0.590

Table 3: Average Adapted Levenshtein Distance between method pairs

5. Identifying Ambiguity

A challenging application arising from being able to find sense definite clusters is the identification of ambiguous words. While such an approach is perhaps exceptionally tricky within an open world, we seek to first show that our approach is valid within the closed world of the dataset and then discuss the applicability of the approach in the case of an open world.

5.1 Hypothesis

Within a closed world for a given term *x* such that *x* has more than meaning within the closed world (dataset) we hypothesised that each sense of *x* will occur in different cluster for a sufficiently large minimum cluster size threshold *t* and sufficiently small maximum term overlap threshold *m*. I.e. each cluster that contains *x* is both sense distinct in its own right, but also sense distinct from any other.

To verify such a hypothesis we selected Method 1 generate clusters. This algorithm was selected as it created the highest number of large clusters therefore enabling a study with a higher threshold (*t*) value. The minimum cluster size for calculating clusters is set to three in order for cluster meaning to be identified sufficiently and the maximum cluster size set to the maximum mined which was 89. In order to verify the hypothesis we first validate it at the extremes by setting *t=89* and *n=1* and manually verifying the semantic cohesion of the cluster. We then set *t=3* and *n=1* and observe a case where *x* is shared between two clusters where the senses of the clusters do not differ.

Having trivially verified our hypothesis holds at the extremes we then seek to prove that as *t* is increased then the proportion of clusters sharing less than *n* common terms with identical senses decreases.

5.2 Evaluation

For the selected clustering method we first generate all clusters, select those pairs that have a single overlapping term ($n=1$) and then assign the pairs to three bins depending on their size, with both clusters in the pair required to fit within the bins size allocation to be included. The first bin contained pairs with clusters of size greater than 25, the second cluster of size between 25 and 10 and the third contained clusters of size between 10 and 3. From the bins a random 15 cluster pairs from each bin were chosen for human evaluation. Human evaluation consisted of a two expert evaluators extracting what they considered the sense of the cluster while being exposed to only the single cluster in the pair and common term at one time. The common term and cluster pairs were then recombined and two further, unconnected, experts deciding if the clusters are conceptually different. The experts were allowed to assign one of three values, 1 (clusters conceptually distinct), 0 (clusters conceptually the same) or 0.5 (not so clear). The 0.5 measure was necessary due to the fuzzy nature of conceptual similarity. The three judges were then shown each others scores and allowed to discuss before assigning a final agreed rating resulting in score out of 15 for each cluster.

SizeOfCluster>25, n=1	*10<SizeOfCluster<25 n=1*	*SizeOfCluster<10 n=1*
14/15 (93.3%)	11.5/15 (76.6%)	11/15 (73.3%)

Table showing the number (%) correctly identified clusters containing an ambiguous terms from a randomly selected 15 per group.

While further evaluation is needed to statistically validate the result, these preliminary results indicate that, within the closed world of the dataset terms appearing across multiple clusters of at least size t have a much higher likelihood of being truly ambiguous terms. As t gets large, within the dataset, this likelihood becomes 1. That is, as the proportion of overlap is smaller compared to the cluster size, any overlapping terms are increasingly likely to be ambiguous terms.

While further evaluation is needed to statistically validate the result the preliminarily results indicate that, within the close world of the dataset terms appearing across multiple clusters of at least size t have a much higher likelihood of being truly ambiguous terms. As t gets large, within the dataset, this likelihood increases to 1. In the dataset this is at least 16 terms for a threshold value of t, as observed by hand based the evaluation of our sample above. It is of further note that 68.75% (11 out of 16) of these discovered ambiguous terms are ones not currently identified by Wikipedia based mining methods, such as that proposed by Sanderson [8].

5.3 Moving from a closed to an open world

With a closed world assumption we were able to forget to some degree the problem of unseen data, i.e. simply because the clustering algorithm has formed two distinct clusters does not mean they necessarily are ambiguous – in an open world we may simply not have seen the connection yet. So given such a possibility, the question is, can such a method be applied in the real, open world? The answer is perhaps twofold. Firstly, the ability to definitively detect ambiguous words is one open to debate not just at a pragmatic level but also at a philosophical level (as discussed below). Therefore it seems unlikely that algorithms that identify all ambiguity can be developed. Secondly the results show that the likelihood of the shared term being truly ambiguous being increased as t becomes large. In the open world exceptionally large datasets are available leading to very large values of t. In addition the required value of t could possibly linked to factors such as the number of meaning per word or the number of possible concepts formulations leading to a relatively stable, practically small value of t. So while the open world poses greater challenges for detecting ambiguous terms automatically, the observations from this closed dataset indicate sufficient promise for the cluster overlap method to warrant further investigation.

6. CONCLUSIONS AND ONGOING WORK

The main purpose of this work as outlined in the original proposal to the workshop[7] was to trial the coselection method in a real and substantial dataset, gathered over a broad range of users, with the aim of discovering whether it was feasible to firstly generate sense-coherent clusters of terms, and secondly to discover whether proportionally small overlaps could be used as indicators of ambiguous terms.

The prior trial in a controlled environment showed that the principle of generating sense-coherent clusters was feasible [9], so it remained to see whether the principle would also work in true clickthrough data. The collected data from a university source (the Teesside data) formed good clusters but contained very little ambiguity. The RFD 2006 dataset has provided an excellent opportunity to evaluate the principle in the most general form of such data.

In the experiment reported here, it was shown that sense-coherent clusters were generated by all five of the methods investigated. Interestingly the coherence was marginally strongest in the method that incorporated no sense-separation (Method 0) although this may have been an artifact of the clusters selected for human evaluation - these were selected randomly from each of the five methods, but if we had selected specifically clusters incuding the terms that were previously identifed as ambiguous, we may have observed Method 0 being relatively less sense-singular than the other methods. Ongoing work includes the selection of another set of clusters to maximise the ambiguity so that we can isolate the effect of ambiguity on the different methods, so the experiment reported here at least provides a baseline for comparison.

In summary, the use of the coselection method over the RFP 2006 dataset has given some valuable insights into a number of potential issues that were not obvious from the prior analyses over alternative datasets. There is promise that this method can cluster URLs into sense-singular agglomerations and hence to automatically separate out senses of ambiguous terms, and to be able to do it without explicit human intervention.

7. REFERENCES

[1] E. Agirre. L. Marquez and R.Wicentowski , Proc. Fourth International Workshop on Semantic Evaluations (SemEval-2007), 2007, Assoc. Computational Linguistics, http://www.aclweb.org/anthology/W/W07/W07-20}

[2] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 407–416, 2000. DOI= http://doi.acm.org/10.1145/347090.347176

[3] W.S. Chan, W. T. Leung and D. K. Lee, Clustering Search Engine Query Log Containing Noisy Clickthroughs. Proc.

[7] See http://sl.cis.unisa.edu.au/~gavin/doc/WSCD09-Ashman.pdf

International Symposium on Applications and the Internet (SAINT'04), IEEE, 2004.

[4] N. Craswell and M. Szummer. Random walks on the click graph. Proc. SIGIR 07, pp 239–246, ACM. DOI= http://doi.acm.org/10.1145/1277741.1277784

[5] [McCarthy, D. 2007. Word Sense Disambiguation: Algorithms and Applications, Eneko Agirre and Philip Edmonds (editors). COMPUTATIONAL LINGUISTICS-ROCHESTER- 33, no. 2: 255.

[6] S. Patwardhan, S. Banerjee, and T. Pedersen. 2007. UMND1: Unsupervised Word Sense Disambiguation Using Contextual Semantic Relatedness. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)}, 390-393. Association for Computational Linguistics}.

[7] Pedersen and Kulkarni. Unsupervised Discrimination of Person Names in Web Contexts. Proceedings of the Eighth International Conference on Intelligent Text Processing and Computational Linguistics, pp. 299-310, 2007.

[8] M. Sanderson. Ambiguous queries - test collections need more sense, Proc. SIGIR 2008, pp 499-506, ACM. DOI= http://doi.acm.org/10.1145/1390334.1390420

[9] Truran, Goulding & Ashman. Co-active Intelligence for Information Retrieval, Proceedings of ACM Multimedia '05, 547-55, 2005. DOI=http://doi.acm.org/10.1145/1101149.1101273

[10] J.-R. Wen, J.-Yun Nie and H.-J. Zhang. Query Clustering Using User Logs, Transactions on Information Systems, 20(1), ACM, 2002. DOI= http://doi.acm.org/10.1145/503104.503108

[11] J.-R. Wen and H.-J. Zhang. Query Clustering in the Web Context, in Information Retrieval and Clustering, Kluwer, 2002.