Word Sense Disambiguation and Information Retrieval

Mark Sanderson
Department of Computing Science,
University of Glasgow,
Glasgow G12 8QQ
United Kingdom
(email: sanderso@dcs.gla.ac.uk)

Abstract

It has often been thought that word sense ambiguity is a cause of poor performance in Information Retrieval (IR) systems. The belief is that if ambiguous words can be correctly disambiguated, IR performance will increase. However, recent research into the application of a word sense disambiguator to an IR system failed to show any performance increase. From these results it has become clear that more basic research is needed to investigate the relationship between sense ambiguity, disambiguation, and IR.

Using a technique that introduces additional sense ambiguity into a collection, this paper presents research that goes beyond previous work in this field to reveal the influence that ambiguity and disambiguation have on a probabilistic IR system. We conclude that word sense ambiguity is only problematic to an IR system when it is retrieving from very short queries. In addition we argue that if a word sense disambiguator is to be of any use to an IR system, the disambiguator must be able to resolve word senses to a high degree of accuracy.

1 Introduction

Word ambiguity is not something that we encounter in every day life, except perhaps in the context of jokes. Somehow, when an ambiguous word is spoken in a sentence, we are able to select the correct sense of that word without considering alternative senses. However, in any application where a computer has to process natural language, ambiguity is a problem. For example, if a language translation system encountered the word ‘bat’ in a sentence, should the translator regard the word as meaning: an implement used in sports to hit balls; or a furry, flying mammal?

The field of IR is no exception to this problem. For example, a manager of an on-line news retrieval system reported (in a personal communication with the author) that the recent change in British Prime Minister has caused problems. A number of users had tried to retrieve articles about the Prime Minister using the query ‘major’. This query caused many articles about ‘John Major’ to be retrieved. However, in addition many more articles were retrieved where ‘major’ was used as an adjective or as the name of a military rank.

From this example, it seems reasonable to assume that an IR system will improve its performance if the documents it retrieves are represented by word senses rather than words. Recently, research was conducted to investigate this method of document representation. The researchers used a word sense disambiguator (a program that attempts to resolve the senses of ambiguous words) to disambiguate an IR test collection. However, experiments using this disambiguated collection showed a drop in retrieval performance.

As a consequence of this unexpected result, it was clear that a more basic investigation of the significance of ambiguity to IR was required. It is the results of this investigation that are presented here.

The structure of this paper is as follows. A brief review of previous research is presented in Section 2, followed by an outline of the experimental objectives. Section 3 describes the experimental methods used. This is followed in Section 4 by a description of results. Finally, Section 5 covers conclusions and future work.

2 Previous Research

The automatic disambiguation of word senses is a problem that has been studied for many years - Gale, Church and Yarowsky [1] cite work dating back to 1950. Early attempts to build disambiguators [2, 3, 4] relied on a combination of hand built lexicons and rules. Although working well for the examples they were programmed for, researchers were never able to 'scale up' the disambiguators to work on large disambiguation problems.

However in 1986 Lesk [5] built a disambiguator that used the textual definitions of word senses in an on-line dictionary to provide sense evidence. By using this large reference work, Lesk's disambiguator had the potential to be applied to large scale problems. The disambiguation technique Lesk used is in fact similar to techniques used in IR. To disambiguate a word *w* appearing in a certain context (for example, the 20 words surrounding *w*), the definitions of all the potential senses of *w* were looked up in the online dictionary. These definitions could be thought of as a small collection of documents. Disambiguation was a ranked retrieval of the definitions using the context as a query. The sense defined by the top ranked definition was chosen as the sense of *w*.

Since Lesk's paper a bewildering array of disambiguators have been built: Cowie [6], Black [7], Wallis [8] and Demetriou [9] have made further use of dictionaries; Zernik [10] built a disambiguator using a morphological analyser; Hearst [11] used learning based on human evidence; Dagan [12] used bilingual corpora; Church [13] tried aligned bilingual corpora; Voorhees [14] and Sussna [15] used the WordNet thesaurus; and Yarowsky [16] used a combination of Roget's thesaurus and Grollier's encyclopaedia to produce one of the better performing disambiguators to date.

2.1 Disambiguation and IR

The first attempt to use a disambiguator with an IR system was by Weiss [2]. Using his disambiguator to resolve the senses of five ambiguous hand picked words in the ADI collection, Weiss reported a 1% improvement in IR performance.

Some of the most extensive research on ambiguity and IR was performed by Krovetz and Croft [17], who used the CACM and TIME test collections. For each of the standard queries in these collections, they performed a retrieval. For each retrieval, they examined the match between the intended sense of each query word and that word's sense in a number of the retrieved documents. This manual investigation involved the study of thousands of these query/ document word sense matches (or mismatches). Amongst other things, the study found that: a sense mismatch was more likely to happen when the document was not relevant to the query; and further, that sense mismatches occurred more often when there were a small number of words in common between the query and document. They concluded that the impact of sense ambiguity on IR was not dramatic, but that disambiguating word senses was probably beneficial to retrieval when there were few words in common between the document and the query.

The first large scale tests of applying a disambiguator to an IR system were performed by Voorhees [14] and Wallis [8]. Voorhees built a sense disambiguator based on the WordNet thesaurus [18]. She applied the disambiguator to the CACM, CISI, CRAN, MED and TIME collections. Unfortunately retrieval experiments run on these disambiguated collections resulted in a drop in IR performance.

Wallis used a disambiguator as part of a more elaborate experiment which replaced the words in a text collection by the text of their dictionary definitions. This was done so that synonymous words (which have similar dictionary definitions) would be represented in a similar manner, and therefore documents containing these synonymous words would be retrieved together. When replacing a word by its definition, a disambiguator was used to select the definition that most represented the word. Wallis performed tests on the CACM and TIME collections, but found no significant improvement in IR performance.

The results of both Voorhees and of Wallis are surprising as it would seem reasonable that if ambiguity were resolved, IR performance would increase. One of the problems faced by them was a lack of reliable performance figures for their disambiguators: for example, Voorhees reported problems establishing the correct (ie the intended) sense of some of the words in the standard queries. Such problems make it difficult to establish just what went wrong.

2.2 Evaluation of Disambiguators

Evaluation has always been a problem in disambiguation research, as the only way to evaluate a disambiguator's performance has been through the manual checking of its output. As this is such a time consuming process, most disambiguators have only been evaluated on a handful of words.

Yarowsky [19] reported on a novel technique for evaluating disambiguators that is completely automatic. The method involved the introduction to a text collection, of artificially created ambiguous words, called pseudo-words. The creation of such a word (in this case a size 2 pseudo-word) is performed by replacing all occurrences of two words, for example 'banana' and 'kalashnikov', by a new ambiguous word 'banana/kalashnikov'. The source of evidence used by the disambiguator (eg lexicon) would be updated to reflect the union of the two words. The disambiguator is then applied to each occurrence of the new word. Evaluation of the disambiguator's output is a trivial matter as we know beforehand the correct sense of each occurrence of the word.

However, like any simulation, there are limitations. The method chosen to form pseudo-words from individual words is one of random selection. Therefore, the various senses of a pseudo-word are unlikely to be closely related. This differs from a proportion of actual ambiguous words whose senses are related in some manner. The significance of this difference is unclear, and therefore it can not be claimed that the ambiguity introduced exactly matches the ambiguity found in real situations.

Note that the technique of introducing ambiguity is not new to the field of IR, the transformation that word stemmers perform can be thought of as introducing ambiguity to a collection. Although of course, unlike the process presented here, the aim of a stemmer is to improve performance.

3 The Experimental Technique

Although Yarowsky invented psuedo-words solely for the purpose of evaluating disambiguators, his method would seem well suited to the examination of the relationship between sense ambiguity and IR. To conduct this examination, the performance of an IR system retrieving from a test collection is first noted. Then, ambiguity is introduced into the collection using pseudo-words. The performance of the IR system retrieving from this additionally ambiguous collection can be compared to the performance figures gained from the initial retrieval.

Pseudo-words allow the experimenter to vary, at will, the precise amount of ambiguity in a collection. So for example, levels of ambiguity that far exceed the levels in standard test collections could be studied. However, the primary advantage of using pseudo-words is that the disambiguation of the pseudo-words can be precisely controlled by the experimenter. Therefore the effects on retrieval performance of a disambiguator operating at varying levels of accuracy can also be studied. It is this use of pseudo-words to simulate ambiguity in a test collection that forms the basis of the experiments presented in this paper.

4 Test Collection

Before explaining experimental detail and results, it is necessary to describe the test collection that was used. The collection chosen for the experiments was the Reuters text categorisation collection (created to test the Construe system [20], later modified by Lewis [21]) which consists of 22,173 documents taken from the Reuters newswire. The Reuters collection was chosen in preference to the standard IR test collections (CACM, Cranfield, LISA, TIME, NPL etc.) because it is significantly larger than them and in addition, it was felt that the usage of English in Reuters was less specialised than in many of the test collections. The latter is an important factor for planned disambiguation experiments which will use a standard English dictionary to provide evidence of word senses.

The main difference between Reuters and an IR test collection is that Reuters doesn't have a set of standard queries with corresponding relevant documents. However each document in Reuters is tagged with a number of manually assigned subject codes. It is these codes that allow us to use Reuters as a test collection for comparing document representation methods. This use of Reuters was first described by Lewis and it is his method, with some modifications, that is described here.

First, **R** is defined as the set of all documents in the Reuters collection. This set is then partitioned into two subsets of equal size: **Q** (the query set) and **T** (the test set). The method used to partition **R** was chosen to be a random assignment of documents into one of the two subsets. This method ensured that groups of documents covering common themes would be evenly distributed to both **Q** and **T***.

Next, **S** is defined as the set of all subject codes that have been assigned to at least one document in **Q** and at least one document in **T**. If we pick one of the subject codes from **S**, we can now perform a retrieval. (The retrieval system used in these experiments was developed specifically for this work. It is based upon the probabilistic weighted term model as described in [22].)

For example suppose we perform a retrieval for the subject code 'crude'. First, all documents in **Q** tagged with 'crude' are selected. Then by performing relevance feedback using the selected documents, word/weight pairs are generated to form a query. This query is used to retrieve from the **T** set. The resulting ranked document list is examined to see where in the ranking, documents tagged with 'crude' appear. The position of the tagged documents is used to produce precision/recall figures. A conservative interpolation technique (outlined in [23]) is used to transform these figures into precision values at ten standard recall levels (0.1, 0.2, …, 1.0).

This process is repeated for each subject code in **S**, each time producing another set of precision values. These precision values are then averaged to give an overall set of values for each of the ten standard recall levels.

So by partitioning Reuters and using the subject codes, all the components of a classic IR test collection are created.
• the collection to be searched - **T**
• a set of queries - generated from **Q**, for each element of **S**
• a set of relevant documents for each query - documents in **T** tagged with the respective element of **S**.

The use of relevance feedback to generate the queries in place of verbose user generated queries means that the form of retrieval can be likened to an iteration of relevance feedback during a retrieval session.

4.1 Data Reduction

When performing a retrieval experiment using Reuters the question arises, how many query words should be generated by the relevance feedback process? It is clear from the work of Hughes [24] and Harman [25] that in a given situation there is an optimum number of words to use. As it was thought that such optimum numbers may be dependent on the amount of introduced ambiguity, the number of query words added was made a variable of the retrieval experiments. Therefore, the experimental results are expressed in three variables: precision (*p*), recall (*r*) and the number of query words added (*w*). The results can be plotted on a three-dimensional graph as shown in Figure 1. From the graph we can see that for all recall levels, the precision is low at *w*=1, with a rapid rise peaking at around *w*=5, before falling away as *w* increases.

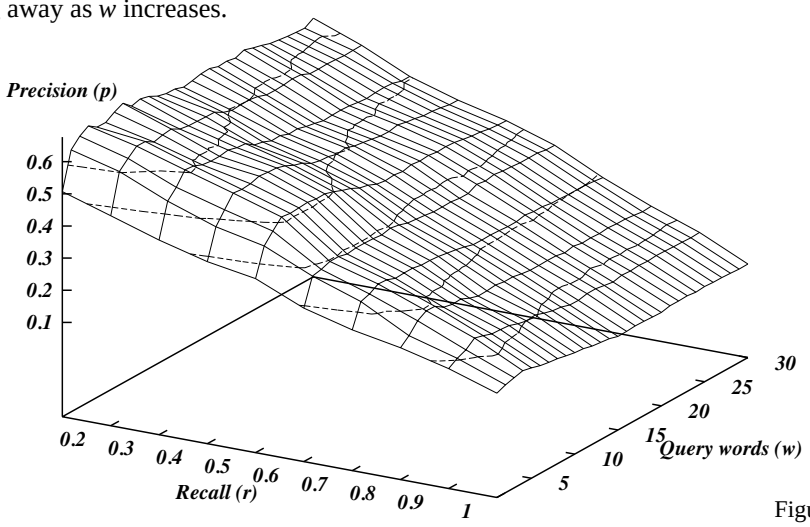

Figure 1

* This differs from Lewis who partitioned the collection based on the document's creation date. Such a partitioning was necessary for testing a newswire categorisation system, however this was not a factor of the experiments presented here.

Unfortunately it was found that three-dimensional graphs become difficult to read when the results of several retrieval experiments were plotted together. What was needed was a two dimensional plot of *w* against a variable expressing retrieval performance, in other words reduce the *p*/*r* figures to a single number. The method used to calculate this number is illustrated with the following example. To calculate the retrieval performance of the *p*/*r* figures tabulated in Figure 2, for each of the ten pairs of *p*/*r* numbers a corresponding *f* measure is calculated. The formula for *f* is,

$$f = \frac{1}{1/2(1/p) + 1/2(1/r)}$$

This measure is discussed in detail in [23]*.

Recall (*r*)	Precision (*p*)	F-measure (*f*)
0.1	0.592995	0.171140
0.2	0.544545	0.292552
0.3	0.472835	0.367091
0.4	0.432949	0.415823
0.5	0.398068	0.443249
0.6	0.326031	0.422488
0.7	0.278630	0.398600
0.8	0.224293	0.350358
0.9	0.165700	0.279872
1.0	0.107376	0.193929

Figure 2

Of the ten *f* measures calculated (Figure 2), the maximum (f-max) is selected as the retrieval performance figure. Applying this data reduction method, the graph in Figure 1 can now be plotted in two dimensions (Figure 3).

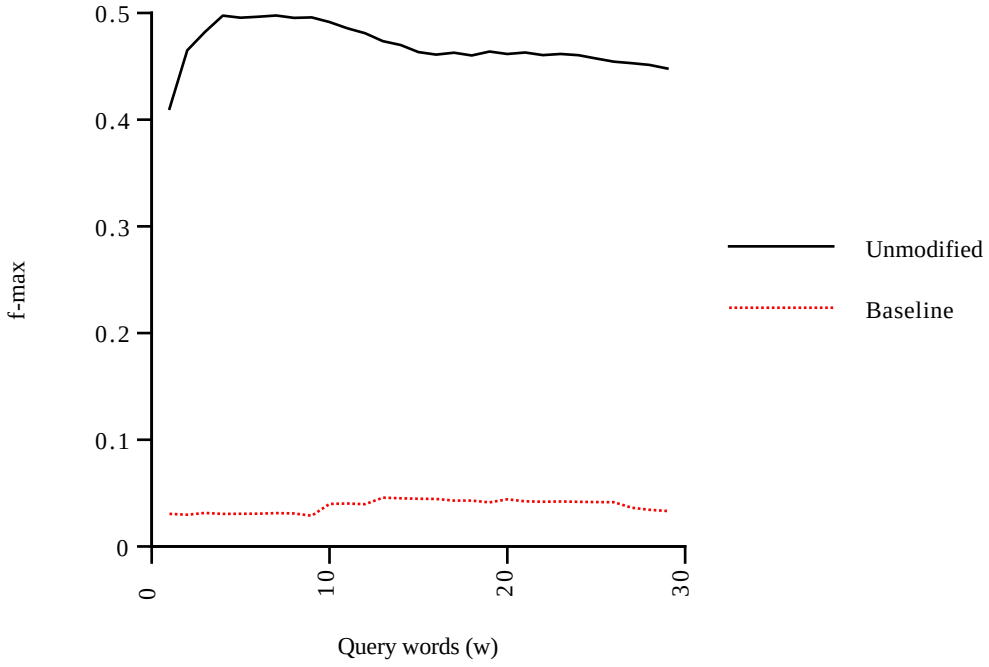

Figure 3

* In fact van Rijsbergen defines a measure called E, however F is simply defined as 1-E.

4.2 Random Case Test

Before any retrieval experiments were performed, it was necessary to establish how well the Reuters subject codes indicated document content. In other words, are the documents in set **Q**, marked with the subject code 'crude', good sources of evidence for retrieving similarly marked documents in set **T**?

The experimental method used to test this was identical to the method outlined above except for an additional step: when documents tagged with a certain subject code were selected from the set **Q**, a random set of documents were selected (from **Q**) and used to form the query instead. Figure 2 shows the result of this 'random case' experiment along with the result of an experiment using the subject codes as normal. As can be seen the random case is significantly worse than the method using the subject codes. In addition to establishing the utility of the subject codes as document content indicators, this experiment provides a 'baseline' which gives a scale to compare the differences between subsequent experimental results.

5 Experimental Results

Two sets of experiments were run: the first were concerned with the effect on IR performance of the introduction of additional ambiguity into the Reuters collection using pseudo-words; the second set of experiments studied the effect on IR performance of disambiguating pseudo-words (introduced into the collection) with a disambiguator operating at varying levels of accuracy.

5.1 Effects of Ambiguity on Performance

In the first experiment all words in the Reuters collection were paired to produce size 2 pseudo-words. The result of the retrieval experiment run on this additionally ambiguous collection is shown in Figure 4. As can be seen, when the result is compared to the retrieval experiment run on the unmodified collection, there is little difference in retrieval performance.

As this experiment showed only a small drop in performance, it was decided that more ambiguity needed to be introduced into the collection by creating larger pseudo-words. The creation of such pseudo-words is no different to the method outlined above. For example, to create a size three pseudo-word, all occurrences of the words: 'banana', 'kalashnikov', and 'anecdote' would be replaced by the pseudo-word 'banana/kalashnikov/anecdote'.

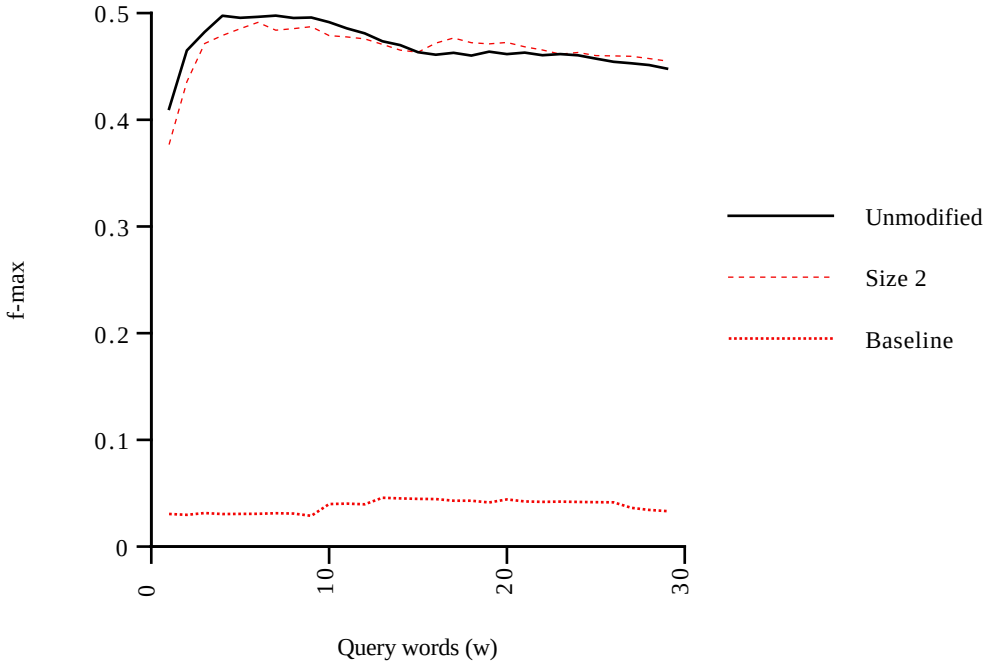

Figure 4

Four further experiments were run where ambiguity was introduced into the collection using pseudo-words of sizes three, four, five and ten. The results of these experiments are shown in Figure 5. As can be seen, IR performance is remarkably resistant to the introduced ambiguity. This is quite a striking result when we consider that in the final experiment (size ten pseudo-words) the number of distinct words in the Reuters collection was reduced from around 40,000 to 4,000.

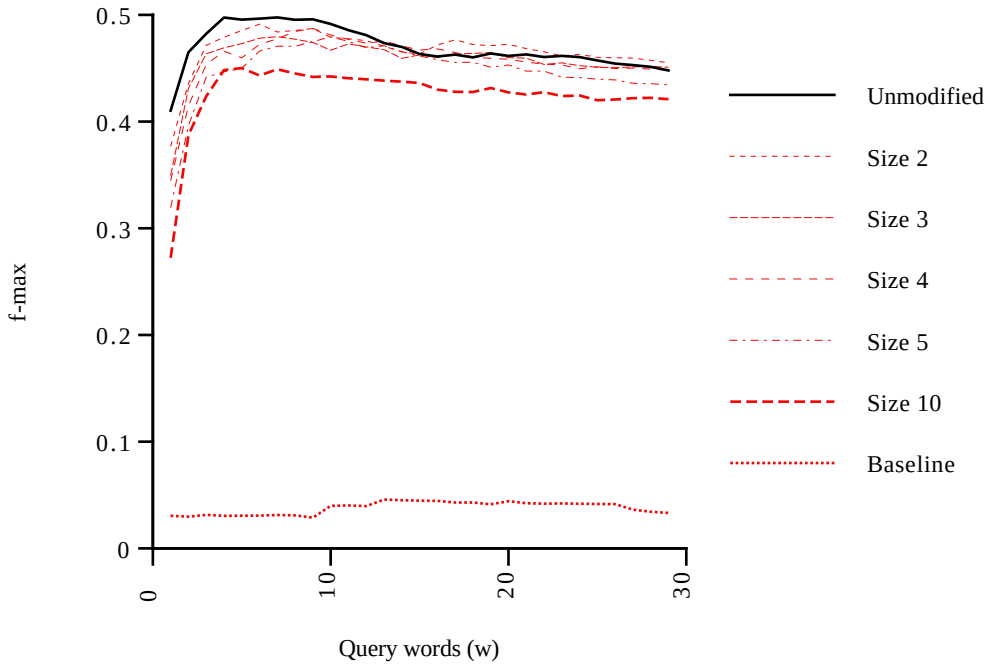

Figure 5

When comparing in detail the difference in performance between retrievals from the unmodified collection and retrievals from an ambiguous collection (Figure 6), we can see that the difference is greatest for retrievals based on queries of one or two words. Once the number of words in the query increases, the difference in performance quickly reduces. This result would seem to indicate that the degree of word collocation (ie the number of query words

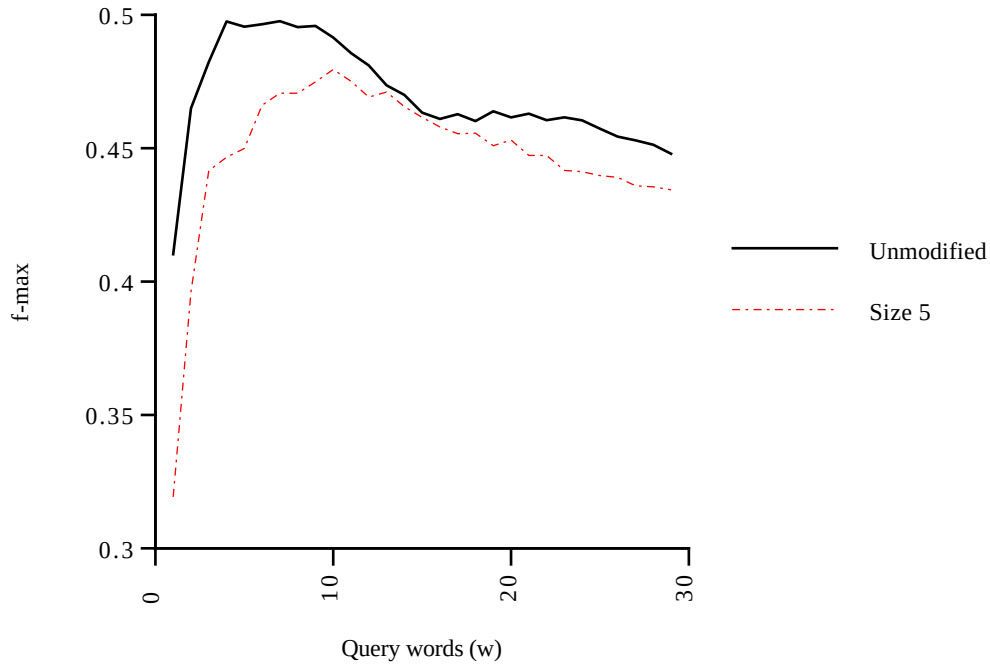

Figure 6

occurring in a retrieved document) plays an important role in the impact of sense ambiguity to IR. This concurs with the findings of Krovetz and Croft. Intuitively this is perhaps not too surprising, after all, if a document is retrieved by matching on the query words: 'mammal', 'flying', 'vampire' & 'bat', it is unlikely that this particular use of 'bat' refers to the sporting implement.

5.2 Disambiguating Ambiguity

The final set of experiments investigated the effects on performance of a pseudo-word disambiguator operating at varying levels of accuracy. For these experiments ambiguity was introduced into the collection using size five pseudo-words. This additionally ambiguous collection was then disambiguated, but with a controlled amount of error. A retrieval was then run on the 'erroneously disambiguated' collection.

This experiment was performed a number of times, each time with the percentage of correct disambiguations set to a different value. The results of two of these experiments are shown in Figure 7. As can be seen from the graph,

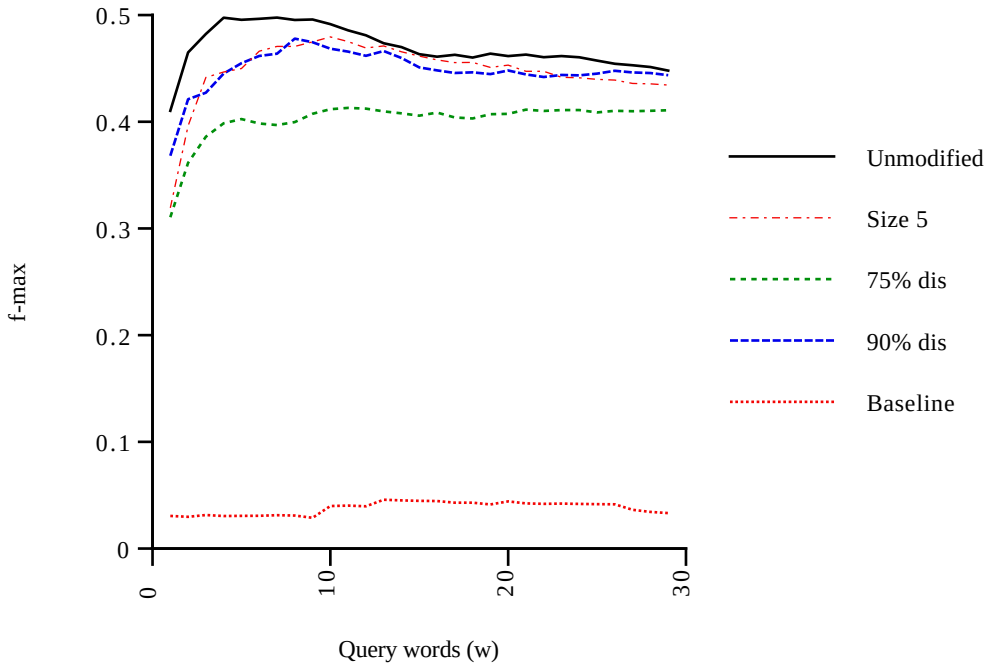

Figure 7

disambiguation accuracy has a dramatic effect on performance. When the introduced ambiguity is disambiguated with an accuracy of 75%, the retrieval performance is actually worse than performance using the ambiguous collection. With disambiguation at 90% accuracy, performance is similar to that of the ambiguous collection, although a small improvement can be seen for retrievals based on queries composed of one or two words. There is some anecdotal evidence to suggest that in general, tools built for computational linguistics tasks need to operate at, at least 90% accuracy before they are of practical use.

6 Conclusions

Using the novel experimental technique of introducing and removing ambiguity into a test collection in a controlled manner, insights into the significance of ambiguity to IR have been gained. In general, we can conclude that the performance of such systems is insensitive to ambiguity but very sensitive to erroneous disambiguation.

One area to which these results may be pertinent is that of bilingual IR systems. Such systems consist of document collections written in a foreign language but searched with queries constructed in a user's native language. Such an approach is of obvious interest in the common situation where a person's ability to read a foreign language is greater than their ability to write it. It may be thought that the amount of ambiguity introduced by the automatic translation

process would adversely affect the retrieval performance. However, the results presented here suggest that this introduced ambiguity may not, in fact, be such a problem.

Overall, the results presented in this paper appear to confirm 'common sense' beliefs, such as the ability of collocation to resolve word sense ambiguity and the high accuracy required of a disambiguator, with perhaps a little surprise as to the degree to which IR systems are resilient to ambiguity. It is hoped that this refining of the general appreciation of word sense ambiguity may be useful in identifying which areas justify further investigation within the context of IR.

7 Acknowledgements

Thanks go to: Fabrizio Sebastiani, Alison Cawsey, Keith van Rijsbergen, Bob Krovetz and especially Iain Campbell for their constructive comments on earlier drafts of this paper.

8 References

1. Gale W, Church KW, Yarowsky D. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. Proceedings of the ACL, 1992; 30:249-256

2. Weiss SF. Learning to disambiguate. Information Storage and Retrieval, 1973; 9:33-41

3. Kelly E, Stone P. Computer recognition of English word senses. North-Holland Publishing Co., Amsterdam, 1975

4. Small S, Rieger C. Parsing and comprehending with word experts (a theory and its realisation). In: Strategies for Natural Language Processing, Lehnert WG, Ringle MH (Eds), LEA, 1992, pp 89-148

5. Lesk M. Automatic sense disambiguation: how to tell a pine cone from an ice cream cone. Proceedings of the SIGDOC Conference 1986; 24-26

6. Cowie J , Guthrie J, Guthrie L. Lexical disambiguation using simulated annealing. Proceedings of COLING Conference, 1992;359-365

7. Black E. An experiment in computational discrimination of English word senses. IBM Journal, 1988; 32:185-194

8. Wallis P. Information retrieval based on paraphrase. Proceedings of PACLING Conference, 1993

9. Demetriou GC. Lexical disambiguation using constraint handling in Prolog (CHIP). Proceedings of the European Chapter of the ACL, 1993; 6:431-436

10. Zernik U. TRAIN1 vs. TRAIN2: Tagging word senses in corpus. Proceedings of RIAO 91, Intelligent Text and Image Handling, 1991; 567-585

11. Hearst MA. Noun homograph disambiguation using local context in large text corpora. Proceedings of the 7th conference, UW Centre for the New OED & Text Research Using Corpora, 1991; 7

12. Dagan I, Itai A, Schwall U. Two languages are more informative than one. Proceedings of the ACL, 1991: 29:130-137

13 Church KW. Using bilingual materials to develop word sense disambiguation methods. Proceedings of ACM SIGIR Conference, 1992; 15: 350

14. Voorhees EM. Using WordNet™ to disambiguate word sense for text retrieval. Proceedings of ACM SIGIR Conference, 1993;16:171-180

15. Sussna M. Word sense disambiguation for free-text indexing using a massive semantic network. Proceedings of CIKM, 1993
16. Yarowsky D. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. Proceedings of COLING Conference, 1992; 454-460

17. Krovetz R, Croft WB. Lexical Ambiguity and Information Retrieval. ACM Transactions on Information Systems, 1992;10

18. Miller G. WordNet: an on-line lexical database. International Journal of Lexicography, 1990; 24:513-523

19. Yarowsky D. One sense per collocation. Proceedings of ARPA Human Language Technology Workshop, 1993

20. Hayes PJ. Intelligent high volume text processing using shallow, domain specific techniques. Working Notes, AAAI Spring Symposium on Text-Based Intelligent Systems, 1990:134-138

21. Lewis DD. Representation and learning in information retrieval. PhD Thesis, COINS Technical Report 91-93 Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003, 1991

22. Robertson SE, Sparck-Jones K. Relevance weighting of search terms. Journal of the American Society for Information Science, 1976;27:129-146.

23. van Rijsbergen CJ. Information retrieval (second edition). London: Butterworths, 1979

24. Hughes GF. On the mean accuracy of statistical pattern recognisers. IEEE Transactions on Information Theory, 1968;14:55-63

25. Harman D. Relevance feedback revisited. Proceedings of ACM SIGIR Conference, 1992;15:1-10