Word sense disambiguation and information retrieval

Mark Sanderson

Submitted for the degree of Ph.D.

to the University of Glasgow

from the Department of Computing Science

September 1996

UNIVERSITY
of
GLASGOW

Acknowledgements

Only after the working on and writing of a thesis, does one understand why most thesis acknowledgements are annoyingly full of deep squelchy love and heart felt thanks to everyone, from a distant relation to a faithful pet. Perhaps, after all the intense processing by the rational and scientific parts of one's brain, the emotional part feels a bit left out? Perhaps not.

I wish to thank from the bottom of my heart the following people for each providing some or all of the following traits: inspiration, relaxation, support, goading, criticism, discussion, reading, correcting, and snogging. They are, the IR group; Bob Krovetz; the GUM club; my parents; my second cousin twice removed; and my pet fish. Particular thanks go to my first supervisor Keith van Rijsbergen, to my second supervisor Alison Cawsey, to Iain Campbell, and to Pam Sanderson - thanks guys.

This thesis was funded in part by a grant from the British Library.

Telling quote

I read in comp.risks that a (nameless) library had some problems when upgrading from paper card catalogues to an on-line index. The software they used had a standardized set of keywords, and it replaced ‘Madonna’ with ‘Mary, Blessed Virgin, Saint’, causing reclassification of recent works by Ms. Ciccone.

--Matthew Haines (haines@isi.edu)

Abstract

In recent years, great advances have been made in the speed, accuracy, and coverage of automatic word sense disambiguators - systems that given a word appearing in a certain context, can identify the sense of that word. This research has prompted a number of investigations into the relationship between information retrieval (IR) and lexical ambiguity. The work presented in this thesis is such an exploration. It goes beyond previous research, however, by studying not only the relationship of ambiguity to IR, but also that of disambiguation to IR.

Starting with a review of previous research that attempted to improve the representation of documents in IR systems, this research is reassessed in the light of word sense ambiguity. It will be shown that a number of the attempts' successes or failures were due to the noticing or ignoring of ambiguity.

In the review of disambiguation research, many varied techniques for performing automatic disambiguation are introduced. Research on the disambiguating abilities of people is presented also. It has been found that people are inconsistent when asked to disambiguate words and this causes problems when testing the output of an automatic disambiguator.

The first of two sets of experiments to investigate the relationship between ambiguity, disambiguation, and IR, involves a technique where ambiguity and disambiguation can be simulated in a document collection. The results of these experiments lead to the conclusions that query size plays an important role in the relationship between ambiguity and IR. Retrievals based on very small queries suffer particularly from ambiguity and benefit most from disambiguation. Other queries, however, contain a sufficient number of words to provide a form of context that implicitly resolves the query word's ambiguities. In general, ambiguity is found to be not as great a problem to IR systems as might have been thought and the errors made by a disambiguator can be more of a problem than the ambiguity it is trying to resolve.

In the complementary second set of experiments, a disambiguator is built and tested, it is applied to a document test collection, and an IR system is adjusted to accommodate the sense information in the collection. The conclusions of these experiments are found to broadly confirm those of the previous set.

Table of contents

List of figures

List of tables

1 Introduction

An *information retrieval* (IR) system retrieves from a document collection, those documents that are relevant to a user's query. Although the collection can consist of any media type, this thesis is concerned only with the retrieval of text documents, and, more specifically, retrieval of such documents using a document ranking method.

1.1 Word sense ambiguity

An IR system is affected by the characteristics of text, one such characteristic is *word sense ambiguity.* Most words are ambiguous to some degree, what sense a word occurrence has depends on the context it appears in. For some words, their senses are unrelated, for example the word 'bat' could refer to an implement used in sports to hit balls or a flying mouse like animal. For most words however, their senses are related (e.g. through metaphor), the word 'crash' for example can refer to a physical event or the value of shares in a stock market dropping. As IR systems process written text, they are affected by word sense ambiguity. An example of such an effect was reported in a personal communication with the author. A manager of an on-line news retrieval system found queries about the current British Prime Minister were causing problems with their IR system. 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, but in addition many more articles were retrieved where 'major' was used as an adjective or as the name of a military rank.

Word sense ambiguity is not something encountered by people in every day life, except perhaps in the context of jokes. Somehow, when an ambiguous word is used in a sentence, people are usually able to select the correct sense of that word without conscious effort. This manual *word sense disambiguating* (WSD) ability has been investigated, an overview of which can be found in Hirst *[Hirst 86]*. Choueka and Lusignan *[Choueka 85]*, working with the French language, found that people could accurately determine the sense of a particular word from reading the previous two words alone. Miller *[Miller 54]* briefly describes similar work by Kaplan using the English language which seems to draw similar results to those of Choueka and Lusignan. These works show that accurate disambiguation can be performed without exposure to the wider context in which an ambiguous word appears.

Automatic WSD systems have been studied for many years - Gale, Church, and Yarowsky *[Gale 92a]* cite work dating back to 1950. For many years disambiguators could only accurately disambiguate the text of tightly focused subject areas. The nature of their design pre-

vented their ‘scaling up’ to process more general texts. In recent years this situation has changed and the accuracy, speed, and scalability of disambiguators has improved.

This improvement is such that it is now possible to apply a disambiguator to a document collection of wide ranging texts and expect it to disambiguate the text accurately. Retrieval can be performed on such a collection using an IR system, and because the words of that collection are represented as word senses, the quality of document retrieval can reasonably be expected to improve. It is an investigation of this possibility that forms the basis of this thesis.

1.2 Chapter breakdown

Chapters 2, 3, & 4 introduce and review the subjects of word sense disambiguation and IR. Chapter 2 contains a brief explication of the core concepts of IR. Chapter 3 outlines techniques that try to improve a document’s representation in IR and relates them to word senses. Chapter 4 presents a review of research in word sense disambiguation.

Chapter 5 describes the technique used to simulate, introduce, and control ambiguity in a document collection. The technique is used to measure the impact of such ambiguity on the effectiveness of an IR system. The experiments are described and conclusions drawn.

Chapter 6 outlines the design and choice of strategy used to construct the disambiguator to be tested. Initial pre-tests are presented that were performed on the disambiguator to establish its suitability.

Chapter 7 describes the time-consuming testing of the disambiguator’s accuracy against manually determined senses. The results of the tests are presented also.

In Chapter 8, the disambiguator is incorporated into an IR system and the effectiveness of that system is measured.

Chapter 9 concludes the thesis with a brief description of the contributions of this work and suggests possible areas of future research.

There are two appendices that describe related work performed during the core work of the thesis.

2 Core concepts in IR

The core concepts presented in this chapter are those concepts of IR that are referred to later in this thesis. The chapter starts with a description of the basic configuration of an IR system and the representation of documents and queries within it. This is followed by an explanation of a method used for calculating the relevance of documents to queries. A process of query reformulation known as relevance feedback is detailed and, finally, an evaluation of the retrieval effectiveness of an IR system is outlined. Those requiring a full introduction to the field of information retrieval are referred to the relevant texts *[Van Rijsbergen 79]*, *[Frakes 92]*.

2.1 Configuration of an IR system

We start with an abstracted view of the working of an IR system (Figure 1). Here, a retrieval starts with a user query. This query is first converted into an internal representation to allow it to be processed by the system. The transformed query is then matched against a collection of documents also stored in this representation. The system assigns to each document in this collection a score that indicates the relevance of that document to the entered query. The documents in this collection are ranked by their assigned *relevance score*, with the highest scoring documents being presented in rank order to the user.

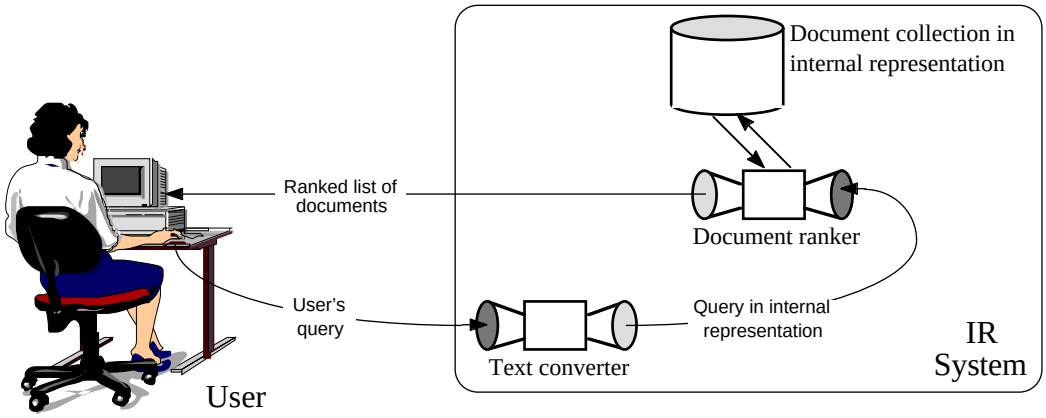


Figure 1. Configuration of an IR system.

2.2 Document and query representation

The relevance score must be calculated using a representation of the document collection and query that a computer can process and yet is good enough that the relevance score is meaningful. The representation typically used by systems is a set of features derived from the document collection. Each document in the collection is represented as a list of its features, the query is also represented in this manner. The most common feature set used is the set of words in the document collection. Before becoming features however, these words will typi-

cally have the case of their letters normalised. It is also likely that certain types of words such as prepositions, determiners, pronouns, etc. will be removed from the feature set. These removed words are known as *stop words*. Once the words have been processed into a feature set they are often referred to as *terms*.

An additional subtly to the representation of documents is the assignment of a numerical weight to all terms in a document collection. The weight assigned to a term occurring in a certain document is an attempt to quantify that term's importance to the subject of that document. There are many methods for calculating the weight of a term. Most are statistical, based on the term's frequency of occurrence within a collection, known as the *inverse document frequency* (*idf*), and on its frequency of occurrence within the document, known as the *term frequency* (*tf*). The term weighting function shown below is typical of such a method. The weight resulting from this function is often referred to as a *tf•idf* weight.

$$w_{ij} = \frac{\log(freq_{ij} + 1)}{\log(length_j)} \bullet \log\left(\frac{N}{n_i}\right) \quad (1)$$

w_{ij} = tf•idf weight of term i in document j
$freq_{ij}$ = frequency of term i in document j
$length_j$ = number of unique terms in document j
N = number of documents in collection
n_i = number of documents term i occurs in

Let us look at an example, imagine that we wish to retrieve from a collection of newswire articles, of which the example document shown in Figure 2 is a member. Each document in this collection is first transformed into a set of features: a set of terms with a *tf•idf* weight assigned to each term. Table 1 shows the feature set derived from the example document sorted by term weight.

Given the query 'bank practice in Amsterdam or Rotterdam', the system will transform this query into a set of features {bank, practice, amsterdam, rotterdam} and calculate a relevance score for each document in the collection. A simple but effective way to calculate this score relative to a certain query is to sum the weights of the query terms contained within each document. So for the query {bank, practice, amsterdam, rotterdam}, the relevance score of the document in Figure 2 would be calculated as shown in Table 2.

Once an initial retrieval has taken place, users might want to reformulate their query in the light of reading the documents retrieved from their initial query. We shall now look at a process that supports this reformulation.

```
 PATTERN-ID 27 TRAINING-SET
 1-APR-1987 04:18:30.07
TOPICS: corp-news cbond    END-TOPICS
PLACES: netherlands    END-PLACES
PEOPLE:     END-PEOPLE
ORGS: eib    END-ORGS
EXCHANGES:     END-EXCHANGES
COMPANIES:     END-COMPANIES

RM
f0238reute
u f BC-EIB-PLANS-300-MLN-GUI   04-01 0059

EIB PLANS 300 MLN GUILDER BOND ISSUE DUE 1995
    AMSTERDAM, April 1 - The European Investment Bank is
planning a 300 mln guilder 6.25 pct bullet bond due 1995, lead
manager Amsterdam-Rotterdam Bank NV said.
    The issue will be priced April 7 and subscriptions close
April 9. The payment date is May 14 and the coupon date May 15,
Amro Bank said.
 REUTER
```

Figure 2. Example document from newswire service.

Word	tf	idf	tf*idf	Word	tf	idf	tf*idf	Word	tf	idf	tf*idf
amsterdam	15	47	705	nv	9	45	405	close	9	27	243
guilder	15	46	690	bank	19	19	361	april	19	12	228
date	15	33	495	issue	15	24	360	manager	9	25	225
bond	15	31	465	plan	15	23	345	price	9	20	180
amro	9	50	450	coupon	9	38	342	mln	15	10	150
eib	9	50	450	european	9	33	297	pct	9	14	126
bullet	9	49	441	payment	9	31	279	said	15	4	60
rotterdam	9	47	423	lead	9	29	261	reuter	9	2	18
subscriptions	9	47	423	investment	9	28	252				

Table 1. Feature set derived from document in Figure 2.

bank	361	
practice	0	(not in document)
amsterdam	705	
rotterdam	423	
Rel. score	1489	

Table 2. Calculation of relevance score.

2.3 Relevance feedback

In IR *relevance feedback* implies a process of redirecting an IR system's output (documents) back into that system's input (query) to produce another, more accurate output (more relevant documents). A user indicates to the system which of the documents from the collection, just retrieved, are considered to be relevant to the user's query. The system selects a set of terms

characteristic of the relevant documents, and adds these terms to the user's query[1]. It can be expected that the addition of these terms will result in a query that more closely reflects the user's search requirements. The term selection process works as follows. Each term in the set of documents considered relevant is assigned a score that indicates how characteristic that term is of those documents when compared to the rest of documents in the collection. The highest scoring of these terms are selected.

$$w_i = \log\left(\frac{r_i(N - n_i - R + r_i)}{(R - r_i)(n_i - r_i) + 0.5}\right) \quad (2)$$

w_i = score assigned to term i

R = number of documents marked as relevant

N = number of documents in the collection

n_i = number of documents in which term i occurs

r_i = number of marked relevant documents in which term i occurs

0.5 is added to the denominator to avoid divide by zero

Equation 2 shows a classic function used in determining the score of a term. It measures the ratio of a term's frequency of occurrence[2] in the non-relevant documents, against its frequency of occurrence in the documents indicated as relevant. If these two frequencies are similar, the term's score is low. If, however, a term occurs much more frequently in the relevant documents than it does in the non-relevant, it is given a high score.

We illustrate this scoring process in the following example. Let us imagine that a user is perusing a list of retrieved documents, retrieved in response to the query 'nuclear waste dumping'. This user decides that three of these documents are relevant to his information need and indicates his preference to the system. Fragments of these relevant documents are shown in Figure 3. Normally, the term scoring would be applied to all terms in these documents. For this example, however, just three terms have been highlighted to illustrate the working of the function. It should be remembered that this function requires the documents to be represented as a set of features. For clarity, the documents are shown as text fragments, not feature sets.

Using Equation 2 we can see that in order to compute the score of a particular term, it is necessary to know two pairs of numbers: that term's frequency of occurrence in the relevant documents and in the document collection as a whole; and the size of the relevant document set

1. Relevance feedback can also refer to the simple re-weighting of a user's existing query terms with no additional terms being added. This variant of the process is not referred to here.
2. The frequency of occurrence of a term in a set of documents is defined as the ratio of two numbers, the number of documents containing the term against the number of documents not containing the term.

Document 1
…The experience of **Billingham** showed that disposal sites should in future be well away from populationcentres.

The extra cost involved would be worthwhile if public acceptance of the need for disposal could beachieved…

Document 2
…The costs of developing onshore sites are estimated at nearly 200 m pounds sterling each, but an offshoresite could cost much more, Nirex **said**.

The sites would be for materials stored mainly at Drigg, Cumbria, and in concrete silos at nuclear powerstations, factories and research establishments.

The highly **radioactive** material, which would have been buried about 1,000 feet underground at **Billingham**, includes the metal packaging of nuclear fuel rods.

The lowest level waste to be buried in clay at Elstow or elsewhere, includes tools, glasscontainers, plasticwrapping, pipes and discarded protective shoes and clothing…

Document 3
…Next week's meeting at the International Maritime Organisation will be considering the report, which has been completed but did not come to any firm recommendations. The officials will have to decide whether themoratorium should come to an end.

British officials **said** yesterday the report provided no justification for a ban on the disposal of **radioactive** waste at sea…

Figure 3. Fragments of three documents marked as relevant

and the size of the collection. The value of these numbers for each of the three highlighted terms is shown in Table 3.

term	r_i	R	n_i	N	w_i
said	2	3	13,236	20,000	0.01
billingham	2	3	25	20,000	3.23
radioactive	2	3	368	20,000	2.03

Table 3. Computation of term score.

Taking the term 'said' first, it has a slightly lower frequency of occurrence in the relevant documents than in the non-relevant, consequently the function computes a low score. The other two terms, however, have a much higher frequency of occurrence in the relevant documents than in the non-relevant which results in a high score. These terms are strong candidates for being added to the query. We can see from this example how relevance feedback can select potentially useful query terms that might never have been thought of by a user: 'billingham' for example is the name of a possible site for nuclear waste dumping in the UK. Sanderson *[Sanderson 91]* and Stanfill *[Stanfill 86]* both document the utility of relevance feedback to the retrieval process.

Now that the basic workings of an IR system have been described, a method used to evaluate a system's retrieval effectiveness is presented.

2.4 Evaluation

The evaluation of an IR system calls upon many research areas such as Human Computer Interaction (HCI), algorithms, statistics, etc. The only aspect of evaluation that concerns this thesis, however, is the evaluation of the quantity of relevant documents a system retrieves with respect to a query. This aspect of evaluation is known as *retrieval effectiveness* and numerous measures have been devised to compute it. An explanation and discussion of several can be found in Van Rijsbergen *[Van Rijsbergen 79]*. Of all these measures, the most often used for evaluating retrieval effectiveness are the complementary measures *precision* and *recall*. Precision measures the proportion of retrieved documents that are relevant, and recall measures the proportion of relevant documents that have been retrieved. In both cases at some cutoff point in a document ranking. They are formally defined as follows.

$$Precision = \frac{\text{retrieved relevant documents}}{\text{retrieved documents}} \tag{3}$$

$$Recall = \frac{\text{retrieved relevant documents}}{\text{all relevant documents}} \tag{4}$$

The classic method of using recall and precision to evaluate the effectiveness of an IR system is to measure precision at a number of standard recall values (i.e. recall=0.1, 0.2, 0.3, …, 1.0). These measurements result in a set of recall precision figures. Typically these figures are presented as a graph, an example of which is shown in Figure 4. The monotonically decreasing line shown in Figure 4 is typical of the shape of a recall precision (RP) graph.

Although it is possible to derive objective information about a system's effectiveness from such a graph, its main use is as a means of comparison. Figure 5 shows a graph of the RP figures of two retrieval systems. In this graph we see that system 2 has a higher precision than system 1, at all values of recall. Before we can state with confidence that system 2 is the more effective system however, some form of statistical significance test would have to be performed on the recall/precision figures.

As useful as these two measures are, there is one problem, to calculate recall, the total number of relevant documents in a collection must be known. The only way to obtain this value with complete certainty is to manually assess the relevance of every document in the collection. Because this is such a labour intensive task, a number of document collections have been created where this value has been determined for a set of standard queries. Many of these so

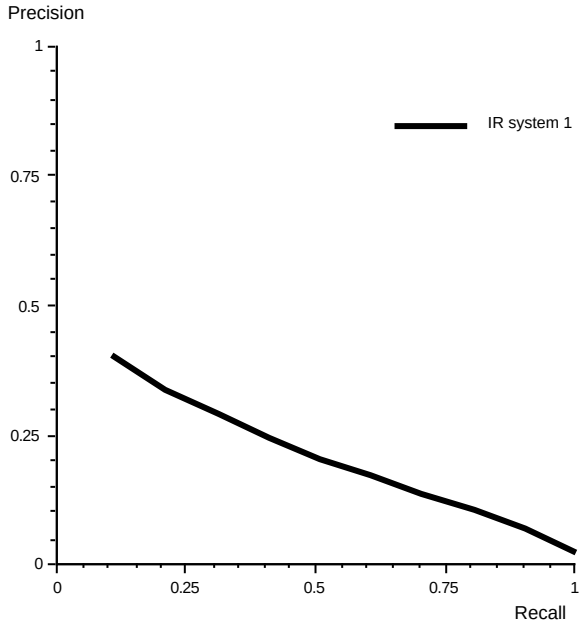


Figure 4. Classic monotonically decreasing line of a RP graph.

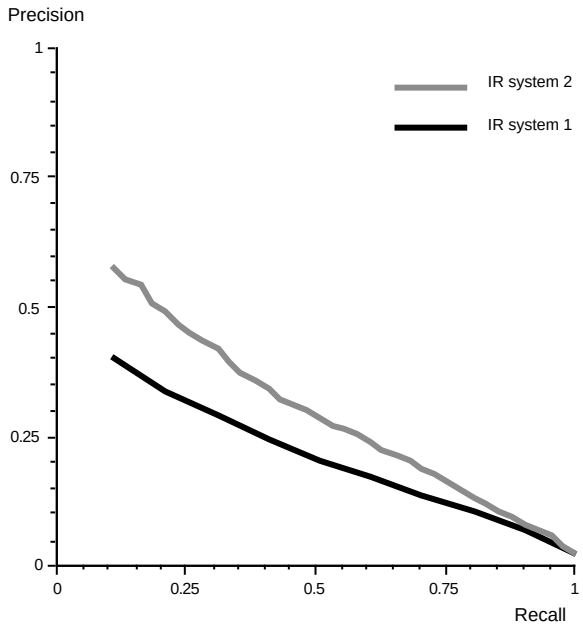


Figure 5. Comparing effectiveness of two IR systems.

called *test collections* have been placed in the public domain (Sparck Jones and Van Rijsbergen *[Sparck Jones 76]* provides a survey of some) thus providing researchers with a means to quickly evaluate their IR system's retrieval effectiveness. In addition, by using these collections researchers can directly compare their IR system to others.

As test collections get larger, the calculation of the total number of relevant documents becomes harder, due to the increasing manual effort required in assessing the relevance of every document. Eventually the size of test collections is such that the human resources

required to calculate this value are too great and it can only be estimated in some manner. Random sampling of a document collection has been used successfully by Blair and Maron *[Blair 85]*. Another technique to achieve this estimation, known as the pooling method, involves submitting a test collection query to a number of IR systems each using a different retrieval strategy. The top ranked documents retrieved by each system are manually assessed for relevance. Those judged to be relevant are regarded as the complete set of relevant documents for that query. For this technique to work, it is important that the systems mutually retrieve all or nearly all of the relevant documents.

For many years IR researchers have had at their disposal approximately eight publicly available test collections. Although at the time of their creation, these collections stretched the computing resources of IR researchers, nowadays when compared to the document collections most operational systems retrieve from, these eight are dwarfed in size, see Table 4. Blair and Maron *[Blair 85]* found that effectiveness varies with collection size. This suggests that results based on retrievals from small test collections may not hold when applied to larger collections. Because of this, small test collections are increasingly falling out of favour with IR researchers.

Collection name	No of docs	Bytes per doc	Col size (Mb)
adi	82	466	0.04
medline	1,033	1,079	1.10
time	423	3,663	1.50
cranfield 1400	1,400	1,203	1.60
cacm	3,204	717	2.20
cisi	1,460	1,526	2.20
npl	11,429	283	3.10
lisa	5,872	610	3.40
Commercial collection	75,900	2,853	206.50

Table 4. Comparing the size of test collections.
The first eight are test collections (measurements taken from the Virginia disc one *[Virginia disc 90]*) and the last is a typical document collection used in a commercial IR system.

In recent years a set of much larger test collections have been created, consisting of several hundred thousand documents that occupy giga bytes of storage. These collections are collectively known as the TREC collections. Their popularity is demonstrated by the yearly conferences *[Harman 95]* dedicated to presenting IR research using just the TREC collection. These

conferences have almost become competitive with research groups vying with each other to produce the most effective system for these collections.

This concludes the introduction to the core concepts of IR that are relevant to this thesis.

3 Enhancing a document's representation in an IR system

No matter how effective an IR system is at retrieving, scope for improvement is always sought on many fronts, e.g. better methods of calculating relevance scores, improved user interfaces, etc. Much of this thesis is concerned with an attempt to improve document and query representations (by disambiguating words) so this chapter first focuses on other language based approaches taken to improving these representations. It will then be shown that an awareness of word sense ambiguity can help to explain the failures of some of these attempted improvements.

3.1 Stemming

When thinking about possible improvements to an IR system, one of the most obvious areas to be addressed is the morphological variance of words. Users entering the query word 'work', will in all likelihood expect a system to retrieve documents containing the words 'works' or 'worked' as well. To enable this form of matching, when processing document and query representations, a word *stemmer* is employed to normalise the morphological variants of a word into a common root form.

An English word stemmer is mainly composed of transformation rules for the removal of suffixes (e.g. 's', 'ed', 'ing', 'ion'). For the morphological variants of many words this suffix stripping process is all that is necessary. For variants such as 'absorption' and 'absorb', however, additional rules, known as conflation rules, are required to complete the normalisation into a root form. Some variants do not adhere to any standard rules of transformation. To stem these, a table listing the transformation of each individual variant is required. Irregular verbs (e.g. ran/run, brought/bring, and worn/wear) would be listed in such a table. Other possible additions to a stemmer could include prefix removal (e.g. 'kilo', 'milli', 'micro'), or the ability to recognise proper nouns so that they are not stemmed (e.g. the town 'Inverkeithing', or the company[3] 'Thinking Machines'). Molto & Svenonius *[Molto 91]* provide details on an algorithm to detect such nouns.

A popular stemming algorithm is the Porter stemmer *[Porter 80]* which, apart from its stemming ability, is distinctive amongst stemmers due to its simplicity and small size. As available computational power increases, there is a trend towards using stemmers that employ large word lists to check the validity of applying stemming rules to variants. Examples of such stemmers are the WordNet stemmer *[Miller 90]*, *[WordNet]* and Krovetz's dictionary based

3. Company names have an unusual attribute in relation to ambiguity as copyright and trademark laws aid in preventing them from becoming ambiguous words.

stemmer *[Krovetz 93]* which Krovetz has shown to be one of the best stemming algorithms to date.

In terms of improving retrieval effectiveness, stemming does not make a large difference. Frakes *[Frakes 92]* in his chapter on stemming provides a summary of retrieval effectiveness experiments, the broad conclusion of which is that at worst stemming does not harm effectiveness. It would appear that stemming offers both benefits and costs: stemming results in more potentially relevant documents being retrieved, but by unifying the morphological variants of a word, semantic differences will be lost. Stemming the variant 'training' to 'train' for example blurs the distinction between a process of self improvement and a means of transportation.

3.2 Phrase representation

Another possible method of improving retrieval effectiveness is to represent documents and queries by phrases as well as words. In order to do this, a means of identifying phrases in text is required. There are two methods of identification, those using statistics and those using *natural language processing* (NLP) techniques. The statistical methods try to find pairs of words that co-occur near to each other in documents more often than is expected by chance. A distance measure between term pairs is often employed along with an ignoring of word order to ensure that phrases like 'information retrieval' and 'retrieval of information' are regarded as equivalent. The NLP based methods parse the grammatical structure of text, identifying certain syntactic patterns such as noun and verb phrases. Salton and Buckley *[Salton 89]* have compared the phrase identification accuracy of these two methods and have concluded that they are roughly equivalent. All that separates them is the computational power required to implement them, Salton and Buckley's implementation of a statistical phrase identification method required less computation than the NLP method.

The change in retrieval effectiveness resulting from the use of phrases in document and query representations has been found to vary. Smeaton *[Smeaton 87]* used NLP methods to perform phrase identification and showed some improvement in retrieval effectiveness. In a comparative study, Fagan *[Fagan 87]* showed that statistical methods improved an IR system's retrieval effectiveness better than NLP methods did. He found both phrase identification methods behaving erratically however, showing large improvements on some test collections and next to no improvement on others. More recently Lewis *[Lewis D.D. 91]* also investigated representing documents with phrases identified using NLP techniques. Working with a test collection of Reuters newswire articles, Lewis found the use of these phrases caused a reduction in retrieval effectiveness. One possible cause for this drop in effectiveness is the inaccuracy of the NLP systems used by Lewis.

The inconclusiveness of the utility of phrase representation in the research presented here is reflected in a recent TREC conference, TREC-3 *[Harman 95]*. An examination of the working of the systems with the highest effectiveness reveals that some use phrase representation methods (e.g. CLARIT *[Evans 95]*) and some do not[4] (e.g. OKAPI *[Robertson 95]*). Therefore, one can only conclude that the advantages of this representation method remain to be demonstrated.

3.3 Parts of speech tagging of text

One NLP system that performs its task to a high degree of accuracy is a *parts of speech tagger.* Such a system assigns grammatical tags (noun, verb, determiner, etc.) to the words of a corpus as well as determining if a word is being used as the head or the modifier of a phrase. Using large manually tagged corpora as training data, statistically based taggers such as the CLAWS system *[Garside 87]* are reported to tag text with an accuracy of over 95%.

There have been two investigations that examined the benefits of adding grammatical tags to the representation features of documents and queries (*[Smeaton 92]*, *[Sacks-Davis 90]*). Effectiveness might improve once the calculation of a system's relevance score is altered to give preference to documents that contain query words in the same grammatical form as is found in the query. Both Smeaton and Sacks-Davis applied a parts of speech tagger to the documents and queries of the CACM collection. Both researchers reported that their alterations resulted in no increase in effectiveness. Quite what caused this failure is not entirely clear. There is the ever present possibility that the taggers used in these experiments were not accurate enough, but perhaps the answer is that simply adding syntactic structure to a document and doing no more with that structure is of little use. Perhaps only when that structure is used to derive semantic information, can a benefit be found.

Smeaton and Sheridan *[Smeaton 91]* extended this work to discover if any benefit could be gained from parsing whole sentences and representing those sentences by the resulting syntactic structure. They examined how to calculate the degree of match between the parse tree structures (they called Tree Structure Analytics (TSA)) of two sentences. In addition to counting the number of words in common between the two sentences, parts of the syntactic structure were taken into account to assess the importance of a particular word to a sentence's meaning or the significance of word order. As their system could only process single sentences, they ran tests on a collection of document titles. Early results from testing showed little or no improvement in effectiveness and so this line of investigation was abandoned.

4. Note that those that do not use phrases in their retrieval algorithms do measure some form of query word proximity within a document.

3.4 Matching areas within a document

It is quite common for IR systems to regard a document as a bag of words, discarding the document's structure and the location of words within that structure. This results in a loss of the context of each word in a document and therefore a loss of information about the sense of each word occurrence. Some systems retain word location information and allow users to specify a minimum separation between query words within retrieved documents. Although probably not often thought of in this way, when a user specifies that all query words must occur in close proximity, much is being implicitly said about the sense of those words. For example if the query words were 'racket' and 'tennis', it is unlikely that documents would be retrieved that contain occurrences of 'racket' referring to a clamour.

There have been attempts in IR to use word location to improve retrieval effectiveness. Recently Hearst and Plaunt *[Hearst 93a]* reported on their method of splitting documents in a collection into sub-sections. The relevance score of a document was calculated by summing individual relevance scores calculated for each of its sub-sections. This had the effect of giving preference to documents that contained occurrences of query terms in close proximity to each other. Different definitions of a sub-section were tested, of which two methods were found to work equally well: defining sub-sections to be paragraphs; and defining subject changes in a document (such changes were detected using a sentence similarity measure) as sub-section boundaries. Incorporation of each of these methods into an IR system resulted in an improvement in retrieval effectiveness of between 19% to 28%.

The test collection Hearst and Plaunt used was deliberately biased towards large documents (>1,500 words i.e. >3 pages of text) which are more likely to benefit from word location information than smaller documents as there is a higher probability of query terms being spread more widely in a large document than in a small one. Therefore, one must consider the possibility that the improvement Hearst and Plaunt report was exaggerated by the type of collection used. Nevertheless, their result provides strong evidence that preferring documents containing query terms in close proximity to each other is a useful retrieval technique.

3.5 Are word senses their undoing?

Of the approaches that have failed to elicit an improvement in retrieval effectiveness there are a number that have employed methods that rely on words being representations of a single sense. Recognising that this reliance is unrealistic sheds light on the reasons for these methods' downfall. This section outlines a number of retrieval methods that may fall into this category.

Using relevance feedback to expand a query with terms from relevant documents has been shown to improve retrieval effectiveness. This expansion however can only take place once relevant documents have been retrieved. Research has been conducted to examine ways in which an initial query might be automatically expanded. The idea underlying this research can be illustrated with the following example: if a query contains the word 'bank' and this word is found to co-occur frequently in the document collection with the word 'economic', then it would seem reasonable to automatically expand the query with this additional word. Unfortunately, Smeaton and Van Rijsbergen *[Smeaton 83]* showed the application of this expansion technique failed to produce any significant improvement in retrieval effectiveness. Both times the researcher concluded that, for their experiments, there was insufficient co-occurrence information to provide accurate enough expansion.

When thinking of word senses, however, another explanation arises. The automatic expansion of a query in the manner described here does not take into the account the sense of the query word. To continue the example, it may well be that the user entered the query word 'bank' intending the river sense of that word and the automatic expansion of this query with 'economic' resulted in a degraded retrieval. Similar problems occur when the words of queries or collection documents are expanded with synonyms procured from a thesaurus. If no heed is taken of word senses, thesaural expansion inevitably becomes a process of adding all the synonyms of all the senses of the word being expanded. Such wide ranging expansion is likely to be an obstacle to improving retrieval effectiveness. This has been confirmed in the experimental work of Lalmas *[Lalmas 96]*.

Appendix B of this thesis describes a process known as document imaging, which is an attempt to improve retrieval effectiveness by reweighting the terms of a document based on co-occurrence information within a document collection. Although it remains to be seen if retrieval effectiveness will be detrimentally affected by the interaction of this process and word sense ambiguity, the discussion in this appendix shows how ambiguity is the cause of unexpected retrieval results.

3.6 Summary

The aim of this chapter has been to provide the reader with an introduction to text based information retrieval and then to present language based attempts to improve the representation of documents and queries. During this presentation the importance of word sense ambiguity to IR was illustrated by the success of approaches that had considered this important factor and by the indifferent results of those that had ignored it.

4 Word sense disambiguation research

This chapter contains a review of WSD research and is divided into two broad classes: disambiguation based on manually created rules; and disambiguation using evidence derived from large corpora. Following on from this review is a discussion of the problems associated with testing the accuracy of disambiguators and finally a review of research into WSD and IR is presented.

4.1 Disambiguation based on manually generated rules

The majority of disambiguation systems until the 1980s were based on manually created rules for sense selection and much effort was required to build them (a more complete review of these disambiguators can be found in Hirst *[Hirst 86]*). Because of this, the systems described here are mainly demonstrators of a technique rather than practical 'ready to use' disambiguators. For anyone contemplating the construction of a disambiguator capable of processing thousands of different words, which is one of the aims of the work in this thesis, manually devising sense selection rules is not practical. Therefore, the work presented in this section is intended for historical interest only.

An early example of disambiguation is the research of Weiss *[Weiss 73]*. He manually constructed a set of rules to disambiguate five words. These rules were of two types, general context rules, and template rules. A general context rule would state that an ambiguous word occurrence had a certain sense if a particular word appeared near that ambiguous word. For example, if the word 'print' appeared near to the word 'type' then its sense was likely to be related to printing. The more specific template rules stated that an ambiguous word occurrence was a certain sense if a particular word appeared in a specific location relative to that occurrence. For example, if the word 'of' appeared immediately after the word 'type', then the sense of that occurrence was likely to be the 'variety of' sense.

Following limited testing, Weiss found that template rules were better at determining sense than the context rules, and so applied them first. To create these rules, Weiss examined 20 occurrences of an ambiguous word, and then tested these manually created rules on a further 30 occurrences. These tests were performed for five ambiguous words. The accuracy of the resulting disambiguator was of the order of 90%. Weiss examined the erroneous disambiguations and found them to be mostly idiomatic uses.

A larger disambiguator was built by Kelly & Stone *[Kelly 75]* who manually created a set of rules for 6,000 words. They consisted of contextual rules similar to those created by Weiss, in addition to rules for checking certain grammatical aspects of a word occurrence. In some

instances the grammatical category of a word is a strong indicator of its sense, for example ‘the train’, ‘to train’. The grammar and context rules were grouped into sets so that only certain rules were applied in certain situations. Conditional statements controlled the application of rule sets. Unlike Weiss’ system, this disambiguator was designed to process a whole sentence at the same time. It could vary the order in which the words of a sentence were disambiguated by stopping the disambiguation of one word, trying to disambiguate other words in the sentence, and then returning to the original word to discover if disambiguation could now be completed. The system, however, was not a success and Kelly and Stone reported:

> …we applied these techniques very energetically to real human language, and it became absolutely clear that such a strategy cannot succeed on a broad scale.

Another approach to disambiguation was tried by Small & Rieger *[Small 82]* using what they called ‘word experts’, which were essentially programs. Their idea was to build an ‘expert’ for each ambiguous word. When disambiguating words in a sentence the expert of each of these words would be invoked. An expert would examine its word’s context, make decisions about the possible senses of that word and publicise these decisions to the other experts. If, when processing its evidence, an expert could do no more, it would become ‘dormant’ and wait for other word experts in the sentence to publicise their decisions. This additional evidence would hopefully provide further clues to the dormant expert to enable it to ‘awake’ and finish disambiguating its word. There is no mention of testing this disambiguator and it would seem from the report of this work is that Small & Rieger got no further than the process of building the experts: at one point in their paper they stated:

> the expert for the word ‘throw’ is currently six pages long … this is large, but it should be ten times that size

The disambiguators described so far have been based on rules for determining word senses that were manually created. As we have seen from the work of Kelly & Stone and of Small & Rieger, when such disambiguators were extended to work on larger vocabularies, the effort involved in building them became too great and the resulting disambiguators showed little success. Since the mid 1980s, however, disambiguation research has moved away from manually created rules towards automatically generated rules based on disambiguation evidence derived from existing corpora available in machine readable form and it is this form of disambiguation that is now discussed.

4.2 Disambiguation using evidence from machine readable corpora

The first corpus based disambiguation was by Lesk *[Lesk 88]*. He used the textual definitions of a dictionary to provide evidence for his disambiguator. His use of the dictionary can be shown with a simplified example. Suppose we wished to resolve the sense of the occurrence of ‘ash’ in the following sentence.

> There was *ash* from the coal fire.

To disambiguate ‘ash’, its dictionary definition was looked up and the individual senses of this word (two in this case) were identified.

> ash(1): The soft grey powder that remains after something has been burnt.
>
> ash(2): A forest tree common in Britain.

Next the definitions of each of the context words in the sentence (apart from stop words) were looked up.

> coal: A black mineral which is dug from the earth, which can be burnt to give heat.
>
> fire: The condition of burning; flames, light and great heat.

What followed was a process similar to ranked retrieval: the individual dictionary sense definitions of ‘ash’ were regarded as a small collection of documents (a collection of two in this case); and the definitions of the context words of ‘ash’, which are ‘coal’ and ‘fire’, were regarded as a query. The two sense definitions were ranked by a scoring function based on the number of words co-occurring between a sense’s definition and the definitions of all context words. The top ranked definition was chosen to be the sense of this occurrence of ‘ash’. In this example, sense one of ‘ash’ was chosen, although only because the word ‘burnt’ appeared in both the definitions of ‘coal’ and of ‘ash’ sense one. The selection of senses based on such a small number of matching words is more likely to be error prone and there is the increased possibility of there being no matching words. This problem with Lesk’s disambiguation technique will be addressed later.

Lesk performed some limited testing and reported a disambiguation accuracy of between 50% and 70% which in comparison with later disambiguators, was not high. The importance of Lesk’s work, however, was to demonstrate the use of existing corpora as sense disambiguation evidence and by doing so, to raise the possibility of building, without much effort, a disam-

biguator capable of resolving the senses of a great many words. Lesk's work prompted much varied research in this area, using many different corpora types and sense selection methods, and highlights of this research are reviewed below.

One of the motivations for conducting this review is to find the method of disambiguation that will be best suited to a set of WSD and IR experiments to be described later in this thesis. Therefore, throughout this review comments will be made on the suitability of each disambiguation method with respect to these experiments. They are categorised here in this review by corpora type, starting with dictionaries and some of the 'low level' problems associated with them, then moving onto disambiguators based on manually sense tagged corpora, followed by multilingual, and finally thesaurus based disambiguators.

4.2.1 Machine readable dictionaries

Many machine readable dictionaries became available to researchers when publishers released into the public domain typesetting files normally used for producing the paper copy of a dictionary. These files were never intended for lexical analysis and, as seen in Figure 6, they are full of typesetting instructions: in this figure these take the form '*nn'. Wilks *[Wilks 90]* describes this situation as having a *Machine Readable Dictionary* (MRD) but needing a *Machine Tractable Dictionary* (MTD) to facilitate meaningful access to the information by a computer program.

```
metallic
1 M0088800 !< me *80 tal *80 lic
3 m9!"t *67 lIk
5 adj !<
7 100 !< Wa5 !< MT-- !< ----S
8 *45 a *44 of or like metal : *46 metallic colours !| metallic coins *45 b
*44 partly of metal : *46 a metallic mixture
7 200 !< !< ---- !< ----T
8 with a ringing quality (of sound) : *46 a sharp metallic note
```

Figure 6. Extract from typesetting file of dictionary.

Fortunately the fields of a dictionary entry (grammatical code, phonetic spelling, definition, etc.) are usually sufficiently structured by the typesetting instructions to provide a lexicon that can be exploited to distil information from the entry. Much work has explored the production of programs that transform dictionary typesetting files into a form usable by lexical analysis programs. For example Neff & Boguraev *[Neff 91]* designed a grammar and built a parsing system that successfully transforms MRDs into a lexical database.

Looking at the example definition we can see some of the possible transformations. For example the lines that are labelled with the number '8' contain written definitions of a word

sense followed by examples of use. Utilising the typesetting instructions it is possible to separate these two components. The examples of use are always displayed (for this dictionary) in italics,'*46' is an instruction to use italics and so the presence of this instruction on this line is an indication of the start of examples of use (metallic colours, metallic coins, a metallic mixture).

Given that methods do exist to reliably transform a MRD into a MTD, we now return to the review of disambiguation research that uses them.

4.2.2 Dictionaries and disambiguation

As we saw in the example disambiguation of an occurrence of 'ash' using Lesk's technique, only one word was found to co-occur between the definition of the context words and the dictionary sense definitions of 'ash'. Lesk acknowledges this as a problem and mentions that his disambiguator was unable to process a number of ambiguous word occurrences because no words co-occurred between the context and ambiguous word definitions. He suggested one solution might be to use dictionaries with larger definitions such as the Oxford English Dictionary *[Simpson 89]*. This idea however was never tested.

With a different approach to the problem, Wilks et al. *[Wilks 90]* used a technique of expanding a dictionary definition with words that commonly co-occurred with the text of that definition; the idea being that commonly co-occurring words are semantically related to those in the definition. This co-occurrence information was derived from all definitions in the dictionary and all definitions were expanded in this manner. Wilks chose to expand the Longmans Dictionary of Contemporary English *[Longman 88]* (LDOCE). This dictionary was intended for people for whom English is a second language, therefore, all its definitions were written using a simplified vocabulary of around 2,000 words. Wilks stated that the use of this vocabulary produced a large number of word co-occurrences. In addition, the vocabulary contained few synonyms which would have been a distracting element in the co-occurrence calculations. Figure 7 (taken from Wilks's paper) displays graphically some of the definition words found to commonly co-occur in LDOCE.

Using this network and a definition of the word 'bank' shown in Figure 8, we look at how Wilks' method expanded the geographical definition of this word. The word 'river' is expanded with the words 'flood', 'across', 'bridge'; and the word 'lake' is expanded with 'shore'. Thus the definition contains additional semantically related words that increase the evidence upon which a disambiguator can draw, therefore reducing the problems that had been encountered in Lesk's disambiguator.

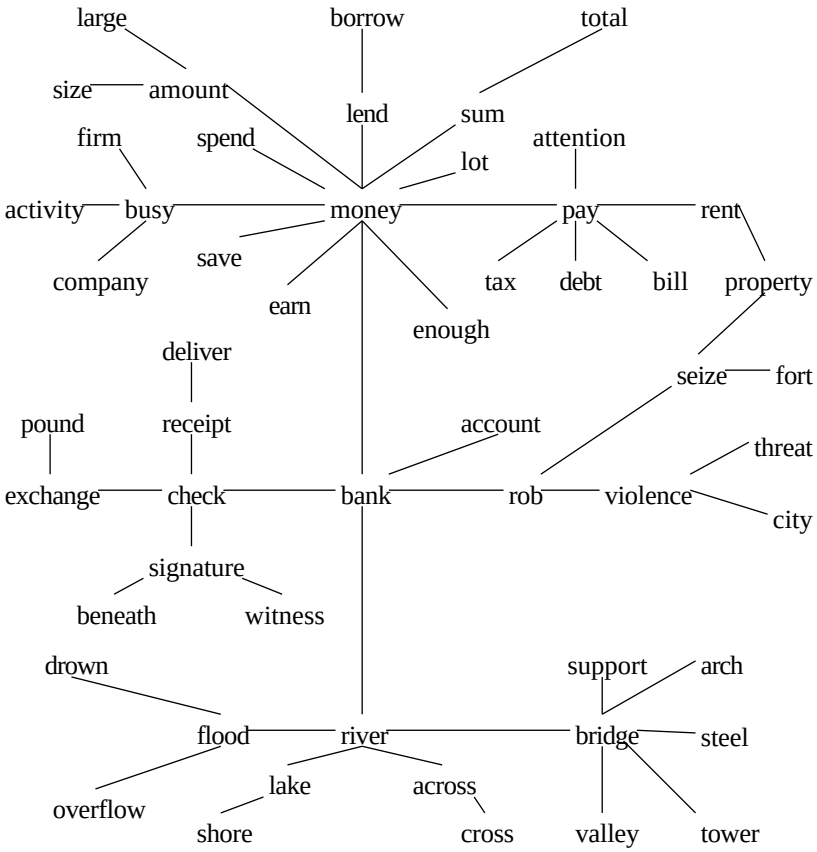


Figure 7. Commonly co-occurring words in LDOCE.

bank (n)
1 Land along the side of a river, lake.

Figure 8. Definition of a geographical sense of ‘bank’.

Like Lesk’s method, the disambiguation process Wilks used is similar to ranked retrieval: given an ambiguous word occurrence, the definitions of that word’s senses were treated as a small document collection and the context words of that occurrence were treated as a query. Each definition was assigned a score based on the number of definition words found in the context. The definitions were ranked by their score with the top scoring sense definition being selected as the correct sense.

Wilks tested the accuracy of his disambiguator on the word ‘bank’ as it appeared in around 200 sentences. The disambiguator was judged correct if it selected the same sense that Wilks had chosen when manually disambiguating the sentences. The senses of ‘bank’ are defined in LDOCE at two levels of granularity, as is shown in Figure 9. At the fine grained level LDOCE defines 13 senses for ‘bank’. Wilks reported that his system selected the correct sense of ‘bank’ 53% of the time. At the coarse level, LDOCE groups the 13 fine grained senses of

‘bank’ into five senses (I - V). Wilks reported that his system correctly selected these senses 85% of the time.

I **bank** (n)
1 Land along the side of a river, lake.
2 Earth which is heaped up in a field or garden, often making a border or division.
3 A mass of snow, clouds, mud.
4 A slope made at bends in a road or race-track, so that they are safer for cars to go round.
5 *sandbank.*

II **bank** (v)
6 Of a car or aircraft to move with one side higher than the other, when making a turn *bank up*.

III **bank** (n)
7 A row, of oar s in an ancient boat or *keys* on a *typewriter*.

IV **bank** (n)
8 A place in which money is kept and paid out on demand, and where related activities go on *street*.
9 In a place where something is held ready for use, *organic* products of human origin for medical use.
10 A person who keeps a supply of money or pieces for payment or use in a game of chance.
11 Break the bank to win all the money that the *bank* {4}3 has in a game of chance.

V **bank** (v)
12 To put or keep money in a bank.
13 To keep one's money in the stated bank.

Figure 9. LDOCE definition of ‘bank’.

Further work on this disambiguator was performed by Guthrie et al. *[Guthrie 91]* who exploited a set of subject categories assigned to many of the sense definitions in LDOCE[5]. The scope of these categories is wide ranging, from the general such as ‘automotive’ or ‘economics’, to the more focused like ‘golf’ and ‘gambling’. The word ‘bank’, defined in Figure 9, has a number of its sense definitions assigned a category: definitions 4 and 6 are assigned the ‘automotive’ category; 10 is assigned the ‘gambling’ category; and 8 is assigned ‘economics’. As can be seen here, the assignment of categories appears to be inconsistent as definition 11 should be assigned the ‘gambling’ category also.

The method of disambiguation used by Guthrie was identical to Wilks with the exception that, during the definition expansion process, a definition assigned a certain subject category was only expanded with co-occurring words present in the other definitions assigned the same category. Some of the more focused categories were assigned to so few definitions that little or

5. These classifications only appear in the machine readable version of the dictionary.

no word co-occurrence was present and expansion could not take place. As Guthrie had arranged the categories into a semantic hierarchy, it was possible to incorporate definitions of a tightly focused category into a more general category allowing the expansion process to proceed. No tests for this disambiguator were reported.

4.2.3 Disambiguating more than one word at a time

So far the corpora based disambiguators described here have been designed to work on one word occurrence at a time. Yet the sense of an occurrence is defined by the senses of the other words in its context. In their work with LDOCE, Wilks et al. *[Wilks 90]* noted that it would be desirable to disambiguate a whole sentence simultaneously. But they pointed out that, to exhaustively check every permutation of word sense assignments in a typical sentence would involve examining hundreds of thousands of sense combinations. As a solution, they suggested using the technique of *simulated annealing* which has been applied successfully to computing problems that are prone to combinatorial explosion.

This suggestion was taken up by Guthrie et al. *[Cowie 92]*. Using similar sense disambiguation techniques to their previous work (see above), they built a disambiguator that attempted to simultaneously resolve all the ambiguous words in a sentence. They tested their disambiguator on a total of 67 sentences and reported an accuracy of 47% when resolving to the LDOCE fine grain senses. Disambiguation based on the coarse grain senses was performed with an accuracy of 72%. Unfortunately they did not compare their system with their previous category based disambiguator or with the Wilks disambiguation results. Thus it is hard to decide on the merits of their technique.

Work with the same aim of simultaneously disambiguating whole sentences was undertaken by Demetriou *[Demetriou 93]*, who also used LDOCE. However, his disambiguation algorithm exhaustively checked every sense combination, thus succumbing to the explosion of possible senses to be considered. The disambiguation accuracy he reported was 58% but, as he tested on a different set of sentences from Guthrie, comparison of these disambiguators' accuracy is not possible.

Dictionary based disambiguators have never shown particularly high levels of accuracy. It is questionable therefore if this type of disambiguator will ever be anything more than an experimental system.

4.2.4 Manually tagging a corpus

Another approach to automatic disambiguation is to manually disambiguate words in a corpus and use this data to train a disambiguator. The first large scale study of this approach was

undertaken by Black *[Black 88]*. Working with LDOCE fine grained senses[6], she selected five ambiguous words that had at least three senses within the same part of speech and for each word, manually disambiguated 2,000 occurrences of that word. Black randomly partitioned each set of occurrences into a training set, consisting of 1,500 occurrences and a test set, consisting of the remaining 500. Three disambiguation strategies Black had created were applied to the five training sets. Each of the strategies exploited different features of the training sets: one method used the subject categories contained within LDOCE; the other two automatically generated sense selection rules similar to those used by Weiss. Each method's success at disambiguating each test set was measured: the subject classification method achieved 45% accuracy; the other two performed approximately the same, at 72% and 75%.

The approach of manual sense tagging to train a disambiguator has also been used by Hearst *[Hearst 91]*. Training her system for a certain ambiguous word involved manually disambiguating a number of occurrences of that word. The disambiguator gathered lexical and grammatical clues from the context of these occurrences to build up information to help discriminate between the senses of the word. Once the system had undergone this *supervised training*, Hearst tried *unsupervised training* on untagged word occurrences to try to improve the system's accuracy. Here the disambiguator would attempt to disambiguate an occurrence and gather from its context the same type of lexical and grammatical clues gathered during supervised training.

Testing was performed on occurrences of six ambiguous words that Hearst had manually disambiguated for this purpose. Results from this testing were reported for the disambiguator's accuracy after supervised training on different numbers of tagged word occurrences (maximum of 70) and for the disambiguator's accuracy after both supervised and unsupervised training. Overall, the results showed that the more training word occurrences there were, the better the disambiguation. Unsupervised training was found to work, although a sufficient number of manually tagged word occurrences was required to start up the disambiguator. Hearst reported a disambiguation accuracy ranging from 73% up to 100%, though the perfect disambiguation was for one word only.

The use of supervised and unsupervised training may yet prove to be the most effective approach to building a disambiguator. Since the completion of the experimental work of this thesis, Yarowsky has reported on a disambiguator based on these techniques which has a disambiguation accuracy of >96% *[Yarowsky 95]*. This approach has also been successful for

6. It is not clearly stated in the paper what type of word sense was used, but fine grained senses are the most likely.

parts of speech tagging systems *[Garside 87]* that perform quite similar tasks to disambiguators[7]. As the construction of such a disambiguator would be likely to require manually tagging many examples of every distinct ambiguous word to be disambiguated, this is too great an effort for one person to undertake and so, for the purposes of this thesis, this approach to disambiguation was disregarded.

4.2.5 Language translation dictionaries and multilingual corpora

In the field of machine translation, a program translates a sentence into a target language and, for each word in that sentence, the translation program is faced with a selection of candidate words to translate to. Typically this choice of translations reflects the senses of those words. The word ‘bat’ for example, can be translated into German as ‘Schlagholz’ (sporting sense) or as ‘Fledermaus’ (mammalian sense).

Dagan et al. *[Dagan 91]* built a disambiguator to improve the accuracy of that choice. The evidence used by their disambiguator was a translation dictionary - for translating from a source language into a target language - and a large corpus written in the target language. The disambiguation method used was as follows. Given a sentence, written in the source language, containing words that are translatable in a number of ways, the disambiguator generated every combination of those translated words (no mention was made of the combinatorial explosion resulting from testing all combinations). Taking each combination in turn, it examined the target language corpus counting how often that combination occurred in the corpus. The combination occurring most often was chosen as the correct translation and the senses reflected in this translation were selected.

The Hebrew sentence shown in Figure 10, for example, contains three ambiguous words which cause there to be 27 possible English translations of the sentence. To decide which of these to use, an English language corpus is examined to find the combination of translations that occur most commonly in that corpus. In this example the correct translations are ‘increases’, ‘progress’, and ‘talks’.

Dagan et al. tested their disambiguator on two language translations: German to English; and Hebrew to English. In total they attempted to disambiguate 159 word occurrences (105 Hebrew, 54 German), 54 of which were discarded because there were insufficient examples of the translations in the target language corpus. Of the remaining 105 (73 Hebrew, 32 German) a disambiguation accuracy of 75% was achieved for German/English based disambiguation and 92% for Hebrew/English. They attributed the lower accuracy of the German/English

7. Parts of speech taggers assign grammatical tags to words based on the context in which those words appear.

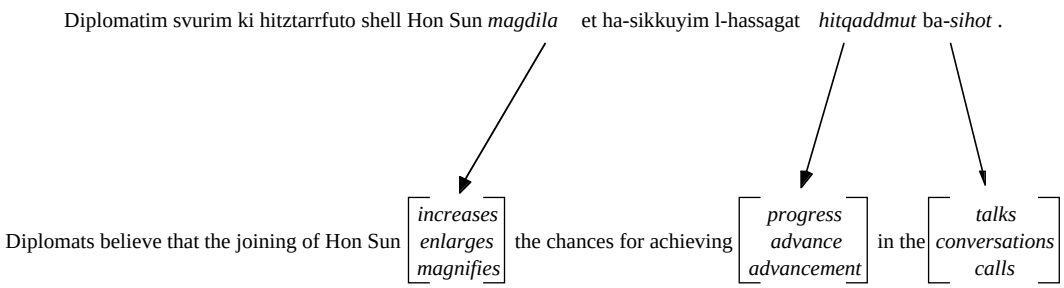


Figure 10. The possible translations of a Hebrew sentence into English.

translation to the low number of occurrences of potential translation words in the English corpus.

4.2.6 Bilingual corpora

A bilingual corpus consists of two corpora, one a translation of the other. One of the best known bilingual corpora is the Canadian Hansard 500Mb of transcriptions in French and English of the proceedings of the Canadian Parliament. After some processing, these corpora were used to train a disambiguator built by Gale et al. *[Gale 92b]*. They found that, because the two corpora were direct translations of one another, the sentence and paragraph structure of the two were almost identical. They were able, using an automated technique (outlined in *[Gale 91]*) to align with a high degree of accuracy, each sentence in one corpus with that sentence's translation in the other corpus. The resulting *aligned bilingual corpus* could then be used to discover how a word, appearing in a particular sentence, was translated into the other language. Using the same principle exploited by Dagan, that translations of a word generally reflect the senses of that word, the words of this aligned corpus could be automatically sense tagged.

The disambiguation technique devised to exploit this corpus is similar to the manual tagging techniques outlined above. When training the disambiguator for a particular word, for each sense of that word, the contexts of all the occurrences of that word tagged with that sense were gathered. The words contained in these contexts (apart from stop words) were regarded as *clue words*: disambiguation evidence for that sense of that word. Disambiguation was (again) similar to IR, the context of an ambiguous word occurrence was regarded as a query and the sets of clue words for each of the senses of the ambiguous word were regarded as a small collection of documents. Gale et al. performed limited testing on their disambiguator, processing the word 'bank', Gale et al.'s trained disambiguator achieved an accuracy of around 92%.

Gale et al.'s & Dagan's bilingual corpora based methods, while innovative, are limited in the number of senses they can resolve as they both rely on sense distinctions being reflected in

language translation. In the reporting of their disambiguators, neither author addressed this issue and therefore the extent this limitation is unclear. As there are other techniques, yet to be described, that achieve equal or higher levels of accuracy, but avoid the potential limitations of language translation, these disambiguation techniques were rejected for use in the experiments in this thesis.

4.2.7 Thesauri

A thesaurus is perhaps one of the more obvious candidates as a source of evidence for an automatic disambiguator and a number of researchers have used this type of reference work in disambiguation research. One of the most popular thesauri currently available is WordNet, *[Miller 90], [Miller 95], [WordNet]* which was compiled at Princeton University and is in the public domain. This thesaurus was designed for use in computer based work and so does not have any of the readability problems associated with some of the MRDs mentioned above. Each of WordNet's 90,000 words is assigned to one or more *synsets*. A synset is a set of words that are synonyms of each other and together, these words define the synset and its meaning. The synsets to which a particular word is assigned, constitute the individual senses of that word. These synsets are linked to form a semantic network, an example fragment of which is shown in Figure 11. As can be seen the links between synsets are formed by semantic relations, the most prevalent of which are the two complementary hierarchical relations, the *hypernym* or is-a relation (e.g. a cabin is a type of house), and the *hyponym* or instance-of relation (e.g. the class of houses has an instance of the type cabin). There are three other relations used to link synsets in the semantic network: the *meronym* or has-part/has-member relation (e.g. a house has a part attic); the *holonym* or member-of/part-of relation (e.g. an attic is a part of a house); and the *antonym* or is-opposite relation (e.g. black is the opposite of white). WordNet is composed of four such semantic networks, one for each major grammatical category: noun, verb, adjective, and adverb.

Sussna *[Sussna 93]* chose WordNet's semantic network of noun synsets as a source of evidence for a disambiguator. His aim was to use this network to enable him to calculate a *semantic distance* between any two words in the network. To achieve this, he assigned a weight to all the synset relations in the semantic network. The strength of weight assigned to a relation reflected the semantic similarity expressed by that relation. For example, the synonymy relations within a synset were assigned the highest weight, whereas antonym relations would be assigned the lowest weight. The semantic distance between two synsets could be calculated by summing the weights attached to the relations making up the shortest path between those two synsets. Sussna made no mention of the potentially expansive search required to find this path in the network.

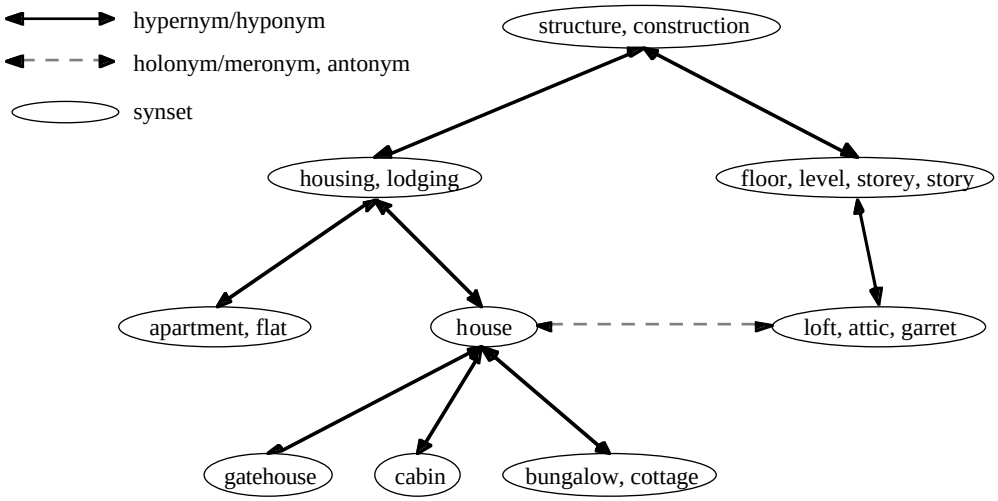


Figure 11. Fragment of the WordNet semantic network.

The disambiguation method Sussna used is similar to the methods outlined in previous sections: given an ambiguous word appearing in a certain context, all the synsets (senses) containing that word were looked up in WordNet. Each synset was given a score calculated as the sum of semantic distances between the context words and that synset. The synsets were ranked by their score, with the top ranked chosen as the sense of the ambiguous word occurrence. Sussna tried his disambiguation technique in a number of configurations. The main parameters he varied were, the size of context used when disambiguating a word and the number of words disambiguated simultaneously. When disambiguating more than one word at a time, Sussna's technique examined every sense combination and so, like Demetriou's method, encountered problems of the sense combinations increasing exponentially.

Testing was performed on ten documents taken from the TIME collection *[Virginia disc 90]*. Within these documents 319 ambiguous word occurrences were selected and manually disambiguated by Sussna. The disambiguator resolved these occurrences with an accuracy of 56%. It was reported that a context of 41 words produced the best disambiguation accuracy. In addition, simultaneously disambiguating words improved accuracy but due, to the number of sense combinations growing exponentially, the disambiguator was limited to processing concurrently no more that ten ambiguous words.

Sussna also tested the ability of humans at disambiguating word occurrences in five of the TIME documents. The disambiguation accuracy achieved by Sussna's subjects was 78%. Such a figure raises the question, if the subjects in these experiments were this poor at disambiguating, one might wonder how correct were Sussna's manual disambiguations. However, in a personal communication to the author, Sussna stated that the conditions he had allowed for himself when disambiguating were quite different from conditions in which he placed his

experimental subjects. They had been restricted to using the same evidence his automatic disambiguator had used, which was a small window of the stemmed words surrounding the ambiguous word. Sussna, on the other hand, had access to the full document in which the word occurred. Because of this difference, Sussna stated his trust in his manual disambiguation accuracy. The issue of measuring disambiguation accuracy will be returned to in a subsequent section of this chapter.

In looking for a disambiguation design to use in experiments later in this thesis it is tempting to choose a WordNet based technique because of its availability and usability. The disambiguator described here, however, is not particularly accurate. What is required is a better disambiguation method that could be implemented with WordNet, so other methods were investigated.

Using Roget's thesaurus[8] *[Kirkpatrick 88]* and the Grolier Multimedia Encyclopedia *[Grolier]*, Yarowsky *[Yarowsky 92]* built a disambiguator which is one of the most accurate to date. The disambiguator is based on the 1042 semantic categories into which all words in Roget are placed. These are broad categories covering areas like, tools-machinery or animals-insects. Figure 12 shows some of the words placed into the tools-machinery category. Yarowsky's disambiguator would attempt to resolve an ambiguous word to one of these categories. For example, it would decide if the sense of an occurrence of the word 'crane' was the tools-machinery or the animal-insect categories.

> Tool, implement, appliance, contraption, apparatus, utensil, device,
> gadget, craft, machine, engine, motor, dynamo, generator, mill, lathe,
> equipment, gear, tackle, tackling, rigging, harness, trappings, fittings,
> accoutrements, paraphernalia, equipage, outfit, appointments,
> furniture, material, plant, appurtenances, a wheel, jack, clockwork,
> wheel-work, spring, screw, turbine, wedge, flywheel, lever, bicycle,
> pinion, crank, winch, crane, capstan, windlass, pulley, hammer,
> mallet, mattock, mall, bat, racket, sledge hammer, mace, club,
> truncheon, pole, staff, bill, crow, crowbar, pole axe, handspike,
> crutch, boom, bar, pitchfork, …

Figure 12. Some words placed into the tools-machinery category.

In acquiring evidence to decide which semantic category (sense) an ambiguous word occurrence should be assigned, a set of clue words - one set for each semantic category - was derived from a grammatically tagged Grolier Encyclopedia. To derive one of these clue word sets for a category, every occurrence of every word in that category was looked up in Grolier and the context of each occurrence (the 100 words surrounding that occurrence) was gathered.

8. Note this is not the 1911 version of Roget's thesaurus available in the public domain but a recent and more extensive version obtained from its publishers through special agreement.

For example: the category tools-machinery contains 348 words which occurred 30,924 times within Grolier. The contexts of each of these occurrences were gathered. Figure 13, taken from Yarowsky's paper, shows a small sample of these occurrences with part of their context.

CARVING .SB The gutter	**adz** has a concave blade for form
uipment such as a hydraulic	**shovel** capable of lifting 26 cubic
on .SR Resembling a power	**shovel** mounted on a floating hul
uipment, valves for nuclear	**generators**, oil-refinery turbines
00 BC, flint edged wooden	**sickles** were used to gather wild
1-penetrating carbide-tipped	**drills** forced manufacturers to fi
ent heightens the colors .SB	**Drills** live in the forests of equa
traditional ABC method and	**drill** were unchanged, and dissa
nter of rotation .PP A tower	**crane** is an assembly of fabricat
rshy areas .SB The crowned	**crane**, however, occasionally

Figure 13. Contexts of tools-machinery words taken from Grolier. Sentence and paragraph boundaries are labelled.

It is from these contexts that the clue words are selected. A process similar to relevance feedback is used (the selection of words from a set of documents). For each context word, its frequency of occurrence within all contexts is compared to its frequency of occurrence within Grolier encyclopaedia as a whole. All of the context words are assigned a score based on this comparison of frequencies. The highest scoring words are used as the clue words for their semantic category. Yarowsky reports deriving around 3,000 clue words for each category. Some of these words selected for the two categories tools-machinery and animal-insect are shown in Figure 14. Note as Yarowsky stated in his paper:

> …these are not a list of members of the category; they are the words which (sic) are likely to co-occur with the members of the category.

tools-machinery
tool, machine, engine, blade, cut, saw, lever, pump, device, gear, knife, wheel, shaft, wood, tooth, piston, …

animal-insect
species, family, bird, fish, breed, cm, animal, tail, egg, wild, common, coat, female, inhabit, eat, nest, …

Figure 14. Some of the clue words derived for two semantic categories.

In testing, Yarowsky trained his disambiguator for 12 ambiguous words. Several hundred occurrences of each of these words were manually disambiguated. The accuracy of the disambiguator varied, but on average it resolved word senses with an accuracy of 92%. The test words were selected because they had been used in other disambiguation research and, there-

fore, some comparison was possible between this and other work. However, comparisons were not exact because none of the other researchers had tried to disambiguate using the Roget definitions of word sense.

4.2.8 Summary

The aim of this review of disambiguation techniques has been to show the wide range of techniques used by researchers and an indication of which technique is deemed most appropriate for the later experiments. A fuller discussion on which of these techniques is best suited for use in an IR system will be addressed in Chapter 6.

4.3 Testing a disambiguator

If there is one theme to be drawn from the disambiguation research presented so far, it is that testing a disambiguator is problematic. There are almost no standard 'pre-disambiguated' corpus available, so researchers are often faced with the time consuming task of manually disambiguating all the occurrences of the words to be tested. It is generally not possible to share disambiguated text between research projects because it is likely that the definitions of word sense each project uses will differ. For example, WordNet defines 15 senses of 'bank' where as LDOCE defines 13 and there is no clear correlation between them. When reviewing the research in this chapter, the lack of common sense definitions makes it difficult to compare the accuracy of different disambiguators.

Even if a research project has the resources to manually disambiguate a corpus for the purposes of testing a disambiguator's accuracy, the manual identification of word senses is not a simple matter. To illustrate, when working with the LDOCE fine grained senses, Wilks et al. *[Wilks 90]* found that sometimes no sense definition correctly described the sense of word occurrences they were manually disambiguating. This observation was later supported in a study performed by Kilgarriff *[Kilgarriff 91]* who, using a set of 83 words, tried to assign a single fine grained LDOCE sense to each word as it appeared in a number of contexts. Kilgarriff judged that 60 of the words had at least one occurrence that was not described by a single LDOCE sense. He was not any more specific about his results however, because he did not consider his study large enough to be statistically significant.

The topic of word senses, as defined in dictionaries, was examined by Jorgensen *[Jorgensen 90]*. She selected a number of ambiguous words and for each word, extracted a set of sentences containing that word from a corpus. She then asked a number of subjects to partition each set of sentences into clusters, so that the sentences in each cluster would contain the same word sense. One week later she asked the same subjects to repeat the task, but this time

showed them dictionary definitions for each ambiguous word. She found that the agreement between the subject's two partitions was 68%.

Jorgensen's results and the previous examples would seem to illustrate that the manual disambiguation of an ambiguous word is by no means an easy task. This perhaps should come as no surprise as it is not hard to believe that the possible senses of a word are merely ill defined areas within a continuum of meaning for that word. Any attempt to assign one of a set of cleanly defined sense categories to a particular word occurrence is inevitably a somewhat artificial task.

4.3.1 Discussion

The fact that manual sense disambiguation is not a simple matter is further compounded by the related issue of the consistency in disambiguation across people. Although work in manual disambiguation consistency is new, the signs are that it is not high. To demonstrate this point Ahlswede et al. *[Ahlswede 93]* compared the output of ten different disambiguators: seven humans; two well known disambiguation algorithms; and an algorithm that randomly selected senses. Ahlswede applied these ten disambiguators to the same ambiguous text - he did not specify the size or nature of this text - and compared their output. The results of this comparison showed on average 66% agreement between the disambiguators. At the time of writing this thesis he is conducting a much larger study involving 100 people.

Inevitably Ahlswede's research into manual disambiguation calls into question the reported accuracy of the disambiguators reviewed here, because the accuracy of all disambiguators was benchmarked against the output of manual disambiguators. Often, little is said by researchers on the method of testing their disambiguators and sometimes it is not possible to tell if the manual disambiguations were performed by one person or many, or if there was any cross checking of the disambiguation. The reported accuracy of disambiguators may be affected by the unreliability of manual disambiguation where disambiguation research resolves to fine grained senses. Anyone contemplating the measurement of a disambiguator's accuracy should be aware of such problems.

To address the problem of evaluating disambiguation accuracy, Yarowsky *[Yarowsky 93]* reported a novel technique that is completely automatic. The method involved the introduction to a corpus, of artificially created ambiguous words, called *pseudo-words*. The creation of such a word was performed by replacing all occurrences of two words, for example 'banana' and 'kalashnikov', by a new ambiguous pseudo-word 'banana/kalashnikov'. The source of evidence used by the disambiguator being tested, was adjusted to reflect the union of

the two words and the disambiguator was applied to each occurrence of this new word. Evaluation of the disambiguator's output was a trivial matter because it was known beforehand the correct *pseudo-sense* of each occurrence of the pseudo-word. Pseudo-words form the basis of one of the experiments presented in this thesis and they are discussed in detail in Section 5.1.

4.4 Word sense disambiguation and IR

As outlined in the previous chapter, information retrieval researchers have often looked to natural language processing for techniques that might improve the effectiveness of an IR system and word sense disambiguation may be one such technique. One of the first mentions of actually using a disambiguator to try to improve the representation of a document collection was made by Weiss *[Weiss 73]*. He reported that experiments using the SMART IR system *[Salton 83]* had shown that resolving all ambiguous words in a document collection would only result in a 1% improvement in retrieval effectiveness. As Weiss does not describe this experiment in detail, one can only speculate on the reason for this small improvement.

When thinking about WSD and IR one can imagine two scenarios that to some extent conflict with each other.

- If we imagine having a disambiguator capable of accurately disambiguating a document collection and a retrieval system capable of working with such a collection, it is easy to believe that retrieval effectiveness would improve due to this more sophisticated document representation.

- It is also easy to imagine a scenario where disambiguation is unlikely to be of use. If a user were to enter a query of many semantically similar words, for example 'bank economic financial monetary fiscal' then, for any document containing all five words, it is unlikely that the occurrence of 'bank' in that document will refer to the margin of a river. Therefore, disambiguation of this document is not necessary, due to the self disambiguating nature of the large query[9].

These ideas on IR and WSD have, to a certain extent, been confirmed by the work of Krovetz and Croft *[Krovetz 92]* who have conducted some of the most extensive research on ambiguity and IR to date. Using two traditional IR test collections, CACM *[Virginia disc 90]* and TIME, they performed a retrieval for each of the standard queries in these collections. For each

9. From this example, we see that relevance feedback can be thought of as a form of manual disambiguation: if a user enters a short ambiguous query, performs a retrieval, and then chooses a number of documents he or she considers to be relevant, the sense of each original query word contained in those documents is likely to be similar to the sense intended by the user.

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 query document word sense matches or mismatches. They found that, when the document was not relevant to the query a sense mismatch was more likely to occur and, that sense mismatches occurred more often when there were a small number of query words in the document. They concluded that the impact of sense ambiguity on retrieval effectiveness was not dramatic, but that disambiguating ambiguous words would probably improve retrieval effectiveness when there were few query words occurring in the document.

The first experiment using a corpus based disambiguator with an IR system was by Zernik *[Zernik 91]*. He disambiguated the occurrences of 20 words within a corpus, performed a retrieval and examined the change in what he called 'retrieval accuracy': presumably some form of precision based evaluation measure. When retrieving with a query composed of 30 words, Zernik reported no change in 'retrieval accuracy'. For retrievals based on a one word query, however, Zernik stated that 'accuracy [was] improved by up to 50%'. This is further evidence supporting Krovetz and Croft's conclusions.

When seeking the first large scale experiments where a corpus based disambiguator is applied to a document collection for subsequent use by an IR system, we turn to two separate pieces of research which were carried out concurrently: namely the work of Voorhees *[Voorhees 93]* and the work of Wallis *[Wallis 93]*.

Voorhees built a sense disambiguator based on the WordNet thesaurus's synset network of nouns. To disambiguate a word appearing in a certain context, the synsets of that word were ranked according to a score based on the number of words co-occurring between the ambiguous word's context and the *hood* of the synset being scored. The definition of a hood is explained in detail in Section 6.1.2. Unfortunately, no testing of this disambiguator's accuracy was performed. Using a modified version of the SMART IR system, she compared retrieval effectiveness on a disambiguated test collection, against the effectiveness on that collection in its original ambiguous state. The collections she ran these tests on were CACM, CISI, CRANFIELD 1400, MEDLINE, and TIME (*[Virginia disc 90]*). For each of these collections, retrieval effectiveness was found to be consistently worse when retrieving from the disambiguated collections. Voorhees reported that her experiments were hampered by deficiencies in the test collections themselves. For a number of the shorter test collection queries, she found it impossible to determine the intended sense of the words within those queries. The major deficiency in Voorhees work, however, was the lack of testing of her disambigua-

tor's accuracy. Without this information it is hard to assess what caused the drop in retrieval effectiveness.

Wallis used a disambiguator, based on the design of Wilks et al. *[Wilks 90]*, as part of a more elaborate experiment that represented the words of a document collection by the text of their dictionary definitions. This was done so that synonymous words, which it was hoped would have similar dictionary definitions, would be represented in a similar manner and therefore documents containing synonymous words would be retrieved together. In his paper, Wallis illustrated this representation method using the example words 'ocean' and 'sea': the definitions of which are shown in Figure 15. As can be seen, these synonymous words do indeed share a number of words in their definitions and so for these two words at least, Wallis' representation method would be of benefit.

ocean (n)
1 The *great* mass of *salt water* that *covers* most of the *earth*.

sea (n)
1 The *great* body of *salt*y *water* that *covers* much of the *earth*'s surface

Figure 15. Definitions of two synonymous words.

When replacing a word occurrence by the text of its definition, if the word was ambiguous a number of definitions would exist, one for each sense of that word. In these cases a disambiguator was used to select the definition that best defined the sense of the ambiguous word occurrence. Wallis performed tests on the CACM and TIME collections, but found no significant improvement in retrieval effectiveness when using his technique. More recently Richardson and Smeaton *[Richardson 95]* attempted a similar approach using the representation of words in WordNet. Unfortunately their attempt was less successful than Wallis as they reported a drop in effectiveness.

It is clear from the results of Voorhees, Wallis, and Richardson et al. that using a disambiguator to improve the effectiveness of an IR system it is not a simple matter. It was decided that before any attempt was made to use a disambiguator with an IR system, a greater understanding of the issues involved in WSD and IR was required. For example, how much retrieval effectiveness is affected by ambiguity and, perhaps more crucially, what effect disambiguation might have on effectiveness. These questions will be addressed by the set of experiments described in the next chapter.

5 Retrieving from an additionally ambiguous collection

The original intention of the research in this thesis was as follows: build a disambiguator; apply it to the words of a test collection; perform retrieval experiments; and examine the resulting retrieval effectiveness. In the light of the work of Wallis and of Voorhees (who reported little effectiveness change and a drop in effectiveness respectively) it was clear that a greater understanding of the relationship between word sense ambiguity, disambiguation accuracy, and IR effectiveness was required before the originally intended research was to be attempted.

Using a technique that introduces additional sense ambiguity into a test collection, this chapter presents an experiment that goes beyond previous work to reveal the influence that ambiguity and disambiguation accuracy have on the effectiveness of an IR system. This chapter is an expansion of work previously presented by the author *[Sanderson 94]*.

The chapter's structure is as follows. First, a brief reminder of the workings of the technique used to introduce additional ambiguity is presented along with a discussion of the validity of this technique. In amongst this section is also an outline of the experimental method. The main components of the experiment are described: namely the IR system; and the document collection. As the collection used is an unconventional choice for IR experiments, its usage is described in some detail. Following on from this, the experimental results are presented along with an analysis and detailed discussion that first attempts to explain the results and then compares them with a potentially contrasting set of results published by Schütze and Pedersen *[Schütze 95]*. Finally, the implications of the results from the experiment are discussed.

5.1 Pseudo-words

In order to undertake an experiment of this kind, it is necessary to have a document collection composed of words that have been disambiguated. However as Krovetz and Croft found (reviewed in Section 4.4), a significant amount of effort is required to manually disambiguate thousands of occurrences of ambiguous words. Although such an effort can produce accurate results, it is a time consuming exercise. As a consequence, this restricts the number of occurrences that one can process. So for the experiments in this chapter an alternative method was sought.

Yarowsky's novel technique of using pseudo-words (artificially created ambiguous words) to evaluate a disambiguator, is completely automatic. Although invented solely for the purpose of evaluating disambiguators, it became apparent that these words could be used as substitute ambiguous words in a retrieval experiment. An investigation of the effects of ambiguity on

retrieval effectiveness would be performed as follows. First, the effectiveness of an IR system retrieving from a test collection would be noted. Then, ambiguity would be introduced into the collection using pseudo-words. The effectiveness of the system retrieving from this additionally ambiguous collection would be compared to the effectiveness figures gained from the initial retrieval. The difference in effectiveness would be an indication of the impact of ambiguity on an IR system. Pseudo-words would allow the experimenter to vary, at will, the amount of additional ambiguity in a collection. Levels of ambiguity that far exceed the levels in standard test collections could be studied. In addition, as pseudo-words are automatically generated, there are no 'manual overheads' when applying experiments to other test collections.

The primary advantage of using pseudo-words is that the correct pseudo-sense of every word is known, therefore, one can simulate the effects of a disambiguator by restoring these words to their original state. In itself, not a particularly fruitful procedure, however, as was shown in the review of disambiguators (Section 4.2), disambiguation is an erroneous process and this error can be simulated by occasionally restoring the pseudo-words to an incorrect pseudo-sense. This allows one to measure the influence on retrieval effectiveness of a (simulated) disambiguator operating at experimentally controlled levels of accuracy.

Given these controllable and flexible properties, pseudo-words were prime candidates to be used in the experimental manner just described. Nevertheless, it was necessary to ensure that these artificially ambiguous words were as realistic a simulation of ambiguity as possible.

5.1.1 The realism of pseudo-word ambiguity

Three attributes of ambiguous words were identified and pseudo-words were analysed to examine how realistically these attributes are simulated. This analysis is now presented.

Relatedness of sense

The method chosen to form pseudo-words from individual words is one of random selection. Using such a method, it is likely that the various pseudo-senses of a resulting pseudo-word will not be related. This differs from a proportion of actual ambiguous words whose senses are related in some manner. To illustrate, the word 'surf' can be used to refer to the physical action of surfing ocean waves, but equally it is used metaphorically to refer to the casual browsing of information. Clearly these senses are related in a manner not simulated by pseudo-words formed by random word selection.

How important the unrelatedness of pseudo-senses is, can perhaps be determined by how often the relatedness of an ambiguous word's senses is exploited by authors. It can be

expected that humorous prose will use related senses, but it was felt that this literary device is not used much in general writing. Therefore, it was believed that this failing in pseudo-words would not have a significant impact on the IR experiments presented in this chapter.

Context of senses

Given that the random conflation of two words results in a pseudo-word whose two pseudo-senses are unrelated, it also follows that they will be used in different contexts. This aspect of pseudo-words was examined in ambiguous words. It quickly became apparent that there are many examples of ambiguous words whose senses appear in very different contexts. For example it is easy to imagine the contexts that the two senses of 'surf' appear in, will be different. In fact most corpus based disambiguators (as described in Section 4.2) assume that each sense of a word will be surrounded by a unique, wide (40-100 words), context. The Yarowsky disambiguator uses this unique context assumption and has a reported disambiguation accuracy of about 90%. Therefore, it was concluded that having pseudo-words whose pseudo-senses are used in different contexts makes for a more realistic simulation of ambiguity.

Frequency of occurrence of the senses of a word

The final attribute of ambiguous words to be considered here, is the distribution of the frequency of occurrence of an ambiguous word's senses: for example, are the senses of an ambiguous word used equally often; is one sense used to the exclusion of others; or is there some other form of distribution? An experiment was conducted to measure this distribution in both ambiguous words and pseudo-words. As this comparison provides an insight into the results of the main experiments to be presented in this chapter, the details of this comparison are presented later (Section 5.6.1). The result of the comparison, however, was to show that the distribution of the frequency of occurrence of the senses of both pseudo-words and ambiguous words was the same, and it was concluded that pseudo-words simulate this aspect of ambiguity realistically.

Summary

In indentifying three attributes of ambiguous words, it has been established here that pseudo-words simulate two of the three realistically. Of the other attribute, relatedness of sense, it is believed that the inability to simulate this with pseudo-words will not significantly impact on the results of the IR experiments.

5.2 The retrieval system

The retrieval system used in all the experiments presented in this thesis is one developed by the author. The indexing and retrieval modules of this system are based on the probabilistic retrieval model first proposed by Robertson and Sparck Jones *[Robertson 76]*. For the experiments to be presented here, the indexing module of the system used a *binary occurrence weighting model* (i.e. assigning an *idf* weight to the terms of a document collection). It is generally accepted however, that retrieval systems adopting this type of weighting scheme have a lower effectiveness than those using *within document frequency weighting* (i.e. a *tf•idf* weight), and perhaps with hindsight the choice of weighting model was not the best. Nevertheless, the aim of these experiments is not to produce results to be contrasted with others, rather it is to compare the effectiveness of the same system indexing and retrieving from different versions of a test collection. There is no evidence to suggest that the results of these experiments were unduly influenced by adopting the binary occurrence weighting model.

5.3 The test collection

The collection used for the experiments was ‘Reuters 22,173’, which was created for testing a text categorisation system *[Hayes 90]*, and was later modified by Lewis *[Lewis D.D. 91]* for use as a IR test collection. It consists of 22,173 documents taken from the Reuters newswire service.

It was chosen for use in the experiments of this chapter in preference to the standard IR test collections (CACM, CRANFIELD 1400, LISA, TIME, NPL etc.) as it is at least an order of magnitude larger than them, both in number of documents and in overall size (20Mb). An additional consideration in choosing this collection was that the usage of English in Reuters was felt to be less specialised than that of many of the other test collections. This was considered to be an important factor for the later disambiguation experiments (in Chapter 8.3) that use an English language reference work that contains mainly standard uses of words.

5.3.1 Using Reuters as an IR test collection

The main difference between Reuters and conventional IR test collections is that it does not have a set of standard queries with a corresponding set of relevant documents. Each document in Reuters is tagged with a number of manually assigned subject codes. It is these codes that allows Reuters to be used as a test collection for comparing document representation methods. Lewis was the first to suggest using the collection in this manner and it is his method, with some modifications, that is described here.

R was defined as the set of all documents in the Reuters collection. This set was then partitioned into two subsets of equal size: *Q* the query set, and *T* the test set. The partitioning method was a random assignment of documents into one of the two subsets[10]. This ensured that groups of documents covering common themes were evenly distributed between both *Q* and *T*. *S* was defined as the set of all subject codes that were assigned to at least one document in *Q* and at least one document in *T*. In all, there were 111 subject codes in *S*. With the three sets, *Q*, *T*, & *S*, Reuters could be used as a test collection, an example of how this was done is now illustrated.

A retrieval took place for each subject code in *S*, for this example the code 'acq' (i.e. acquisitions) is used. First, all documents in *Q* tagged with 'acq' were identified. Then, by performing query expansion by relevance feedback based on the identified documents, term/weight pairs were selected from these documents to form a query[11]. (The size of the query could be varied, a typical size was ten terms.) This query was used to retrieve from the documents in set *T*. The ranked document list resulting from this retrieval was examined to see where in the ranking, documents tagged with 'acq' appeared. These tagged documents were regarded as relevant to the query and their position in the ranking was used to produce a set of recall/precision figures[12]. This retrieval process was repeated for all subject codes in *S*, resulting in one set of recall/precision figures for each code. These were then averaged to provide a single set of figures that measured retrieval effectiveness.

In summary, by using Reuters in this manner, all the components of a classic IR test collection were created:

- the collection to be searched - *T*;
- a set of queries - generated through term selection from *Q*, for each member of *S*;
- a set of relevant documents for each query (i.e. subject code) - documents in *T* tagged with the respective element of *S*.

10. Lewis, who was interested in studying a news wire filtering system, partitioned the collection chronologically: all documents written before a certain date were placed in the query set (*Q*); all the written after that date were placed in the test set (*T*).
11. 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.
12. A pessimistic interpolation technique (outlined in *[Van Rijsbergen 79]*) is used to transform these figures into precision values at ten recall levels (0.1, 0.2, …, 0.9, 1.0). The common practice of using eleven recall levels was rejected for reasons outlined in Section 5.3.3.

5.3.2 Cleaning the collection

Before Reuters could be used for experimentation, some preprocessing and cleaning of the collection was necessary. In his thesis, Lewis describes a series of steps he went through to remove errors from the Reuters collection before using it in his experiments. These errors were mostly of a syntactic nature such as incorrectly formatted subject codes. Unfortunately, the collection made available *[Reuters]* still had many of these errors present[13] and they had to be corrected.

During this cleaning process it was noted that a few articles within the collection were duplicated. As the collection is an archive of a newswire service, this duplication may be due to articles being resent several hours after they were first relayed. Subsequent to the main experiments described in this chapter, an experiment was performed to estimate the number of duplicate articles within Reuters. The results and conclusions of this are reported in Appendix A.

5.3.3 Data reduction

As discussed in Section 4.4, it was anticipated that the influence of ambiguity on the retrieval effectiveness of an IR system will be dependent on the size of query. Therefore it was decided that query size should be a parameter of the experiments. As the queries for the Reuters collection are generated by relevance feedback, it is possible to vary the number of the terms generated from relevance feedback and thus vary the size of the query. This means, however, that experiments involving a varying query size will produce results that are expressed in three variables: recall (r), precision (p), and the size of query (s, count of the number of query terms). The results of a series of retrievals is plotted on a three-dimensional graph as shown in Figure 16. From the graph we can see that for all recall levels, the precision is low at s=1, with a rapid rise peaking at around s=5, before falling away as s increases. It was found however that three-dimensional graphs are difficult to read when the results of several retrieval experiments were plotted together. What was needed was a two dimensional plot of s against a variable expressing retrieval effectiveness, in other words reduce each set of RP figures to a single number. What follows is a short discussion of a method that was considered and rejected for the data reduction. This is then followed by a description of the method chosen. For more detailed reading on this subject the reader is directed to chapter seven of Van Rijsbergen's book *[Van Rijsbergen 79]*.

13. It would appear that the collection made available by Lewis was not the cleaned version described in his thesis but an earlier partly processed version. This has now been rectified and those wishing to use the Reuters collection can now obtain a cleaned version.

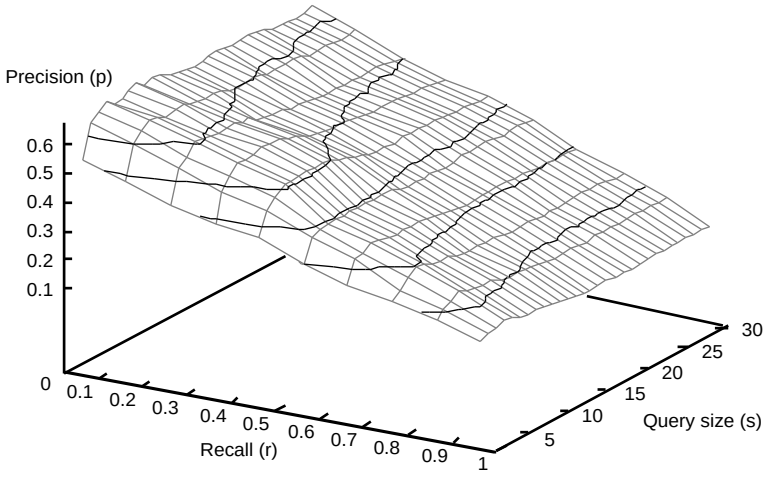


Figure 16. Plot of precision, recall, and query size. The contours show levels of precision.

Average precision

A commonly used reduction method is to average the precision values measured at each standard recall value. Calculating such an average produces a value that is directly proportional to the area under a RP graph.

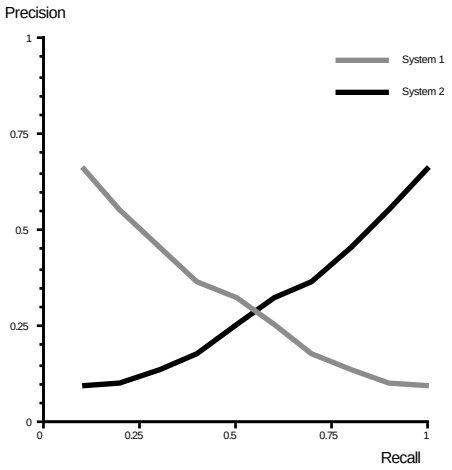


Figure 17. Two RP graphs with the same average precision.

Although it is possible to imagine two very different RP graphs that, when compared using average precision, would appear to be identical (see Figure 17), these extreme cases are very unlikely. This is because RP graphs from an IR system will almost always take the form of a monotonically decreasing line. Because of this, average precision is reasonably effective for summarising the difference between two such lines. It does, however, have two disadvantages.

- Each precision value is given equal weight when calculating the average, despite the fact that a change in retrieval effectiveness can cause different magnitudes of change in precision depending on what value of recall the precision is measured at. For example, suppose that within a document collection there are ten documents relevant to a given query. Two retrieval systems being tested on this collection each produce a document ranking, the top portions of which are shown in Figure 18. As we can see, two relevant documents appear in each ranking. System two, however, is slightly better than system one as the relevant documents within it are each one rank position higher.

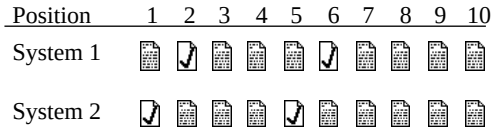


Figure 18. Top ten documents from two rankings. Ticks indicate relevant documents.

If we measure precision at recall=0.1 (the first relevant document), for system 1 precision is 0.5 but for system 2 it is 1.0. If, however, we measure precision at recall=0.2, for system 1 it is 0.33 and for system 2 it is 0.4, a smaller difference. From this example we can see that small changes in the top part of a document ranking can cause large changes in precision[14]. The effect lessens, however, as we move down the document ranking, which means that changes in precision measured at low recall are less significant than similar changes in precision measured at higher recall.

- The second disadvantage can be illustrated in Figure 19 which shows the RP lines of two retrieval systems: system 1 and system 2. The average precision of the two systems is the same. Nevertheless, some users might prefer one system over the other, for example system 1 over system 2 because of its better precision at high recall. There is no established method of adjusting average precision to accommodate such a preference.

From this description of problems with average precision, we can see that to improve on it, an alternative method should have the following two features. First, that it should regard RP points with a level of importance that reflects more accurately each point's impact on retrieval effectiveness. Secondly, that it should be possible to adjust this method to reflect a user's preferences on the importance of recall against precision.

14. The common practice of using interpolation methods to produce a measure of precision at recall value 0 is likely to be influenced strongly by this effect and for this reason, its usefulness as part of an effectiveness measure is questionable.

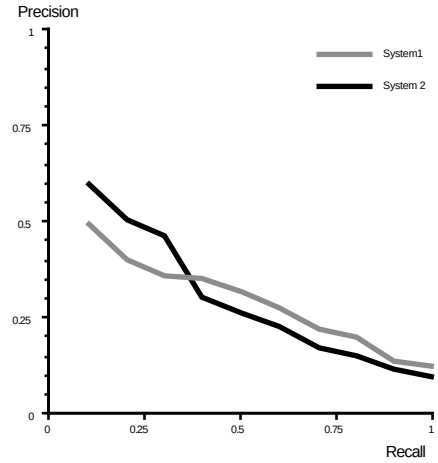


Figure 19. Two RP graphs with the same average precision.

The *f* measure

One such measure that has the features highlighted above is the *f* measure created by Van Rijsbergen *[Van Rijsbergen 79]*. It is defined in Equation 5.

$$f = \frac{1}{\alpha\left(\frac{1}{p}\right) + (1-\alpha)\left(\frac{1}{r}\right)} \qquad (5)$$

$$p = \text{Precision} \quad r = \text{Recall} \qquad 0 \le \alpha \le 1$$

The aim of this measure is to produce a number that indicates the effectiveness of the retrieval system for a given pair of recall and precision values. This measure has the range [0..1], where 1 indicates the highest retrieval effectiveness. The variable α is set to indicate a preference for either precision or recall. It is common for α to be set to 0.5 to indicate equal importance for both. Note that in his book Van Rijsbergen betrays his preference for distance functions by discussing a metric called *e* which is the inverse of *f*.

$$f = 1 - e \qquad (6)$$

To illustrate the relationship between *f* and recall & precision, Figure 20 shows the values of recall and precision that solve Equation 5 when *f*=0.2, 0.3, 0.4, 0.5, 0.75, 1.0, and α=0.5. Plotted over this is a set of RP figures taken from an IR experiment (shown in black). These figures along with corresponding *f* measures are tabulated in Table 5.

The *f* measure addresses the problem of changes in precision having a variable impact on retrieval effectiveness. As can be seen in the graph: an improvement in precision at high recall will result in a larger increase in *f* than if the same precision improvement were to happen at lower recall. It is hoped that this characteristic of *f* closely resembles most people's notion of

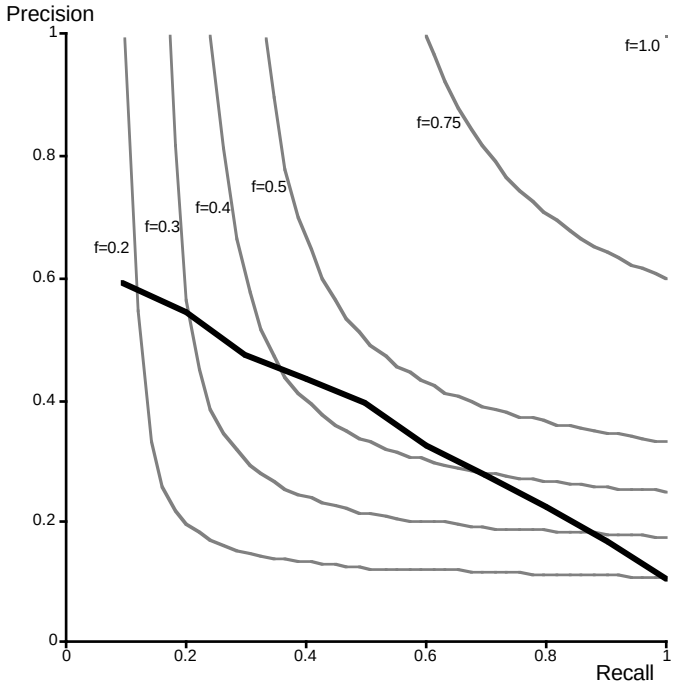


Figure 20. The relationship of *f* to recall and precision, α=0.5.

r	*p*	*f*
0.10	0.59	0.17
0.20	0.54	0.29
0.30	0.47	0.37
0.40	0.43	0.42
0.50	0.40	0.44
0.60	0.33	0.42
0.70	0.28	0.40
0.80	0.22	0.35
0.90	0.17	0.28
1.00	0.11	0.19

Table 5. Tabulation of RP figures graphed in Figure 20.

retrieval effectiveness. From the graph we can see that for the plotted RP graph, the highest *f* measures are for those points that have the best balance between recall and precision.

As already mentioned, the variable α can be adjusted to alter the preference given to precision or recall. The graphs in Figure 21 & Figure 22 show this by plotting *f* with α=0.125 (emphasising recall) and α=0.875 (emphasising precision) respectively. Tables 6 & 7 show the *f* measures for the same set of RP figures used above.

Because of the monotonically decreasing values of precision in a set of RP figures, the position of the maximum *f* measure, f_{max}, tends to change when α is varied: when α<0.5, f_{max} is generally found at high recall values; and when α>0.5, f_{max} is generally found at low recall. So

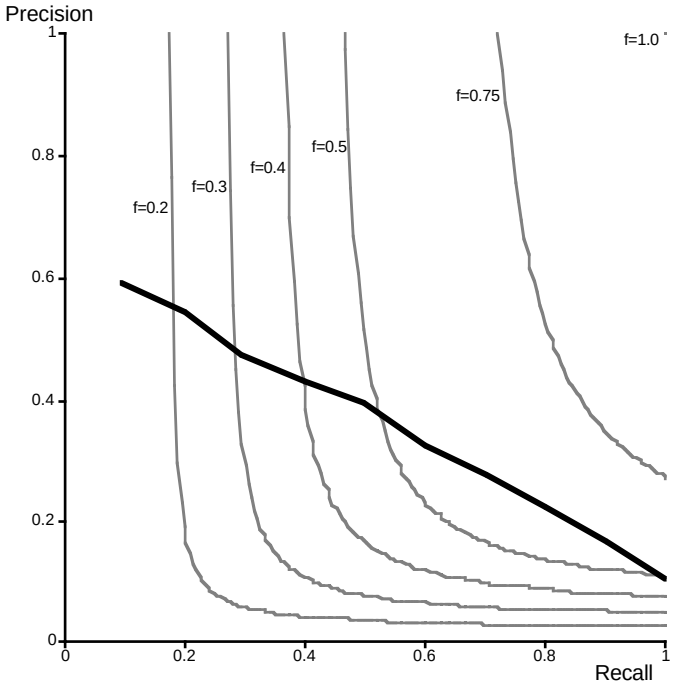


Figure 21. The relationship of *f* to recall and precision, α=0.125.

r	*p*	*f*
0.10	0.59	0.11
0.20	0.54	0.22
0.30	0.47	0.31
0.40	0.43	0.40
0.50	0.40	0.48
0.60	0.33	0.54
0.70	0.28	0.59
0.80	0.22	0.61
0.90	0.17	0.58
1.00	0.11	0.49

Table 6. Tabulation of RP figures graphed in Figure 21.

in effect by varying α we can vary the influence different parts of a RP line have on the resulting set of *f* measures.

5.3.4 Reducing the set of *f* measures

By using *f*, a set of RP figures is reduced to ten *f* measures. A further reduction to a single value is required. One could take the average of the ten measures but, as can be seen in Figure 23, the distribution of a typical set of *f* measures (taken from the graph in Figure 20) is not a normal distribution. Average and standard deviation, are only meaningful when applied to that type of distribution, so an alternative statistic was sought.

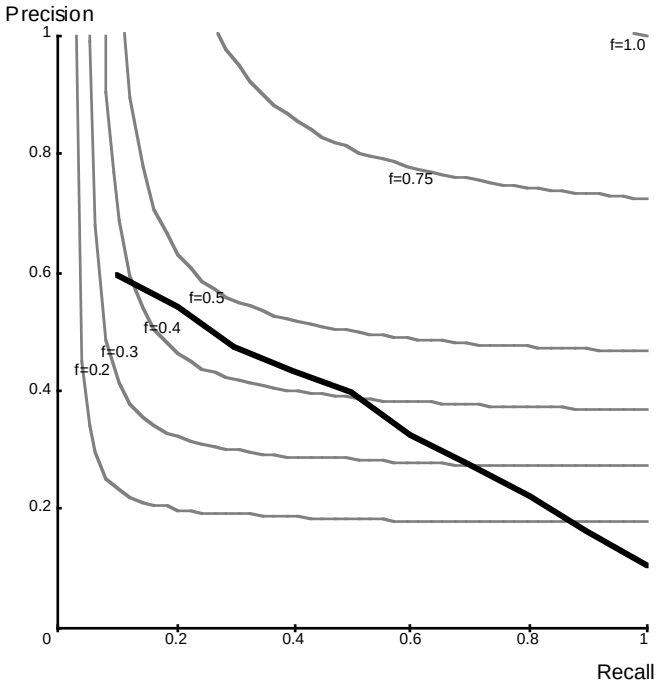


Figure 22. The relationship of *f* to recall and precision, α=0.875.

r	*p*	*f*
0.10	0.59	0.37
0.20	0.54	0.45
0.30	0.47	0.44
0.40	0.43	0.43
0.50	0.40	0.41
0.60	0.33	0.35
0.70	0.28	0.30
0.80	0.22	0.25
0.90	0.17	0.18
1.00	0.11	0.12

Table 7. Tabulation of RP figures graphed in Figure 22.

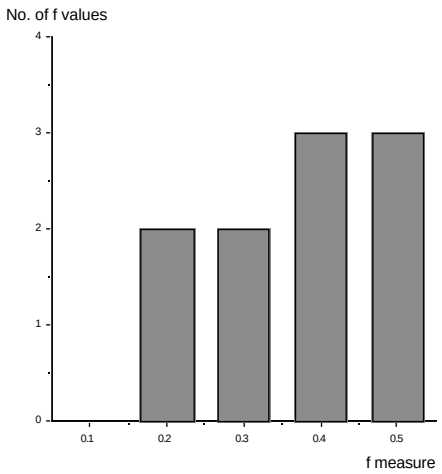


Figure 23. Discrete distribution of a set of f measures.

The aim was to produce a statistic with the same properties as average and standard deviation: i.e. a number indicating where the main concentration of measures is; and a number indicating the spread from that concentration. On inspection of the *f* measures of many sets of RP figures, it was apparent that the main concentration of measures occurred near to f_{max}, and consequently it was chosen as the primary statistic. To compute the spread of measures from f_{max}, a statistic called σ_{max} was devised (Equation 7). Its method of calculation is similar to that of standard deviation except that differences are measured with respect to f_{max}.

$$\sigma_{max} = \sqrt{\frac{\sum_{i \in N} (f_{max} - f_i)^2}{|N|}} \qquad (7)$$

$$N = \text{set of f measures}$$

However σ_{max} has a different characteristic to that of standard deviation: as f_{max} increases, it is to be expected that σ_{max} will increase also. To explain this, an example is used. Figure 24 and the accompanying Table 8 show the RP figures of two IR systems. The values of f_{max} for these two systems, 0.49 against 0.40, reflects the superiority of system 1 over system 2. Despite a large difference in precision between the two systems at all values of recall, the *f* measures computed for recall values 0.1 and 0.2 are almost the same (values highlighted in the table). This similarity is due to the behaviour of the *f* function at low recall. Because of this behaviour, the lowest values of *f* are almost constant. Therefore if f_{max} increases, σ_{max}, the spread of values from f_{max}, inevitably increases. This can be seen when measuring the effectiveness of the two example systems: system 1, f_{max}=0.49 & σ_{max}=0.15; system 2, f_{max}=0.40 & σ_{max}=0.11. Because of this characteristic of σ_{max}, its main use is as a means of discrimination when f_{max} values are similar.

5.4 Establishing the upper and lower bounds of effectiveness

When comparing the retrieval effectiveness of an IR system against the effectiveness of another, sensible comparisons can only be made in the presence of upper and lower bounds.

As all of the intended experiments will involve degrading the quality of the test collection by introducing additional ambiguity, it seems natural that the upper bound should be the effectiveness of the system retrieving from the Reuters test collection without any introduced ambiguity. The method used to establish a lower bound is to measure the retrieval effectiveness of an IR system using a retrieval strategy of randomly selecting documents. Figure 25 shows the plot of these two bounds, as can be seen the lower bound is significantly worse than the upper bound. Note that the upper bound reaches an optimum value and then drops away as the size

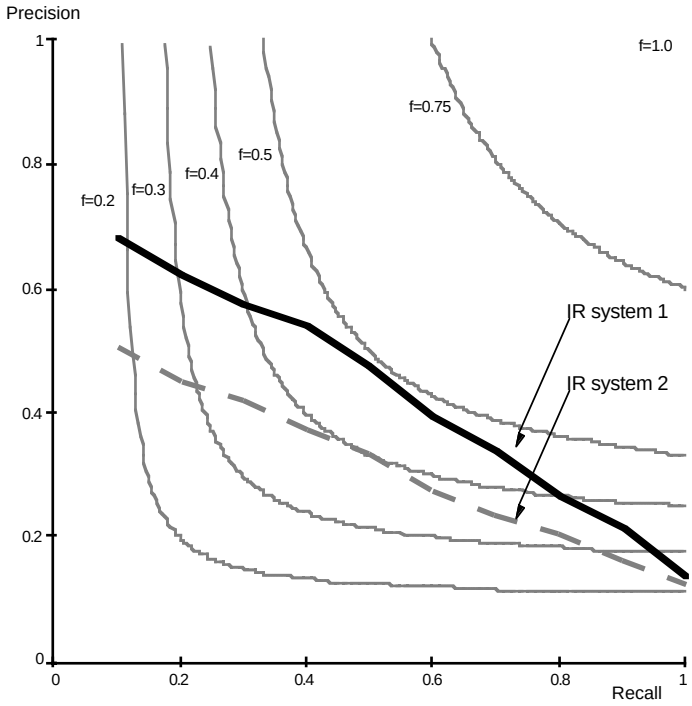


Figure 24. Precision and *f* measures of two IR systems.

	IR system 1		IR system 2	
r	*p*	*f*	*p*	*f*
0.10	0.68	**0.17**	0.51	**0.17**
0.20	0.63	**0.30**	0.45	**0.28**
0.30	0.58	0.39	0.42	0.35
0.40	0.55	0.46	0.37	0.39
0.50	0.48	0.49	0.33	0.40
0.60	0.39	0.48	0.28	0.38
0.70	0.34	0.46	0.24	0.35
0.80	0.27	0.40	0.21	0.33
0.90	0.22	0.35	0.16	0.28
1.00	0.14	0.24	0.12	0.22

Table 8. Precision and *f* measures of two IR systems.

of query increases. (This is a well known trait of IR systems, it is discussed by Hughes *[Hughes 68]* and by Harman *[Harman 92]*.) There is little point in comparing the retrieval effectiveness of the upper bound with other work using the Reuters collection such as Lewis *[Lewis D.D. 91]* or Apté et al. *[Apté 94]* as those researchers used this collection in different manners from each other and from this work. These differences were mainly in the method used to create *Q* the query set, and *T* the test set of the collection. For the experiment described here, this is process is outlined in Section 5.3.1.

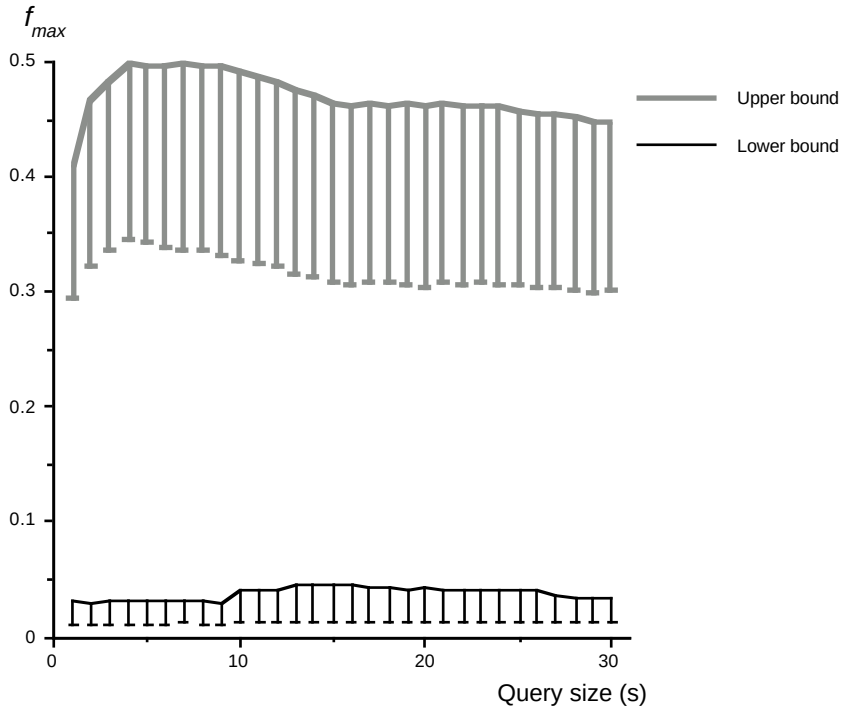


Figure 25. Upper and lower bounds on retrieval effectiveness.

5.5 Start of the experiments

The experiments now start with an investigation of the retrieval effectiveness of an IR system as increasing amounts of ambiguity are added to the Reuters collection using pseudo-words.

5.5.1 Effects of ambiguity on effectiveness

In the first experiment, all words in the Reuters collection were randomly paired to produce size two pseudo-words. The result of the experiment run on this additionally ambiguous collection is shown in Figure 26. As can be seen, when the result is compared to the retrieval experiment run on the unmodified collection, there is little difference in retrieval effectiveness.

As this experiment showed only a small drop in effectiveness, 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 already outlined in Section 4.3.1. 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’.

Two further experiments were run where ambiguity was introduced into the collection using pseudo-words of sizes five and ten. The results of these two experiments are shown in Figures 27 & 28. Considering that introducing pseudo-words of size ten into the Reuters collection reduces the number of distinct terms in that collection from 40,000 to 4,000, the relatively small decrease in retrieval effectiveness caused by this introduction is striking[15].

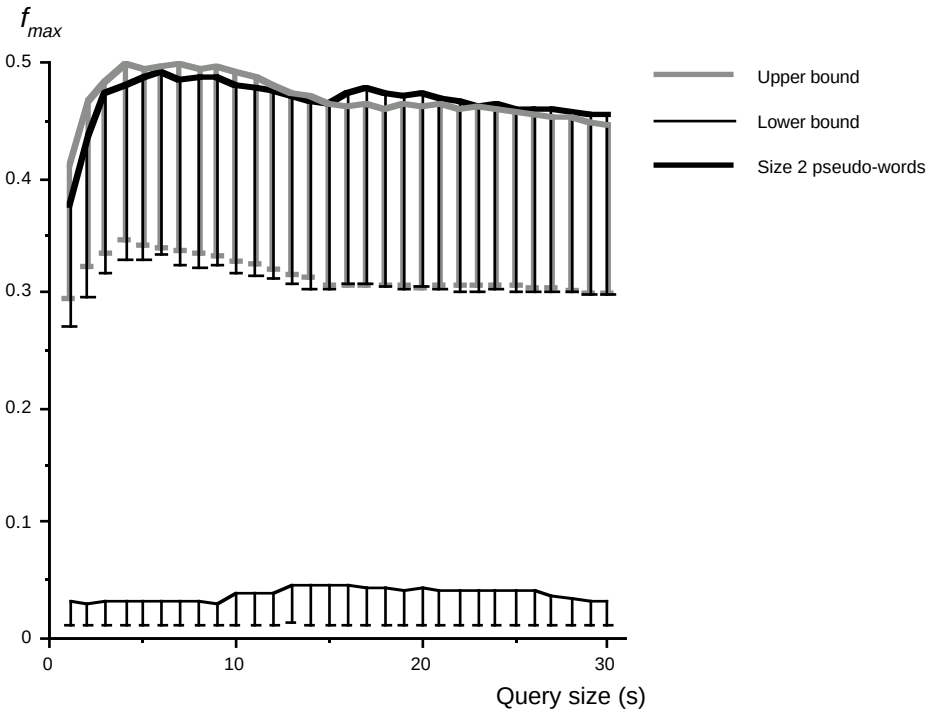


Figure 26. Introducing size two pseudo-words into the Reuters collection.

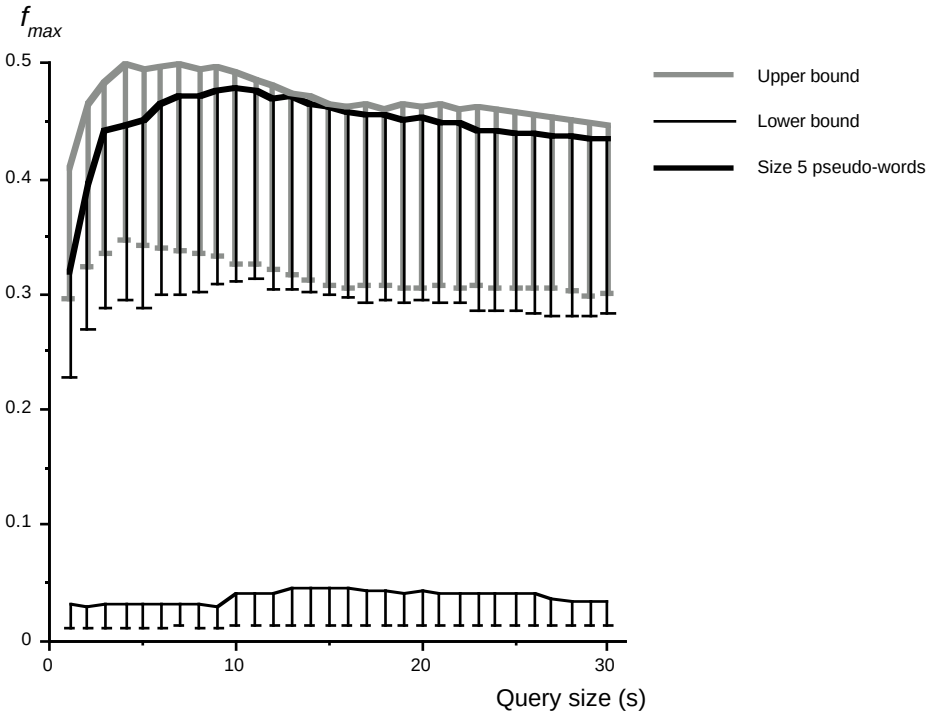


Figure 27. Introducing size five pseudo-words into the Reuters collection.

15. Since conducting the work, it has come to light that experiments of this kind were previously carried out by Burnett et al. *[Burnett 79]* who were performing experiments using document signatures. They were investigating how best to generate a signature from a document. One of their experiments involved randomly pairing together words in the document in the same way that size two pseudo-words are created. They noted that retrieval effectiveness was not affected greatly by this pairing. This seems to be in agreement with the results presented here.

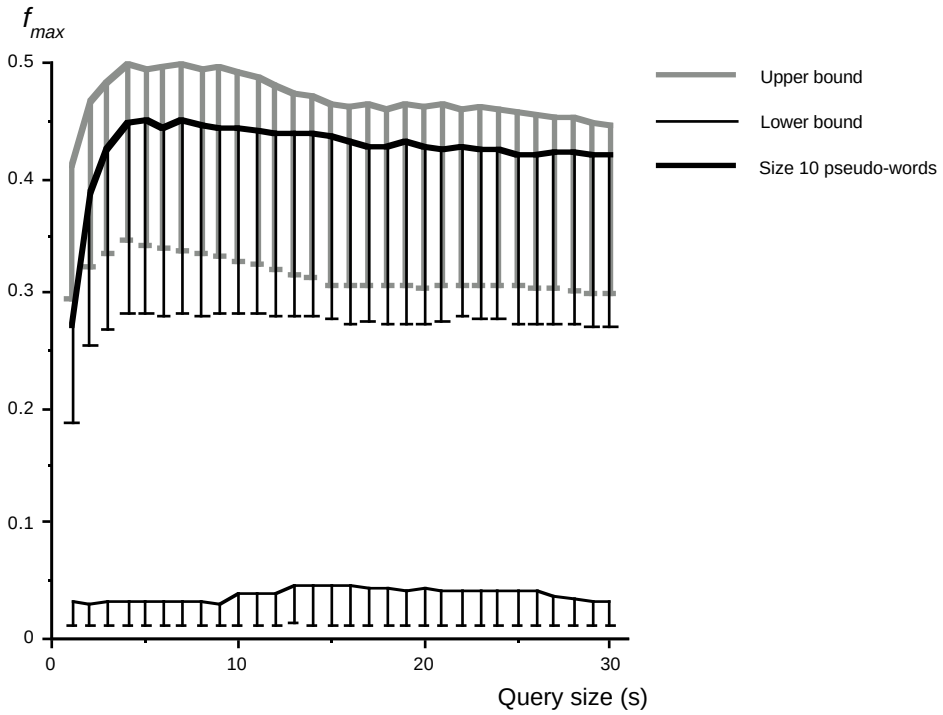


Figure 28. Introducing size ten pseudo-words into the Reuters collection.

As can be seen from the experimental results graphed so far, σ_{max} indicates that the introduction of ambiguity is having little or no effect on the spread of f measures from f_{max}. It was found that σ_{max} varied little throughout the experimentation and as the result graphs are somewhat cluttered by its inclusion, it will not be shown from now on.

5.5.2 Query size

When comparing in detail the difference in effectiveness between retrievals from the unmodified collection and retrievals from an ambiguous collection (Figure 29), 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 effectiveness reduces. This result is consistent with the idea that the degree of word collocation (i.e. the number of query words occurring in a retrieved document) plays an important role in the impact of sense ambiguity on effectiveness, previously outlined in Section 4.4.

5.5.3 Disambiguating ambiguity

The final set of experiments investigated the influence on retrieval effectiveness of a pseudo-word disambiguator operating at varying levels of accuracy. The general method for performing this procedure was introduced in Section 5.1, the details of this experiment are described here.

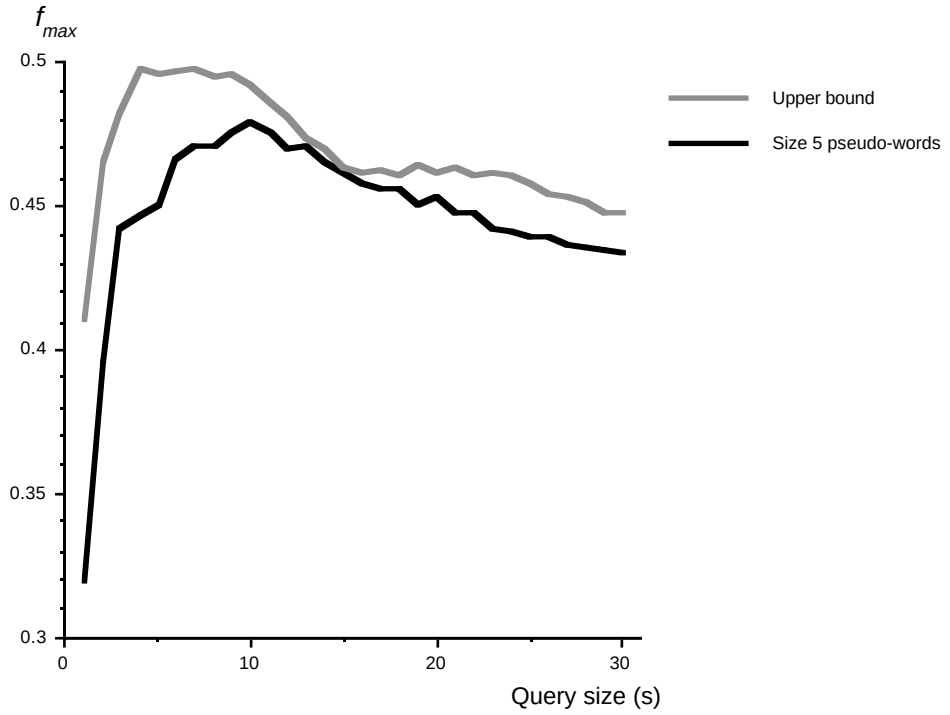


Figure 29. A 'close up' of the top half of Figure 27. Note the *y* origin of this graph is 0.3.

Additional ambiguity was introduced into the Reuters collection using pseudo-words. For these experiments, size five pseudo-words were introduced. These words were then restored to their original state (i.e. *pseudo-disambiguated*) with a controlled amount of error. The method used to decide when an error would occur was by random selection. When such an error occurred, the incorrectly chosen pseudo-sense was selected randomly as well. A retrieval was then performed on the erroneously disambiguated collection and the effectiveness of the retrieval system was measured.

The results these experiments are shown in Figure 30. As can be seen, disambiguation accuracy has a dramatic effect on effectiveness. When the introduced ambiguity is disambiguated with an accuracy of 75% (25% error), the effectiveness is actually worse than that using the ambiguous collection. With disambiguation at 90% accuracy, effectiveness is similar to that of the ambiguous collection, although a small improvement can be seen for retrievals based on queries composed of one and two words. It would appear that errors made by a disambiguator can have a much more significant effect on retrieval effectiveness than ambiguity itself. The only time a disambiguator offers improvements in retrieval effectiveness is for short queries which, from the results of previous experiment, appear to be most affected by ambiguity.

5.5.4 Other collections

These experiments were repeated on two of the more traditional test collections: the CRANFIELD 1400 and the CACM. The results of these experiments are shown in the following four

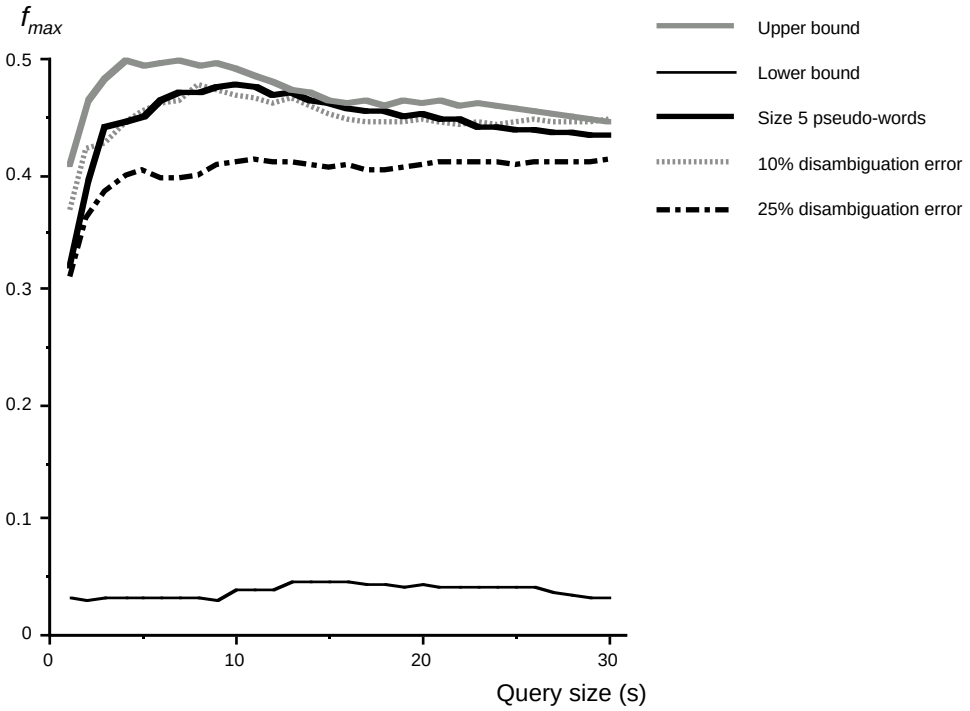
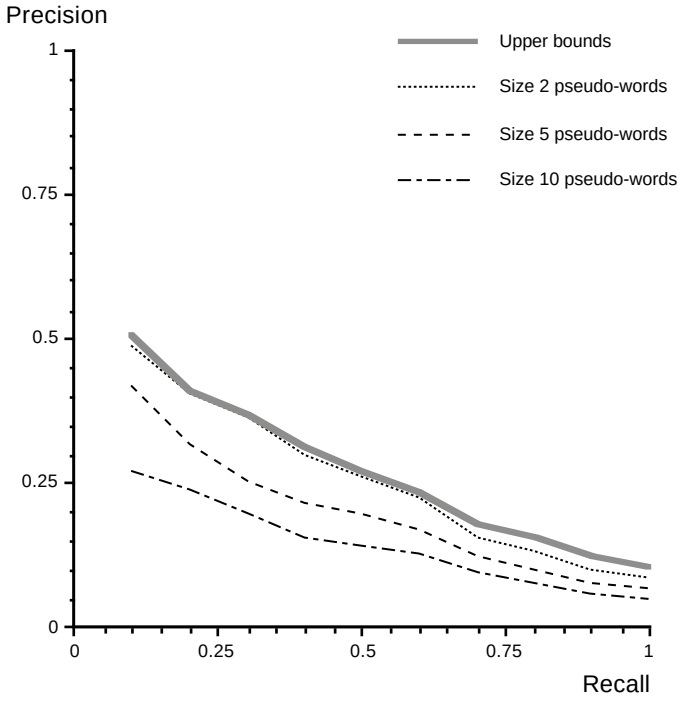


Figure 30. Erroneously disambiguating pseudo-words in Reuters.

graphs: Figures 31 & 33 show the effect on retrieval effectiveness of introducing additional ambiguity into the collections; Figures 32 & 34 show the effect of erroneously disambiguating size five pseudo-words. For these collections, query size was not varied. Therefore, effectiveness is shown using 'classic' RP graphs. In order to provide continuity with the previous experiments, each graph is accompanied by a table showing the corresponding f_{max} measures.

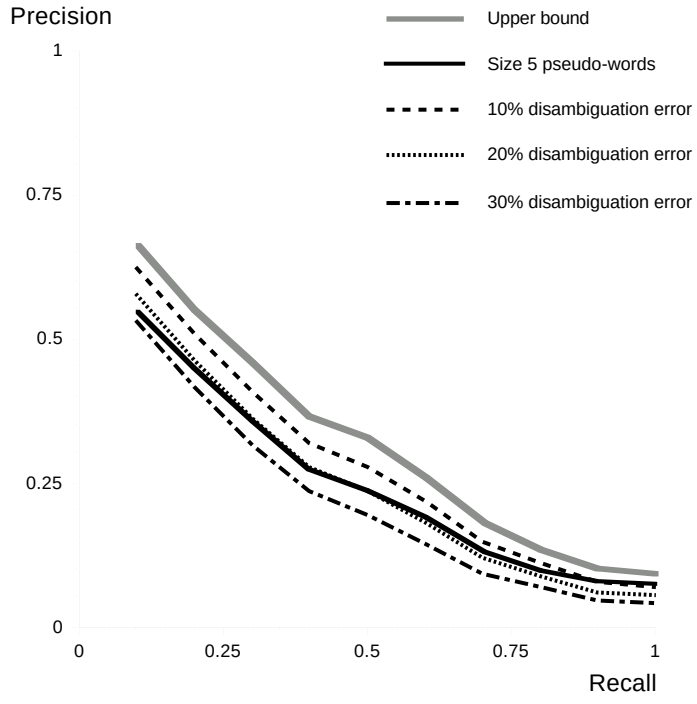
As can be seen, in a similar manner to the experiments on the Reuters collection, retrieval effectiveness is not affected as much by the additional ambiguity as might be expected. However, in the case of the erroneous disambiguation experiments, we can see that the 'break even' point (where effectiveness on a disambiguated collection and an ambiguous collection are equal) is at a disambiguation error of ≈20% rather than ≈10% found in the Reuters collection experiment.

It is believed that reason for the difference is due to the difference in the queries between Reuters and the other two collections. Because the queries in Reuters are generated from relevance feedback, the Reuters queries will predominately contain good terms that discriminate well between relevant and non-relevant documents. This is quite different from the manually generated queries of the CACM and CRANFIELD 1400 collection, where it is quite possible that only a small number of the terms in those queries are good discriminators.



	f_{max}
Upper bound	0.39
Size 2 p/w	0.38
Size 5 p/w	0.32
Size 10 p/w	0.28

Figure 31. Pseudo-words of size two to ten in the CRANFIELD 1400 collection.

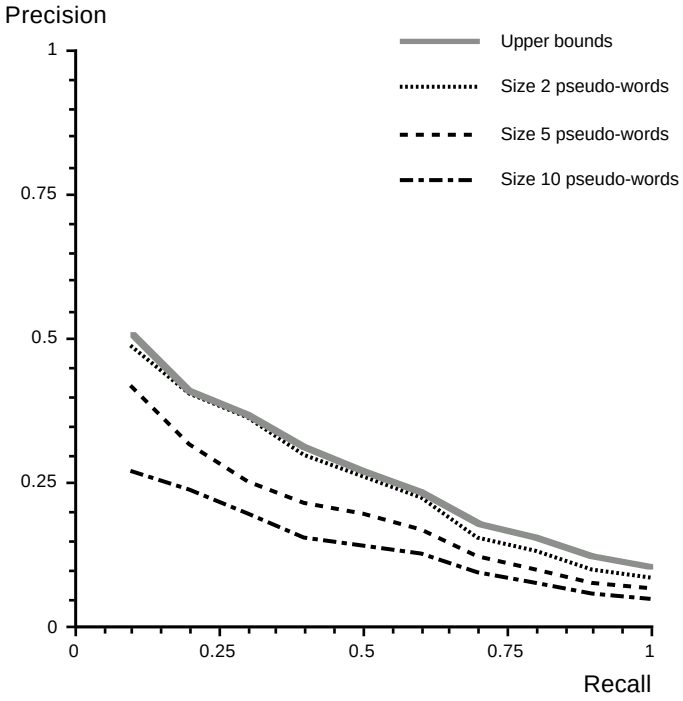


	f_{max}
Upper bound	0.39
Size 5 p/w	0.32
10% dsm. err.	0.36
20% dsm. err.	0.33
30% dsm. err.	0.31

Figure 32. Erroneously disambiguating pseudo-words in CRANFIELD 1400.

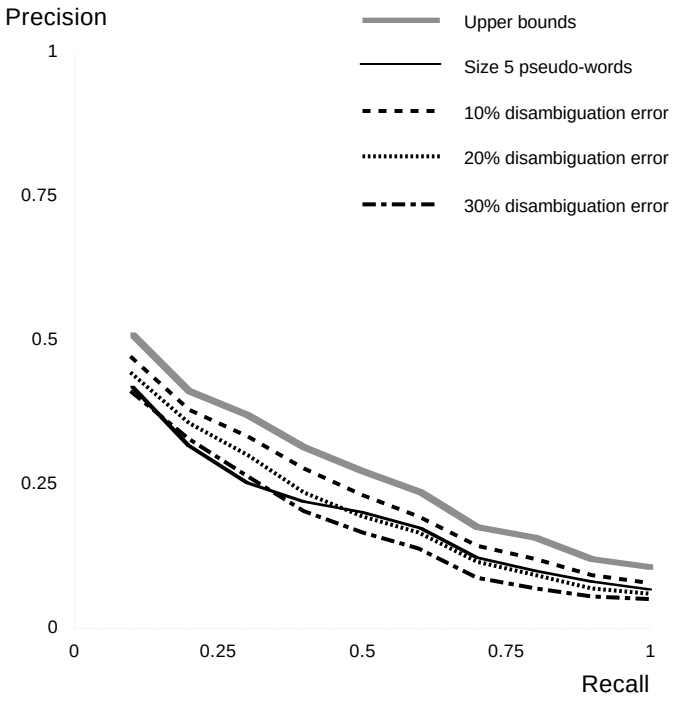
5.6 Analysis and discussion

In this section, the make up of pseudo-words and their pseudo-senses is analysed in order to provide an insight into the experimental results. This will be followed by a description of



	f_{max}
Upper bound	0.35
Size 2 p/w	0.34
Size 5 p/w	0.28
Size 10 p/w	0.24

Figure 33. Pseudo-words of size two to ten in the CACM collection.



	f_{max}
Upper bound	0.35
Size 5 p/w	0.28
10% dsm. err.	0.32
20% dsm. err.	0.30
30% dsm. err.	0.28

Figure 34. Erroneously disambiguating pseudo-words in CACM.

other work published since these experiments were performed that seems to contradict the results presented here. It will be shown that this apparent difference is due to different forms of word sense ambiguity.

5.6.1 Examining the make up of pseudo-words

The example pseudo-words shown in the previous sections have in some ways been unrepresentative. The two component words of the pseudo-word ‘banana/kalashnikov’ are familiar to most people, and part of that familiarity is perhaps an expectation that the frequency of occurrence of these two words is similar. Although helping to explain the principles underlying pseudo-words, this example is atypical. To understand why, we need to examine in more detail the components of pseudo-words, namely words themselves.

Words have very different frequencies of occurrence within a document collection. This can be demonstrated by examining the CACM document collection which contains approximately 7,500 distinct words occurring 100,000 times. Figure 35 shows the distribution of the frequency of occurrence of these words. It can be seen in this graph that the distribution is skewed. Creating pseudo-words by random selection from such a distribution of words is likely to result in pseudo-words composed of multiple pseudo-senses with a similar skew. This can be tested. Sets of pseudo-words of size 2, 3, 4, 5, and 10 were created from the words of the CACM collection, and the distribution of the frequency of occurrence of their pseudo-senses, was examined. For each of these pseudo-words, it was found that one pseudo-sense accounted for the majority of occurrences of the pseudo-word of which it was part. This is shown in Table 9 which displays the percentage of occurrences accounted for by a pseudo-word’s most commonly occurring pseudo-sense. From these figures, it was concluded that the distribution of the frequency of occurrence of these pseudo-senses was indeed skewed.

The example pseudo-word given previously can now be seen to be a little artificial as its components appear to have relatively similar frequencies of occurrence. A more typical pseudo-word (randomly selected from the set generated from the CACM collection) is ‘meet/hoc’ (‘hoc’ from the adjective ‘ad hoc’). The frequencies of occurrence of its two pseudo-senses are 16 and 3.

Given this distribution, we can begin to formulate reasons for the results of the experiments in Section 5.5. It has been generally accepted that words with a medium frequency of occurrence are those that overall have the greatest impact in resolving relevant documents from non-relevant during retrieval *[Luhn 58]*. Therefore, we should concentrate on the effect on these words when considering the impact of pseudo-words on IR effectiveness. If a pseudo-word contains a pseudo-sense based on a medium frequency word, there is a high probability that its other pseudo-senses will be based on low frequency words. Therefore, its most common pseudo-sense will account for the majority of occurrences of that pseudo-word. This means that such a pseudo-word will, in its effect, be little different from that of the medium

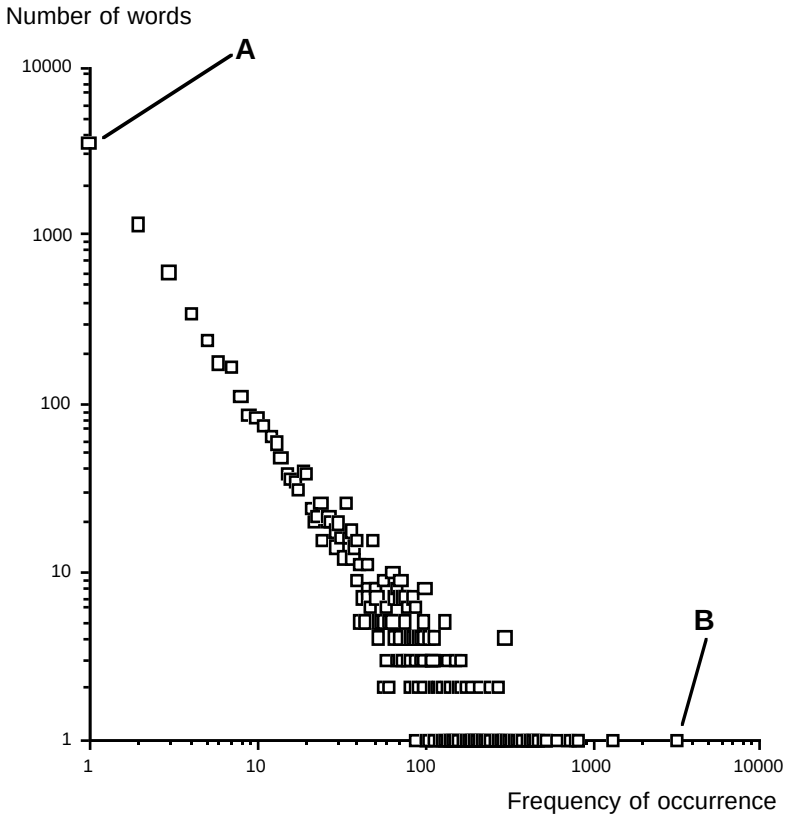


Figure 35. Distribution of the frequency of occurrence of words in the CACM collection. Graph plotted on a logarithmic scale. Point A shows that around 3,600 of the words (about half of all words in the collection) occur in the collection only once. Point B shows that one word occurs around 3,000 times in the collection.

No. of senses	Most common sense (%)	
2	92	{50}
3	85	{33}
4	80	{25}
5	78	{20}
10	65	{10}

Table 9. Percentage of occurrences accounted for by most common pseudo-sense of a pseudo-word. The figures in brackets (shown for comparison) are the percentage that would result if pseudo-senses occurred in equal amounts. Measurements made on the CACM collection.

frequency word that is its main component. The relatively small drop in retrieval effectiveness after pseudo-words were introduced into the Reuters collection now seems less surprising.

Given the skewed distribution of pseudo-senses, it would not be unreasonable to wonder how realistic a simulation of real ambiguous words it is. In their study of the testing of disambiguators, Gale et al. *[Gale 92a]* stated that if a disambiguator used a strategy of selecting the most commonly occurring sense, it would be correct 75% of the time. This suggests that the senses of ambiguous words have a similar distribution to pseudo-words. It is possible to measure the frequency distribution of word senses using the SEMCOR sense tagged corpus

which is released with WordNet *[WordNet]*. It is a 100,000 word corpus consisting of around 15,000 distinct words. All word occurrences were manually tagged with senses as defined in the Wordnet thesaurus (v1.4). Using this corpus, we can plot the distribution of the frequency of occurrence of ambiguous word senses (Figure 36). From Figures 35 & 36, we can see that *senses* in the SEMCOR corpus have a skewed frequency distribution similar to that of the *words* in the CACM collection.

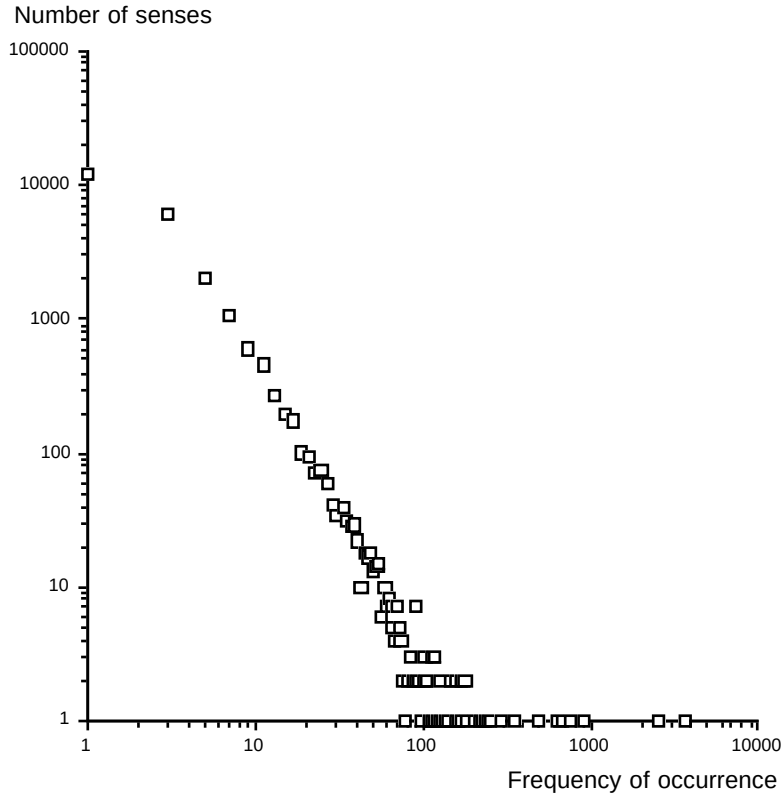


Figure 36. Distribution of the frequency of occurrence of senses in the SEMCOR corpus. Graph plotted on a logarithmic scale.

No. of senses	Size of set	Most common sense (%)
2	3145	92 {50}
3	1697	85 {33}
4	1046	79 {25}
5	640	72 {20}
6	448	68 {17}
7	275	63 {14}
8	200	60 {13}
9	141	60 {11}
10	93	53 {10}

Table 10. Percentage of occurrences accounted for by the most common sense of a word. The figures in brackets (shown for comparison) is the percentage that would result if senses occurred in equal amounts. Measurements made on the SEMCOR corpus.

As was done with pseudo-words, the distribution of the frequency of occurrence of word senses was examined, Table 10 displays the percentage of occurrences accounted for by a word's most common sense. The percentage was computed for separate sets of words, the set a word belongs to is defined by the number of senses that word has. As can be seen, a word's most common sense accounts for the majority of that word's occurrences. The figures in Tables 9 & 10 shows a strong similarity. From this comparison, it was concluded that the distribution of a pseudo-word's pseudo-senses is a realistic simulation of an ambiguous word.

5.6.2 Other work

After the work in this chapter was completed, Schütze and Pedersen *[Schütze 95]* published work on word sense ambiguity and IR, the conclusions of which seemed to contradict the work described here. They had built a disambiguator, applied it to the words of a test collection and achieved a 7-14% improvement in effectiveness: the first published results showing a disambiguator working successfully with an IR system. Although the improvement they reported was not ruled out by the results of the pseudo-word experiments, it was not expected and warrants an explanation.

The reason for this apparent contradiction in results appears to be due to the type of ambiguity resolved by Schütze and Pedersen's disambiguator. Their disambiguator does not use a dictionary or thesaurus as a source of word sense definitions, instead it uses only the corpus to be disambiguated. The disambiguation method is as follows. For each word in the corpus, the context of every occurrence of that word within the corpus is examined and common contexts are clustered. For example, if we take the word 'ball', we might find that within a corpus of newspaper articles, this word appears in a number of common contexts: a social gathering; and perhaps a number of different sports such as tennis, football, or cricket. For Schütze and Pedersen's disambiguator each one of these common contexts constitutes an individual 'sense' of the word. This is where we see what is unusual about this disambiguator: these 'senses' are quite different from the senses one will find in a dictionary. It is unlikely for instance, that a dictionary would distinguish between different types of the sporting sense of 'ball'.

A further difference in Schütze and Pedersen's disambiguator is that it only attempts to identify the common 'senses' of a word: Schütze and Pedersen state that a common context is only identified as a 'sense' if it occurs more than fifty times in the corpus. In doing this, the disambiguator avoids the problem that dictionary based disambiguators face of identifying the sense of a word from a long list of candidate senses (defined in the dictionary) many of which will not actually appear in the corpus.

Therefore, it can be concluded that the difference in results between the work presented in this chapter and the work presented by Schütze and Pedersen is due to the significant difference in definition of what constitutes a sense of a word. From the effectiveness results reported by Schütze and Pedersen, it would appear that in the context of IR, identifying different *uses* of a word is a good strategy. Uses are easier to identify than senses, as only relatively common ones are found. Uses are flexible, in that they show up the subtle distinctions of use contained in a particular corpus unlike senses which are fixed by their dictionary definitions. However, there are two aspects of Schütze and Pedersen's technique that remain unresolved.

- First, word senses are fixed by the dictionary they appear in but word uses are entirely defined by the collection being analysed. Across different collections, a word's uses could be quite different In the context of querying an IR system employing Schütze and Pedersen's technique, it may be confusing for users to identify different query word uses depending on which collection they are retrieving from.

- Second, the test collection Schütze and Pedersen used was an early version of the TREC collection which is well known for having particularly large queries (>100 words per query is common). So large in fact that their disambiguator could identify the uses of query words automatically, no manual identification was required. Uses are defined only by their cluster of surrounding context words. It might be instructive to test how easily a user could identify a word's use from these context words, especially, as pointed out by the authors, many of the context words are likely to be proper nouns. For example, the context words of the tennis use of 'ball' are likely to include the names of many tennis players. Unless users know who these players are, they might not be able to deduce the meaning of that use. This contrasts with a dictionary based disambiguator that would display to a user that dictionary's sense definitions. It would be hoped that a user could understand the meaning of a word sense from its definition given that definitions were written for this very task.

In conclusion, Schütze and Pedersen have produce clear evidence that paying heed to the 'senses' of a word can bring benefits to retrieval effectiveness. The technique they used, however, was based on word uses and not senses. Potential problems with this technique in relation to users have been suggested; their significance, however, cannot be assessed until further investigation of word uses takes place.

5.7 Conclusions

Using the novel experimental technique of introducing and removing ambiguity in a test collection, insights into the significance of ambiguity to retrieval effectiveness were gained. It

was anticipated that query size would play an important role in these experiments. Therefore, the Reuters 22,173 collection was selected because the size of its queries could be varied at will. As anticipated, the conclusions of the first set of experiments were that for short queries, ambiguity reduced effectiveness significantly. For longer queries, however, ambiguity was not as large a problem as might have been expected. The use of a disambiguator on the test collection was found to improve the effectiveness of a retrieval system but only if its disambiguation was accurate. The amount a disambiguator improved effectiveness varied depending on the size of the query, with short queries benefiting most from disambiguation. This refining of the general appreciation of word sense ambiguity will be used to identify areas that justify further investigation within the context of this thesis.

6 Design and pre-testing of the disambiguator

This and the following chapter describe the design, and testing of an automatic word sense disambiguator in preparation for the final experiment: to test the retrieval effectiveness of an IR system working with such a disambiguator. This chapter describes the design of the disambiguator and the pre-testing of that design. More substantive tests on the disambiguator to establish its accuracy are described in Chapter 7.

6.1 The design of the disambiguator

For a number of natural language processing applications, such as grammatical tagging, fully functioning tools have been made publicly available. In the field of word sense disambiguation, however, no publicly available disambiguators yet exist. This means that for these experiments, a word sense disambiguator had to be built. No attempt was made to devise a completely new disambiguation method, rather, existing techniques were examined to find the most suitable.

As was shown in Chapter 4, many different methods of disambiguation exist. In choosing which of these to use, three factors were considered: the reported accuracy of the disambiguator; the availability of the text resources (dictionary, bilingual corpus, thesaurus, etc.) that would be needed to implement it; and most importantly, the intended use of the disambiguator, namely to disambiguate a large document text collection for subsequent use in an IR experiment. Resources were available to attempt a number of disambiguation methods: the various techniques devised by Wilks, Guthrie, and their colleagues using the Longmans dictionary; as well as the WordNet disambiguation methods of Voorhees and of Sussna. These techniques were rejected, however, as one of the disambiguators reviewed promised to be particularly well suited to the IR experiments. This was Yarowsky's disambiguation technique, which was reported as being one of the most accurate disambiguators to date. To understand why this disambiguator is well suited to IR, we need to first briefly recall Yarowsky's disambiguation method. A fuller explanation of his disambiguator is found in Section 4.2.7.

6.1.1 Recalling Yarowsky's method

The functional similarity of disambiguators to IR systems has already been highlighted: senses are like a document collection represented as a set of features; context is like a query and is similarly represented; and the process of disambiguation is like ranked retrieval. What distinguishes disambiguation methods from each other is the process used to gather sense representation features, known here as clue words. Typically, disambiguators will gather clue words directly from a reference work like a dictionary or thesaurus. Yarowsky's disambigua-

tor, however, uses a thesaurus (Roget's thesaurus), as a source of what we shall call *seed words*, one set of these seeds is generated for each word sense. The seeds contained in each set are in fact synonyms of the associated sense. For example, seeds of the economic sense of 'bank' might include 'depository' and 'lender'. To gather a set of clue words for a particular sense, the contexts of all the occurrences of all that sense's seed words are looked up in a large text corpus. Figure 37 shows some example contexts of the seed words 'lender', 'depository', and 'bank'.

Corporate bonds provide maximum safety to **lenders** while offering a steady income in the form of interest
money market of Antwerp. In return, the **lenders** were given monopoly rights and political protection.
and placed them in government **depositories** located in major cities
saving-and-lending institution, serving as a **depository** primarily for the money of individuals
has been the site of the U.S. Gold Bullion **Depository**. The gold is stored in concrete
highly developed monetary system with **banks** and credit, as did ancient Greece
World Health Organization and the International **Bank** for Reconstruction and Development

Figure 37. Example contexts of seed words.

Such contexts are analysed using a statistical technique similar to relevance feedback to gather words that have a higher frequency of occurrence in the contexts than in the corpus as a whole. It is those words that are then used as a set of clues for that sense. Clues of the economic sense of 'bank' gathered from the contexts, might be 'gold', 'credit', 'development', etc. This process of gathering clues is repeated for every sense of every word to be disambiguated.

By gathering clues directly from the corpus the disambiguator is in effect being 'trained' to disambiguate senses as they are used in that corpus. Therefore, it will be most accurate when applied to words used in the same linguistic style of the corpus. Yarowsky used the Grolier Multimedia Encyclopedia for this purpose. It was probably chosen by him as it is large and covers a wide range of topics written in a non-specialist style. Beyond that, there is nothing special about Grolier. One could just as easily use other corpora. In building a disambiguator for general use, Grolier seems a good choice. For the intended use here, we want a disambiguator that is specialised for the document collection we will be retrieving from, one that will pick up on the linguistic style and cultural references of that particular collection. Using Yarowsky's disambiguation method, this can be achieved by using the collection as the source of sense clue words.

To illustrate why this strategy might be advantageous, let us imagine the situation of a disambiguator gathering clues for the economic sense of 'bank'. If the corpus we wished to disambiguate was a collection of British newspaper articles from the mid 1990's, we might find that in these articles good clues are the surnames 'Portillo' and 'Clarke' who were British govern-

ment treasury ministers at the time. By training the disambiguator on this corpus, those words are likely to be gathered as clues. They would not have been gathered had the disambiguator been trained on a general corpus like Grolier. The only limitation to this customised training strategy is the size of the corpus, one that is too small will contain insufficient word occurrences for good sense clues to be gathered. Quite what constitutes a small corpus remains to be determined.

As Yarowsky's disambiguation method is customisable to a particular document collection and because it has been shown to work well, it was decided that a disambiguator based on this method was best suited for the experiments of this chapter. The first stage was to implement and pre-test this disambiguator and it is this process that is now described, starting with the issues arising from the implementation.

6.1.2 Implementing Yarowsky's method

In choosing to implement Yarowsky's method there was a problem. The thesaurus he used, (the 1977 edition of Roget) was obtained through a private arrangement with the publisher and is not in the public domain. The most obvious replacement was WordNet, as it is large, freely available, designed to be used in computing projects, and is provided with a stemmer that transforms morphological variants into the correctly formed root words contained in the thesaurus. However, the organisation and grouping of words in WordNet is different from Roget and if Yarowsky's method was to be implemented using WordNet it was necessary to first decide if these differences were problematic. Earlier it was stated that Yarowsky used the synonyms of a word sense as its seeds. That was a slight simplification. Yarowsky, in fact, used more than just synonyms, his disambiguation technique used seeds from the broad semantic categories found in Roget. These categories cover wide areas such as tools & machinery, or animals & insects. WordNet does not have these large categories, which is unfortunate as Yarowsky's disambiguation method requires large numbers of seeds for each sense. So the possibility of constructing large semantic categories in WordNet was investigated. As was described in Section 4.2.7, all words in WordNet are connected by relations, the most common being synonymy (a set of words related by synonymy is called a synset) and the hierarchical relations hypernymy and hyponymy. It was decided to determine if those three relations could be exploited to obtain something similar to the Roget semantic categories. Looking at a part of the WordNet hierarchy in Figure 38, we can see the structure surrounding the synset holding one of the senses of the word 'bank', this time a river sense. By traversing the structure above and below 'bank', we could perhaps generate a set of words that are in the same broad semantic category.

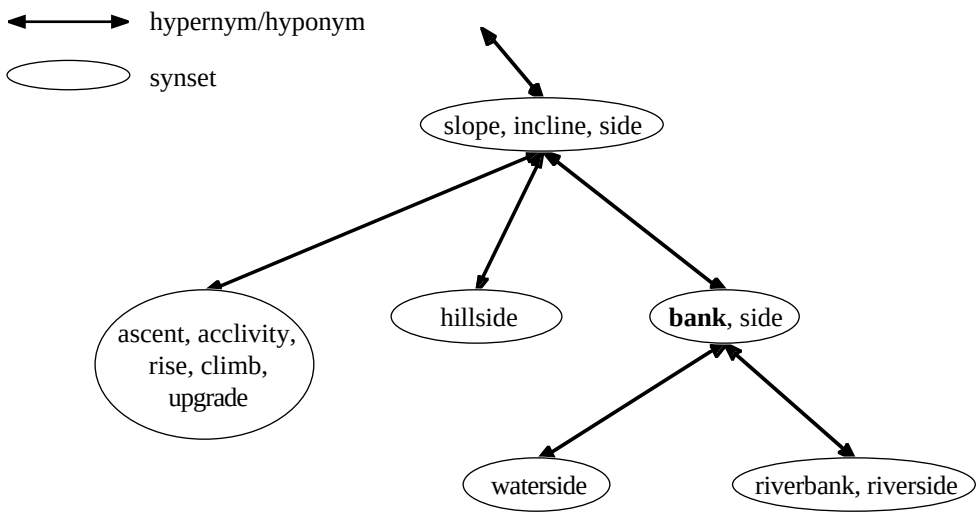


Figure 38. Fragment of the WordNet hierarchy.

Other research that has used the WordNet relations in this manner was examined with the aim of discovering the most appropriate strategy for generating large categories. Only two pieces of research were found that described a traversal over the WordNet relations: the research of Voorhees *[Voorhees 93]* and of Hearst *[Hearst 93b]*. Since the strategies devised by them are quite similar, for the sake of brevity only one is described here, that of Voorhees. She used a traversal strategy to gather from WordNet, clues (not seeds) for a disambiguator, a description of which, can be found in Section 4.4.

Voorhees' method of gathering clue words for a particular word sense was to gather all the words contained in, what she called, the *hood* of that sense. The hood of a sense contained in a synset *s* is defined by Voorhees as follows.

> To define the hood of a given synset, *s*, consider the set of synsets and the hyponymy & hypernymy relations in WordNet as the set of vertices and directed edges of a graph. Then the hood of a given synset *s* is the largest connected sub graph that contains *s*, contains only descendants of an ancestor of *s*, and contains no synset that has a descendent that includes another instance of a member of *s*.

Voorhees did not test the accuracy of the disambiguator that used the clue words but she did apply it to a number of test collections and performed retrieval experiments on those collections. The results showed that retrieval effectiveness was lower when retrieving from the disambiguated collection than retrieving from the ambiguous collection. One could infer from this result that using Voorhees' hood technique is not a promising line of enquiry. When assessing this work, however, it is important to be aware that her use of WordNet was as a

source of clues whereas the use of WordNet with the Yarowsky disambiguation method is as a source of seeds. This difference is important, as it affects the type of words one might wish to gather. As Voorhees was gathering clues for a particular sense, it follows that she was looking for words from WordNet that would commonly occur *in the context* of that sense. When gathering seeds, however, one is looking for words that would commonly occur *in place* of that sense. In fact, when one considers that a thesaurus is a reference work that relates similar words, one can begin to see that Voorhees' traversal technique might be better suited to gathering seed words. Therefore, it was decided to use her technique as the basis for the seed gathering process of the disambiguator. It was felt, however, that there were certain features of Voorhees' traversal strategy that could be improved upon and these will now be discussed and explained.

Similarity of hoods

Voorhees defined that the hoods of a word's senses must be disjoint. This has the effect of preventing the hoods from being semantically similar. Quite why Voorhees elected to do this is not clear as it is often the case that the senses of a word, as defined in a dictionary or thesaurus, are similar and there seems little to be gained from suppressing this. Therefore, the seed word gathering method to be used for the disambiguator in this thesis will allow its hoods to overlap.

Size of hoods

The other feature to be changed was the size of hoods. Voorhees placed no restriction on a hood's size and there was no attempt to ensure that the hoods belonging to the senses of a word were of similar size. Both of these factors are important to consider. If one generates a hood that is too big, many of the outlying words in that hood are unlikely to contribute to the process of discriminating one sense from another, in fact those words are likely to introduce error into the process. Ensuring that the hoods are the same size across all the senses of a word is also likely to be important. If we imagine a word with two senses, the first with a hood containing a great many words, the second containing just a few, it follows that a disambiguator discriminating between these two senses is more likely to pick the first sense just by chance. Therefore, the same number of seeds will be gathered for all senses of a word. Just how many should be gathered, will be determined through testing.

6.1.3 The adopted traversal strategy

Now that the seed word gathering strategy has been defined, we shall go through an example. When gathering seed words for a particular word sense, the synset corresponding to that sense

is first looked up in WordNet. As has already been stated, all synsets are linked into the WordNet hierarchical structure, and by traversing this structure, seeds can be gathered. The six diagrams in Figure 39 show the progression of the seed gathering process over a part of the WordNet hierarchical structure. Each dot in the structure represents a synset. The gathering process is as follows. Starting at the initial synset (shown as a black dot in tree 1), the process first traverses down the hierarchy (following hyponym links), level by level, gathering seeds as it goes (see trees 2 and 3). If not enough seeds are gathered in this traversal, the process moves up one level (following a hypernym link) from the initial synset (tree 4). From here, it starts again traversing down the hierarchy, level by level, gathering more seeds (trees 5 and 6). To avoid retraversing areas of the hierarchy, the process does not traverse down a link that was previously traversed up. When traversing synsets along a level of the hierarchy, the order in which the process traverses them is defined by the order in which they are stored in WordNet. The seed word gathering process stops as soon as the required number of seeds is gathered.

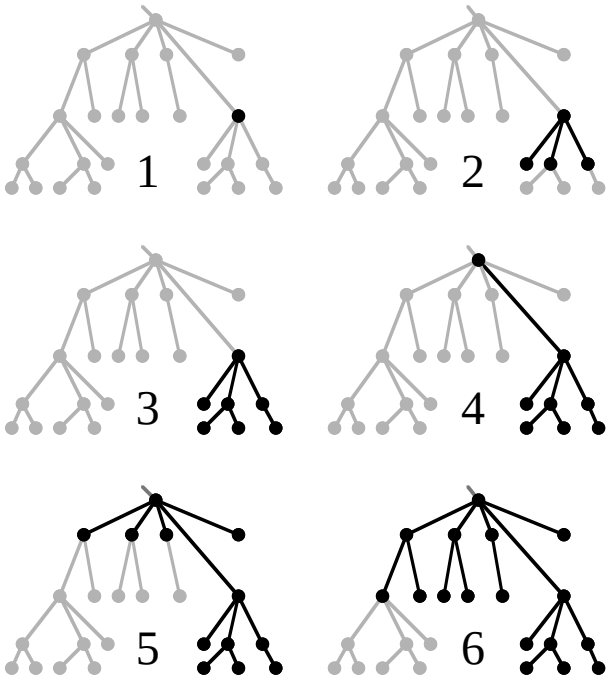


Figure 39. Traversal strategy over WordNet semantic hierarchy.

Now that we have defined the disambiguation method and shown how it will be adapted to work with available resources, the next stage in development of this disambiguator was to implement it and then pre-test it.

6.2 Pre-testing of the disambiguator using pseudo-words

As it was anticipated that full testing of the accuracy of the disambiguator would be a time consuming process, it was decided to perform a quick pre-test using pseudo-words to establish

if the disambiguator worked at all. As will be seen, it could, and so these first tests were extended with the aim of establishing how the accuracy of the disambiguator was affected by its two main parameters, the number of seed words generated and the number of clue words gathered. Although these tests showed the disambiguator to be working, its accuracy was not impressive and so before starting the second stage of testing, an alternative method of using the disambiguator's output was devised.

These pre-tests were based on Yarowsky's disambiguator testing device, pseudo-words: artificial ambiguous words created by the concatenation of two or more ordinary words. The corpus into which pseudo-words were introduced was the Reuters document collection. That corpus was chosen as it would be the collection that the finished disambiguator would be applied to. It was decided to test the disambiguator on five pseudo-words of size two. That size was chosen because at the time of testing it was the largest size the disambiguator could process in a tolerable amount of time (later versions of the disambiguator were faster). It was not felt that this restriction on the number of pseudo-senses was a problem as the main aim of these tests was to establish if the disambiguator worked at all.

Table 11 shows data on the occurrences of the pseudo-words that were tested and the proportion of those occurrences for which each pseudo-sense of each pseudo-word accounted. As can be seen, for four of the pseudo-words the two pseudo-senses occurred in roughly equal numbers, the other pseudo-word was composed of two words with a 75/25% split in their number of occurrences.

pseudo-word	telegraph/relationship		support/help		impact/space		discuss/cent		buyer/publish	
num occs	111	107	1367	1244	463	141	614	704	368	360

Table 11. The five pseudo-words used in initial testing of the accuracy of the disambiguator.

6.2.1 Does the disambiguator work?

The first part of these experiments was a test to see if the disambiguator was working at all. The disambiguator was set to disambiguate all the occurrences of the five pseudo-words. Its two main parameters, the number of seed words gathered and the number of clue words generated were set to 150 and 1,000 words, respectively. These values were similar to the parameters used by Yarowsky in his implementation of the disambiguator. The results of the test are shown in Table 12.

For each occurrence of a pseudo-word, the disambiguator could make one of two choices: it could disambiguate that occurrence, and that disambiguation could be correct or incorrect; or it could judge that the evidence was such that a choice between pseudo-senses could not be

	p-w	telegraph/relationship		support/help		impact/space		discuss/cent		buyer/publish		av
1	num occs	111	107	1367	1244	463	141	614	704	368	360	
2	ambig	1	1	77	42	7	5	7	20	3	10	
3	correct	54	100	500	863	347	65	480	601	108	259	
4	incorrect	56	6	790	339	109	71	127	83	257	91	
5	%disam	99%	99%	94%	97%	98%	96%	99%	97%	99%	97%	**97%**
6	%correct	50%	94%	42%	73%	76%	50%	79%	88%	30%	75%	
7			72%		57%		63%		84%		52%	**66%**

Table 12. Initial pseudo-word disambiguation experiments.

made. In this case the word occurrence would be left ambiguous. For this initial experiment, a pseudo-word occurrence would be left if the disambiguator had assigned identical confidence scores to each of the pseudo-word's two pseudo-senses. Row 2 in the table shows the number of pseudo-word occurrences left ambiguous. Where the disambiguator disambiguated, rows 3 and 4 show its accuracy. Row 5 shows the percentage of the occurrences where a disambiguation took place, and of those, row 6 shows what percentage were correct. Row 7 shows the overall accuracy of disambiguation for each pseudo-word. As can be seen, accuracy is variable. For three of the pseudo-words, 'telegraph/relationship', 'impact/space', and 'discuss/cent', the disambiguator appears to be working to some extent. For the other two, 'support/help', and 'buyer/publish' there is no discernible disambiguation taking place. Across all the different configurations of the disambiguator that were tested, the accuracy over these five pseudo-words was always found to follow this pattern. The right most column of row 7 of the table shows the unweighted average accuracy of the disambiguator, 66%.

An investigation was conducted to try to improve this accuracy by making stricter the decision criteria the disambiguator used when deciding to disambiguate or not. The best way found to achieve this was to set a minimum percentage difference between the confidence scores of the two pseudo-senses. For example, requiring that the higher scoring pseudo-sense of a pseudo-word occurrence was at least 20% higher than that of the other pseudo-sense. Table 13 shows the accuracy of the disambiguator over five minimum differences: 0% (the same difference as was used in the experiment of Table 12), 10%, 20%, 40%, 60%. As can be seen, there is a trade off when using this strategy: as the minimum difference increases, the disambiguation accuracy increases to some extent, but at the high price of the number of occurrences where disambiguation takes place, falling off significantly. Given that this strategy was only producing a relatively small increase in disambiguation accuracy for this significant drop in disambiguations, the use of minimum differences was not pursued further.

p-w	telegraph/relationship		support/help		impact/space		discuss/cent		buyer/publish		av
num occs	111	107	1367	1244	463	141	614	704	368	360	
0%											
ambig	1	1	77	42	7	5	7	20	3	10	
correct	54	100	500	863	347	65	480	601	108	259	
incorrect	56	6	790	339	109	71	127	83	257	91	
%disam	99%	99%	94%	97%	98%	96%	99%	97%	99%	97%	**97%**
%correct	50%	94%	42%	73%	76%	50%	79%	88%	30%	75%	
		72%		57%		63%		84%		52%	**66%**
10%											
ambig	3	21	474	415	80	27	60	52	30	43	
correct	99	49	324	641	305	52	455	579	98	243	
incorrect	9	37	569	188	78	62	99	73	240	74	
%disam	97%	80%	65%	67%	83%	81%	90%	93%	92%	88%	**84%**
%correct	92%	57%	36%	77%	80%	46%	82%	89%	29%	77%	
		74%		57%		63%		85%		53%	**66%**
20%											
ambig	9	37	861	768	153	57	111	85	57	84	
correct	97	39	166	385	255	45	425	559	83	222	
incorrect	5	31	340	91	55	39	78	60	228	54	
%disam	92%	65%	37%	38%	67%	60%	82%	88%	85%	77%	**69%**
%correct	95%	56%	33%	81%	82%	54%	84%	90%	27%	80%	
		75%		57%		68%		87%		54%	**68%**
40%											
ambig	24	73	1290	1181	333	106	227	175	119	162	
correct	86	19	29	49	111	19	343	496	62	173	
incorrect	1	15	48	14	19	16	44	33	187	25	
%disam	78%	32%	6%	5%	28%	25%	63%	75%	68%	55%	**43%**
%correct	99%	56%	38%	78%	85%	54%	89%	94%	25%	87%	
		77%		58%		70%		91%		56%	**70%**
60%											
ambig	46	96	1349	1238	428	126	366	319	180	217	
correct	64	6	10	4	29	9	229	375	47	134	
incorrect	1	5	8	2	6	6	19	10	141	9	
%disam	59%	10%	1%	0%	8%	11%	40%	55%	51%	40%	**27%**
%correct	98%	55%	56%	67%	83%	60%	92%	97%	25%	94%	
		77%		61%		71%		95%		59%	**73%**

Table 13. Results of five pseudo-word disambiguation experiments.
Measuring the accuracy of the disambiguator over five minimum differences. Layout of this table is similar to that of Table 12.

There remained to be tested, however, the effect on the disambiguator's accuracy of varying its two main parameters: the number of seed words gathered; and the number of clue words that were subsequently generated. The experiments that measured this are now described.

6.2.2 Altering the disambiguator's two main parameters

The first parameter varied was the number of seed words. Figure 40 graphs the effect on disambiguation accuracy of this variation. The effect was measured over a number of clue word settings. It appears that altering the number of seed words has no consistent effect on the disambiguator's accuracy.

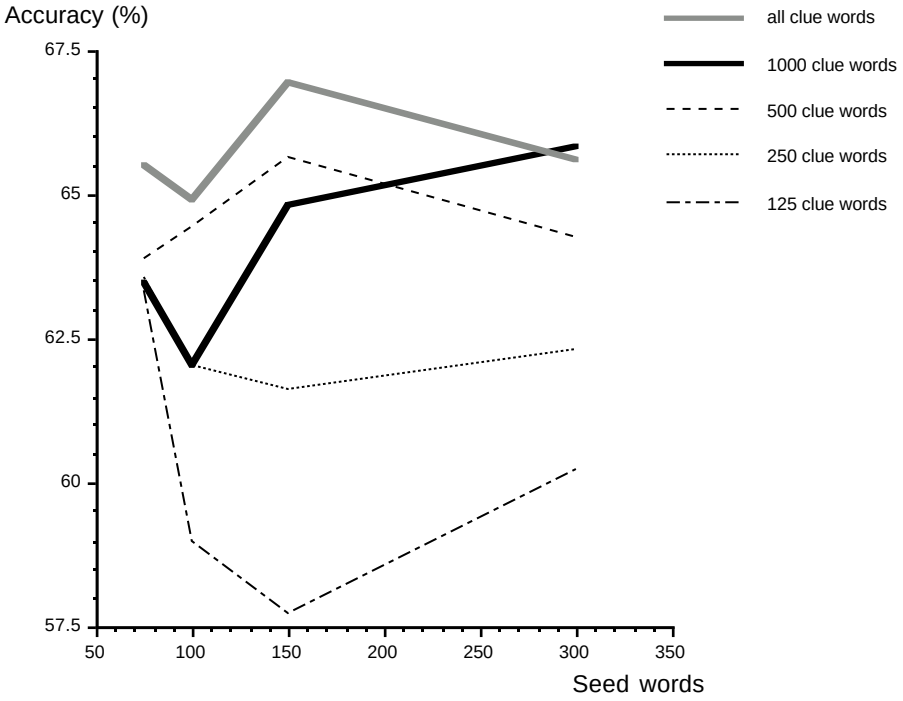


Figure 40. Accuracy of the disambiguator against number of seed words. Note the origin of the y-axis is at 57.5%.

The second parameter to be varied was the number of clue words and this produced a consistent change in accuracy, as can be seen in Figure 41. Here it can be said with some confidence that as the number of clues is increased, the accuracy increases also, and this occurs across all seed word settings.

6.2.3 Summary

These experiments have shown the disambiguator working to some extent. When the main parameters of the disambiguator were varied, its accuracy was affected by less than 10%. Varying the number of clue words seemed to have a consistent effect on the disambiguator's accuracy, and this will be the main parameter that will be examined in further testing.

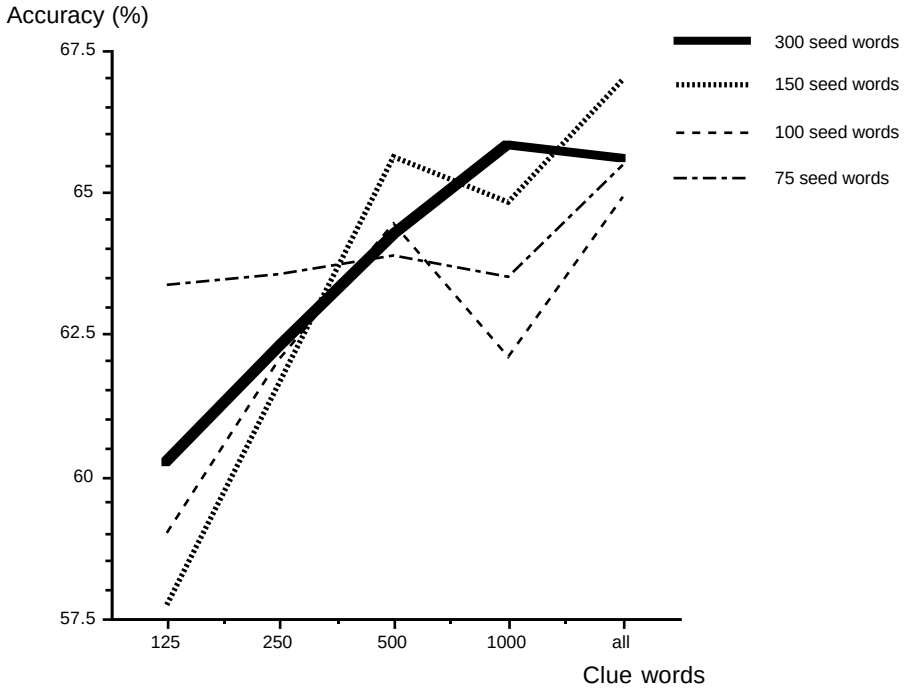


Figure 41. Accuracy of the disambiguator against number of clue words. Note the origin of the y-axis is at 57.5%.

A further result of the experiments was to confirm previous research *[Yarowsky 93]* showing that pseudo-words were a good method for pre-testing a disambiguation strategy.

6.2.4 Conclusions of the experiments: a change of tack

The results of the pseudo-word experiments seem to indicate that the accuracy of the disambiguator is not at the levels that the results in Chapter 5 suggest are required for successful use in an IR system. Therefore, if the disambiguator is to be used at all, it will have to be integrated into an IR system in manner different to that envisaged in the experiments of that chapter.

Almost all disambiguators are used to select one sense for each word occurrence. As was discussed in Section 4.3, the senses defined in dictionaries and thesauri are somewhat arbitrary delineations of the meanings of a word. It is possible that a word will be used in such a way that its meaning (as perceived by a reader) fits a number of the senses found in a dictionary or thesaurus. Even if a reader can categorise a word occurrence into a single pre-defined sense, research (in Section 4.3) has shown that other readers may categorise that same occurrence into another sense.

When choosing the sense of a particular word occurrence, a disambiguator assigns a score to each of the senses of that word. The score indicates the disambiguator's confidence of that sense being the intended sense of that word occurrence. Therefore, rather than representing an

occurrence by just the highest scoring sense, one could represent an occurrence by all the senses, each weighted by its confidence score. In using such a representation method, the problems discussed above are addressed. Therefore, this *full-sense representation* method was adopted for use in the next stage of testing: measuring the accuracy of the disambiguator.

7 Disambiguation accuracy of real words

The next stage in building the disambiguator was to measure its accuracy on actual ambiguous words taken from the document collection on which the final IR experiments will be run. It was hoped that this test would provide a more representative idea of the disambiguator's accuracy. The chapter first discusses potential changes to the disambiguator's design now that it is disambiguating real words. Next, the method of testing it is described and this is followed by the results of the accuracy tests.

7.1 Issues raised when disambiguating real words

Disambiguating real ambiguous words raises the possibility of using other NLP tools to preprocess the words before disambiguation take place, the most obvious tool being a grammatical tagger, a number of which have been released into the public domain. Using such a tool, the words to be disambiguated would be tagged with their grammatical category (noun, verb, adjective, pronoun, determiner, etc.) thus reducing the number of senses from which the disambiguator must choose. The output of the disambiguator is intended for use by an IR system which is generally more influenced by semantic rather than syntactic similarity. Indeed those who have applied a grammatical tagger to a test collection with the aim of improving the representation of that collection and thus increase retrieval effectiveness have so far had little success *[Sacks-Davis 90]*, *[Smeaton 92]*.

It is unclear if this syntactic partitioning of senses is desirable, for example, if we take the word 'bank', WordNet defines it as having fifteen senses: nine for the noun; six for the verb (shown in Figures 42 & 43). Suppose the disambiguator disambiguated a noun tagged occurrence of the word 'bank' as noun sense 8, an economic sense, and then disambiguated a verb tagged occurrence of 'bank' as verb sense 5, another economic sense. There is no structure in WordNet to show that these two senses are semantically related, an IR system trying to measure some form of similarity between these two occurrences would have nothing to go on. If the tagger were not used and we were to adopt the full-sense representation of an ambiguous word as discussed in Section 6.2.4, it is likely when disambiguating the noun and verb occurrence of the word 'bank', that both economic senses (noun sense 8 and verb sense 5) would both be assigned high confidence scores to both occurrences. By basing sense scoring in this way, word occurrences gain a fuller sense representation. One could even think of this as a form of sense conflation. Therefore, it was decided not to use a grammatical tagger in these experiments.

Now that this final issue was resolved, measuring the disambiguator's accuracy could proceed.

Sense 1
bank, side -- (*sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the side of the river and watched the currents"*)

Sense 2
depository financial institution, bank, banking concern, banking company -- (*a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home"*)

Sense 3
bank -- (*a long ridge or pile; "a huge bank of earth"*)

Sense 4
bank -- (*an arrangement of similar objects in a row or in tiers; "he operated a bank of switches"*)

Sense 5
bank -- (*a supply or stock held in reserve esp for future emergency use; "the Red Cross has a bloodbank for emergencies"*)

Sense 6
bank -- (*the funds held by a gambling house or the dealer in some gambling games; "he tried to break the bank at Monte Carlo"*)

Sense 7
bank, cant, camber -- (*a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force*)

Sense 8
savings bank, coin bank, money box, bank -- (*a container (usually with a slot in the top) for keeping money at home; "the coin bank was empty"*)

Sense 9
bank, bank building -- (*a building in which commercial banking is transacted; "the bank is on the corner of Nassau and Witherspoon"*)

Figure 42. Nine senses of the noun ‘bank’.

Sense 1
bank, tip laterally -- (*of boats and aircraft*)

Sense 2
bank, enclose with a bank -- (*as of roads*)

Sense 3
bank, have an account, keep money -- (*do business with a bank*)

Sense 4
bank -- (*be in the banking business*)

Sense 5
deposit, bank -- (*put into a bank account*)

Sense 6
bank, cover with ashes -- (*of fires, to control the rate of burning*)

Figure 43. Six senses of the verb ‘bank’.

7.2 Measuring the disambiguator’s accuracy

Whichever ambiguous words were chosen for the testing of the disambiguator, their occurrences in Reuters would have to be disambiguated manually. Given the recent literature outlining the inconsistency of human sense disambiguation, reviewed in Section 4.3, it seemed

that to employ just one person performing this supposedly straightforward task would not provide a sufficiently accurate set of disambiguated words. Therefore, it was decided that the senses would be manually disambiguated by two people working independently from each other.

Assessing the accuracy of the automatic disambiguator when disambiguating pseudo-words was easy, with pseudo-words there was always one correct pseudo-sense to be chosen: the disambiguator was either right or wrong. When assessing the disambiguator against human sense tagging, there is a problem: what if (as was expected) the two manual disambiguators tagged a word occurrence with different senses? To accommodate this uncertainty in sense tagging, a metric was devised that measured the correlation between the disambiguator's and the human's sense tagging answers and this is now described.

7.2.1 The similarity measure

In trying to decide what sort of similarity measure to use, one first needs to see what is to be measured. Table 14 shows the possible output of an automatic disambiguator after processing a word occurrence with five senses. We can see that a confidence score is assigned to each of the senses by the disambiguator.

Sense	Score
1	136
2	136
3	0
4	165
5	150

Table 14. Confidence scores assigned by a disambiguator for a word with five senses.

One can think of manual sense tagging in terms of confidence scores as well: by selecting a sense for a word in a certain context, the manual tagger has assigned a maximum score to that sense and implicitly has assigned zero to all other senses. When combining the tagging results of multiple manual taggers, we can add their sense scores together, see Table 15.

Sense	Tagger A	+	Tagger B	=	Sum
1	0		0		0
2	0		0		0
3	0		0		0
4	1		0		1
5	0		1		1

Table 15. Combining the output of two manual taggers.

Treating manual tagging data in this manner allows for inconsistent tagging by people, but it also allows the taggers more freedom when tagging word occurrences: they can assign multiple senses or even a ranking of senses to a word occurrence. In Table 16 tagger B is fairly certain that the word occurrence is sense 2 but is not completely sure and wants to express the possibility that the occurrence may be sense 5.

Sense	Tagger A +	Tagger B +	Tagger C =	Sum
1	0	0	0	0
2	1	0.75	0	1.75
3	0	0	0	0
4	0	0	0	0
5	0	0.25	1	1.25

Table 16. Combining the output of three manual taggers, one of whom is not completely sure which sense the word occurrence is.

When comparing human and automatic disambiguation it would seem sensible that the correlation measure calculates the relative similarity in the scores assigned to each word sense. The measure that was chosen was called the variation distance and it is now explained.

To compute the variation distance, the scores assigned to each set of word senses are first normalised. As can be seen in Table 17, the confidence scores assigned to each sense set are scaled so that they sum to one. In the case where all senses scores of a set are zero, i.e. no judgment has been made on any of the senses of a word occurrence, these senses are assigned an equal score which is normalised to sum to one. The variation distance between these two normalised sets of senses is measured as follows. Taking each sense in turn, the absolute difference between the two scores assigned to that sense is calculated. These differences are summed to give the variation distance between the two sets of senses. This distance measure is defined in Equation 8.

Sense	disam1	disam2	|Diff|
1	0.23	0.00	0.23
2	0.23	0.00	0.23
3	0.00	0.00	0.00
4	0.28	0.50	0.22
5	0.26	0.50	0.24
			+
	variation	distance	0.92

Table 17. Calculation of the variation distance.

$$\text{variation distance} = \sum_{s \in S} \left| disam1_s - disam2_s \right| \qquad (8)$$

S = set of senses of a word

When sense sets are normalised to sum to one as shown here, the variation distance has the range [0..2]. The value of zero indicates an exact correlation between the sense sets, two indicates no correlation.

7.3 The words to be disambiguated

The next stage in the testing was to select and then manually disambiguate the occurrences of a number of words. The selection of these words was performed manually. Though many of the candidate words had a large number of senses, many of these senses were obscure uses and were judged unlikely to appear in the Reuters collection. Therefore, it was decided that for a word to be selected, at least two of its senses should have a reasonable chance of being used in the Reuters collection. This proved to be quite a restrictive factor, reducing the candidate words to a small number. Five words were manually chosen that satisfied this criteria: ‘assembly’, ‘carrier’, ‘duty’, ‘maintenance’, and ‘platform’. They occurred a total of 736 times in Reuters.

7.3.1 The manual tagging

As has already been stated, two manual taggers were used to disambiguate the five words. These words were disambiguated with respect to the word sense definitions of WordNet. There was a question of just how to present the sense definitions to the manual taggers as WordNet defines the intended meaning of a sense in a number of ways. The three most common are by its synonyms, by a written definition known as the *gloss*, and by that sense’s position in the WordNet hypernym hierarchy. An example of all three definitions is shown in Figure 44. Clearly the gloss is the most explicit form of sense definition, the other two define the sense more implicitly.

Initially it was not realised that the gloss definitions existed, so the manual taggers were asked to disambiguate the 736 occurrences of the five ambiguous words using just the synonyms and hierarchy parts of the sense definition. Once this mistake had been realised, the same two manual taggers disambiguated all the word occurrences again, this time the sense definitions were embellished with the gloss. Neither tagger discussed the experiment with the other until they had completed both runs. Over both of these runs, for each of the occurrences of each of the ambiguous words, the manual taggers were shown the full paragraph in which that occurrence appeared, an example of which is shown in Figure 45. The taggers were asked to select one or a number of the WordNet senses that best fitted the word occurrence. On occasion, the taggers assigned more than one sense to an occurrence, though this happened less often than anticipated. In total the taggers assigned multiple senses to 3% of the occurrences.

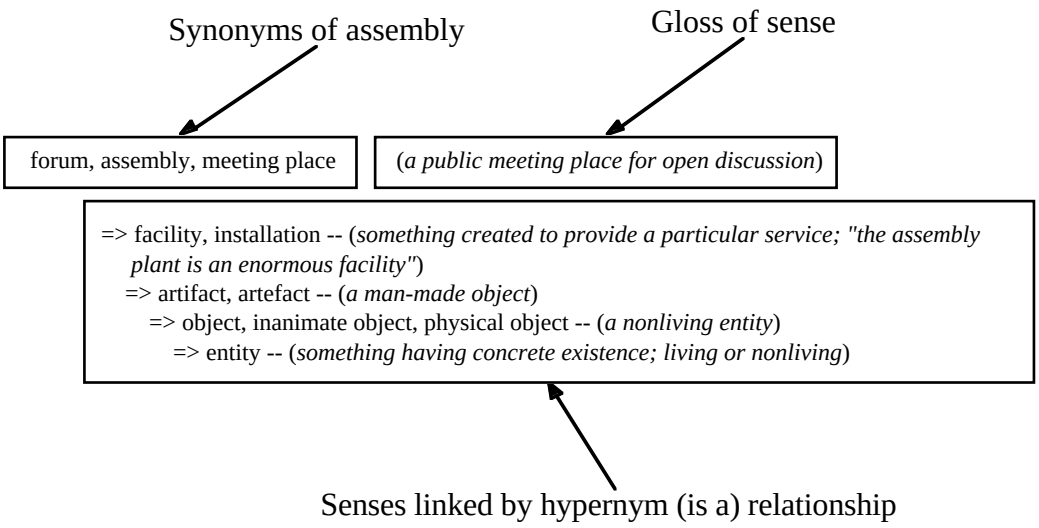


Figure 44. A sense of the word 'assembly' as defined in WordNet.

Prime Minister Li Gun mo told the eighth Supreme People's **assembly** in Pyongyang that North Korea intends to increase international trade by 220 pct in the period 1987 93 gross industrial output by 90 pct and agricultural production by 40 pct according to the North Korean Central News Agency monitored here. REUTER

Figure 45. The occurrence of an ambiguous word as shown to manual taggers.

Across both of these tagging runs (with and without the gloss) a comparison was made of the sense tagging consistency between the two. The results of this comparison are shown in Table 18 where for each word, it shows the number of occurrences of that word, the number of senses that word has, the number of senses the two manual taggers agreed on for each of the two runs, and the number that would be agreed on had they randomly selected senses. For these comparisons, a word sense match meant that the taggers agreed exactly on the sense(s) they thought a word occurrence was being used in.

Word	Occs	Num sns	No gloss		With gloss		Random selection	
assembly	116	5	26	*22%*	92	*79%*	23	*20%*
carrier	155	10	125	*81%*	145	*94%*	16	*10%*
duty	243	3	190	*78%*	204	*84%*	81	*33%*
maintenance	113	5	93	*82%*	96	*85%*	23	*20%*
platform	109	4	9	*8%*	92	*84%*	27	*25%*
Total	**736**		**443**	***60%***	**629**	***85%***	**170**	***23%***

Table 18. The consistency of tagging between the manual taggers.

As can been seen from the table, the level of agreement between the taggers is higher when the gloss is included in the definition (85% over 60%). In discussions with the taggers after com-

pleting their tasks, both said that they better understood the intended sense of the definitions after reading the gloss. Nevertheless, they still failed to agree on the senses of over 100 word occurrences.

Table 19 shows the number of word occurrences that the taggers changed their minds on between the two runs. As can be seen, they altered the tags on approximately the same number of occurrences, although these changes happened for different words: tagger A re-tagged the occurrences of 'platform' the most; whereas tagger B changed 'assembly'.

Word	**Occs**	**Num sns**	**Tagger A**		**Tagger B**	
assembly	116	5	38	*33%*	68	*59%*
carrier	155	10	11	*7%*	24	*15%*
duty	243	3	32	*13%*	40	*16%*
maintenance	113	5	14	*12%*	19	*17%*
platform	109	4	103	*94%*	15	*14%*
Total	**736**		**198**	***27%***	**166**	***23%***

Table 19. The consistency of each tagger across the two runs.

Note that levels of consistency between taggers is all that one can address here. It is not possible to objectively measure the correctness of the taggers as this would require a perfectly disambiguated text to compare against and as has already been established (see Section 4.3.1) this cannot be achieved as manual sense taggers are inconsistent when conducting this subjective task.

Now that a set of words were manually disambiguated, and a metric was devised to measure the accuracy of the disambiguator, its testing could proceed.

7.4 Does the disambiguator disambiguate real words?

An initial test of the disambiguator was conducted to see if it worked at all on real words. Accuracy is measured using the variation distance. This metric, however, is not familiar: it ranges between zero and two, but it is not clear what constitutes a good score within that range. To judge the accuracy of the disambiguator, the value of its variation distance needs to be assessed in the context of values resulting from other disambiguation strategies. Those used here were randomly selecting word senses, and always selecting the most common sense of a word, as defined in WordNet. Table 20 shows the variation distances for the three disambiguation strategies over the five words tested. The random sense selection strategy provides a lower bound with which to compare the other disambiguation strategies. The different values

of variation distance for this strategy across the five words is directly proportional to the number of senses each word has. For example, ‘duty’ has the smallest number of senses (three), therefore the variation distance for this word is the smallest, as random sense selection is correct one in three times.

word	number of senses	random selection	most common	disam-biguator
assembly	5	1.54	1.84	1.43
carrier	10	1.78	1.57	1.53
duty	3	1.24	1.87	1.05
maintenance	5	1.54	0.20	1.45
platform	4	1.45	0.19	0.40
	average	1.51	1.13	1.17
	av without platform	**1.53**	**1.37**	**1.36**

Table 20. The accuracy of the disambiguator against two simplistic disambiguation strategies.

Comparing the results of the random sense scoring strategy and the disambiguator in its various configurations shows that in every case the disambiguator is more accurate than the random strategy. This would seem to indicate that the disambiguator is working, though hardly at an ideal level.

To give some idea of the workings of the disambiguator, Figure 47 shows the top ranked clue words gathered for the five senses of ‘assembly’ (shown in Figure 46). For senses 1 to 3, these words appear to be good clue words for their respective senses, for senses 4 and 5, however, it is harder to see how the clue words will help to identify their senses.

Sense 1
assembly -- (*a group of machine parts that fit together to form a self-contained unit*)

Sense 2
fabrication, assembly -- (*the act of constructing something (as a piece of machinery)*)

Sense 3
assembly -- (*a group of persons gathered together for a common purpose*)

Sense 4
forum, assembly, meeting place -- (*a public meeting place for open discussion*)

Sense 5
assembly, assemblage, gathering -- (*the social act of assembling*)

Figure 46. The five senses of ‘assembly’.

We can see from Table 20 that the strategy of always selecting the most common sense of a word has variable success. For two of the words, ‘maintenance’ and ‘platform’, this strategy is the best, but for ‘assembly’ and ‘duty’, it is the worst. Although on average it performs

1	2	3	4	5
drill	design	senate	libyan	meeting
grinder	fabrication	court	troop	fight
sander	niagara	congress	chad	election
hammer	casting	house	field	war
field	mohawk	tribunal	main	insurance
equipment	valve	delaware	capture	indemnity
system	owner	chancery	airport	gathering
machinery	ball	chamber	worth	virginia
polisher	composition	leader	strip	casualty
division	acknowledge	crowd	ouadi	office
certain	consumer	diet	north	mutual
plant	nuclear	legislature	libya	celina
microcomputer	engineer	meeting	forum	battle
percussion	suit	chapter	depot	party
machine	mile	democratic	army	concentration
power	steel	approval	facility	mobilization
brown	plant	robert	storage	congregation
tool	involve	class	force	resistance
application	point	parliament	northern	engagement
business	manufacture	committee	link	convention
product	facility	fail	network	minister
network	power	win	schedule	speech
combine	restraint	require	central	vote
own	recording	reagan	report	instrument
press	invention	budget	store	transfer
datum	intestine	file	datum	nakasone
26	formation	symbolize	cash	session
versatile	exagerrat	quintette	warehouse	defense
vancouver	defeather	president	territory	oppose
treadmill	syracuse	multitude	racetrack	estate
southwest	suppress	household	operation	visit
simulator	puncture	gathering	helikopte	drill
operation	optimism	entourage	antitumor	award
northwest	forestry	accugraph	treasure	meet
mainframe	erection	sentence	surround	legislation

Figure 47. Top 50 clue words for each of the five senses of the word ‘assembly’.

well, this strategy is unlikely to be of any use in an IR context. Using it to disambiguate a document collection would add no information. All occurrences of each word would be assigned the same sense.

The bottom row of Table 20 shows the average variation distance without the values for ‘platform’. These were found to be so different from those of the other words, they considerably affected the overall average[16]. So much so, it was feared that decisions about the best configuration of the disambiguator would be based on the disambiguator’s accuracy on this word alone. Therefore, it was decided to ignore the measures for this word and base all decisions about the disambiguator on its disambiguation accuracy of the other four.

16. An examination of the clue words the disambiguator derived for ‘platform’ revealed that the unusual measures were due to just one clue word which was acting as a particularly good indicator for the most common sense of ‘platform’.

7.5 Measuring the disambiguator's accuracy on real words

From the results of pseudo-word tests, it was found that the disambiguator's accuracy varied depending on the number of clue words used. It was important to try to establish the best clue word configuration at this point, as the speed of the disambiguator was such that there would only be time for one disambiguation of the Reuters collection. In a series of tests, the accuracy of the disambiguator was measured with different values in an attempt to establish the number of clues that would produce the best disambiguation accuracy. The results of these tests are shown in Table 21. The table shows the variation distance measures for each of the

	random selection	most common	25	50	100	150	200
assembly	1.54	1.84	1.37	1.43	1.50	1.38	1.31
carrier	1.78	1.57	1.56	1.53	1.57	1.71	1.73
duty	1.24	1.87	1.03	1.05	1.08	0.68	0.82
maintenance	1.54	0.20	1.46	1.45	1.47	1.61	1.58
platform	1.45	0.19	1.41	0.40	0.97	1.10	1.23
av without platform	**1.53**	**1.37**	**1.35**	**1.36**	**1.41**	**1.34**	**1.36**

Table 21. Accuracy against number of clue words to be used by disambiguator.

five words (although 'platform' was ignored), over the two simplistic disambiguation strategies, and the five clue word configurations (25-200) of the automatic disambiguator.

Unlike the pseudo-word experiments, there is no clear trend in the disambiguation accuracy as the number of clue words is varied. By examining the figures in more detail, however, a factor was found that does vary consistently with the number of clue words: the number of times the disambiguator makes a judgment on at least one of the senses of a word occurrence, i.e. where at least one of the senses is assigned a confidence score. This factor was examined.

Table 22 shows the number of word occurrences on which the disambiguator made a judgement. It shows for each of the five words across the five clue word configurations, the accuracy of the disambiguator and next to each measure, the percentage of word occurrences that the disambiguator attempted to disambiguate. The accuracy is measured over just those word occurrences where disambiguation was attempted. For example, the disambiguator configured to gather 25 clue words only attempted 55% of the ambiguous word occurrences and had an overall accuracy of 1.18 for those occurrences.

In this table we can see that as the number of clue words is increased, the number of word occurrences attempted by the disambiguator also increases but this in turn results in a decrease in accuracy. As this decrease is not too large, it was decided that occurrences attempted

	random selection	25		50		100		150		200	
assembly	1.54	1.34	*87%*	1.42	*90%*	1.50	*100%*	1.38	*100%*	1.31	*100%*
carrier	1.78	1.45	*67%*	1.47	*78%*	1.54	*89%*	1.71	*100%*	1.73	*100%*
duty	1.24	0.59	*34%*	0.74	*43%*	0.92	*61%*	0.68	*100%*	0.82	*100%*
maintenance	1.54	1.35	*48%*	1.34	*52%*	1.42	*65%*	1.62	*79%*	1.57	*91%*
platform	1.45	1.44	*28%*	0.4	*100%*	0.97	*100%*	1.10	*100%*	1.23	*100%*
av without platform	**1.53**	**1.18**	***55%***	**1.24**	***62%***	**1.34**	***76%***	**1.35**	***96%***	**1.36**	***98%***

Table 22. Re-examination of tests results, concentrating on the number of ambiguous words the disambiguator makes a judgement on.

should have a higher priority over accuracy when choosing the number of clue words. Therefore, 150 clue words was chosen as the best compromise between these two factors.

7.6 Summary

This and the previous chapter have described the preparation for the following experiments: retrieving from a disambiguated collection. A disambiguator was constructed, and a means of testing it devised. The choice of disambiguation strategy was one that attempted to train the disambiguator to the linguistic style and cultural references of the corpus to be disambiguated. Implementation of the disambiguator was adjusted to allow it to work with available resources (i.e. WordNet). A means of testing its accuracy was devised. In this test a method of more fully representing the senses of a word occurrence was used along with a means of measuring the correlation between two such representations. This method was found to work well the sense tagging inconsistencies of manual disambiguators. It was also anticipated that the method would better represent the possible senses of a word occurrence. The disambiguation accuracy test revealed the disambiguator to be working, though not as well as was hoped for.

8 Retrieving from a disambiguated collection

This chapter describes the final experiment: a test of the retrieval effectiveness of an IR system working with an automatic word sense disambiguator. It first reports the issues arising from the disambiguator's application to the Reuters collection, followed by the necessary adjustments made to the IR system to allow it to process this sense tagged collection. The results of the experiments are presented, and conclusions are drawn.

8.1 Disambiguating the documents of the Reuters test collection

It has been noted by a number of researchers that when a word is used a number of times in a document, there is a very high chance that it will be used in the same sense throughout that document. Yarowsky *[Yarowsky 95]* has documented this feature and has also built a disambiguator that exploits it. Through its use he has shown an improvement in disambiguation accuracy. Given this confirmation that the 'one sense per document' rule holds, it was decided to use it when disambiguating the Reuters collection as it would simplify the disambiguation process. One aspect that is not completely clear from Yarowsky's paper is how large a document can be for the rule to still hold. As the corpus he disambiguated was a collection of newspaper articles, however, it was judged that the rule would apply to the articles of the Reuters collection.

8.2 Adjusting the IR system to accommodate senses

In order to use the word sense information from the disambiguator, the IR system had to be adjusted. The system is based on the binary probabilistic model of IR proposed by Robertson and Sparck Jones *[Robertson 76]*. This model cannot easily accommodates the full-sense representation produced by the disambiguator. An ad hoc approach was taken to adjust the IR system so it could handle this sense information.

8.2.1 Devising a method to accommodate senses

There is very little other work on accommodating disambiguation information in an IR system. Most *[Voorhees 93]*, *[Zernik 91]* have adopted the approach taken in Chapter 5 of using a *single sense representation*: replacing a word by a single word sense. Schütze & Pedersen *[Schütze 95]* however (reviewed in Section 5.6.2), used a *multi-sense representation*. They have shown this approach to improve retrieval effectiveness when compared to that gained from using a single sense. In their system, a word occurrence was represented by a fixed number of its highest scoring senses. These senses were treated as being of equal importance. When comparing the representations of two word occurrences, if there were any senses in

common between the two, the occurrences were judged to have matched. As has already been described, the ‘senses’ that Schütze & Pedersen’s disambiguator worked with are quite different from those found in dictionaries or thesauri. As such, it was unclear if the binary multi-sense matching scheme was appropriate for this work. Nevertheless, given the success of their system, it was decided to try it. It was anticipated that the full-sense representation coupled with the variation distance described in Section 7.2 would provide a more subtle means of measuring the similarity of word senses and this was chosen as the main representation method.

8.2.2 Implementing a method to accommodate senses

The main part of the IR system that needed to be adjusted to accommodate the sense representation was the document relevance score function. The definition of the adjusted function is shown in Equation 9, as can be seen it is a sum over those terms that co-occur in both query and document. The calculation of the *idf* of a term was left as normal. Though it may have been beneficial to use a more complex weighting function that included sense information, this possibility was not explored. The degree of sense match between a query and document term is measured by the *cor* function. Like *idf*, this measure has the range [0..1], one indicates an exact correlation between senses, zero indicates no correlation. The definition of this measure varies depending on the type of sense matching scheme adopted: either the full-sense/variation distance (Equation 10), or the single sense/binary matching scheme (Equation 11)[17]. By combining the *idf* and sense information in this manner, the *idf* of a term will still feature in a document’s relevance score even if there is no sense match.

$$\sum_{t \in Q \cap D} idf(t) + cor[senses(t, Q), senses(t, D)] \tag{9}$$

Q = set of terms in query
D = set of terms in document
t = a term
$senses(t, N)$ = sense representation of term t as it occurs in N
idf = inverse document frequency of a term
$correl$ = correlation measure between two sense representations, [0..1]

$$cor(s_1, s_2) = 1 - \frac{vd(s_1, s_2)}{2} \tag{10}$$

s = sense representation of a term

vd = variation distance between two sense representations, [2..0]

17. An initial version of this measure was more flexible, allowing a match between the top n ranked senses. Experimental results showed this measure to be little different from the simpler version shown here.

$$cor(s_1, s_2) = bin(s_1, s_2) \quad (11)$$

s = sense representation of a term

$$bin = \begin{cases} 1, \text{ if the top ranked sense matches between term occurrences} \\ 0, \text{ otherwise} \end{cases}$$

8.3 The disambiguation experiments

With a disambiguated version of the Reuters collection and an adjusted IR system, the final set of experiments could now proceed: to test if a disambiguator could be used to improve the effectiveness of an IR system. This chapter describes the execution and results of these experiments along with the conclusions drawn.

8.3.1 Recalling the experimental set up

The use of Reuters as a test collection is unconventional in that its queries are automatically generated using relevance feedback. An advantage of this method is that the size of these queries can be easily varied by adjusting the number of words relevance feedback produces. In Chapter 5 it was shown how query size played an important role in the relationship between ambiguity and retrieval effectiveness. Because of this, effectiveness was shown in relation to query size. It was measured using f_{max}, with α set at various values to alter its emphasis on recall or precision.

Figure 48 shows a graph with two plots of retrieval effectiveness against query size. Here f_{max} is measured with α=0.5 which results in an equal emphasis on recall and precision. The two sets of effectiveness figures in this graph are an IR system ignoring disambiguation information, and a randomly generated lower bound. As in Chapter 5, these will appear in all the f_{max} graphs shown below.

8.3.2 The results

The first experiment to be conducted was also the main experiment of the thesis, comparing the effectiveness of an IR system ignoring disambiguation information against two systems using it. The results are shown in Figure 49. The first, using a single sense/binary matching scheme labelled ‘Binary correlation’, and the second, a full-sense/variation distance based measure labelled ‘Full correlation’. As can be seen, across almost all query sizes the incorporation of the disambiguation information reduces effectiveness.

As disappointing as this result is (to the author anyway), it would seem to be a reflection of the disambiguation simulation results in Chapter 5. They showed accuracy to be important in determining whether the application of a disambiguator would improve or degrade retrieval

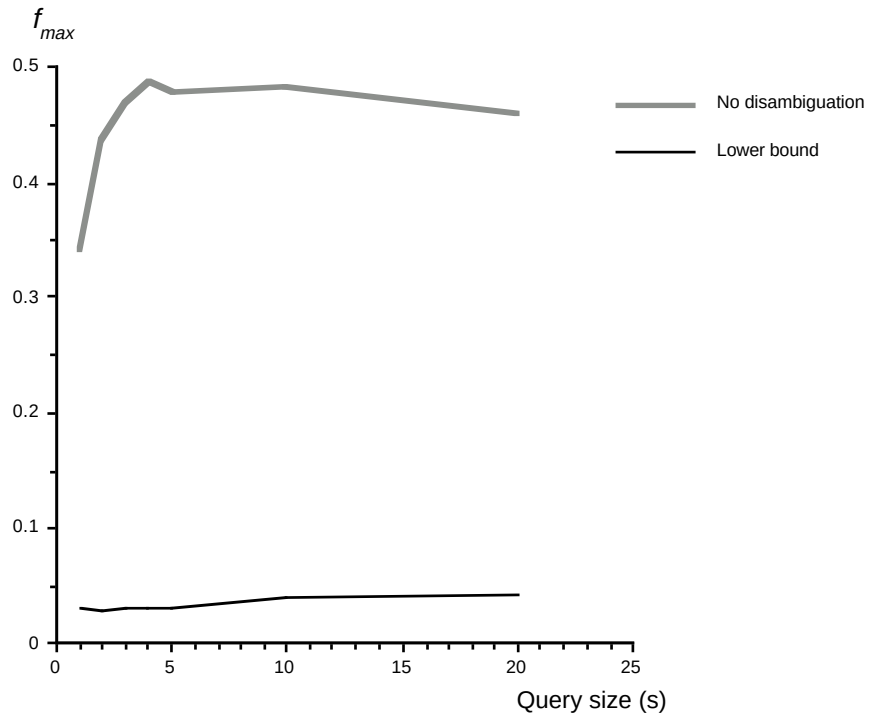


Figure 48. Upper and lower bounds on retrieval effectiveness for this set of experiments.

effectiveness. Testing of the disambiguator's capabilities have shown it was far from perfect. It would appear that the disambiguator was not accurate enough.

One thing to notice from this graph is that for most of the query sizes tested, better effectiveness was achieved using the binary correlation measure instead of the full correlation measure. If we recall Equation 9 and Equation 11 we can see that by using the binary correlation measure, whenever the correlation is zero, only the *idf* is used in the calculation of a document's relevance score. From the graph we can see that using just the *idf* produces better effectiveness. So, by using the binary correlation measure, less use of the disambiguation information is made and therefore it does less badly.

Looking at single word queries

There was a part of the graph, however, where disambiguation appeared to improve effectiveness and the figures for this case, namely single word queries, were plotted as a standard RP graph for closer examination, see Figure 50.

From the graph we can see that across all levels of recall, the precision of the IR system using the full correlation measure is higher than that of the system that does not use the disambiguation information. Again, this result would seem to confirm the disambiguation simulation results of the previous chapter that indicated that disambiguation would be most beneficial to an IR system when it was retrieving from short queries. Although it has only been shown for

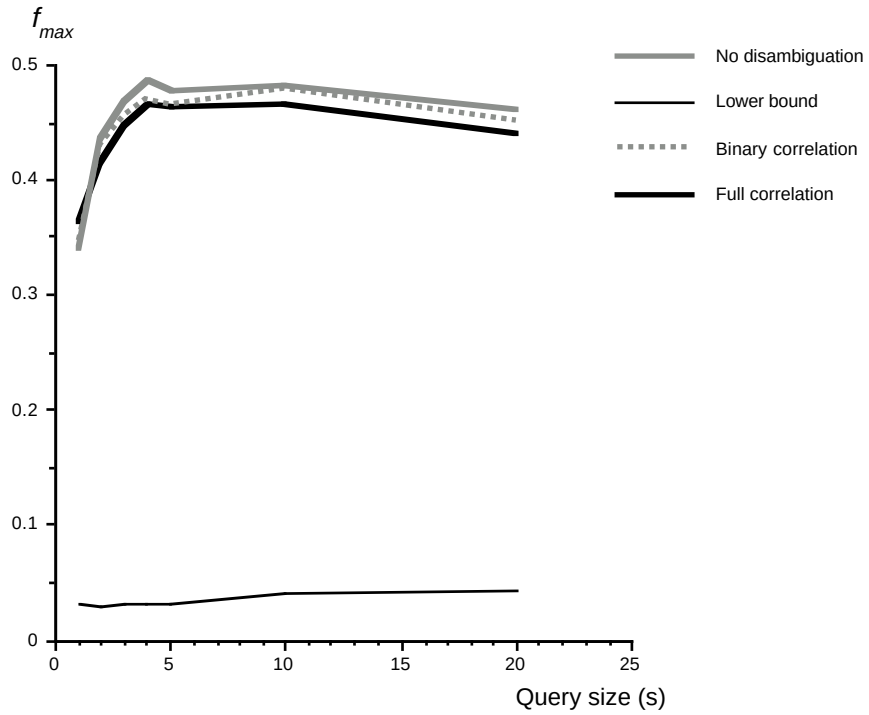


Figure 49. Effectiveness when using, and not using, disambiguation information, α=0.5.

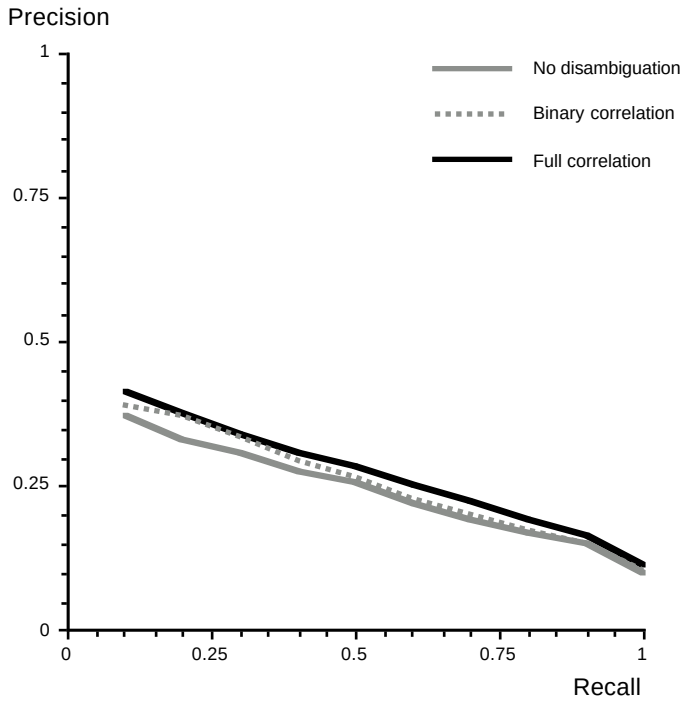


Figure 50. Effectiveness for one word queries.

this one type of query, it is believed that this result constitutes the first large scale experiment where a disambiguator based on predefined senses (i.e. senses defined in some language reference work: dictionary, thesaurus, etc.) has been shown to improve the effectiveness of an IR system. In addition, the graph shows that, in this single word query case, the use of the full correlation measure produces better effectiveness than the binary correlation.

Examining the results more closely

While analysing the effectiveness figures for the various query sizes, it was noticed that the system using the binary correlation measure resulted in slightly higher precision at low recall than the system using no disambiguation information. This improvement can be seen in the RP graph shown in Figure 51. This graph shows the effectiveness when retrieving on a five word query.

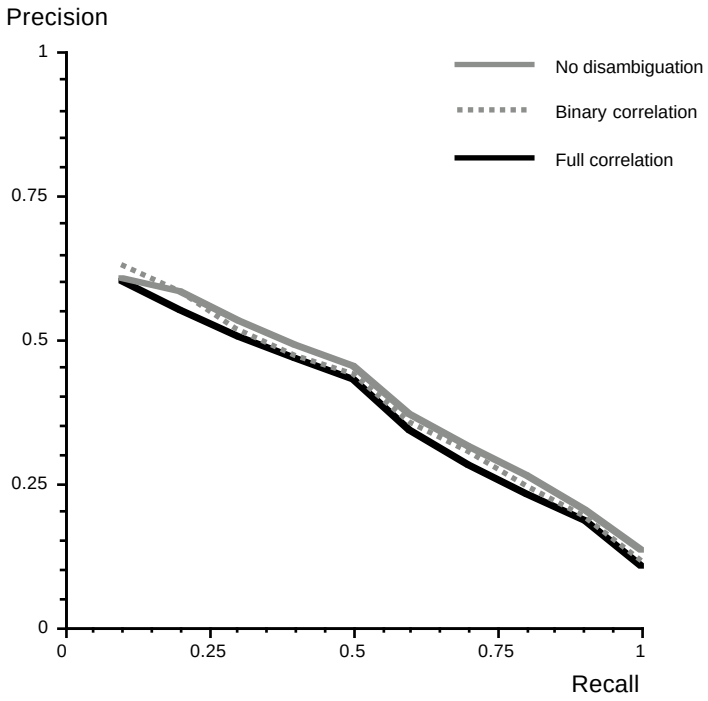


Figure 51. Effectiveness for five word queries.

This slight improvement was unlikely to show up in the f_{max} effectiveness graph in Figure 49 as α was set to 0.5, causing f_{max} to place equal emphasis on recall and precision. By replotting, the graph with α=1.0, emphasising precision over recall, the improvement can be examined (see Figure 52). Setting α to this value, causes f_{max} to become the highest precision value of a set of RP figures. This has the effect of measuring an IR system's effectiveness at the top of a document ranking.

As can be seen in Figure 52, some form of improvement is gained by using the binary correlation measure. It is, however, very small and as has already been stated, precision measured at low recall is particularly sensitive to small changes in the position of relevant documents in a ranking. As we can see from Figure 51, the improvement in precision measured at recall value 0.1 is approximately 0.025. The average number of relevant documents per query in Reuters is 82, so at recall value 0.1, eight relevant documents have been retrieved. The precision of the system without disambiguation is 0.65 and the precision of the disambiguation based system

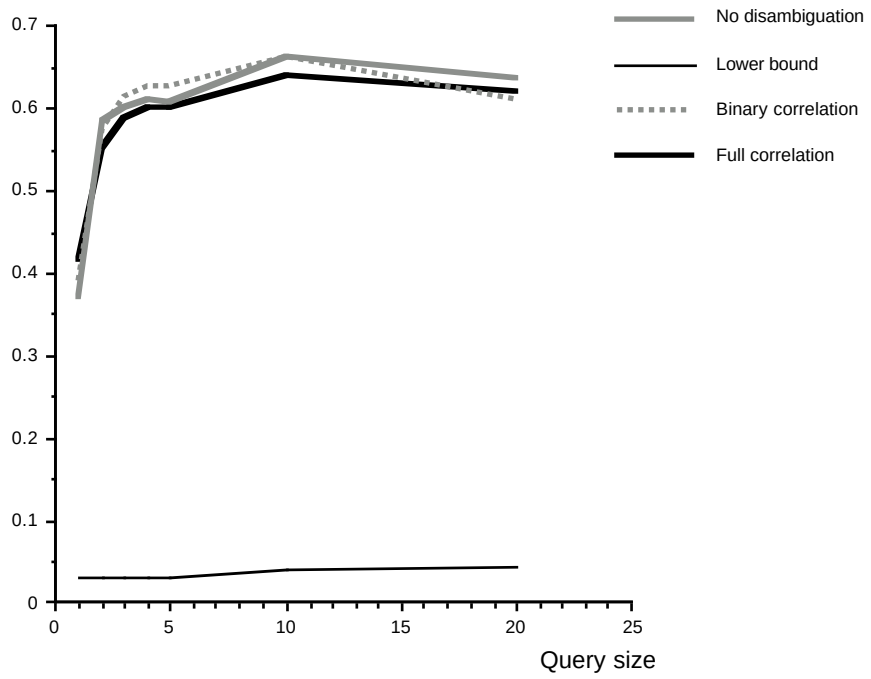


Figure 52. Effectiveness when using, and not using, disambiguation information, α=1.0.

with a binary correlation measure is 0.675. These precision values show that the eight relevant documents are among the first twelve documents retrieved. The 0.025 precision difference means these eight documents appear on average half a rank position higher. What might cause this very slight improvement is not clear. The result does fit, however, with the notion of disambiguation as a precision enhancing method for IR.

8.4 Conclusions

In this chapter the effectiveness of an IR system retrieving from a disambiguated collection was measured. The main result of these experiments is the drop in retrieval effectiveness resulting from the use of the sense information produced by the disambiguator. As has already been stated, it is believed that the relatively poor accuracy of the disambiguator is to blame for this and Section 9.2 contains some suggestions on how this might be improved. Despite the errors in its output, use of the disambiguator's sense information was found to improve retrieval effectiveness for very short queries and this improvement is believed to be the first demonstration of disambiguation (based on a dictionary or thesaurus) being beneficial to retrieval effectiveness.

Another aspect of the experiments was the comparison of two methods for accommodating sense information in an IR system: full-sense/variation distance; and single-sense/binary-matching. As only a single case of using disambiguation information was shown to be benefi-

cial to retrieval effectiveness, no conclusions can be made on the merits of one method over another. The result of this one case, however, showed that use of the full-sense/variation distance method produced better effectiveness than use of the other. This is countered slightly by the result showing the single-sense/binary-matching method marginally improving effectiveness measured at low recall.

Although taken on their own, these results are disappointing, in the wider context of the other ambiguity experiment of the thesis (see Chapter 5), the results are complementary. Together, both experiments paint the following picture of the impact of ambiguity and disambiguation on retrieval effectiveness. Ambiguity is not as significant a problem to effectiveness as might have been thought, except for retrievals based on short queries. Disambiguation is only useful to an IR system if the disambiguator is accurate. The benefits of a disambiguator are greatest for retrievals based on short queries.

9 Contributions and future work

This chapter presents a description of the contributions arising from the work presented in this thesis and concludes with a discussion of possible future work.

9.1 Contributions of the work

The contributions of this work are as follows: retrieval effectiveness experiments that incorporate variable query size; a pseudo-word based testing methodology; experiments to determine to what extent pseudo-words realistically simulate ambiguous words; a full-sense representation of the senses of a word; and the conclusions drawn from the experimental results.

9.1.1 Experiments with variable query size

As it was anticipated that query size would play an important role in the relationship of ambiguity and IR, the experiments were based on a test collection where query size could be varied. The subsequent results were expressed as three sets of figures: recall, precision, and query size. To allow better analysis of these figures, a method of reducing the recall/precision figures to a single number was formulated. Based on Van Rijsbergen's f measure, f_{max} was argued as being a better statistic for this task than the widely used standard: average precision.

Size of queries is an experimental variable rarely examined in IR experiments and yet from the results presented here, clearly can play a significant role in deciding if a retrieval technique is useful or not. It would be hoped that this aspect of retrieval is paid closer attention in the future.

9.1.2 Pseudo-word testing methodology

A testing methodology based on pseudo-words was devised. It proved to be a fast and effective means of exploring the relationships between retrieval effectiveness, ambiguity, and disambiguation. Using this methodology, a number of experiments were conducted, the results of which provided a much greater understanding of these relationships. Because of these initial experiments, it was possible to be better prepared for the much larger ambiguity experiments undertaken afterwards. In general, it would appear that the use of simulation in IR experiments can provide good initial information to an experimenter, provided that the simulation is accurate.

9.1.3 Appropriateness of pseudo-words

As pseudo-words are a simulated form of lexical ambiguity, a study was undertaken to examine the correctness of the simulation. Factors such as the relatedness of sense and the context

of senses were considered as well as an analysis of the frequency of occurrence of senses. The conclusions of these studies was that pseudo-words provide a good simulation of ambiguous words.

9.1.4 Representation and matching of word senses

As other research had shown that the manual disambiguation of a text by a number of people would produce inconsistent results, a testing strategy for the disambiguator was devised that would allow for this. Rather than opt for the normal procedure of presuming that there is a correct way to disambiguate a text. The accuracy of the disambiguator was measured in terms of how close it came to the consensus produced by the manual disambiguators. A word occurrence was represented by all its senses, each weighted by a confidence score. Using the variation distance, the correlation of a disambiguator's output to that of manual disambiguators could be measured. It was also argued that the full-sense representation of a word was a more accurate model of the manner in which word senses are used. This representation method and the variation distance were also used to measure the similarity of query and document word occurrences in the retrieval experiments. Although far from conclusive, the results provided some evidence that the use of full-sense/variation distance produced better retrieval effectiveness than a single-sense/binary matching scheme.

9.1.5 Conclusions drawn from experiments

The results of both sets of experiments presented in this thesis proved to be complementary. The main contributions derived from them are as follows.

- Query size has been shown to play an important role in determining the impact of ambiguity on retrieval effectiveness. Retrievals based on queries composed of one or two words were considerably affected by ambiguity, those based on longer queries were much less affected.

- The errors made by disambiguators were found to have a significant impact on effectiveness, so much so, that disambiguation was only worth performing when the accuracy of the disambiguator was high. The only time that disambiguation was found to provide any utility to retrieval effectiveness was for retrievals based on short queries.

- The analysis of the frequency distribution of word senses mentioned in Section 9.1.3 went some way to providing an explanation for these results. This analysis showed that the skewed distribution of the senses of a word caused ambiguity to be not as significant to retrieval effectiveness as might have been thought.

9.2 Future work

From the work of this thesis, a number of areas of investigation may provide further research. They are of two types: the first three sections here describe aspects of the experiments presented in the thesis that might be worthy of expansion; the sections following on present new work that could stem from these experiments.

9.2.1 Use better resources for the existing disambiguation strategy

Without doubt, the accuracy of the disambiguator was poor. This is most likely due to failings in the resources used to train the disambiguator, namely the WordNet thesaurus (source of seed words) and the Reuters document collection (source of clue words).

The attempt to generate broad semantic categories from the structure of WordNet was only tested in a very limited manner. A more in-depth examination of these categories might provide information on a more effective means of generation. It is also possible that new thesauruses containing the categories required by the disambiguation method are now available online.

Another possible reason for the poor disambiguation accuracy is the relatively small size of the Reuters collection (~25Mb). The disambiguation strategy used relied on gathering clue words from the collection text. It is possible that this text did not provide a sufficiently large source of clue words for all word senses. How large a collection needs to be for this type of disambiguation strategy is unclear. Yarowsky *[Yarowsky 92]* used a 60Mb corpus which is comparable in size to Reuters, but in more recent work *[Yarowsky 95]* he has used corpora an order of magnitude larger.

9.2.2 Use a better disambiguator

It may be beneficial to repeat the experiments using any improved disambiguation strategy that is subsequently devised. Already one such strategy exists, Yarowsky's new disambiguator based on unsupervised learning *[Yarowsky 95]* has a number of qualities that make it a more attractive disambiguator than the one chosen for use in this thesis *[Yarowsky 92]*: a pragmatic quality is that it does not use for its training, reference works that are hard to come by; its more important quality, however, is that it would appear to be more accurate than previous strategies (reported accuracy of 96%). The only disadvantage is that the disambiguator may only work well when disambiguating large corpora.

9.2.3 Use other collections

The choice of Reuters as test collection for the final set of retrieval experiments was driven by the need for variable query sizes; a feature not possessed by traditional test collections. There is now the promise of the TREC collection *[Harman 95]* having this variability. The queries of TREC-6 are expected to be between one and three words in length which will complement well the other longer queries of TREC. It would be of interest to repeat the final retrieval experiments on this collection. Given the discussion on the need for a larger test collection in Section 9.2.1, it may well be that TREC will be a collection on which improvement in retrieval effectiveness is shown through the use of disambiguation.

The following sections describe areas of possible other work.

9.2.4 Using the analysis of word sense frequencies

It would appear that the reasons for the relatively low impact of lexical ambiguity on retrieval effectiveness is due to the skewed frequencies of occurrence of the senses of ambiguous words (see analysis in Section 5.6.1). It may be possible to use this analysis to explain the impact on effectiveness caused by other processes. For example, a speech recognition system produces output that is similar to that of a disambiguator: for every word spoken to a recogniser it outputs a list of candidate recognised words each with an attached uncertainty value. An analysis of the frequency of occurrence of these words might reveal similarities to the skewed frequencies found for word senses. If this were the case, some of the impacts on retrieval effectiveness reported in this thesis could be applicable to an IR system retrieving from spoken documents.

9.2.5 Other approaches to accommodating sense information

As was mentioned in Section 8.2 there may have been better ad hoc methods of incorporating the sense information into the document relevance score function. There may also be advantages in developing a theoretical model of retrieval that incorporates the full sense representation of words. If such a model were developed, its handling of the uncertainty embodied in this representation could find applications beyond disambiguation. For example, optical character recognition and speech recognition both produce texts with similar representation issues to those addressed here: i.e. they produce lists of candidate recognised words each with an attached uncertainty value.

9.2.6 Targeting the use of disambiguation

As was seen from the results of Chapter 8.3, the use of disambiguation information in a retrieval system was found to improve retrieval effectiveness under certain conditions but

found to degrade it under others. Improvement occurred for short queries and there were indications that retrieval of documents in the top part of a ranking also benefited. This raises the possibility of targeting an IR system's use of disambiguation information only to situations where those conditions hold. It remains to be seem whether it would be possible to formally define what constitutes a short query, or for that matter, what the 'top part' of a document ranking is. Nevertheless, this area could be a promising line of enquiry.

9.2.7 Conduct user experiments

Although many of the retrieval experiments performed in this thesis have used a newer test collection and investigated often unexamined aspects of retrieval, they are still very much in the mould of traditional fully automatic experimentation. As convenient as they are, they provide little or no information on how users might react to an IR system that incorporates disambiguation information. The results of the experiments have identified certain conditions where the use of such information might improve effectiveness. Therefore, it is important to discover if users would benefit from these improvements. For the identified conditions, this would mean testing if user queries are short enough, if they contain words that need to be disambiguated (perhaps user queries are all proper nouns), and if users are interested in only the 'top part' of a document ranking. In addition, users of such a system would be required to define the sense(s) of query words; would they be willing to do this? Even with the insights and advances gained through the work presented here, these further questions need to be addressed before final conclusions can be drawn on the utility of word sense disambiguation to IR.

A Duplicate detection in the Reuters collection

While conducting some preliminary experiments with the Reuters collection, it was discovered that contained within it were a number of documents that were exact duplicates of each other (see Figure 53). A short study was conducted to try to discover how many such documents there were. The results of this study revealed that the notion of a duplicate document was not as simple as first thought.

PATTERN-ID 21689 PUBLISHED-TESTSET
24-APR-1987 07:23:50.50
V
f0474reute
u f BC-BANK-OF-JAPAN-INTERVE 04-24 0085

BANK OF JAPAN INTERVENES IN TOKYO MARKET

TOKYO, April 24 - The Bank of Japan intervened just after the Tokyo market opened to support the dollar from falling below 140.00 yen, dealers said.

The central bank bought a moderate amount of dollars to prevent its decline amid bearish sentiment for the U.S. Currency, they said.

The dollar opened at a record Tokyo low of 140.00 yen against 140.70/80 in New York and 141.15 at the close here yesterday. The previous Tokyo low was 140.55 yen set on April 15.
REUTER

PATTERN-ID 1682 TRAINING-SET
23-APR-1987 20:21:46.09

RM
f3091reute
b f BC-BANK-OF-JAPAN-INTERVE 04-23 0086

BANK OF JAPAN INTERVENES IN TOKYO MARKET

TOKYO, April 24 - The Bank of Japan intervened just after the Tokyo market opened to support the dollar from falling below 140.00 yen, dealers said.

The central bank bought a moderate amount of dollars to prevent its decline amid bearish sentiment for the U.S. Currency, they said.

The dollar opened at a record Tokyo low of 140.00 yen against 140.70/80 in New York and 141.15 at the close here yesterday. The previous Tokyo low was 140.55 yen set on April 15.
REUTER

Figure 53. Reuters documents referring to the same event whose body texts are identical.

The contents of this appendix are as follows. A brief review of previous duplicate detection research will be presented, followed by a description of the methods and results of the duplicate detection work conducted here.

A.1 Other duplicate research

A.1.1 Bibliographic databases

In a bibliographic database, the main task is not to find exact duplicate records, rather it is to find those that refer to the same work but differ in some manner. Differences are typically due to inaccurate or inconsistent data entry. One such detection method was developed by Ridley *[Ridley 92]* who adopted a two stage technique. First, all records in a database were assigned a number generated from a *hashing function* that used as its input, fields of a bibliographic record. Any records that had the same hashing number were examined in greater detail in the second stage. This entailed a comparison of fields by customised processes: i.e. the author field process looked for missing initials; the title field process looked for a missing suffix. Detection techniques of this kind are supported by the work of O'Neill et al. *[O'Neill 93]* who manually examined duplicate bibliographic records to find which fields were most likely to differ.

A.1.2 Electronic publishing

As electronic publishing becomes more common, the potential problems of copyright violation and of plagiarism will increase. Most efforts devised to combat these problems concentrate on attempts to prevent or at least make it difficult for people to copy electronic documents. However, the detection of duplicates or partial duplicates is another approach. Brin et al. *[Brin 95]* proposed a system where electronic publishers store in a centralised database, *signatures* of all their published works. A signature would in some way summarise a document. The owners of this database could continually scan other electronic document collections looking for duplicates that might violate their copyright.

The method that Brin et al. proposed for building these signatures involved the breaking up of documents into what they call chunks. They suggest that these could be sentences, paragraphs, or some form of interleaved text unit. Each chunk of a document is passed to a hashing function that produces a number (quite how this function works is unclear from the paper). All numbers of that document are concatenated to form a signature. Detection of duplication is simply a process of comparing the hash numbers of two document signatures and looking for an unexpectedly high number of matches.

A method similar to this was adopted for the Reuters based work presented here. As only duplicate documents were of interest, the size of chunk was chosen to be a whole document, and the hashing function was a term selection method based on *idf* weights. This detection method is now described.

A.2 The duplicate detection for Reuters documents

During the building of an IR system *[Sanderson 91]*, the following was noted. Performing relevance feedback based on a single document, resulted in a query composed of terms from that document alone. A retrieval based on that query almost always resulted in a document ranking whose relevance scores were distributed in the manner shown in Figure 54. The highest relevance score was assigned to the document that relevance feedback was based on. All other retrieved documents were assigned a significantly lower score. It was hypothesised that

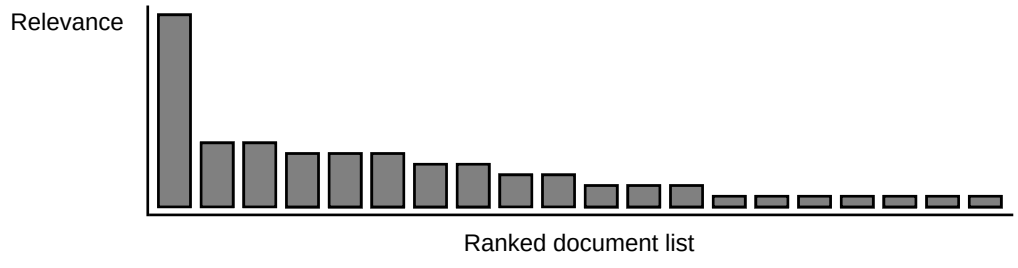


Figure 54. Relevance scores assigned to a document ranking.

a query generated from relevance feedback based on a single document would uniquely identify that document. The only exception to this would be if there was an exact duplicate of it.

It was a detection method based on this hypothesis that was tested in these experiments. It works as follows. For each individual document in a collection, generate a query using relevance feedback based on just that document[1], perform a retrieval and analyse any other documents with a high relevance score to discover if they are duplicates. If such a duplicate is found by this method, it is described here as one document *retrieving* another. Although this was found to work well, after some informal testing, further modifications to the method were made and they are now described[2].

A.2.3 First modification

The first modification arose when documents such as the pair in Figure 55 were found. As can be seen, one is a longer version of the other. Unfortunately, for document pairs of this type, the shorter would retrieve the longer as a potential duplicate even though it is not. This happens because all the words in the shorter version of the document (from which relevance feedback generates a query) appear in the longer version. Therefore, it was decided that two documents were exact duplicates only if the first document retrieved the second and the second retrieved the first. This would hopefully avoid the type of document pair shown here. After conducting the experiments, it was realised that this modification would probably not have been necessary if the term weighting scheme, used in retrieval, had been based on within document frequencies and document length normalisation.

A.2.4 Second modification

The second modification occurred when the type of document pair in Figure 56 was found. As can be seen, these documents are almost identical but they refer to different events. It would appear that for a number of regular events, like the financial transactions reported in Figure 56, the Reuters staff have a standard set of templates that they use for such events. To avoid this type of document pair it was decided that potential duplicates had to be relayed within 48 hours of each other.

A.3 Testing the method

To test the effectiveness of the duplicate detection method, potential duplicates of every document in the Reuters collection were retrieved and placed into one of three sets: documents

1. It was found that queries composed of 20 terms were large enough to accurately find the duplicates.
2. Since conducting this work, Kirriemuir *[Kirriemuir 95]* has investigated this area and has devised a broadly similar method, although it is less exhaustive in its pursuit of duplicates.

PATTERN-ID 10068 TRAINING-SET
14-MAR-1987 23:23:04.16
RM
f0844reute
r f BC-UNION-LEADERS-TOUR-YU 03-14 0104

UNION LEADERS TOUR YUGOSLAVIA TO QUELL STRIKE

BELGRADE, March 15 - Yugoslav trade union leaders are touring the country in an attempt to quell a wave of strikes following a partial wages freeze, official sources said.

Eyewitnesses in the northern city of Zagreb reported far more police on the streets than normal after the city and areas nearby experienced the biggest wave of strikes in the country in recent memory.

National newspapers in Belgrade have given few details of the strikes. But Zagreb papers said thousands of workers went on strike and thousands more were threatening action over pay cuts.

Official sources said there were also strikes at a Belgrade medical centre, a food factory in Sambor, and enterprises in Nis, Leskovac and Kraljevo, as well as other towns.

They said national union officials were travelling throughout the country to speak to meetings in an attempt to restore calm.

But trade union leaders were avoiding making statements to the press and had not made their stand on the strikes clear.

Western diplomats said the strikes appeared to be spontaneous and without any unified orchestration.
REUTER

PATTERN-ID 10256 TRAINING-SET
16-MAR-1987 09:46:11.84
C G T M
f2044reute
d f BC-UNION-LEADERS-TOUR-YU 03-16 0120

UNION LEADERS TOUR YUGOSLAVIA TO QUELL STRIKE

BELGRADE, March 16 - Yugoslav trade union leaders are touring the country in an attempt to quell a wave of strikes following a partial wages freeze, official sources said.

Eyewitnesses in the northern city of Zagreb reported far more police on the streets than normal after the city and areas nearby experienced the biggest wave of strikes in the country in recent memory.

National newspapers in Belgrade have given few details of the strikes. But Zagreb papers said thousands of workers went on strike and thousands more were threatening action over pay cuts.

Western diplomats said the strikes appeared to be spontaneous and without unified orchestration.
Reuter

Figure 55. Documents referring to the same event where one is a longer version of the other.

PATTERN-ID 12705 TRAINING-SET
2-MAR-1987 11:44:41.93
V RM
f0060reute
b f BC-/-FED-ADDS-RESERVES-V 03-02 0060

FED ADDS RESERVES VIA CUSTOMER REPURCHASES

NEW YORK, March 2 - The Federal Reserve entered the U.S. Government securities market to arrange 1.5 billion dlrs of customer repurchase agreements, a Fed spokesman said.

Dealers said Federal funds were trading at 6-3/16 pct when the Fed began its temporary and indirect supply of reserves to the banking system.
Reuter

PATTERN-ID 19586 TRAINING-SET
9-MAR-1987 11:49:35.16
V RM
f0663reute
b f BC-/-FED-ADDS-RESERVES-V 03-09 0060

FED ADDS RESERVES VIA CUSTOMER REPURCHASES

NEW YORK, March 9 - The Federal Reserve entered the U.S. Government securities market to arrange 2.5 billion dlrs of customer repurchase agreements, a Fed spokesman said.

Dealers said Federal funds were trading at 6-3/16 pct when the Fed began its temporary and indirect supply of reserves to the banking system.
Reuter

Figure 56. Documents whose body text is very similar but each refers to a different event.

pairs that appeared to be duplicates but reported different events; documents pairs where one was a longer version of the other; and documents pairs that were exact duplicates. The accuracy with which documents were placed in each set was then measured.

A.3.5 The first set: documents that report different events

In examining each document pair in this set, the following test question was asked,

> Do these documents refer to a different event?

In Table 23 we can see that 88% of pairs passed this test which indicates that the modification was effective in partitioning this type of document from exact duplicates. The four pairs that were incorrectly assigned were exact duplicates relayed more than 48 hours apart.

Passed	30	88%
Failed	4	12%
Total	34	

Table 23. Results of the first document duplicate test.

A.3.6 The second set: documents where one is a longer version of the other

There were 338 pairs in this set. Rather than check every pair, a quarter of them was randomly selected and examined. The test question applied while inspecting each pair was,

> Do these two documents refer to the same event and is one of them a longer version of the other?

As can be seen in Table 24, 84% of the pairs inspected passed this test, indicating a reasonably effective detection of this form of document pair. Most of the pairs incorrectly assigned to this set referred to different events. If a chronological test like that used above had been applied, these pairs would have been eliminated. The others incorrectly assigned were documents referring to distinct events that were relayed within a short time of each other, for example, hourly stock exchange reports. Quite how one would eliminate this type of pair without resorting to a collection specific solution is not clear.

Passed	71	84%
Failed	14	16%
Total	85	

Table 24. Results of the second document duplicate test.

A.3.7 The final set: documents that are exact duplicates of each other

These were document pairs that passed both modifications: each document retrieves the other, and they were relayed within 48 hours of each other. The number of pairs identified was 955. Rather than manually check all, a quarter was randomly selected and examined. The test question applied while inspecting each pair was,

> Do these documents refer to the same event and are the body texts within them identical?

As can be seen in Table 25, all of the document pairs examined passed this test.

Passed	240	100%
Failed	0	0%
Total	240	

Table 25. Results of the final document duplicate test.

A.4 Analysis of results

These results indicate that the duplicate detection method used in these experiments has a high precision in its identification of the three classes of duplicate defined here. The main objective of this work was to identify exact duplicates: four were incorrectly identified as time based duplicates; 955 such pairs (accounting for 4.5% of the Reuters collection) were correctly identified.

B Sense resolution properties of logical imaging

The evaluation of an implication by imaging is a logical technique developed in the framework of modal logic. Its interpretation in the context of a 'possible worlds' semantics is appealing for IR. In 1994, Crestani and Van Rijsbergen proposed an interpretation of imaging in the context of IR based on the assumption that 'a term is a possible world'. This approach enables the exploitation of term-term relationships that are estimated using an information theoretic measure.

Recent analysis of the probability kinematics of logical imaging in IR have suggested that this technique has some sense resolution properties. In this appendix we will present this new line of research[3].

B.1 Introduction

In their recent papers Crestani and Van Rijsbergen *[Crestani 95a]*, *[Crestani 95b]* described a technique called retrieval by logical imaging which originates from the theoretical field of modal logic. They showed how to apply this technique to an IR system. This application of imaging to IR could be described as a top-down approach: deciding if a technique could be useful to IR from theoretical analysis rather than from a bottom up approach of analysing the results of retrievals.

An investigation was undertaken to understand this technique, not in terms of theory but in terms of the 'nuts and bolts' of IR: words and their meaning. This investigation discovered an unexpected effect that imaging has on certain types of ambiguous words and it is a description and explanation of this effect that constitutes this appendix.

Before describing this effect, an introduction to imaging is provided followed by a discussion of some pertinent aspects of word sense ambiguity and of disambiguators. After this, the effect on word senses caused by imaging is outlined and this is followed by a proposal for an experiment to measure this effect. Finally there is a short discussion and conclusions.

B.2 Logical imaging and possible worlds semantics

Imaging is a process developed in the framework of Modal Logic *[Chellas 80]*. It enables the evaluation of a conditional sentence without explicitly defining the operator '$\rightarrow$'. What it requires is a clustering on the space of events (worlds) by means of a primitive relation of neighbourhood. This semantics is called *possible worlds semantics*, it was proposed by

3. This appendix is adapted from a paper *[Crestani 95c]* coauthored with Fabio Crestani and C.J. van Rijsbergen.

Kripke in *[Kripke 71]*. According to this semantics the truth value of the conditional $y \rightarrow x$ in a world w is equivalent to the truth value of the consequent x in the closest world w_y where the antecedent y is true. The identification of the closest world is done using the clustering. The passage from a world to another world can be regarded as a beliefs revision, and the passage from a world to its closest is therefore equivalent to the least drastic revision of one's beliefs. Using this process it is possible to implement the logical uncertainty principle proposed by Van Rijsbergen in *[Van Rijsbergen 86]*:

> Given any two sentences x and y; a measure of the uncertainty of $y \rightarrow x$ relative to a given data set is determined by the minimal extent to which we have to add information to the data set, to establish the truth of $y \rightarrow x$.

Imaging can be extended to the case where we have a probability distribution on the worlds *[Lewis D. 81]*. A probability distribution over the worlds can be regarded as a measure of the prior uncertainty (or certainty) associated with the beliefs. In this case there is a shift of the original probability P of the world w to the closest world w_y where y is true. Probability is neither created nor destroyed, it is moved from a 'not-y-world' to a 'y-world' to derive P_y a new probability distribution. This process is called deriving P_y from P by imaging on y.

A formal and detailed exposition of the imaging process can be found in *[Stalnaker 81]*, *[Lewis D. 81]*. In the next section we will present how imaging can be used in the context of IR.

B.3 Retrieving documents by logical imaging

The most obvious way of applying imaging to IR would be by considering a document as a possible world, regarding it as a set of propositions with associated truth values. This is the view taken originally by Van Rijsbergen in *[Van Rijsbergen 89]* and followed by others (see *[Amati 92]*). In this view we should evaluate the probability of the conditionals $d \rightarrow q$ by computing a new probability distribution P_d by imaging on d over all the possible worlds, i.e. over all the possible document representations. As pointed out in *[Crestani 95a]*, there are various problems related to this interpretation of imaging in IR. Instead, we propose a different approach. We consider the set of terms *T*, index terms or simply terms used in the document collection, as the set of possible worlds.

In order to apply this approach to imaging in IR we need a different representation of the document space. We use the technique of considering a term represented by a set (a vector) of

documents. This is the inverse of the representation technique most often used in IR where a document is represented as a set of features, namely terms (or index terms). Intuitively this can be understood as

> if you want to know the meaning of a term then look at all the documents in which that term occurs.

This idea is not new in IR (see for example *[Amati 92], [Qiu 93]*) and it has been widely used for the evaluation of term-term similarity. Representing terms in this way, we consider a document *d* true in a term (world) *t* if the term *t* occurs in *d*, and similarly a query *q* true in a term (world) *t* if the term *t* occurs in it. Using a measure of similarity among terms it is easy to determine the closest term t_d to *t* that occurs in the document *d*, or similarly t_q, closest term to *t* that occurs in the query *q*.

According to this interpretation of imaging in IR in *[Crestani 95a]* we proposed a model called *retrieval by logical imaging* that considers a process of imaging on *d* over all the possible terms *t* in *T*. This model has been further improved into the *retrieval by general logical imaging* model in *[Crestani 95b]*. For the purpose of this appendix we will refer to the retrieval by logical imaging model, the simplest of the two, which uses imaging as proposed by Stalnaker in *[Stalnaker 81]*. The properties of imaging that we will present in this appendix with regard to word sense resolution are present in both models.

Retrieval by logical imaging is performed by evaluating the following formula:

$$P(d \rightarrow q) = P_d(q) = \sum_T P(t) I(t_d, q) \tag{12}$$

where

$$I(t_d, q) = \begin{cases} 1 \text{ if } t_d \text{ occurs in } q \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

with t_d as the closest term to *t* that occurs in *d* (i.e. where *d* is true).

This process, called *imaging on d*, causes a transfer of probabilities from terms not occurring in the document *d* (i.e. for which the document *d* is not true) to terms occurring in it (i.e. for which the document *d* is true).

Similarly we can also evaluate $P(q \rightarrow d)$ by imaging on *q*:

$$P(q \rightarrow d) = P_q(d) = \sum_T P(t) I(t_q, d) \tag{14}$$

where

$$I(t_q, d) = \begin{cases} 1 \text{ if } t_q \text{ occurs in } d \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

with t_q as the closest term to t that occurs in q (i.e. where q is true).

Here we consider a process of imaging on q over each possible term t in T so that the probability initially assigned to each term moves from terms not occurring in the query q to terms occurring in the query q.

The application of this technique to IR requires an appropriate measure of similarity and an appropriate probability distribution over the term space *T*. In *[Crestani 95a]* these problems were tackled using a measure of similarity based on a information theoretic measure, the *expected mutual information measure* (EMIM), and the standard IR term weighting technique, *idf*. In the following sections we will assume as given both a measure of similarity and a probability distribution over the term space.

In the following two sections we explain the two processes of imaging on the document and on the query by means of an example.

B.3.1 Evaluation of $P(d \rightarrow q)$ by imaging on d

We assume a set of terms T with a probability distribution P which assigns to each term $t \in T$ a probability $P(t)$ so that $\sum P(t) = 1$. We also use the following notation:

$$I(t, x) = \begin{cases} 1 \text{ if } t \text{ occurs in } x \\ 0 \text{ otherwise} \end{cases} \tag{16}$$

We assume we have a document collection D, with $d \in D$, where the documents are represented by terms in the set *T*. Finally, we assume we have a query q also represented by terms in *T*. Then, as explained in the previous Section, it is possible to evaluate the $P(d \rightarrow q)$ as:

$$P(d \rightarrow q) = P_d(q) \tag{17}$$

$$= \sum_T P(t) I(t_d, q) \tag{18}$$

$$= \sum_T P_d(t) I(t, q) \tag{19}$$

where t_d is the term most similar to t that also occurs in d, and $P_d(t)$ is the new probability distribution over the set of terms appearing in d obtained by imaging on d.

The evaluation of $P(d \rightarrow q) = P_d(q)$ must be repeated for each document in the collection D and it is based on the initial probability distribution over the set of terms T and on the availability of a similarity measure enabling the evaluation of t_d. For a practical example of this evaluation

t	$P(t)$	$I(t,d)$	t_d	$P_d(t)$	$I(t,q)$	$P_d(t) \bullet I(t,q)$
1	0.20	1	1	0.30	1	0.30
2	0.10	0	1	0.00	0	0.00
3	0.05	0	5	0.00	0	0.00
4	0.20	0	5	0.00	1	0.00
5	0.30	1	5	0.55	0	0.00
6	0.15	1	6	0.15	1	0.15
$\sum_t$	1.00			1.00		0.45

Table 26. The evaluation of *P(d→q)*.

let us suppose we have a query q described by the terms t_1, t_4, and t_6. We would like to evaluate the probability of relevance of a document d described by terms t_1, t_5, and t_6. Assuming a vector notation, Table 26 reports the evaluation of $P(d \rightarrow q)$ by imaging on d, that is an estimate of the probability of relevance of the document d to the query q.

The evaluation process is the following:

- Identify the terms occurring in the document d (third column).
- Determine for each term in T the t_d, i.e. the most similar term to t for which $I(t,d) = 1$. This is done using the similarity measure on the term space (fourth column).
- Evaluate $P_d(t)$ by transferring the probabilities from terms not occurring in the document to terms occurring in it (fifth column).
- Evaluate $t(q)$ for each term, i.e. determine if the term occurs in the query (sixth column).
- Evaluate $P_d(t) \bullet I(t,q)$ for all the terms in the query (seventh column) and evaluate $P_d(q)$ by summation (bottom of seventh column).

Figure 57 a shows a graphical representation of this process. As can be seen, each term is represented by a world with its probability measure expressing the importance of the term in the term space T. The shadowed terms occur in document d. We assume a measure of similarity on the term space. Using this information we can now transfer the probability from each term not occurring in the document d to its most similar one occurring in d as depicted in Figure 57.

In Figure 57 the terms with null probability disappear, those occurring in the query *q* are taken into consideration and their new probabilities P_d(t) are summed up to evaluate $P_d(q)$.

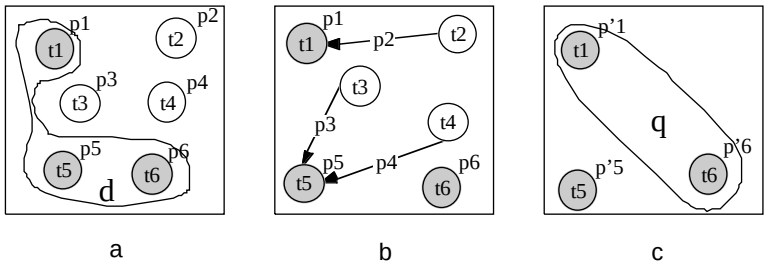


Figure 57. A graphical interpretation of imaging on *d*.

B.3.2 Evaluation of $P(q \rightarrow d)$ by imaging on *q*

Using the same data of the previous example we can now evaluate for documents the probability $P(q \rightarrow d)$. The terminology is analogous to that of the example above, though modified to take into consideration the evaluation of different elements.

The evaluation of $P(q \rightarrow d)$ is obtained as follows:

$$P(q \rightarrow d) = P_q(d) \tag{20}$$

$$= \sum_T P(t) I(t_q, d) \tag{21}$$

$$= \sum_T P_q(t) I(t, d) \tag{22}$$

where t_q is the term most similar to *t* that also occurs in *q*, and $P_q(t)$ is the new probability distribution over the set of terms appearing in *q* obtained by imaging on *q*.

The evaluation of $P(q \rightarrow d)$ must be repeated for each document in the collection *D* and it is based on the initial probability distribution over the set of terms *T* and on the availability of a similarity measure enabling the evaluation of t_q.

Table 27 reports an example of the evaluation of $P(q \rightarrow d)$ which can be structured in the following steps:

- Identify the terms occurring in the query *q* (third column).
- Determine for each term in *T* the t_q, i.e. the most similar term to *t* for which $I(t,q) = 1$ (fourth column).

t	$P(t)$	$I(t,q)$	t_q	$P_q(t)$	$I(t,d)$	$P_q(t) \bullet I(t,d)$
1	0.20	1	1	0.35	1	0.35
2	0.10	0	1	0.00	0	0.00
3	0.05	0	1	0.00	0	0.00
4	0.20	1	4	0.50	0	0.00
5	0.30	0	4	0.00	1	0.00
6	0.15	1	6	0.15	1	0.15
$\sum_t$	1.00			1.00		0.50

Table 27. The evaluation of *P(q→d)*.

- Evaluate $P_q(t)$ by transferring the probabilities from terms not occurring in the query to terms occurring in it (fifth column).
- Evaluate $I(t,d)$ for each term, i.e. determine if the term occurs in the document (sixth column).
- Evaluate $P_q(t) \bullet I(t,d)$ for each term in the document and evaluate $P_q(d)$ by summation (seventh column).

A graphical interpretation of the imaging process in relation to this example is shown in Figure 58.

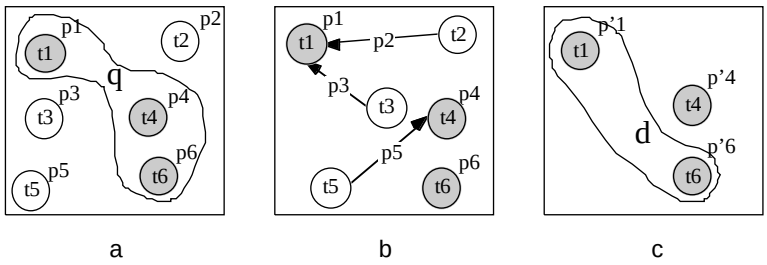


Figure 58. A graphical interpretation of imaging on *q*.

B.4 Word sense disambiguation

Much has already been written in this thesis on the subject of word sense disambiguation. A shared feature of all corpus based disambiguators referred to in this thesis is an assumption that each individual sense of a word will appear in a wide context (typically 40-100 surrounding words) that is distinct from the contexts of its other senses. It is not clear if this assumption is entirely correct as research on human disambiguation has found that people can identify word senses accurately from a much narrower context of 1-5 words. This raises the possibility of having two senses of a word occurring in similar wide contexts but in different narrow contexts. Such a situation probably accounts for some of the errors made by automatic

disambiguators. Nevertheless, the Yarowsky disambiguator (Section 4.2.7) makes this unique context assumption and has a 90% disambiguation accuracy. It is this assumption coupled with the skewed frequency distribution of word senses, highlighted in Section 5.6.1, that is important in the relationship between imaging and the senses of a word. In the following discussion, it is also assumed that similarity is approximated by some form of term co-occurrence measure such as the EMIM.

B.4.3 Imaging and sense ambiguity

As has already been discussed, there are two forms of imaging in IR: imaging on the document $P_d(q)$; and imaging on the query $P_q(d)$. Each form behaves differently with regard to the senses of ambiguous words and we will discuss them separately. To illustrate these discussions, a simplified example will be used.

Let us imagine a document collection in which the word ‘bat’ appears in a number of documents and that the frequency of occurrence of its word senses is skewed. In most documents, the word is used to refer to a sporting implement, but occasionally it is used to refer to a flying mouse like mammal. As the sporting sense of ‘bat’ is predominant, collection words most similar to ‘bat’ will be those similar to this sense. For this example, the words most similar to ‘bat’ are ‘cricket’, ‘baseball’, ‘hit’, and ‘ball’.

Now let us look at two documents from this collection. Document d_1 is represented by ‘bat’ and ‘night’, while document d_2 is represented by ‘bat’ and ‘hit’. Document d_1 uses ‘bat’ in the animal sense (see Figure 59); document d_2 uses it in the sporting sense (see Figure 60). Suppose a user enters the two word query, ‘bat cricket’, how will the two forms of imaging rank these two documents?

Imaging on a document

As we recall, when imaging on a document d, the probabilities of terms not appearing in d are transferred to the terms that do appear in d. Looking at our example, let us first examine d_1. Since the words ‘cricket’, ‘baseball’, ‘hit’, and ‘ball’ are more similar to ‘bat’ than to ‘night’, all their probabilities transfer to this one word (Figure 59). From Table 28 we can see that this transfer results in document d_1 having an estimated probability of relevance of 0.95.

In the case of d_2, this document contains the word ‘hit’. As this word is also similar to ‘cricket’, ‘baseball’, and ‘ball’, the chances are that the probabilities of some of these words are likely to be transferred to ‘hit’ instead of ‘bat’, this is shown in Figure 60. As ‘bat’ is the

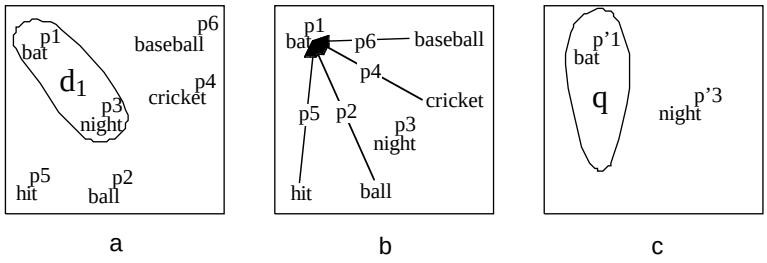


Figure 59. Imaging on document containing animal sense of 'bat'.

t	$P(t)$	$I(t,d_1)$	t_{d_1}	$P_{d_1}(t)$	$I(t,q)$	$P_{d_1}(t) \bullet I(t,q)$
bat	0.20	1	1	0.95	1	0.95
ball	0.10	0	1	0.00	0	0.00
night	0.05	1	3	0.05	0	0.00
cricket	0.20	0	1	0.00	1	0.00
hit	0.30	0	1	0.00	0	0.00
baseball	0.15	0	1	0.00	0	0.00
$\sum_t$	1.00			1.00		0.95

Table 28. Imaging on document containing animal sense of 'bat'.

only query word contained in d_1, this results in d_2 having a lower estimated probability of relevance than d_1 (see Table 29), which means that d_1 is ranked higher than d_2.

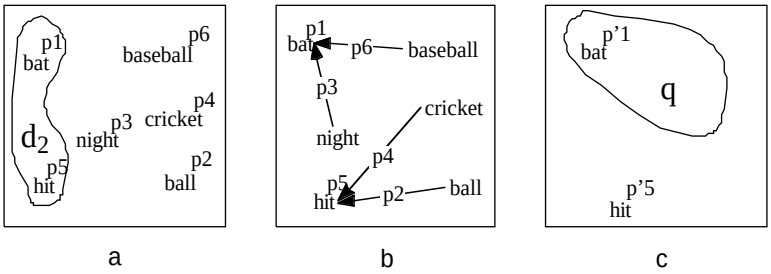


Figure 60. Imaging on document containing sporting sense of bat.

t	$P(t)$	$I(t,d_2)$	t_{d_2}	$P_{d_2}(t)$	$I(t,q)$	$P_{d_2}(t) \bullet I(t,q)$
bat	0.20	1	1	0.40	1	0.40
ball	0.10	0	5	0.00	0	0.00
night	0.05	0	1	0.00	0	0.00
cricket	0.20	0	5	0.00	1	0.00
hit	0.30	1	5	0.60	0	0.00
baseball	0.15	0	1	0.00	0	0.00
$\sum_t$	1.00			1.00		0.40

Table 29. Imaging on document containing sporting sense of 'bat'.

This example seems to show that imaging on a document will give preference to those documents that contain query terms appearing in unusual contexts. In terms of word senses, the supposition is that this form of imaging will rank higher, those documents that hold query terms used in unusual senses.

Imaging on the query

When imaging on a query, the method of probability transfer is similar to imaging on documents except that the transfer is onto the terms in the query. Unlike imaging on documents, the transfer of probabilities to the query terms is the same regardless of what document is being retrieved (see Figure 61). Table 30 shows the estimated probability of relevance for d_2. It is a simple matter to show that d_1 will be assigned the same probability.

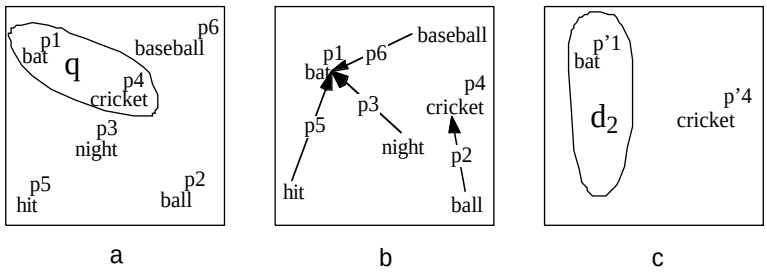


Figure 61. Imaging on query containing sporting sense of 'bat'.

t	$P(t)$	$I(t,q)$	t_q	$P_q(t)$	$I(t,d_2)$	$P_q(t) \bullet I(t,d_2)$
bat	0.20	1	1	0.70	1	0.70
ball	0.10	0	4	0.00	0	0.00
night	0.05	0	1	0.00	0	0.00
cricket	0.20	1	4	0.30	0	0.00
hit	0.30	0	1	0.00	1	0.00
baseball	0.15	0	1	0.00	0	0.00
$\sum_t$	1.00			1.00		0.70

Table 30. Imaging on query containing sporting sense of 'bat'.

B.5 Proposed experimental investigation

As described in Crestani and Van Rijsbergen *[Crestani 95b]* experiments have already been carried out that report an improvement in retrieval effectiveness as a result of retrieving document by logical imaging. We intend to re-examine the retrieval results of those experiments to determine the extent to which the effects described above are to be found. However, it is believed that these effects may not be so clearly observable when queries with a large number of terms are used. As the queries of the test collections Crestani and Van Rijsbergen used are relatively large, it is intended that further tests be performed on these collections using shorter

queries where it is expected that the retrieval results will be more affected by the imaging-sense effect.

B.6 Discussion and conclusions

The effect that document imaging has on documents containing ambiguous query terms is due to the imaging technique being influenced by all the terms of a document and not just those that appear in the query. It is not clear whether this effect of preferring documents containing query terms in unusual senses or contexts is desirable. If a user enters a query term it would seem reasonable to expect him to intend the most common sense. Until the tests outlined above are completed though, we prefer to withhold our judgment.

10 References

A number of these references are URLs. Given their transient nature, it is likely that eventually they will become invalid. Therefore, associated with each URL based reference is additional textual information that may help locate that which is referred to.

Ahlswede 93

T.E. Ahlswede & D. Lorand (1993). Word sense disambiguation by human subjects: Computational and psycholinguistic applications, in Proceedings of a Workshop Sponsored by the Special Interest Group on the Lexicon of the ACL: 1-9.

Amati 92

G. Amati & S. Kerpedjiev (1992). An Information Retrieval logical model: implementation and experiments. Technical Report Rel 5B04892, Fondazione Ugo Bordoni, Roma, Italy, March.

Apté 94

C. Apté, F. Damerau, S.M. Weiss (1994). Towards language independent automated learning of text categorisation models, in Proceedings of ACM SIGIR Conference, 17: 23-30.

Black 88

E. Black (1988). An experiment in computational discrimination of English word senses, in IBM Journal, 32(2): 185-194.

Blair 85

D.C. Blair (1992). Information retrieval and the philosophy of language, in Computer Journal, 35(3): 200-207.

Brin 95

S. Brin, J. Davis, H. Garcia-Molina (1995). Copy detection mechanism for digital documents, in Proceedings of SIGMOD.

Burnett 79

J.E. Burnett, D. Cooper, M.F. Lynch, P. Willett, M. Wycherley (1979). Document retrieval experiments using indexing vocabularies of varying size. - 1. Variety generation symbols assigned to the fronts of index terms, in Journal of Documentation, 35(3): 197-206.

Chellas 80

B.F. Chellas (1980). Modal logic: an introduction. Cambridge University Press, Cambridge, UK, 1980.

Choueka 85
Y. Choueka & S. Lusignan (1985). Disambiguation by short contexts, in Computers and the Humanities, 19: 147-157.

Cowie 92
J. Cowie, J. Guthrie & L. Guthrie (1992). Lexical disambiguation using simulated annealing, in Proceedings of COLING Conference: 359-365.

Crestani 95a
F. Crestani & C.J. van Rijsbergen (1995). Information Retrieval by Logical Imaging. Journal of Documentation, 51(1):1–15.

Crestani 95b
F. Crestani & C.J. van Rijsbergen (1995). Probability kinematics in information retrieval: a case study. Technical Report FERMI/95/1, ESPRIT Basic Research Action, Project Number 8134 - FERMI, Department of Computing Science, Glasgow University.

Crestani 95c
F. Crestani, M. Sanderson & C.J. van Rijsbergen (1995). Sense resolution properties of logical imaging. Proceedings of the BCS IRSG colloquium (17): 277-297.

Dagan 91
I. Dagan, A. Itai, U. Schwall (1991). Two languages are more informative than one, in Proceedings of the ACL, (29): 130-137.

Demetriou 93
G.C. Demetriou (1993). Lexical disambiguation using constraint handling in Prolog (CHIP), in Proceedings of the European Chapter of the ACL, (6): 431-436.

Evans 95
D. A. Evans, N. Milic-Frayling, R. G. Lefferts (1995). CLARIT TREC-4 Experiments, in Proceedings of the 4th TREC conference. Available on-line at http://www-nlpir.nist.gov/TREC/.

Fagan 87
J. Fagan (1987). Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and nonsyntactic methods, in Doctoral Dissertation, Technical Report TR 87-868, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501.

Frakes 92
W.B. Frakes & R. Baeza-Yeates (1992). Information Retrieval: Data structures & algorithms, in Prentice Hall

Gale 91
W.A. Gale & K.W. Church (1991). A program for aligning sentences in bilingual corpora, in Proceedings of the ACL, 29: 177-184.

Gale 92a
W. Gale, K.W. Church, D. Yarowsky (1992). Estimating upper and lower bounds on the performance of word-sense disambiguation programs, in Proceedings of the ACL, 30: 249-256.

Gale 92b
W.A. Gale, K.W. Church, D. Yarowsky (1992). A Method for Disambiguating Word Senses in a Large Corpus, in Computers and the Humanities, 26: 415-439.

Garside 87
R. Garside (1987). The CLAWS word tagging system, in The computational analysis of english: a corpus based approach, R. Garside, G. Leech, G. Sampson Eds., Longman: 30-41.

Guthrie 91
J.A. Guthrie, L. Guthrie, Y. Wilks, H. Aidinejad (1991). Subject-dependent co-occurrence and word sense disambiguation, in Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA.: 146-152.

Grolier
Grolier Multimedia Encyclopedia CD-ROM. Grolier Interactive Inc., 90 Sherman Turnpike, Danbury, CT 06816, USA

Harman 92
D. Harman (1992). Relevance feedback revisited, in Proceedings of ACM SIGIR Conference, 15: 1-10.

Harman 95
D. Harman (1995). Overview of the third text retrieval conference (TREC-3), NIST special publication 500-225.

Hayes 90
P. J. Hayes (1990). Intelligent high volume text processing using shallow, domain specific techniques, in Working Notes, AAAI Spring Symposium on Text-Based Intelligent Systems: 134-138.

Hearst 91
M.A. Hearst (1991). Noun homograph disambiguation using local context in large text corpora, in Proceedings of the 7th conference, UW Centre for the New OED & Text Research Using Corpora.

Hearst 93a
M. A. Hearst & C. Plaunt (1993). Subtopic structuring for full-length document access, in Proceedings of ACM SIGIR Conference, 16: 59-68.

Hearst 93b
M. A. Hearst & H. Schütze (1993). Customising a lexicon to better suit a computational task, in Proceedings of a Workshop Sponsored by the Special Interest Group on the Lexicon of the ACL.

Hirst 86
G. Hirst (1986). Semantic interpretation and the resolution of ambiguity, in Cambridge: Cambridge University Press.

Hughes 68
G.F. Hughes (1968). On the mean accuracy of statistical pattern recognisers, in IEEE Transactions on Information Theory, 14(1): 55-63.

Jorgensen 90
J. Jorgensen (1990). The psychological reality of word senses, Journal of Psychological Research, 19: 167-190

Kelly 75
E. Kelly & P. Stone (1975). Computer recognition of english word senses, in North-Holland Publishing Co., Amsterdam.

Kilgarriff 91
A. Kilgarriff (1991). Corpus word usages and dictionary word senses: What is the match? An empirical study, in Proceedings of the 7th conference, UW Centre for the New OED & Text Research Using Corpora.

Kirriemuir 95
J.W. Kirriemuir & P. Willett (1995). Identification of duplicate and near-duplicate full-text records in database search outputs using hierarchic cluster analysis, in Program - automated library and information systems, 29(3): 241-256.

Kirkpatrick 88
Roget's thesaurus of English words and phrases (1988). New ed. prepared by B. Kirk-patrick. Harmondsworth: Penguin

Kripke 71
S.A. Kripke (1971). Semantical considerations on modal logic. In L. Linsky, editor, Reference and modality, chapter 5, 63–73. Oxford University Press, Oxford, UK.

Krovetz 92
R. Krovetz & W.B. Croft (1992). Lexical Ambiguity and Information Retrieval, in ACM Transactions on Information Systems, 10(1).

Krovetz 93
R. Krovetz (1993). Viewing morphology as an inference process, in Proceedings of ACM SIGIR Conference, 16: 191-202.

Lalmas 96
M. Lalmas (1996). Theories of information and uncertainty for the modelling of information retrieval: an application to situation theory and dempster-shafer's theory of evidence, PhD Thesis, Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.

Lesk 88
M. Lesk (1988). “They said true things, but called them by wrong names” - vocabulary problems in retrieval systems, in Proc. 4th Annual Conference of the University of Waterloo Centre for the New OED.

Lewis D. 81
D. Lewis (1981). Probability of conditionals and conditional probabilities. In W.L. Harper, R. Stalnaker, and G. Pearce, editors, Ifs, The University of Western Ontario Series in Philosophy of Science, 129–147. D.Reidel Publishing Company, Dordrecht, Holland.

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

Longman 88
Longman dictionary of contemporary English, New edition, Longman

Luhn 58
H.P. Luhn (1958). The automatic creation of literature abstracts, in IBM Journal: 159-165.

Miller 54
G. A. Miller (1954). Communication, in Annual Review of Psychology, 5: 401-420.

Miller 90
G.A. Miller (1990). Wordnet: An on-line lexical database, International Journal of Lexicography, 3(4): 235-312.

Miller 95
G.A. Miller (1995). WordNet: A lexical database for English, in Communications of the ACM, 38(11): 39-41.

Molto 91
M. Molto & E. Svenonius (1991). Automatic recognition of title page names, in Information Processing and Management, 27 (1): 83-95.

Neff 91
M.S. Neff & B.K. Boguraev (1991). From machine-readable dictionaries to lexical databases, in Unpublished. Lexical Systems Project, IBM Research, Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

O'Neill 93
E.T. O'Neill, S.A. Rogers & W.M. Oskins (1993). Characteristics of duplicate records in OCLC's on-line union catalogue, in Library Resources & Technical Services, 37(1): 59-71.

Porter 80
M.F. Porter (1980). An algorithm for suffix stripping, in Program - automated library and information systems, 14(3): 130-137.

Qiu 93
Y. Qiu and H.P. Frei (1993). Concept based query expansion. In Proceedings of ACM-SIGIR, pages 160–171.

Reuters
ftp://ciir-ftp.cs.umass.edu/pub/reuters1/. Reuters and Carnegie Group have agreed to allow the free distribution of this data *for research purposes only.* Copyright for auxiliary files and additional annotations resides with David D. Lewis and the Information Retrieval Laboratory, University of Massachusetts.

Richardson 95
R. Richardson & A.F. Smeaton (1995). Using WordNet in a knowledge-based approach to information retrieval, in Dublin City University Technical Report, (CA-0395).

Ridley 92

M.J. Ridley (1992). An expert system for quality control and duplicate detection in bibliographic databases, in Program - automated library and information systems, 26(1): 1-18.

Robertson 76

S.E. Robertson & K. Sparck Jones (1976). Relevance weighting of search terms, in Journal of the American Society for Information Science, 27: 129-146.

Robertson 95

S. E. Robertson, S. Walker, S. Jones, M. M. Beaulieu, M. Gatford, A. Payne (1995). OKAPI at TREC-4, in Proceedings of the 4th TREC conference. Available on-line at http://www-nlpir.nist.gov/TREC/.

Sacks-Davis 90

R. Sacks-Davis, P. Wallis, R. Wilkinson (1990). Using syntactic analysis in a document retrieval system that uses signature files, in Proceedings of ACM SIGIR Conference, 13: 179-191.

Salton 83

G. Salton & M.J. McGill (1983). Introduction To Modern Information Retrieval. The SMART and SIRE experimental retrieval systems, in New York: McGraw-Hill

Salton 89

G. Salton & C. Buckley (1989). A comparison between statistically and syntactically generated term phrases, in TR 89-1027, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501.

Sanderson 91

M. Sanderson & C.J. van Rijsbergen (1991). NRT: news retrieval tool, in Electronic Publishing, EP-odd, 4(4): 205-217.

Sanderson 94

M. Sanderson (1994). Word sense disambiguation and information retrieval, in Proceedings of ACM SIGIR Conference, 17: 142-151.

Schütze 95

H. Schütze & J.O. Pedersen (1995). Information retrieval based on word senses, in Proceedings of the Symposium on Document Analysis and Information Retrieval, 4: 161-175.

Simpson 89
J. Simpson & E. Weiner (1989). The Oxford English dictionary, 2nd ed. Oxford university press.

Small 82
S. Small & C. Rieger (1982). Parsing and comprehending with word experts (a theory and its realisation), in Strategies for Natural Language Processing, W.G. Lehnert & M. H. Ringle, Eds., LEA: 89-148.

Smeaton 83
A.F. Smeaton & C.J. van Rijsbergen (1983). The retrieval effects of query expansion on a feedback document retrieval system, in Computer Journal, 26 (3): 239-246

Smeaton 87
A.F. Smeaton (1987). Using parsing of natural language as part of document retrieval, in Ph. D. Thesis, Department of Computer Science, University College Dublin, Ireland.

Smeaton 91
A. F. Smeaton & P. Sheridan (1991). Using morpho-syntactic language analysis in phrase matching, in Proceedings of RIAO 91, Intelligent Text and Image Handling,: 414-429.

Smeaton 92
A. Smeaton (1992). An evaluation of retrieval performance using simple statistics and SIMPR linguistic processing on a standard collection of texts, in SIMPR technical report, SIMPR-DCU-1992-50.2i

Sparck Jones 76
K. Sparck Jones & C.J. van Rijsbergen (1976). Progress in documentation, in Journal of Documentation, 32(1): 59-75.

Stalnaker 81
R. Stalnaker (1981). Probability and conditionals. In W.L. Harper, R. Stalnaker, and G. Pearce, editors, Ifs, The University of Western Ontario Series in Philosophy of Science: 107–128. D.Riedel Publishing Company, Dordrecht, Holland.

Stanfill 86
C. Stanfill & B. Kahle (1986). Parallel free text search on the connection machine system., in Communications of the ACM, 29(12): 1229-1239.

Sussna 93
M. Sussna (1993). Word sense disambiguation for free-text indexing using a massive semantic network, in Proceedings of the International Conference on Information & Knowledge Management (CIKM), 2: 67-74.

Van Rijsbergen 79
C.J. van Rijsbergen (1979). Information retrieval (second edition), in London: Butterworths.

Van Rijsbergen 86
C.J. van Rijsbergen (1986). A non-classical logic for Information Retrieval. The Computer Journal, 29(6):481–485.

Van Rijsbergen 89
C.J. van Rijsbergen (1989). Toward a new information logic. In Proceedings of ACMSIGIR, Cambridge, USA.

Virginia disc 90
The Virginia disc one CD-ROM, Published by Virginia Polytechnic Institute and State University Press. Editor, Project Director, Principal Investigator Edward A. Fox, Dept. of Computer Science 562 McBryde Hall, VPI&SU, Blacksburg, VA 24061-0106

Voorhees 93
E. M. Voorhees (1993). Using WordNet™ to disambiguate word sense for text retrieval, in Proceedings of ACM SIGIR Conference, (16): 171-180.

Wallis 93
P. Wallis (1993). Information retrieval based on paraphrase, in Proceedings of PACLING Conference, 1.

Weiss 73
S.F. Weiss (1973). Learning to disambiguate, in Information Storage and Retrieval, 9: 33-41.

Wilks 90
Y. Wilks, D. Fass, C. Guo, J.E. Mcdonald, T. Plate, B.M. Slator (1990). Providing Machine Tractable Dictionary Tools, in Machine Translation, 5: 99-154.

WordNet
http://www.cogsci.princeton.edu/~wn/. WordNet was developed by the Cognitive Science Laboratory at Princeton University under the direction of Professor George A. Miller (Principal Investigator). Ongoing development of WordNet is supported by DARPA/ITO (Information Technology Office).

Yarowsky 92

D. Yarowsky (1992). Word sense disambiguation using statistical models of Roget's categories trained on large corpora, in Proceedings of COLING Conference: 454-460.

Yarowsky 93

D. Yarowsky (1993). One sense per collocation, in Proceedings of ARPA Human language technology workshop.

Yarowsky 95

D. Yarowsky (1995). Unsupervised word sense disambiguation rivalling supervised methods, in Proceedings of the ACL, 33.

Zernik 91

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