Natural Language Processing and Information Retrieval

Ellen M. Voorhees

National Institute of Standards and Technology
Gaithersburg, MD 20899 USA
ellen.voorhees@nist.gov

Abstract. Information retrieval addresses the problem of finding those documents whose content matches a user's request from among a large collection of documents. Currently, the most successful general purpose retrieval methods are statistical methods that treat text as little more than a bag of words. However, attempts to improve retrieval performance through more sophisticated linguistic processing have been largely unsuccessful. Indeed, unless done carefully, such processing can degrade retrieval effectiveness.
Several factors contribute to the difficulty of improving on a good statistical baseline including: the forgiving nature but broad coverage of the typical retrieval task; the lack of good weighting schemes for compound index terms; and the implicit linguistic processing inherent in the statistical methods. Natural language processing techniques may be more important for related tasks such as question answering or document summarization.

1 Introduction

Imagine that you want to research a problem such as eliminating pests from your garden or learning the history of the city you will visit on your next holiday. One strategy is to gather recommendations for items to read; that is, to ask for references to documents that discuss your problem rather than to ask for specific answers. Computer systems that return documents whose contents match a stated information need have historically been called *information retrieval* (IR) systems, though lately they are more often called *document retrieval* or *text retrieval* systems to distinguish them from systems that support other kinds of information-seeking tasks.

Information retrieval systems search a collection of natural language documents with the goal of retrieving exactly the set of documents that pertain to a user's question. In contrast to database systems that require highly structured data and have a formal semantics, IR systems work with unstructured natural language text. And in contrast to expert systems, IR systems do not attempt to deduce or generate specific answers but return (pieces of) documents whose content is similar to the question. While IR systems have existed for over 40 years, today the World Wide Web search engines are probably the best-known examples of text retrieval systems. Other examples include systems that support literature searches at libraries, and patent- or precedent-searching systems in law firms. The underlying technology of retrieval systems—estimating the similarity of the content of two texts—is more broadly applicable, encompassing such tasks as information filtering, document summarization, and automatic construction of hypertext links.

Information retrieval can be viewed as a great success story for natural language processing (NLP): a major industry has been built around the automatic manipulation of unstructured natural language text. Yet the most successful general purpose retrieval methods rely on techniques that treat text as little more than a bag of words. Attempts to improve retrieval performance through more sophisticated linguistic processing have been largely unsuccessful, resulting in minimal differences in effectiveness at a substantially greater processing cost or even degrading retrieval effectiveness.

This paper examines why linguistically-inspired retrieval techniques have had little impact on retrieval effectiveness. A variety of factors are indicated, ranging from the nature of the retrieval task itself to the the fact that current retrieval systems already implicitly incorporate features the linguistic systems make explicit. The next section provides general IR background by describing both how current retrieval systems operate and the evaluation methodology used to decide if one retrieval run is better than another. Section 3 provides an overview of recent NLP and IR research including a case study of a particular set of NLP experiments to illustrate why seemingly good ideas do not necessarily lead to enhanced IR performance. The final section suggests some related tasks that may benefit more directly from advances in NLP.

2 Background

Text retrieval systems have their origins in library systems that were used to provide bibliographic references to books and journals in the library's holdings [1]. This origin has had two major influences on how the retrieval task is defined. First, retrieving (pointers to) documents rather than actual answers was the natural extension to the manual processes that were used in the libraries at the time, and this continues to be the main focus of the task. Second, retrieval systems are expected to handle questions on any subject matter included in a relatively large amount of text. This requirement for domain-independence and large amounts of text precluded knowledge-based approaches for text understanding from being incorporated into retrieval systems because the requisite knowledge structures were not available and the processing was too slow. Instead, the majority of information retrieval systems use statistical approaches to compute the similarity between documents and queries. That is, they use word counting techniques and assume that two texts are about the same topic if they use the same words.

A basic understanding of how these current retrieval systems work is required to appreciate how linguistic processing might affect their performance. This section provides a summary of the current practice in IR based on the results of an on-going series of evaluations known as the Text REtrieval Conference (TREC) workshops. The final part of the section describes common practices for retrieval system evaluation.

2.1 The Basics of Current IR Systems

Retrieval systems consist of two main processes, *indexing* and *matching*. Indexing is the process of selecting terms to represent a text. Matching is the process of computing a measure of similarity between two text representations.

In some environments human indexers assign terms, which are usually selected from a controlled vocabulary. A more common alternative is to use automatic indexing where the system itself decides on the terms based on the full text of the document. A basic automatic indexing procedure for English might proceed as follows:

1. split the text into strings of characters delimited by white space, considering such strings to be “words” (tokenization);
2. remove very frequent words such as prepositions and pronouns (removal of *stop words*); and
3. conflate related word forms to a common stem by removing suffixes (stemming).

The resulting word stems would be the terms for the given text.

In early retrieval systems, queries were represented as Boolean combinations of terms, and the set of documents that satisfied the Boolean expression was retrieved in response to the query. While this Boolean model is still in use today, it suffers from some drawbacks: the size of the retrieved set is difficult to control, and the user is given no indication as to whether some documents in the retrieved set are likely to be better than others in the set. Thus most retrieval systems return a ranked list of documents in response to a query. The documents in the list are ordered such that the documents the system believes to be most like the query are first on the list.

The vector-space model is another early retrieval model still in use today [2]. In this model, documents and queries are represented by vectors in T-dimensional space, where T is the number of distinct terms used in the documents and each axis corresponds to one term. Given a query, a vector system produces a ranked list of documents ordered by similarity to the query, where the similarity between a query and a document is computed using a metric on the respective vectors.

Other retrieval models exist, including several different probabilistic models and models based on word proximity. One of the findings of the TREC workshops is that retrieval systems based on quite different models exhibit similar retrieval effectiveness. That is, retrieval effectiveness is not strongly influenced by the specifics of the model used as long as the model incorporates appropriate term weighting. Term weighting, on the other hand, has been shown to have a primary effect on retrieval quality, with the best weights combining term frequency (*tf*), inverse document frequency (*idf*), and document length (*dl*) factors [3]. In this formulation, the *tf* factor weights a term proportionally to the number of times it occurs in the text, the *idf* factor weights a term inversely proportional to the number of documents in the collection that contain the term, and the *dl* factor compensates for widely varying document lengths.

2.2 The TREC Workshops

The relative merit of different retrieval approaches (for example, different weighting schemes) is evaluated using *test collections*, benchmark tasks for which the correct answers are known. Because retrieval performance is known to vary widely across queries, test collections need to contain a sufficient number of queries to make comparisons meaningful. Further, an observed difference in retrieval performance between two systems is generally considered valid only if it is repeatable across multiple collections. Thus statements regarding best practices in IR must be based on hundreds of retrieval runs. TREC provides the necessary infrastructure to support such comparisons [`http://trec.nist.gov`].

The TREC workshops are designed to encourage research on text retrieval for realistic applications by providing large test collections, uniform scoring procedures, and a forum for organizations interested in comparing results. Started in 1992, the conference is co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA). For each TREC, NIST provides a test set of documents and questions. Participants run their retrieval systems on the data, and return to NIST a list of the retrieved top-ranked documents. NIST pools the individual results, judges the retrieved documents for correctness, and evaluates the results. The TREC cycle ends with a workshop that is a forum for participants to share their experiences.

TREC's success depends on having a diverse set of participants. Since the relevance judgments (the "correct answers") are based on pooled results, the pools must contain the output from many different kinds of systems for the final test collections to be unbiased. Also, a variety of different candidate techniques must be compared to make general recommendations as to good retrieval practice. Fortunately, TREC has grown in both the number of participants and the number of different retrieval tasks studied since the first TREC. The latest TREC, TREC-7 held in November 1998, had 56 participating groups from 13 different countries and included representatives from the industrial, academic, and government sectors.

The first TREC conferences contained just two main tasks, *ad hoc* and *routing*. Additional subtasks known as "tracks" were introduced into TREC in TREC-4 (1995). The main ad hoc task provides an entry point for new participants and provides a baseline of retrieval performance. The tracks invigorate TREC by focusing research on new areas or particular aspects of text retrieval. To the extent the same retrieval techniques are used for the different tasks, the tracks also validate the findings of the ad hoc task. Figure 1 shows the number of experiments performed in each TREC, where the set of runs submitted for one track by one participant is counted as one experiment.

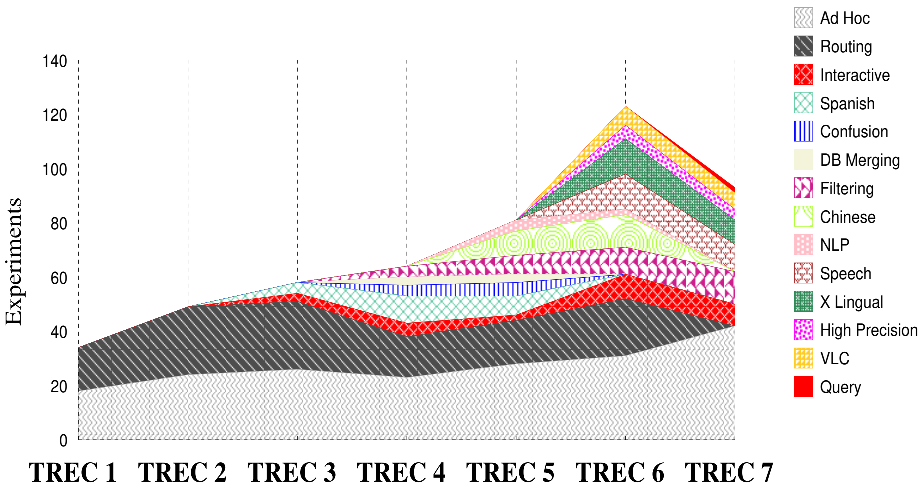

Fig. 1. Number of TREC experiments by TREC task

2.3 Best Practices

Enough different experiments have been run in TREC to support general conclusions about best practices for IR—retrieval techniques incorporated by most retrieval systems because they have been shown to be beneficial [3]. One such practice, term weighting, has already been mentioned as being critical to retrieval success.

Another primary factor in the effectiveness of retrieval systems is good query formulation. Of course, the best way of getting a good query is to have the user provide one. Unfortunately, users don't tend to provide sufficient context, usually offering a few keywords as an initial question. Retrieval systems compensate by performing *query expansion*, adding related terms to the query. There are several different ways such expansion can be accomplished, but the most commonly used method is through *blind feedback*. In this technique, a retrieval run consists of two phases. In the first phase, the original query is used to retrieve a list of documents. The top documents on the list are assumed to be relevant and are used as a source of discriminating terms; these terms are added to the query and the query is reweighted. The second phase uses the reformulated query to retrieve a second document list that is returned to the user.

Two other techniques, the use of passages and phrasing, are now used by most retrieval systems though they do not have as large an impact on the final results as weighting and query formulation do. Phrasing is the determination of compound index terms, i.e., an index term that corresponds to more than one word stem in the original text. Most frequently the phrases are word pairs that co-occur in the corpus (much) more frequently than expected by chance. Generally, both the individual word stems and the compound term are added to the query. Passages are subparts of a document. They are used as a means of finding areas of homogenous content within large documents that cover a variety of subjects.

2.4 Evaluating Retrieval System Effectiveness

Throughout this paper I assume it is possible to decide that one retrieval run is more effective than another. This subsection describes the evaluation methodology used to make this determination.

Retrieval experiments are performed using test collections. A test collection consists of a set of documents, a set of questions (called "topics" in TREC), and, for each question, a list of the documents that are relevant to that question, the *relevance assessments*. Relevance assessments are generally binary (a document is either relevant or not) and assumed to be exhaustive (if a document is not listed as being relevant, it is irrelevant).

A number of different effectiveness measures can be computed using the relevance assessments of a test collection. A very common method of evaluating a retrieval run is to plot *precision* against *recall*. Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved. While a perfect retrieval run will have a value of 1.0 for both recall and precision, in practice precision and recall are inversely related.

The effectiveness of individual queries varies greatly, so the average of the precision and recall values over a set of queries is used to compare different schemes. The precision of an individual query can be interpolated to obtain the precision at a standard set of recall values (for example, 0.0 – 1.0 in increments of .1). The precision at these recall points is then averaged over the set of queries in the test collection. The "3-point" average precision is used below as a single measure of retrieval effectiveness in a case study; this average is the mean of the precision values at each of 3 recall values (.2, .5, and .8).

Another single-valued measure called "(non-interpolated) average precision" was introduced in the TREC workshops and is used to discuss the TREC results below. The average precision for a single topic is the mean of the precision values obtained after each relevant document is retrieved. The mean average precision for a run consisting of multiple queries is the mean of the average precision scores of each of the queries in the run. In geometric terms, the average precision for a single query is the area underneath the uninterpolated recall-precision graph.

3 Current Applications of NLP to IR

Before discussing how NLP is used in IR, it is necessary to define what constitutes "natural language processing". The very fact that retrieval systems operate on natural language text and return useful results demonstrates that, at some level, text retrieval *is* natural language processing. IR systems must at least tokenize the text,[1] which is fairly trivial for English, but is more of a challenge in languages such as German (with its extensive use of compound forms) or Chinese (where there are very few syntactic clues to word boundaries). Many retrieval systems also perform stemming, a type of morphological processing.

Nonetheless, in common usage "NLP for IR" has the more specific meaning of using linguistically-inspired processing to improve text retrieval system effectiveness [4, 5]. In most cases, the NLP has focused on improving the representation of text (either documents or queries) during indexing. Matching the resulting query and document representations then proceeds in the usual way, though special processing may be used to decide if two individual terms match. For example, if index terms are noun phrases, then a partial match may be made if two terms share a common head but are not identical.

This section reviews some of the recent research in applying NLP techniques to information retrieval indexing. The section begins by examining a particular experiment as a case study of the types of issues involved when incorporating NLP techniques within existing retrieval frameworks. It then looks at the research that has been undertaken in the context of the TREC program, especially the NLP track in TREC-5 (1996) [8].

3.1 A Case Study

The case study involves an investigation into using the semantic information encoded in WordNet, a manually-constructed lexical system developed by George Miller and his colleagues at Princeton University [9], to enhance access to collections of text. The investigation took place several years ago and is described in detail elsewhere [10, 11]. It is summarized here to illustrate some of the pitfalls of linguistic processing.

WordNet is a system that reflects current psycholinguistic theories about how humans organize their lexical memories. The basic object in WordNet is a set of strict synonyms called a *synset*. By definition, each synset in which a word appears is a different sense of that word. Synsets are organized by the lexical relations defined on them, which differ depending on part of speech. For nouns (the only part of WordNet used in the experiment), the lexical relations include antonymy, hypernymy/hyponymy (*is-a* relation) and three different meronym/holonym (*part-of*) relations. The *is-a* relation is the dominant relationship, and organizes the synsets into a set of approximately ten hierarchies.

The focus of the investigation was to exploit the knowledge encoded in WordNet to ameliorate the effects synonyms and homographs have on text retrieval systems that use word matching. In the case of homographs, words that appear

[1] Not all systems tokenize the text into words. Systems based on n-grams [6] use word fragments as index terms. Other systems such as the MultiText system [7] do not index at all, but treat the entire document collection as one long string and define queries as arbitrary patterns over the string.

to be the same represent two distinct concepts, such as 'bank' meaning both the sides of a river and a financial institution. With synonyms, two distinct words represent the same concept, as when both 'board' and 'plank' mean a piece of wood. Homographs depress precision because false matches are made, while synonyms depress recall because true matches are missed. In principle, retrieval effectiveness should improve if matching is performed not on the words themselves, but on the concepts the words represent.

This idea of *conceptual indexing* is not new to IR. Controlled vocabularies generally have a canonical descriptor term that is to be used for a given concept. Concept matching has also been used successfully in limited domains by systems such as SCISOR [12] and FERRET [13]; in these systems, meaning structures are used to represent the concepts and sophisticated matching algorithms operate on the structures. Less knowledge-intensive approaches to concept matching have also been developed. For example, abstracting away from the particular words that happen to be used in a given text is the motivation behind latent semantic indexing [14]. The point of our investigation was to see if WordNet synsets could be used as concepts in a general-purpose retrieval system.

Successfully implementing conceptual indexing using synsets requires a method for selecting a single WordNet synset as the meaning for each noun in a text, i.e., a word sense disambiguation procedure. The disambiguation procedure used will not be described here. For this discussion, the important feature of the procedure is that it used the contents of a piece of text (document or query) and the structure of WordNet itself to return either one synset id or a failure indicator for each ambiguous noun in the text. The synset ids were used as index terms as described in the next paragraph.

The experiments used an extended vector space model of information retrieval that was introduced by Fox [15]. In this model, a vector is a collection of subvectors where each subvector represents a different aspect of the documents in the collection. The overall similarity between two extended vectors is computed as the weighted sum of the similarities of corresponding subvectors. That is, the similarity between query Q and document D is

$$\mathrm{sim}(Q, D) = \sum_{\text{subvector } i} \alpha_i \mathrm{sim}_i(Q_i, D_i)$$

where α_i reflects the importance of subvector i in the overall similarity between texts and sim_i is the similarity metric for vectors of type i. For the conceptual indexing experiments, document and query vectors each contained three subvectors: stems of words not found in WordNet or not disambiguated, synonym set ids of disambiguated nouns, and stems of the disambiguated nouns. The second and third subvectors are alternative representations of the text in that the same text word causes an entry in both subvectors. The noun word stems were kept to act as a control group in the experiment. When the weight of the synset id subvector is set to zero in the overall similarity measure, document and query texts are matched solely on the basis of word stems.

To judge the effectiveness of the conceptual indexing, the performance of the sense vectors was compared to the performance of a baseline run (see Table 1).

In the baseline run, both document and query vectors consisted of just one subvector that contained word stems for all content words. The table gives the effectiveness of the baseline run and three different sense-based vector runs for five standard test collections. The five test collections are

CACM: 3204 documents on computer science and 50 queries,
CISI: 1460 documents on information science and 35 queries,
CRAN: 1400 documents on engineering and 225 queries,
MED: 1033 documents on medicine and 30 queries, and
TIME: 423 documents extracted from *Time Magazine* and 83 queries.

Each row in the table gives the average 3-point precision value obtained by the four different retrieval runs for a particular collection, where the average is over the number of queries in that collection. For each of the sense-based vector runs, the percentage change in 3-point precision over the standard run is also given. Thus, the entry in row 'MED', column '211' of the table indicates that the average precision for the MED collection when searched using sense-based vectors 211 (explained below) is .4777, which is a 13.6% degradation in effectiveness as compared to the average precision of .5527 obtained when using standard stem-based vectors.

Table 1. 3-point average precision for sense-based vector runs

	Baseline	110		211		101	
Collection	3-pt	3-pt	%	3-pt	%	3-pt	%
CACM	.3291	.1994	-39.4	.2594	-21.2	.2998	-8.9
CISI	.2426	.1401	-42.3	.1980	-18.4	.2225	-8.3
CRAN	.4246	.2729	-35.7	.3261	-23.2	.3538	-16.7
MED	.5527	.4405	-20.3	.4777	-13.6	.4735	-14.3
TIME	.6891	.6044	-12.3	.6462	-6.2	.6577	-4.6

The three sense-based vector runs differ in the way the subvectors were weighted when computing the overall similarity between documents and queries, and these weights are used to label the runs. The run labeled '110' gives equal weight to the non-noun word stems and the synset ids and ignores the noun word stems. This run represents a true conceptual indexing run. The run labeled '211' gives the non-noun word stems twice the weight given to each of the synset ids and the noun word stems. This run weights the non-noun stems twice to counterbalance the fact that both the noun stems and the noun senses are included. The final run ('101') is a control run— all of the word stems get equal weight and the synset ids are ignored. This is *not* equivalent to the baseline run since the overall similarity measure only counts a term match if the term occurs in the same subvector in both the query and document.

Clearly, the effectiveness of the sense-based vectors was worse than that of the stem-based vectors, sometimes very much worse. As is usually the case with retrieval experiments, examination of individual query results shows that some queries were helped by the conceptual indexing while others were hurt by it. For example, the retrieval effectiveness of MED query 20 was improved by the sense-based vectors. Query 20 requests documents that discuss the effects of 'somatotropin', a human growth hormone. Many of the relevant documents use the variant spelling 'somatotrophin' for the hormone and thus are not retrieved in the standard run. Since the synset that represents the hormone includes both spellings as members of the set, documents that use either spelling are indexed with the same synset identifier in the sense-based run and match the query. In contrast, the retrieval effectiveness of MED query 16 was severely degraded by the sense-based vectors. The query requests documents on separation anxiety in infant and preschool children. It retrieves 7 relevant documents in the top 15 for the standard run but only 1 relevant document in the top 15 for the '110' run. The problem is selecting the sense of 'separation' in the query. WordNet contains eight senses of the noun 'separation'. With few clues to go on in the short query text, the indexing procedure selected a sense of 'separation' that was not used in any document. The query's separation concept could therefore never match any document, and retrieval performance suffered accordingly.

In this particular set of experiments, almost all of the degradation in retrieval performance can be attributed to missing term matches between documents and queries when using sense-based vectors that are made when using standard word stem vectors. The missed matches have several causes: different senses of a noun being chosen for documents and queries when in fact the same sense is used; the inability to select any senses in some queries due to lack of context; and adjectives and verbs that conflate to the same stem as a noun in the standard run but are maintained as separate concepts in the sense-based runs. The importance of finding matches between document and query terms is confirmed by the degradation in performance of the control run '101' compared to the baseline run. The only major difference between the control run, which ignores the senses and just uses the word stems, and the baseline run, which also uses only word stems, is the introduction of subvectors in the '101' run. In the sense-based vectors, stems of words that are not nouns or nouns that are not in WordNet are in one subvector and stems of WordNet nouns are in the other subvector. The extended vector similarity measure matches a word stem in the document vector only if that word stem appears in the same subvector in the query. Therefore, adjectives and verbs that conflate to the same stem as a noun get counted as a match in the baseline run but do not match in the '101' run.

Of course, the fact that the conceptual indexing failed in this one experiment does not mean that concepts are inherently inferior to word stems. A disambiguation procedure that was able to resolve word senses more consistently between documents and queries would have improved the sense-based results above, as would an indexing procedure that could recognize concepts implied by words other than nouns. But the experiment does offer some broader insights into improving word-based retrieval through linguistically selected index terms.

Linguistic techniques must be essentially perfect to help. The state of the art in linguistic processing of domain-independent text (e.g., part-of-speech tagging, sense resolution, parsing, etc.) is such that errors still occur. Thus the effect of errors on retrieval performance must be considered when trying to use these techniques to overcome the deficiencies of word stem indexing. Unfortunately, in the particular case of word sense disambiguation, a common error (incorrectly resolving two usages of the same sense differently) is disastrous for retrieval effectiveness. Sanderson found that disambiguation accuracy of at least 90% was required just to avoid degrading retrieval effectiveness [16]. This is a very high standard of performance for current NLP technology.

Queries are difficult. Queries are especially troublesome for most NLP processing because they are generally quite short and offer little to assist linguistic processing. But to have any effect whatsoever on retrieval, queries must also contain the type of index terms used in documents, or at least have some way of interacting with the documents' index terms.

Nonlinguistic techniques implicitly exploit linguistic knowledge. Even if done perfectly, linguistic techniques may provide little benefit over appropriate statistical techniques because the statistical techniques implicitly exploit the same information the linguistic techniques make explicit. Again using sense disambiguation as an example, in practice homographs are not a major contributor to retrieval failure unless the query is extremely short (one word) or the searcher is interested in very high recall [17]. If a document has enough terms in common with a query to have a high similarity to the query, then the contexts in the two texts are similar and any polysemous words will likely be used in the same sense. In fact, the IR method of computing similarities among texts can be used to build a classifier to discriminate among word senses [18].

Term normalization might be beneficial. Term normalization, i.e., mapping variant spellings or formulations of the same lexical item to a common form, may be one area in which linguistic approaches improve on simple word stems. The use of *somatotropin/somatotrophin* is one example of this effect. Proper nouns are a more general class of lexical items that word stem approaches do not handle very well, but are regular enough to be accurately captured by more sophisticated techniques [19]. Although current IR test collections do not contain enough queries that depend on proper nouns to be able to quantify how much special processing helps, in other retrieval environments such as web search engines providing special processing for names is noticeably better.

3.2 TREC-5 NLP Track

Sense resolution is but one approach to using NLP to improve indexing. The NLP track in TREC-5 invited participants to try any NLP approach on the test collection consisting of almost 75,000 *Wall Street Journal* articles (240MB of text) and TREC topics 251–300. Four groups submitted runs to the track. While the track accepted both automatic and manual runs, only the automatic runs will be discussed here in keeping with the focus of the rest of the paper.

The MITRE group [20] had experience building trainable natural language algorithms for information extraction tasks by participating in the Message Understanding Conferences (MUC). However, TREC-5 was their first entry into TREC, and they were not able to complete all they had hoped to do by the time of the TREC-5 conference. The run they did submit to the NLP track consisted of pre- and post-processing steps applied to a basic SMART[2] statistical run. The preprocessing step aimed to automatically locate and remove from the query statement extraneous material that might mislead a stem-based search. The post-processing step aimed to re-order the ranked output of the SMART search based on learning which were the important keywords and phrases in the query and giving documents containing those terms higher ranks. As implemented for the track, neither process had any appreciable impact (either positive or negative) on the SMART results.

The other three entries in the NLP track tested syntactic phrasing (sometimes in conjunction with other NLP techniques) as a possible improvement over statistical phrases. As noted in Section 2.3, one of the findings of TREC is that phrasing in some form is generally useful. Most systems use statistical phrasing where a "phrase" is any pair of words that co-occur in documents sufficiently frequently. Generally the pair and both the individual word stems are used as index terms. Statistical phrases are clearly only a rough approximation to natural language phrases. Some frequently co-occurring pairs such as 'early fourth' are not phrases at all. Documents containing non-compositional collocations such as 'hot dog' and 'White House' are still (incorrectly) indexed by their component words. Phrases longer than two words are ignored. The internal structure of the phrase is also ignored so that 'college junior' is conflated with 'junior college'. The question is to what extent these problems affect retrieval.

The Xerox TREC-5 NLP track entry directly compared the effectiveness of retrieval runs using statistical phrasing vs. a specific kind of syntactic phrasing [21]. The syntactic phrasing was accomplished by using a light parser to perform a shallow syntactic analysis of text. Pairs of words that the parse found to be in one of the following relations were extracted as phrases: subject-verb, verb-direct object, verb-adjunct, noun modifying noun, adjective modifying noun, adverb modifying verb. Phrases that included a stop word as a phrase component were discarded. For each of the remaining phrases, the component words were stemmed and alphabetically sorted to form the final index term. Figure 2, derived from figures given in the Xerox paper, shows the phrases detected by the statistical and syntactic methods for an example query.

Using the mean average precision measure to evaluate the retrieval runs, the use of the syntactic phrases increased effectiveness 15% as compared to a

[2] SMART is a retrieval system based on the vector space model that was developed at Cornell University.

Original Text (non-stopwords in *italics*):	
	Where and for what *purpose* is *scuba diving* done *professionally*?
Statistical phrases (in WSJ corpus):	
	dive_scub (diving, scuba)
Xerox syntactic phrases:	
	dive_scub (diving, scuba)
	dive_profess (diving, professionally)

Fig. 2. Phrases derived for an example query by both statistical and syntactic methods

baseline run with no phrases (from .200 to .231). Using the statistical phrases improved the mean average precision by only 7% over the same baseline (from .200 to .215), so the syntactic phrases did have a positive effect. But this gain came at a cost in processing time; indexing the 240MB document text took 36 hours longer using the parsing than it did using the statistical methods. Also, the syntactic phrasing was only beneficial when starting with the longer version of the TREC topics. When only the short version of the topics was used (e.g., a single sentence as shown in Figure 2) the syntactic phrasing run *degraded* the baseline effectiveness by 30%.

The the CLARITECH NLP track entry was also an evaluation of the use of syntactic phrases for document indexing [22]. The main goal of the study was to compare different kinds of syntactic phrases to each other, rather than compare syntactic phrases to statistical phrases. The syntactic phrases used by the CLARIT system are noun phrases, and the different types of phrases tested were full noun phrases (e.g., "heavy construction industry group"), adjacent subphrases in the noun phrase (e.g., "heavy construction industry"), and head modifier pairs (e.g, "construction industry", "industry group", "heavy construction").

Four different CLARIT runs were made: a base case consisting of only single words; single words plus head modifier pairs; single words plus head modifier pairs plus full noun phrases; and single words plus all types of phrases. The most effective run was the run that included single words plus head modifier pairs only, which increased mean average precision by 13% over the base case of words only (from .183 to .206). A second set of runs performed after TREC-5 used a more effective query weighting scheme that improved all the runs. With this weighting scheme, the head modifier pairs run was still the most effective, with an increase in mean average precision of 9% over the base case of no phrases (from .221 to .240). These results all used the long version of the topics. Even when using the long version, CLARITECH noted that they did not see as much of an effect on retrieval performance using phrases as expected because the queries contained so few phrases. They also noted that appropriately weighting phrases is an important factor in phrase-based indexing.

The focus of the GE-led TREC group has been on NLP techniques for information retrieval since TREC began [23, 5]. Because their earlier experiments demonstrated that the NLP techniques worked significantly better with longer query statements, much of their TREC-5 work was an investigation into performance of their system when the topic statements were expanded with large amounts of hand-selected document text. Such expansion significantly improves the performance of both statistical and NLP runs, though the NLP runs may get somewhat more of a boost.

TREC-5 was also the year the GE group introduced a stream architecture. In this architecture different independent processes produce index terms for a text and a combination mechanism resolves the various candidate index term sets into one final set. The stream architecture provides a convenient testbed to investigate the relative contributions of the different streams. The group implemented a variety of statistical and linguistic streams including word stems; head modifier pairs (derived from verb object and subject verb combinations in addition to noun phrases); unnormalized noun groups; and names. Similar to the CLARITECH findings, the results of the stream architecture experiments suggested that having some phrases is an improvement over no phrases, but simpler phrases (in this case the unnormalized noun groups) work better than more complicated phrases.

The TREC-5 NLP track participants found the same types of difficulties in trying to improve on statistical IR system effectiveness as were encountered in the case study. Queries are short and therefore don't offer much opportunity to perform processing that will significantly affect retrieval. Large degradation in performance is possible unless the NLP works very well and the term weighting is not disturbed. The statistical phrases capture most of the salient information that can be exploited by syntactic phrases. These are the issues that need to be addressed to improve retrieval effectiveness through linguistic processing.

4 Summary

The explosive growth in the number of full-text, natural language documents that are available electronically makes tools that assist users in finding documents of interest indispensable. Information retrieval systems address this problem by matching query language statements (representing the user's information need) against document surrogates. Intuitively, natural language processing techniques should be able to improve the quality of the document surrogates and thus improve retrieval performance. But to date explicit linguistic processing of document or query text has afforded essentially no benefit for general-purpose (i.e., not domain specific) retrieval systems as compared to less expensive statistical techniques.

The question of statistical vs. NLP retrieval systems is miscast, however. It is not a question of either one or the other, but rather a question of how accurate an approximation to explicit linguistic processing is required for good retrieval performance. The techniques used by the statistical systems are based on linguistic theory in that they are effective retrieval measures precisely because they capture important aspects of the way natural language is used. Stemming is an approximation to morphological processing. Finding frequently co-occurring word pairs is an approximation to finding collocations and other compound structures. Similarity measures implicitly resolve word senses by capturing word forms used in the same contexts. Current information retrieval research demonstrates that more accurate approximations cannot yet be reliably exploited to improve retrieval.

So why should relatively crude approximations be sufficient? The task in information retrieval is to produce a ranked list of documents in response to a query. There is no evidence that detailed meaning structures are necessary to accomplish this task. Indeed, the IR literature suggests that such structures are not required. For example, IR systems can successfully process documents whose contents have been garbled in some way such as by being the output of OCR processing [24, 25] or the output of an automatic speech recognizer [26]. There has even been some success in retrieving French documents with English queries by simply treating English as misspelled French [27]. Instead, retrieval effectiveness is strongly dependent on finding all possible (true) matches between documents and queries, and on an appropriate balance in the weights among different aspects of the query. In this setting, processing that would create better linguistic approximations must be essentially perfect to avoid causing more harm than good.

This is not to say that current natural language processing technology is not useful. While information retrieval has focused on retrieving documents as a practical necessity, users would much prefer systems that are capable of more intuitive, meaning-based interaction. Current NLP technology may now make these applications feasible, and research efforts to address appropriate tasks are underway. For example, one way to support the user in information-intensive tasks is to provide summaries of the documents rather than entire documents. A recent evaluation of summarization technology found statistical approaches quite effective when the summaries were simple extracts of document texts [28], but generating more cohesive abstracts will likely require more developed linguistic processing. Another way to support the user is to generate actual answers. A first test of systems' ability to find short text extracts that answer fact-seeking questions will occur in the "Question-Answering" track of TREC-8. Determining the relationships that hold among words in a text is likely to be important in this task.

Acknowledgements

My thanks to Donna Harman and Chris Buckley for improving this paper through their comments.

References

1. Sparck Jones, K., Willett, P. (eds.): Readings in Information Retrieval. Morgan Kaufmann, San Franciso (1997)
2. Salton, G. Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM. **18** (1975) 613–620
3. Sparck Jones, K.: Further Reflections on TREC. Information Processing and Management. (To appear.)
4. Sparck Jones, K.: What is the Role of NLP in Text Retrieval? In: Strzalkowski, T. (ed.): Natural Language Information Retrieval. Kluwer (In press.)
5. Perez-Carballo, J., Strzalkowski, T.: Natural Language Information Retrieval: Progress Report. Information Processing and Mangement. (To appear.)
6. D'Amore, R.J., Mah, C.P.: One-Time complete Indexing of Text: Theory and Practice. Proceedings of the Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press (1985) 155–164
7. Cormack, G.V., Clarke, C.L.A., Palmer, C.R., To, S.S.L.: Passage-Based Query Refinement. Information Processing and Management. (To appear.)
8. Strzalkowski, T.: NLP Track at TREC-5. Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication 500-238 (1997), 97–101. Also at `http://trec.nist.gov/pubs.html`
9. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
10. Voorhees, E.M.: Using WordNet to Disambiguate Word Senses for Text Retrieval. Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press (1993) 171–180
11. Voorhees, E.M.: Using WordNet for Text Retrieval. In: Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998) 285–303
12. Rau, L.F.: Conceptual Information Extraction and Retrieval from Natural Language Input. In: Sparck Jones, K., Willett, P. (eds.): Readings in Information Retrieval. Morgan Kaufmann, San Franciso (1997) 527–533
13. Mauldin, M.L.: Retrieval Performance in FERRET. Proceedings of the Fourteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. ACM Press (1991) 347–355
14. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. **41** (1990) 391–407
15. Fox, E.A.: Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Unpublished doctoral dissertation, Cornell University, Ithaca, NY. University Microfilms, Ann Arbor, MI.
16. Sanderson, M.: Word Sense Disambiguation and Information Retrieval. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Springer-Verlag (1994) 142–151
17. Krovetz, R., Croft, W.B.: Lexical Ambiguity in Information Retrieval. ACM Transactions on Information Systems. **10** (1992) 115–141
18. Leacock, C., Towell, G., Voorhees, E.M.: Towards Building Contextual Representations of Word Senses Using Statistical Models. In: Boguraev, B., Pustejovsky, J. (eds.): Corpus Processing for Lexical Acquisition. MIT Press (1996) 98–113
19. Paik, W., Liddy, E.D., Yu, E., Mckenna, M.: Categorizing and Standardizing Proper Nouns for Efficient Information Retrieval. In: Boguraev, B., Pustejovsky, J. (eds.): Corpus Processing for Lexical Acquisition. MIT Press (1996) 61–73

20. Burger, J.D., Aberdeen, J.S., Palmer, D.D.: Information Retrieval and Trainable Natural Language Processing. Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication 500-238 (1997), 433–435. Also at http://trec.nist.gov/pubs.html
21. Hull, D.A., Grefenstette, G., Schulze, B.M., Gaussier, E., Schütze, H., Pedersen, J.O.: Xerox TREC-5 Site Report: Routing, Filtering, NLP, and Spanish Tracks. Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication 500-238 (1997), 167–180. Also at http://trec.nist.gov/pubs.html
22. Zhai, C., Tong, X., Milić-Frayling, N., Evans, D.A.: Evaluation of Syntactic Phrase Indexing—CLARIT NLP Track Report. Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication 500-238 (1997), 347–357. Also at http://trec.nist.gov/pubs.html
23. Strzalkowski, T., Guthrie, L., Karlgren, J., Leistensnider, J., Lin, F., Perez-Carballo, J., Straszheim, T., Wang, J., Wilding, J.: Natural Language Information Retrieval: TREC-5 Report. Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication 500-238 (1997), 291–313. Also at http://trec.nist.gov/pubs.html
24. Taghva, K., Borsack, J., Condit, A.: Results of Applying Probabilistic IR to OCR Text. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Springer-Verlag, (1994) 202–211
25. Kantor, P.B., Voorhees, E.M.: Report on the TREC-5 Confusion Track. Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication 500-238 (1997), 65–74. Also at http://trec.nist.gov/pubs.html
26. Garofolo, J., Voorhees, E.M., Auzanne, C.G.P., Stanford, V.M., Lund, B.A.: 1998 TREC-7 Spoken Document Retrieval Track Overview and Results. Proceedings of the Seventh Text REtrieval Conference (TREC-7). (In press.) Also at http://trec.nist.gov/pubs.html
27. Buckley, C., Mitra M., Walz, J., Cardie, C.: Using Clustering and SuperConcepts Within SMART: TREC 6. Proceedings of the Sixth Text REtrieval Conference (TREC-6). NIST Special Publication 500-240 (1998), 107–124. Also at http://trec.nist.gov/pubs.html
28. Mani, I., House, D., Klein, G., Hirschman, L., Obrst, L., Firmin, T., Chrzanowski, M., Sundheim, B.: The TIPSTER SUMMAC Text Summarization Evaluation Final Report. MITRE Technical Report MTR 98W0000138. McLean, Virginia (1998). Also at http://www.nist.gov/itl/div894/894.02/related_projects/tipster_summac/final_rpt.html