Glean: Using Syntactic Information in Document Filtering*

Raman Chandrasekar
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
ramanc@microsoft.com

Srinivas Bangalore
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932
srini@research.att.com

Abstract

In the networked world of the information age, we are exposed to inordinate amounts of information. Search engines and information retrieval systems seek to discern the relevant from the irrelevant information given the context of a user's query. In this paper, we describe a system named Glean, which is based on the idea that coherent text contains significant latent information, such as syntactic structure and patterns of language use, which can be used to enhance the performance of information retrieval systems. We propose a trainable approach that makes use of syntactic information to increase the precision of information retrieval systems. We present results on these improvements to precision under different scenarios: using syntactic information at different granularity, and different sizes of syntactic contexts.

*This work was completed when the authors were at Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA 19104.

1 Search Engines and Syntactic Filtering

In today's networked world, a huge amount of data is available in machine-processable form. Likewise, there are any number of search engines and specialized information retrieval (IR) programs that seek to extract relevant information from these data repositories. Most IR systems and web search engines have been designed for speed, and tend to maximize the quantity of information (recall) rather than the relevance of the information (precision) to the query. As a result, search engine users get inundated with information for practically any query, and are forced to scan a large number of potentially relevant items to get to the information of interest.

The Holy Grail of information retrieval is to somehow retrieve *those and only those* documents pertinent to the user's query. Polysemy and synonymy – the fact that often there are several meanings for a word or phrase, and likewise, many ways to express a concept – make this a very hard task. While conventional IR systems provide usable solutions, there are a number of open problems to be solved, in areas such as syntactic processing, semantic analysis and user modeling, before we develop systems that 'understand' user queries and text collections. Meanwhile, we can use tools and techniques available today to improve the precision of retrieval. In particular, using the approach described in this paper, we can approximate 'understanding' using the syntactic structure and patterns of language use that is latent in documents, to make information retrieval more effective.

In this paper, we show the following:

1. Syntactic information significantly improves the effectiveness of filtering irrelevant documents.

2. A syntactic labeling technique known as *supertagging* is quite effective in filtering documents, and is better than filtering based on part of speech tagging.

3. Filtering performance can be varied by varying the extent of syntactic context.

The paper is organized as follows. In Section 2, we define our task domain. In Section 3, we discuss previous approaches to improving precision in IR. In Section 4 we describe the methodology we adopted towards achieving the task. The nature of syntactic descriptions used, and our method of extracting patterns based on these descriptions is detailed in Section 5. In Section 6, we describe three experiments to demonstrate the three points mentioned above. We discuss issues in implementing such a system in Section 7, and contrast it with MUC/TREC evaluations in Section 8. We conclude with a summary in Section 9.

Before we embark on this, we define some terms. The term *information filtering* has several interpretations. Many researchers use this term to denote the selective dissemination of information, matched to user's profiles of interest (see for instance, [Belkin and Croft, 1992]). In contrast, we view filtering as an adjunct to standard retrieval, where additional information (garnered from a user's query) is used to refine the set of retrieved documents by weeding out potentially irrelevant documents. Note that filtering can be viewed as a form of query expansion, where queries are enriched with information pertaining to the concept of interest. We also use the term *information selection* to refer to the process of *identifying* potentially relevant documents from those already retrieved using some search engine. In this paper, where the context is clear, we use the term *filtering* to refer generically to filtering or selection.

2 Task Definition: Appointments

We are interested in retrieving facts from a sample domain such as *official appointments*. In particular, we are interested in retrieving sentences where the main event is an *appointment* event, such as:

U.S. District Court Judge Michael Moore, a former head of the U.S. Marshals Service and a former U.S. attorney, was appointed to take charge of the [Elian] case after the previous judge became ill. *[CNN]*

Smith-Gardner & Associates (NASDAQ:SGAI), which produces and sells electronic commerce software, said on Wednesday it had appointed its chief executive officer, Gary Hegna, as board chairman. *[MSNBC]*

One method to identify such sentences would be to retrieve sentences with the string *appoint*. However, this is too promiscuous, since sentences that comment on appointments, sentences that include information about appointments in adjunct clauses, sentences that mention *five rooms appointed with Victorian antiques*, etc., are all likely to be retrieved by such a simple pattern.

Clearly, there are syntactic cues which could be used to filter out some of these irrelevant sentences. But this is not simple; besides, the distinction among relevant and irrelevant is often subjective. While all the following sentences contain the word *appoint*, we may consider only the first to be relevant, since sentence (b) is a generic statement about appointments of judges, and sentence (c) makes only a passing reference to an appointment.

a. The Seattle Mariners will meet today to appoint a new manager ....

b. The President appoints judges of the Supreme Court.

c. Fed Vice Chairman Alan Blinder, a Clinton appointee, has argued that a rate cut is necessary to keep the economy from slowing too sharply. *[NYT]*

If we can identify syntactic patterns of interest, we can select documents containing sentences which conform to such patterns. However, hand-crafting such patterns is tedious, time-intensive and expensive. The alternative is to develop an automated method of identifying and using patterns of relevance.

3 Improving Retrieval: Earlier Approaches

Standard retrieval engines use several techniques such as query expansion, thesauri, morphological analysis and stemming techniques [Salton, 1989, Salton and McGill, 1983, Frakes and Baeza-Yates, 1992] to improve its recall, and these are all of significant interest. However, in this paper, we will concentrate on approaches related to improving the precision of an IR system.

Relevance feedback [Rocchio, 1971] is a popular approach which allows the user, through successive interactions, to zero in on what she desires. Latent Semantic Indexing (LSI) [Deerwester et al., 1990] which clusters related documents is another interesting approach to improving IR performance. Query expansion based on linguistic tools such as WordNet [Miller et al., 1990] categories have been suggested in [Voorhees, 1993, Voorhees, 1994]. There have been approaches [Byrd et al., 1995] that provide assistance in constructing better queries based on domain knowledge. [Grefenstette, 1997] suggests a method using natural language (NL) tools to attack the problem of short (one-word) queries. Other approaches [Korfhage, 1991, Hearst and Pedersen, 1996, Hearst and Karadi, 1997] have provided extensive visualization tools to view and prune the results returned by an IR system. Clustering and phrasal analysis have together been exploited in [Anick and Vaithyanathan, 1997]. There have been approaches to improve precision that exploit the structure of documents (based on mark-up languages such as SGML and XML), instead of (or in addition to) the contents of documents. In several of these systems, the problem of information flooding is not adequately handled; the user still has to wade through a lot of information (which may be presented better, though) in order to get to the really relevant material.

There have been a different class of approaches that emphasized the use of information such as collocations or phrases [Fagan, 1987, Croft et al., 1991]) or grammatical relations derived from a parser [Strzalkowski, 1994, Strzalkowski et al., 1997], to improve the precision of IR systems.

However, these approaches require that such higher level information be used for indexing text; they cannot be used with existing collections without reindexing text, a daunting task when terabytes of data are considered.

Robison [Robison, 1970] discusses the idea of making semantic distinctions between words by using the fact that some common syntactic units (such as propositions) occur adjacent to the words whose meaning they specify (or 'govern'). His system is dependent on phrasal units which include specific function words that determine meaning.

There are also ambitious semantics-based approaches such as [Katz, 1997], which propose annotating large numbers of documents for efficient IR. This approach is interesting, but time-consuming and may not be scalable. Besides, human annotations are very subjective, and perhaps of limited utility for very large corpus sizes.

The approach closest to ours is from Microsoft Research [Braden-Harder et al, 1997], where they use matching on semantic representations called "logical forms", scoring and re-ranking to increase precision.

Most other work which mentions document filtering, for example, [Callan, 1996, Mostafa et al., 1997] usually are about selective dissemination of information, using, for example, inference networks, coupled with some machine learning.

4 The Glean Methodology

Our general approach to this problem is illustrated in Figure 1. It consists of two phases, a pattern training phase and a pattern application phase. In the pattern training phase, we manually select, from a corpus of news text, a training set of sentences which we deem to be relevant (say, news about appointments). *This is the only stage where manual intervention is required.* We associate

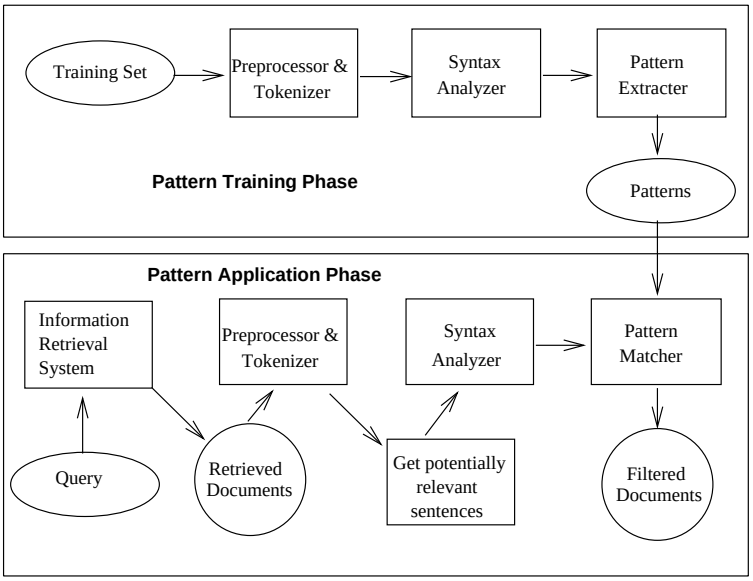

Figure 1: The Glean filtering scheme: An Overview

syntactic descriptions with words of the training sentences in the syntactic analysis phase, from which we identify contextual regularities. This gives us a set of patterns of relevance (which we call *augmented patterns*) for the domain of interest.

These augmented patterns can be used as filters during retrieval. As a first step, we use an IR system or a Web search engine to retrieve documents which are potentially relevant. Sentences which refer to the domain of interest are extracted from these documents and syntactically analyzed. The analyzed sentences are compared against the patterns of relevance obtained in the training phase to determine if the documents containing these sentences should be deemed relevant, or filtered out. Figure 1 provides an overview of the whole process. This system can also be used for ranking instead of filtering documents, by using appropriate ranking parameters such as the number of pattern-matches for a document.

We have developed a tool for information filtering based on this idea, named Glean [Ramani and Chandrasekar, 1
Glean uses the notion of augmented patterns to identify specific structural features in the text, to enhance the precision of information retrieval.

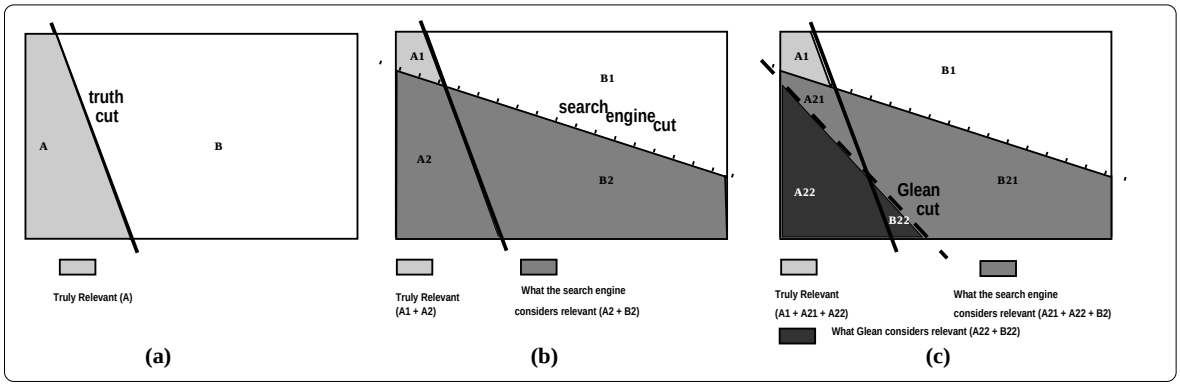

Figure 2: Selection and Filtering in Glean. (a) The truth line divides relevant documents (region A) from irrelevant ones (region B). The problem of selection reduces to the identification of the set of documents represented by A, while the problem of filtering is identifying the set B of documents to be filtered out.
(b) Given a user query, search engines partition the document space in some fashion, shown schematically by a dotted line. This partitioning may miss some relevant documents, as in region A1, and may include many false positives as in region B2.
(c) Glean's filtering reduces the false positives to a smaller set B22, but in the process may miss out some matches A21. The aim in Glean is to minimize areas A21 and B22, ideally to null regions. Thus Glean's task, given regions A2 and B2 (as output from a search engine), is to select regions A22 and B22, or identify regions A21 and B21 that have to be filtered out.

Figure 2 provides an overview of selection and filtering in Glean. If we consider the universe of all documents to be the outer rectangle, we can imagine a query (solid line in Figure 2(a)) dividing it into two regions, corresponding to relevant and irrelevant documents.

When a search engine is invoked on a query, the partitioning (shown by the dotted line in Figure 2(b)) of the space is in general different. The set of documents deemed relevant by the search engine may include some false negatives and false positives, in part because of well-known problems in IR such as polysemy and synonymy. As shown by the dashed line in Figure 2(c), Glean then filters the set of documents deemed relevant by the search engine, and prunes down the number of false positives. In the process, a few actual matches may be lost. The aim in Glean is to duplicate the ideal partitioning, the same as the one shown in Figure 2(a). Since Glean is meant to be a post-processing filter to a search engine, its performance depends on the performance of the underlying search engine. In particular, it is important that the number of false negatives returned

by the search engine is minimal, since Glean has no way to affect this.

In the next section, we discuss two tools for syntactic labeling, and describe how they are used in Glean to induce augmented patterns for use in filtering.

5 Inducing Patterns using Syntactic Labels

The syntactic description of a sentence can be obtained at different levels of detail, using different tools. Although an obvious choice would be to use a parser, parsers are not yet robust and efficient enough. Besides they do not directly associate a syntactic description with each word of a sentence: such descriptions need to be interpreted from the parses produced. In this section, we describe two techniques that directly associate syntactic labels with words: part of speech tagging and supertagging. Given either of these tagging schemes, we show how augmented patterns may be induced from tagged text.

5.1 Part-of-speech Tagging

Part-of-speech (POS) disambiguation techniques (*taggers*) have been used in several NL applications. POS taggers use information from a limited context in deciding which tag(s) to choose for each word in a text. As is well known, these taggers (for example, [Church, 1988, Brill, 1994]) are quite accurate and of significant utility.

We used an N-gram tagger [Church, 1988], with a tagset of 40 tags from the Penn Treebank [Marcus et al., 1993]. This tagset distinguishes some morphological information, such as singular and plural nouns (NN and NNS, NNP and NNPS), and different verbal categories (past, participle, continuous: VBD, VBN, VBG respectively) etc. However, the tagset does not distinguish, for instance, between the use of the word *to* as a preposition and as an infinitival marker, or between *for* as a complementizer and as a preposition. This tagger has been extensively tested,

and is about 95% correct on Wall Street Journal data.

Figure 3(a) depicts the POS tags assigned to each word of the phrase *The woman who was appointed by the Governor* and the sentence *She was appointed by the Governor in 2000.*

```
(a) The/DT woman/NN who/IN was/VB appointed/VBN by/IN the/DT Governor/NNP.
    She/PRP was/VBD appointed/VBN by/IN the/DT Governor/NNP
      in/IN 2000/CD
(b) The/B_Dnx woman/A_NXN who/B_COMPs was/B_Vvx appointed/B_N1nx1V
      by/B_vxPnx the/B_Dnx Governor/A_NXN
    She/A_NXN was/B_Vvx appointed/A_nx1V by/B_vxPnx
      the/B_Dnx Governor/A_NXN in/B_vxPnx 2000/A_NXN
```

Figure 3: (a) POS tags and (b) Supertags assigned to the phrase *The woman who was appointed by the Governor* and the sentence *She was appointed by the Governor in 2000.* Note that the two senses of *appointed* are labeled with different supertags.

5.2 LTAG and Supertagging

The other approach to syntactic annotation is based on Lexicalized Tree-Adjoining Grammar (LTAG) [Schabes et al., 1988] [1] and uses the "supertagging" technique [Joshi and Srinivas, 1994, Srinivas, 1997, Bagalore and Joshi, 1999] described in this section.

Supertags, the primary elements of the LTAG formalism, localize dependencies, including long distance dependencies, and require that all and only the dependent elements of a word be present within the same structure associated with that word. Supertags contain more information (such as sub-categorization and agreement information) than standard POS tags, and there are many more supertags per word than POS tags. For example, a word such as *appointed* would have different supertags corresponding to its use as a transitive verb, in a relative clause, in a passive construction and so on.

Figure 4 depicts the initial set and the disambiguated set of supertags assigned to each word

[1] A wide-coverage English grammar has been implemented in the LTAG framework and this grammar has been used to parse sentences from the Wall Street Journal, IBM manual and ATIS domains [Doran et al., 1994].

Sentence:	the	purchase	price	includes	two	ancillary	companies.
Initial	β_1	α_1	α_2	α_3	β_3	α_4	α_5
Assignment		β_2	α_6	α_7		β_4	α_8
		α_9	α_{10}	α_{11}		α_{12}	α_{13}
		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
Final Assignment	β_1	β_2	α_2	α_{11}	β_3	β_4	α_{13}

Figure 4: (a) A selection of the supertags associated with each word of the sentence *the purchase price includes two ancillary companies* and the most appropriate supertag sequence for the sentence.

of the sentence *the purchase price includes two ancillary companies*, where each α_n and β_n denotes a different supertag, representing various non-recursive and recursive syntactic constructs respectively.

Local statistical information, in the form of a trigram model based on the distribution of supertags in an LTAG parsed corpus, can be used to choose the most appropriate supertag for any given word. The process of assigning the best supertag to each word is termed *supertagging* [Joshi and Srinivas, 1994]. This model of supertagging is very efficient (linear time) and robust: a system trained on 1,000,000 words of Wall Street Journal text and tested on 47,000 words correctly supertagged 92.2% of these words. For a detailed discussion of the supertagger, refer to [Srinivas, 1997, Bagalore and Joshi, 1999].

Figure 3(b) depicts the supertags assigned to each word of the phrase *The woman who was appointed by the Governor* and the sentence *She was appointed by the Governor in 2000.* Note that the supertagger distinguishes between the two uses of the word `appointed`, while the POS tagger does not.

There are about 300 supertags used by the supertagger. However, supertags do not encode morphological information about words (such as number and tense information), although attribute-value pairs are used to represent this information in the LTAG grammar that the supertags are

based on. But supertags distinguish between the two different *to*'s in, for example, *I have to go to Seattle.*

5.3 Inducing and Applying Patterns

We now outline the process by which we induce patterns from tagged sentences, and how we apply these patterns in filtering.

5.3.1 Induction of Patterns

Given a word (or domain) of interest, a set of sentences containing the word or its morphological variants are chosen as training data. These sentences are tokenized and processed to identify named-entities – phrases that denote names of people, names of places, and time phrases. Each of these phrases is converted to one token. These training sentences are then syntactically analyzed using either a POS tagger or a supertagger. The output is analyzed for noun-groups (involving prenominal modifiers) and verb-groups (involving auxiliaries, modals, verbs and adverbs). At this stage, we have an high-level (abstract) view of the structure of each sentence. A sentence, now, is a list of chunks, where each chunk is a word with its tag, or a phrase denoting an entity, noun-phrase or verb-group, along with its tag(s).

To extract a pattern from the processed sentences, we consider a small window around the word of our interest, consisting of a few chunks to the left of the word (the left context) and a few chunks to the right (the right context), not counting punctuation marks. The word and the syntactic labels in this window are then generalized to a small set of augmented patterns, where each augmented pattern is a description involving tags, punctuation symbols and some closed class words. During generalization, the specific modifiers of the word of interest are transformed to match any modifiers in that position. Also, words in the neighboring chunks are replaced by a

catch-all pattern which matches any word/phrase with the same tag; this yields a level of lexical generalization. After this generalization, a pattern got from a phrase such as *she has been appointed CEO* would match a sentence with the segment *the youngest Minister is appointed President*. Thus generalization brings out the syntactic commonality among sentences, and permits an economical description of members of a set of sentences. Note that generalization is likely to increase recall and reduce precision. Figure 5 shows a sample pattern extracted by our system for the word *appoint*.

`Sample pattern:` `\S*/A_NXN \S*:E_VGQUAL appointed:A_nx0Vnx1/E_VG \S*/A_NXN`

Key: \S* refers to any word/phrase; E_VGQUAL is any set of verbal qualifiers; E_VG is a verb group. A_NXN is a noun-phrase supertag, and A_nx0Vnx1 refers to a (transitive) verb preceded and followed by a noun-phrase.

Figure 5: A sample supertag pattern

Augmented patterns are induced for all the sentences in the training data, sorted, and duplicates removed. The resulting patterns are used for filtering.

5.3.2 Application of Patterns

The task in the application phase is to use the patterns induced in the pattern training phase to classify new sentences into relevant and irrelevant ones, corresponding to the tasks of selection and filtering. The relevance of a document is decided on the basis of the relevance of the sentences in the document.

In the pattern application phase, all the (possibly relevant) sentences of each document are subjected to the same stages of processing as the training sentences. Each such sentence is chunked using simple named-entity recognition and then tagged. The tags for the words in each sentence are used to identify noun and verb chunks. The sentence is then matched against the patterns obtained from the training phase. Since this pattern matching is based on simple regular expressions specified over words and supertags, it is extremely fast and robust. A sentence is deemed to be relevant to

a concept word if it matches at least one of the patterns for that word. A document is deemed relevant if it contains at least one relevant sentence.

6 Information Filtering Experiments

We have tried out several experiments to determine the efficacy of using syntactic information in filtering machine-readable documents, using the tools described in the previous section. This section describes three such experiments, which pose the questions in the context of information filtering:

a. Is syntactic information useful?

b. What granularity of syntactic information is useful?

c. How much syntactic context is useful?

6.1 The Utility of Syntactic Information

The objective of this experiment [Chandrasekar and Srinivas, 1997a] is to show that syntactic information improves information filtering performance and to quantitatively measure the improvement in performance.

6.1.1 Training: Creating Augmented Patterns

The training corpus constituted of a corpus of approximately 52 MB of New York Times (NYT) text data comprising of the August 1995 output. The corpus was sentence-segmented, and all sentences from this corpus that contained the word *appoint* or any of its morphological variants were extracted using the Unix tool `grep`. The 494 sentences containing *appoint** were examined manually, and 56 sentences relevant to appointments being announced were identified. Twenty-one

distinct augmented-patterns were then automatically induced from these relevant sentences, using a left context of one chunk and a right context of one chunk.

System	Total Docs	Relevant Docs	Selection			Filtering		
			Total	Correct	Incorrect	Total	Correct	Incorrect
Plain Web Search (Without Glean)	84	28	84	28	56	0	0	0
With Glean filtering	84	28	29	23	6	55	50	5

Table 1: Classification of the documents retrieved for the search query

System	Selection		Filtering	
	Recall	Precision	Recall	Precision
Plain Web Search (Without Glean)	(28/28) = 100%	(28/84) = 33.3%	–	–
With Glean filtering	(23/28) = 82.1%	(23/29) = 79.3%	(50/56) = 89.3%	(50/55) = 90.9%

Table 2: Precision and Recall of Glean for (a) selecting relevant documents and (b) filtering out irrelevant documents

6.1.2 Testing Phase

Given a search expression, Glean fetches the URLs (Uniform Resource Locators) of the documents that match the search expression, using a publicly available search engine. Duplicate URLs are deleted and the document corresponding to each URL is then retrieved. Each retrieved document is then processed using the tools described in the pattern application section. A document is deemed relevant if it matches at least one of the patterns induced for that domain.

For the particular experiment we performed, we used the Alta Vista Web search engine to retrieve the URLs matching a search expression, using the WWW::Search Perl module distributed

from ISI[2]. The document corresponding to each matching URL was downloaded using a simple socket application program, with timeouts to account for busy networks, small bandwidth connections, or failed connections.

To restrict the test set retrieved to a manageable number, we searched the Web using the search expression shown below, where we require that the documents retrieved contain the words/expressions *Fortune 500*, *company* and *CEO*, as well as a form of the word *appoint*:

```
+appoint* +"Fortune 500" +company +CEO
```

A total of 100 URLs matched this query. Documents corresponding to 16 of these URLs were not retrieved due to problems not uncommon on the Web, such as network failure and timeouts. The 84 documents that were retrieved were hand-checked for relevance and 28 documents were found to be relevant. The 84 retrieved documents were also subjected to the filtering process described in the pattern application section. This classified 29 documents as relevant, of which 23 documents matched the hand-classified data. Tables 1 and 2 show the performance of the system in terms of recall and precision for selecting relevant and filtering out irrelevant documents. The first row shows the performance of the Web search engine, while the second row shows the performance of Glean's filtering on the output of the Web search engine.

6.1.3 Discussion

As seen in Tables 1 and 2, Glean filtering increases precision significantly. Compared to the number of sentences in the column under *Total Docs* (which is the total number of documents retrieved by the plain search), the number of documents marked relevant is about a third of the total number. Note that the patterns for the *appoint* concept were extracted from New York Times data and applied with a high degree of success to data retrieved from the Web.

[2] `http://www.isi.edu/lsam/tools/WWW_SEARCH/`

A novelty of our approach is that syntactic information is used as a post-retrieval filter, and hence decoupled from indexing and basic retrieval steps. As a result of this decoupling, this system can be used to filter the output of any IR system.

Errors in supertagging, or even errors in spelling, could lead to mis-categorization of documents. Errors in supertagging during the creation of patterns may cause extremely specific patterns to be created, which may not be a serious problem, since these patterns are not likely to match any of the input. Interestingly the performance of the system is not affected significantly if the supertagger is consistent in its errors, both in the training and test phases.

Filtering may often require information beyond what is available from syntax. For example, we chose not to define sentences which talk about appointments in a very general sense (as in *After the takeover, a new CEO will be appointed*) as being relevant. This may syntactically correspond to a standard sentence about an appointment event, and so it is not easy to filter such sentences out, without additional information.

6.2 Comparing POS Tagging with Supertagging

In the second set of experiments [Chandrasekar and Srinivas, 1997c], we demonstrate that richer syntactic information improves performance of the filtering system. We describe experiments where we use techniques discussed in the previous sections to retrieve relevant documents about appointments. We quantitatively measure the performance of two syntactic labeling techniques – POS-tagging and supertagging – in filtering out irrelevant documents.

POS: `\S*/E_NG \S*:E_VGQUAL appointed:VBN/E_VG \S*/E_NG`
Supertag: `\S*/A_NXN \S*:E_VGQUAL appointed:A_nx0Vnx1/E_VG \S*/A_NXN`
Key: \S* refers to any word/phrase; E_VGQUAL is any set of verbal qualifiers; E_VG is a verb group. A_NXN is a noun-phrase supertag, and A_nx0Vnx1 refers to a verb preceded and followed by a noun-phrase: a transitive verb. E_NG is a tag for a noun phrase, and VBN is a POS tag for a past participle verb.

Figure 6: Sample patterns involving POS tags and Supertags

We performed experiments using the words *appoint*, where the dominant sense is relevant to our task, and *nominate*, where the dominant sense is not relevant to our task. This provided us some insight into the effect of the ratio of the number of relevant documents to the number of irrelevant documents on the performance of our system.

6.2.1 Experiments with Appoint

The training set for this experiment was the same as the one mentioned in the previous section. The 56 training sentences were POS tagged and processed to obtain 20 generalized patterns. Similarly, 21 distinct supertag-based patterns were also induced. Again, left and right contexts were set to one chunk each.

Sample patterns involving POS tags and supertags respectively, which match sentences that contain a noun phrase, followed by the transitive verb *appointed*, possibly qualified by auxiliaries and preverbal adverbs, and followed by a noun phrase, are shown in Figure 6.

The test set for *appoint* was 529 sentences containing the word or a variant, extracted from the July 1995 NYT wire service text. The gold standard was independently created by manually examining these sentences and classifying them into 95 relevant sentences and 434 irrelevant sentences (with respect to the task).

Domain	Total Docs	Relevant Docs	Selection			Filtering		
			Total	Correct	Incorrect	Total	Correct	Incorrect
NYT July95 (Supertags)	529	95	168	77	91	361	343	18
NYT July95 (POS tags)	529	95	73	42	31	456	392	64
Base Case (all appoint*)	529	95	529	95	434	0	0	0

Table 3: Classification of **appoint**$*$ sentences

Domain	Selection			Filtering		
	Recall	Precision	F-score ($\beta = 1$)	Recall	Precision	F-score ($\beta = 1$)
NYT July95 (Supertags)	81%	49%	61	79%	95%	86
NYT July95 (POS tags)	44%	58%	50	90%	86%	88
Base Case (all appoint*)	100%	18%	31	0%	(0/0)	–

Table 4: **Appoint**: Precision and Recall of different filters, for (a) **selection** and (b) **filtering**

The patterns obtained from the training phase were applied to the 529 sentences. The results of these experiments are summarized in Table 3, for the supertag method, the POS tag method and for the base case. The second column in the table gives the count of sentences judged relevant by humans. The columns that follow list judgments made by the program, and the overlap they have with the gold standard. Table 4 shows the recall and precision for selection and filtering, for the word *appoint*. It also shows the F-score measures[3] for β=1, when precision is as important as recall.

6.2.2 Experiments with Nominate

The training corpus for the word *nominate* was collected from the January, February and July 1995 Los Angles Times (LAT) wire service data. The training corpus was partitioned into relevant and irrelevant sets of sentences based on the two broad senses of *nominate* – nomination for an *award/honor* versus nomination for a *post*. We considered sentences that contained the second sense of *nominate* (as the matrix verb) as relevant, and those with the first sense of *nominate* as irrelevant. Of the 190 sentences that mentioned the word *nominate*, only 19 were considered

[3]F-score provides a method of combining recall and precision, and is defined as follows, where β is a parameter that can be set to represent the relative importance of precision to recall:

$$\text{F-score} = \frac{(\beta^2 + 1) * P * R}{(\beta^2 * P) + R}$$

relevant according to our criteria. A total of 7 patterns based on POS tags and a total of 6 patterns based on supertags were extracted from this training set of 19 sentences.

Based on our criteria, the dominant sense of the word *nominate* was regarded as irrelevant. This scenario is common in web searching where the set of irrelevant documents could be extremely large compared to the set of relevant documents.

Domain	Total Docs	Relevant Docs	Selection			Filtering		
			Total	Correct	Incorrect	Total	Correct	Incorrect
LAT (Supertags)	174	12	31	8	23	143	139	4
LAT (POS tags)	174	12	34	4	30	140	132	8
Base Case (all nominate*)	174	12	174	12	162	0	0	0

Table 5: Classification of **nominate**$*$ sentences

Domain	Selection			Filtering		
	Recall	Precision	F-score ($\beta = 1$)	Recall	Precision	F-score ($\beta = 1$)
LAT (Supertags)	67%	25%	36	86%	97%	91
LAT (POS tags)	33%	12%	18	94%	81%	87
Base Case (all nominate*)	100%	7%	13	0%	(0/0)	–

Table 6: **Nominate**: Precision and Recall of different filters, for (a) **selection** (b) **filtering**

The test corpus for the word *nominate* consisted of 174 sentences containing the word or its morphological variant, from the March through June issues of LAT news service. Of these, 12 were hand-tagged to be relevant. The patterns obtained from the training phase using POS tags and supertags were applied to the 174 documents and 4 and 8 documents were tagged as relevant respectively.

We show below two examples each of sentences selected and filtered by the system, using

supertags:

Relevant sentences:

1. *At a convention of his nationalist Power political movement, supporters nominated the 47-year-old for the August 2000 balloting.*
2. *As fate would have it, White this month was nominated to replace John Deutch as deputy defense secretary, putting him in position to implement the commission's proposals.*

Irrelevant sentences:

1. *Nair, Harvard University-educated native of India who directed "Mississippi Masala" and the best foreign language film Oscar-nominated "Salaam Bombay!," said of the 78 speaking roles in "The Perez Family," 76 are by Cubans.*
2. *And Deutch, whom President Clinton has nominated to take over the CIA, indicated that he did not agree with Republican lawmakers who have claimed that the incident warranted a fundamental reappraisal of the U. S. policy of sharing its intelligence with the United Nations .*

The performance of the system on selection and filtering is tabulated in Table 5. Table 6 shows the corresponding recall and precision figures. It also shows F-scores for $\beta = 1$.

6.2.3 POS Tagging & Supertagging: Merits and Demerits

Filtering with either POS tagging or supertagging is better at reducing information overload, than retrieval without filtering.

The tags (POS and supertag labels) employed in our approach serve to categorize different syntactic contexts of a word. Supertags provide a level of descriptive granularity, which is neither at the level of complete parses (which may often be hard or impossible to obtain), nor at the simple level of parts of speech. In general, a richer set of tags leads to better discrimination. Since supertags provide a richer representation than POS tags, it is not surprising that they yield better discrimination.

Because both POS tagging and supertagging use statistical methods, the genre of the training material and the noise in that material may introduce biases. In addition, the vastly bigger tagset for supertagging makes it more prone to mistakes than POS tagging. Further, errors in tagging during the pattern training or pattern application phases can cause erroneous patterns to be created and lead to mis-categorization of documents.

From error analysis, we discovered that over-generalization lowered precision. For example, we achieved better performance by retaining prepositions instead of generalizing them. The size of the window used (one chunk on either side of the domain term) in creating patterns is sometimes inadequate. Syntactic phenomena occurring outside this window is not captured; for example relative clauses are not signaled when a relativizer is more than one chunk away. The effect of increasing the window size (for example, to include two chunks on the left) is discussed in the next section.

6.3 Varying Context in Syntactic Patterns

In this experiment [Chandrasekar and Srinivas, 1997b], we quantitatively measure the performance difference arising from variations in the size of the context around words of interest. Again, we performed experiments using two words: the word *appoint* where the dominant sense is relevant to us, and the word *nominate* where the dominant sense is not relevant for our application.

The test and training sets for this experiment, for the words *appoint* and *nominate*, were the same as those described in the previous section. The experiment was structured as follows. For each of the pairs of training and test sets, the left and right contexts were each varied from 1 to 4. For each of these 16 combinations, the training patterns were induced from the training set, and the patterns tested against the test set. The results of each test was evaluated against the gold standard.

Context L-R	Sample Induced Pattern
1-1	`\S*/A_NXN \S*:E_VGQUAL~appoint:A_nx0Vnx1/E_VG \S*/A_NXN`
1-2	`\S*/A_NXN \S*/\S*?PU\S*? appointed:A_nx0Vnx1/E_VG \S*/A_NXN to/B_vxPnx`
1-3	`\S*/A_NXN \S*:E_VGQUAL~appointed:A_nx0Vnx1/E_VG \S*/A_NXN \S*/A_NXN \S*/B_nx1CONJnx2`
1-4	`\S*/A_NXN \S*:E_VGQUAL~appoint:A_nx0Vnx1/E_VG \S*/A_NXN \S*/A_NXN in/B_vxPnx \S*/A_NXN`
2-1	`\S*/B_ARBs \S*/A_NXN appointed:A_nx0Vnx1/E_VG \S*/A_NXN`
2-2	`\S*/B_COMPs \S*/A_NXN \S*:E_VGQUAL~appointed:A_nx0Vnx1/E_VG \S*/A_NXN \S*/A_NXN`
2-3	`\S*/B_COMPs \S*/A_NXN \S*:E_VGQUAL~appointed:A_nx0Vnx1/E_VG \S*/A_NXN \S*/E_VG \S*/A_NXN`

Table 7: Sample patterns for the word *appoint*, for different context sizes. A context size of 2-1 indicates 2 chunks to the left and 1 chunk to the right.

Table 7 shows patterns (describing one transitive reading of the verb *appoint*) for different values of window sizes (that is, at different levels of contextual description).

The performance of the system in terms of retrieving relevant documents and filtering out irrelevant documents is shown in a series of plots in Figure 7 for the word *appoint*, and Figure 8 for the word *nominate*. In each of these figures, on the left, we see the effect of increasing context on the number of unique patterns created. The next plot shows the precision-recall figures for the selection task, for various context combinations. Similarly, the last plot shows the precision-recall figures for filtering. The results are discussed in the next section.

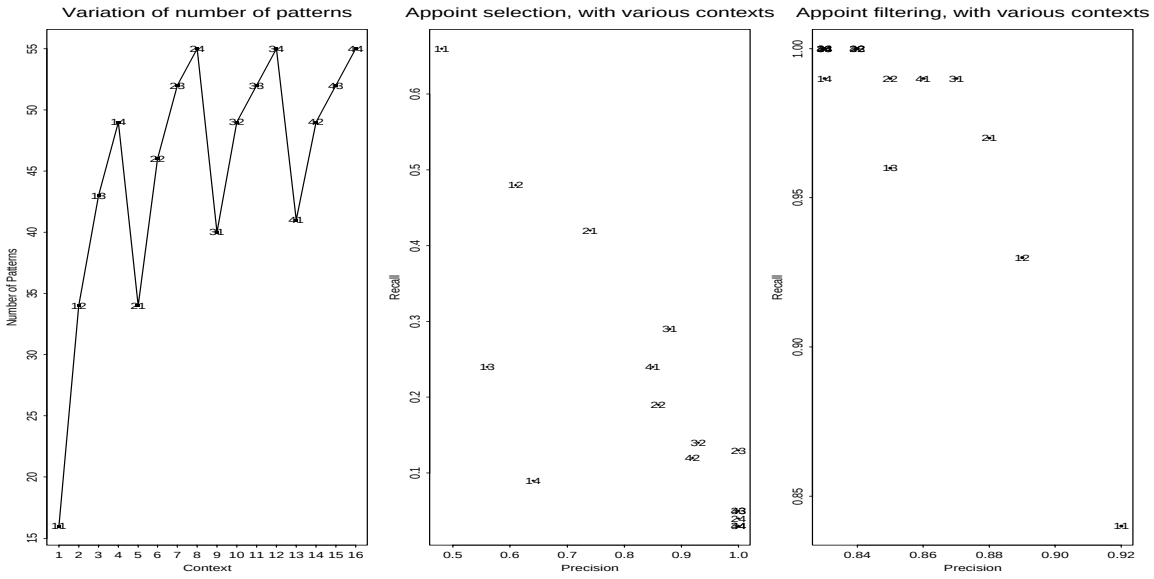

Figure 7: Variation with context for the word *appoint* of (a) number of patterns, and precision and recall for (b) selection and (c) filtering. The X-axis in (a) and selected points in (b) and (c) are labeled with left and right context. For example, the label 3-1 indicates a left context of 3 chunks and right context of 1 chunk. In (a), note that the number of patterns increases with the size of the context. From (b) and (c), a left context of 2 and a right context of 1 is optimal both for filtering and for selection.

6.3.1 Discussion of Experimental Results

As is evident from the tables, the number of patterns generated increases with context size, and saturates at the number of training sentences. Also, increased context increases precision, due to additional constraints; on the other hand it decreases recall, since the additional constraints reduce the number of matching sentences. In contrast, in the filtering task, we find that precision drops and recall increases as the window size increases (see Figure 7). In both the selection and filtering tasks, for both the words studied, we find a left context of 2 chunks and a right context of 1 chunk performs the best. Interestingly, similar findings have been reported in the domain of lexicography, in the work of [Martin et al., 1983].

Figure 9 shows that a left window of two chunks and a right window of one chunk maximizes the F-scores both for selecting relevant documents, and for filtering irrelevant documents. This

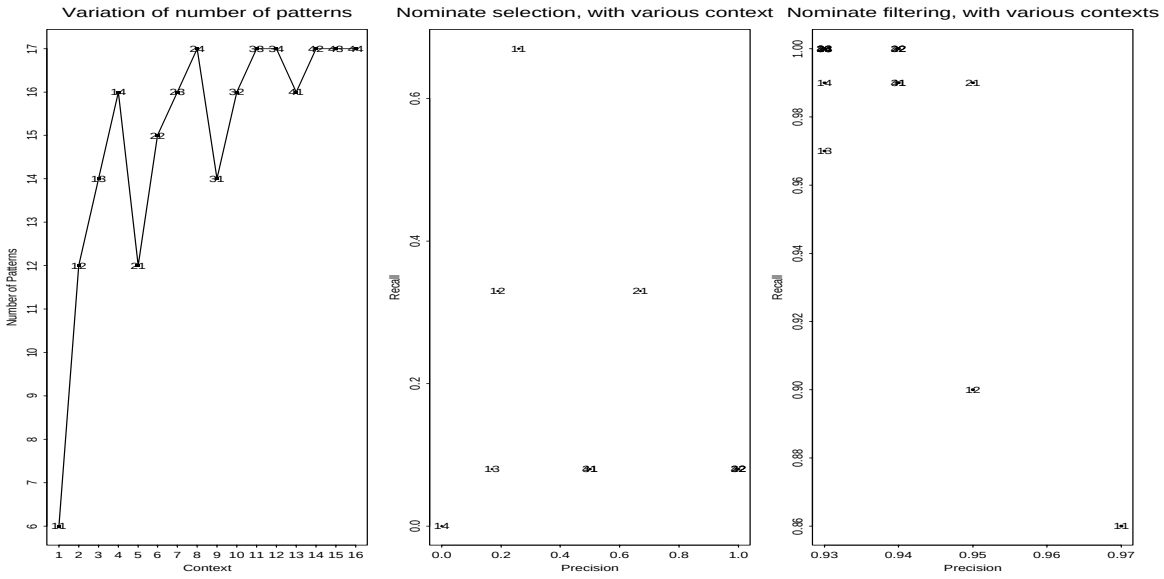

Figure 8: Variation with context for the word *nominate* of (a) number of patterns, and precision and recall for (b) selection and (c) filtering. The X-axis in (a) and selected points in (b) and (c) are labeled with left and right context. For example, the label 3-1 indicates a left context of 3 chunks and right context of 1 chunk. In (a), note that the number of patterns increases with the size of the context. From (b) and (c), a left context of 2 and a right context of 1 is optimal both for filtering and for selection, as was the case for *appoint*.

optimal value of window size linguistically corresponds to the intuition that the left context contains the information such as relative pronouns and complementizers required to discriminate between embedded and main clauses.

The picture for the word *nominate* is different. In the case of *appoint*, 95 out of 529 sentences, or about 18% were relevant. For *nominate*, only (12/174) = 7% (approx.) were relevant. Tightening the pattern (making it more specific) tends to reduce the number of matches to zero very quickly, as the window sizes increase. Thus the dominance of the relevant sense seems to play a major role in the effectiveness of filtering.

7 Implementation Issues

Glean was developed in a research collaboration between the National Centre for Software Technol-

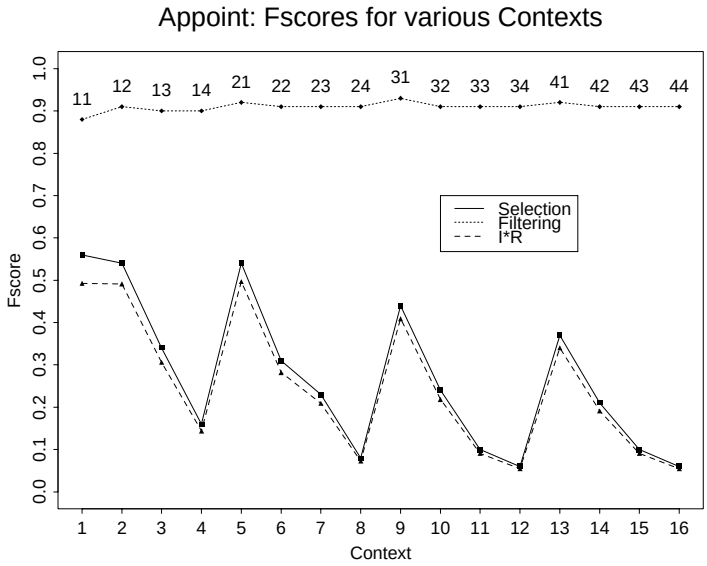

Figure 9: The F-scores (with β=1) for selection and filtering, for the word *appoint*, for various window sizes. The figure also shows (dashed line) the product of the two curves (S*F), which represents the performance on both selection and filtering. Note that a left context of 2 and a right context of 1 is best (F-score$_S$ = 0.54, F-score$_F$ = 0.92).

ogy (NCST), Bombay, the Institute for Research in Cognitive Science (IRCS) and the Center for the Advanced Study of India (CASI), University of Pennsylvania.

A prototype with a Web interface has been developed to demonstrate testing of sentences, text files and Web documents for relevance, as determined by augmented-patterns corresponding to a particular concept. A component of the system allows the user to obtain or add augmented-patterns for specific concepts, given appropriate training data. Users may also create patterns on-the-fly by specifying exemplar sentences (or selecting them from a corpus of sentences); this is useful in accommodating differences in definitions of concepts across users.

In its most general form as a information filtering device, Glean obtains from the user a query which consists of: a query expression, one or word(s) of interest, and optionally, the size of left and right contexts. Augmented patterns are looked-up or generated for each concept word. The system then runs the query on a specified search engine, obtains URLs for matching documents, and access and checks these documents for relevance.

The system is implemented as a series of programs in interpreted Perl, efficiently coded as a combination of server processes and asynchronous code segments. We are trying out various techniques, including relevance feedback methods, to make it simpler for users to create augmented patterns.

There are several other methods by which retrieval efficiency could be enhanced. For example, we have proposed server-side filtering as an innovative method for improving the efficiency of retrieval on the Web [Chandrasekar et al., 1997]. The idea here is to move the filtering code to the site of potentially relevant documents, which would be useful if the filtering code is substantially smaller then the document(s) to be processed at any site. Other methods may easily be incorporated into Glean, since it has been designed with an open architecture.

8 Comparison with MUC and TREC

We now contrast our work against tasks in the Message Understanding Conferences (MUC) [MUC, 1996] and the Text Retrieval Conferences (TREC) [Voorhees and Harman, 1998].

MUC tasks are typically set up to extract information in the form of a predefined template from an *a priori* specified set of documents and topics. For example, the task may be to fill in management change templates, which show *who* replaced *whom* in *what* post in *which* organization for *what* reason. The information that is needed to fill these fields may be explicitly present, but distributed all over a given document. In some cases, this information may have to be inferred using sophisticated natural language techniques. But fundamentally, it is assumed that the documents provided are relevant, and likely to provide the fillers for the templates.

Our task is vastly different. Given a set of seemingly relevant documents from a search engine, we seek to identify precisely those documents which are relevant to our topics of interest, using

sentence-level syntactic information. Our task is not to extract information, but to determine whether a document is relevant or not, based on a few sentences in the document which contain words related to the concepts of interest. The attempt here is to approximate semantic information using syntactic methods. Glean is meant to work on the output of IR systems to refine their results. The output of Glean, in turn, may be used in information extraction tasks such as those in MUC.

On the other hand, queries in TREC tasks are fairly complex, and resemble queries that users may pose to (for instance) research librarians. Ideally, a system would need to 'understand' these queries, decide on relevant keywords to search for, eliminate keywords based on features such as negation and parentheticals, and arrive at a query or queries which retrieve documents directly relevant to the query. The task we attempt in Glean is thus much closer to the TREC task.

TREC participants have used a variety of techniques, including phrase detection and thesaurus-based query expansion, to refine their queries. Attempts have been made, with some success, to improve performance using sophisticated linguistic tools, for example to determine relative relevance of query words, and the operators to be used in queries. Significant research has focused on the relative utilities of the terms in the *topic*, *description* and *narrative* components of TREC queries.

9 Summary

We demonstrated in this paper the utility of syntactic information in improving the performance of IR systems. We outlined some approaches to this problem, and described the ideas underlying a system named Glean, that uses syntactic knowledge to increase the precision of information retrieval. We provided quantitative results on the performance improvement gained by using richer syntactic descriptions (supertags) than by using simple parts of speech. We also examined the effect of increasing syntactic context on the effectiveness of information filtering. The system has been

tested on a collection of newswire documents, and has achieved recall and precision figures of 88% and 97% for filtering out irrelevant documents. We also propose mechanisms such as server-side filtering for efficient filtering over the Web.

References

[Anick and Vaithyanathan, 1997] Anick, P. G. and Vaithyanathan, S. (1997). Exploiting clustering and phrases for context-based information retrieval. In *Proc. ACM SIGIR-97*, pages 314–323.

[Bagalore and Joshi, 1999] Srinivas Bangalore and Aravind Joshi. (1999). Supertagging: An Approach to Almost Parsing. In *Computational Linguistics*, 25:2, 1999.

[Belkin and Croft, 1992] Belkin, N. and Croft, W. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12).

[Braden-Harder et al, 1997] Lisa Braden-Harder, Simon Corston, Bill Dolan, Lucy Vanderwende (1997). Using Natural Language to Improve Precision in Information Retrieval *MS Internal Report, October 1997.*

[Brill, 1994] Brill, E. (1994). Some advances in transformation-based part of speech tagging. In *Proceedings of AAAI94.*

[Byrd et al., 1995] Byrd, R., Ravin, Y., and Prager, J. (1995). Lexical assistance at the information-retrieval user interface. In *Fourth Annual Symposium on Document Analysis and Information Retrieval.*

[Callan, 1996] Jamie Callan (1996). Document Filtering with Inference Networks In *Proceedings of SIGIR'96, Zurich.*

[Chandrasekar and Srinivas, 1997a] Chandrasekar, R. and Srinivas, B. (1997a). Gleaning information from the Web: Using syntax to filter out irrelevant information. In *Proceedings of AAAI 1997 Spring Symposium on NLP on the World Wide Web*.

[Chandrasekar and Srinivas, 1997b] Chandrasekar, R. and Srinivas, B. (1997b). Using supertags in document filtering: The effect of increased context on information retrieval effectiveness. In *Proceedings of Recent Advances in NLP (RANLP) '97*, Tzigov Chark.

[Chandrasekar and Srinivas, 1997c] Chandrasekar, R. and Srinivas, B. (1997c). Using syntactic information in document filtering: A comparative study of part-of-speech tagging and supertagging. In *Proceedings of RIAO '97*, pages 531–545, Montreal.

[Chandrasekar et al., 1997] Chandrasekar, R., Srinivas, B., and Sarkar, A. (1997). Sieving the Web: Improving search precision using information filtering. Submitted for publication.

[Church, 1988] Church, K. W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *2nd Applied Natural Language Processing Conference*, Austin, Texas.

[Croft et al., 1991] Croft, B. W., Turtle, H. R., and Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. In *Proceedings of the* 14^{th} *Annual International Conference on Research and Development in Information Retrieval (SIGIR '91)*, pages 32–45, Chicago, USA.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Socety for Information Science*, 41(6):391–407.

[Doran et al., 1994] Doran, C., Egedi, D., Hockey, B. A., Srinivas, B., and Zaidel, M. (1994). XTAG System - A Wide Coverage Grammar for English. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan.

[Fagan, 1987] Fagan, J. L. (1987). *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Nonsyntactic Methods.* PhD thesis, Cornell University. TR87-868.

[Frakes and Baeza-Yates, 1992] Frakes, W. B. and Baeza-Yates, R. S. (1992). *Information Retrieval: Data Structures and Algorithms.* Prentice Hall.

[Grefenstette, 1997] Grefenstette, G. (1997). SQLET: Short Query Linguistic Expansion Techniques, palliating one-word queries by providing intermediate structure to text. In *Proceedings of RIAO '97*, pages 500–509, Montreal.

[Hearst and Karadi, 1997] Hearst, M. A. and Karadi, C. (1997). Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proc. ACM SIGIR-97*, pages 246–255.

[Hearst and Pedersen, 1996] Hearst, M. A. and Pedersen, J. O. (1996). Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proc. ACM SIGIR-96.*

[Joshi and Srinivas, 1994] Joshi, A. K. and Srinivas, B. (1994). Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan.

[Katz, 1997] Katz, B. (1997). Annotating the world wide web using natural language. In *Proceedings of RIAO '97*, pages 136–155, Montreal.

[Korfhage, 1991] Korfhage, R. R. (1991). To see, or not to see – is that the query? In *Proc. ACM SIGIR-91*, pages 134–141.

[Marcus et al., 1993] Marcus, M. M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19.2:313–330.

[Martin et al., 1983] Martin, W., Al, B., and van Sterkenburg, P. (1983). On the processing of a text corpus: from textual data to lexicographical information. In Hartmann, R., editor, *Lexicography: Principles and Practices*, Applied Language Studies Series. Academic Press, London.

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

[Mostafa et al., 1997] Mostafa, J., S. Mukhopadhyay, M. Palakal and W. Lam (1997). A multi-level approach to intelligent information retrieval: model, system, and evaluation. *ACM Trans. Information Systems*, 15(4): 368-399.

[MUC, 1996] MUC (1996). *Message Understanding Evaluation and Conference: Proceedings of the 6th ARPA Workshop*. Morgan Kaufmann.

[Ramani and Chandrasekar, 1993] Ramani, S. and Chandrasekar, R. (1993). Glean: a tool for automated information acquisition and maintenance. Technical report, National Centre for Software Technology, Bombay.

[Robison, 1970] Robison, H. R. (1970). Computer-detectable semantic structures. *Information Storage and Retrieval*, 6:273–288. Pergamon Press.

[Rocchio, 1971] Rocchio, J. (1971). Relevance Feedback in Information Retrieval. In Salton, G., editor, *The Smart System – Experiments in Automatic Document Processing* . Prentice-Hall Inc., Englewood Cliffs, NJ.

[Salton, 1989] Salton, G. (1989). *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley.

[Salton and McGill, 1983] Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

[Schabes et al., 1988] Schabes, Y., Abeillé, A., and Joshi, A. K. (1988). Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. In *Proceedings of the* 12^{th} *International Conference on Computational Linguistics (COLING'88)*, Budapest, Hungary.

[Srinivas, 1997] Srinivas, B. (1997). *Complexity of Lexical Descriptions and its Relevance to Partial Parsing.* PhD thesis, University of Pennsylvania, Philadelphia, PA.

[Strzalkowski, 1994] Strzalkowski, T. (1994). Document indexing and retrieval using linguistic knowledge. In *Proceedings of RIAO '94*, pages 131–145, New York.

[Strzalkowski et al., 1997] Strzalkowski, T., Lin, F., Perez-Carballo, J., and Wang, J. (1997). Building effective queries in natural language information retrieval. In *Proc. 5th Conference on Applied Natural Language Processing*, pages 299–306, Washington. ACL.

[Voorhees and Harman, 1998] Voorhees, E. and Harman, D., editors (1998). *The Sixth Text REtrieval Conference.* Department of Commerce, National Institute of Standards and Technology. Proc. 6th TREC, Gaithersburg, MD, Nov. 19-21, 1997.

[Voorhees, 1993] Voorhees, E. M. (1993). Using WordNet to disambiguate word senses for text retrieval. In *Proc. ACM SIGIR-93*, pages 171–180.

[Voorhees, 1994] Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *Proc. ACM SIGIR-94*.