December 1996

# Using Syntactic Information in Document Filtering: A Comparative Study of Part-of-Speech Tagging and Supertagging

R. Chandrasekar
*University of Pennsylvania*

B. Srinivas
*University of Pennsylvania*



University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-29.

# Using Syntactic Information in Document Filtering: A Comparative Study of Part-of-Speech Tagging and Supertagging


## Abstract

Any coherent text contains significant latent information, such as syntactic structure and patterns of language use. This information can be exploited to overcome the inadequacies of keyword-based retrieval and make information retrieval more efficient. In this paper, we demonstrate quantitatively how syntactic information is useful in filtering out irrelevant documents. We also compare two different syntactic labelings-- simple Part-of-Speech (POS) labeling and Supertag labeling-- and show how the richer (more fine-grained) representation of supertags leads to more efficient and effective document filtering. We have implemented a system which exploits syntactic information in a flexible manner to filter documents. The system has been tested on a large collection of news sentences, and achieves an F-score of 89 for filtering out irrelevant sentences. Its performance and modularity makes it a promising postprocessing addition to any Information Retrieval system.



## Comments

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-29.


*Institute for Research in Cognitive Science*

# Using Syntactic Information in Document Filtering: A Comparative Study of Part-of-speech Tagging and Supertagging

**R. Chandrasekar**
**B. Srinivas**

**University of Pennsylvania**
**3401 Walnut Street, Suite 400A**
**Philadelphia, PA 19104-6228**

**December 1996**

**Site of the NSF Science and Technology Center for Research in Cognitive Science**

**IRCS Report 96--29**

# Using Syntactic Information in Document Filtering: A Comparative Study of Part-of-speech Tagging and Supertagging

**R. Chandrasekar***
Institute for Research in Cognitive Science &
Center for the Advanced Study of India

**B. Srinivas**
Department of Computer &
Information Science

University of Pennsylvania, Philadelphia, PA 19104
{mickeyc,srini}@linc.cis.upenn.edu

## Abstract

Any coherent text contains significant latent information, such as syntactic structure and patterns of language use. This information can be exploited to overcome the inadequacies of keyword-based retrieval and make information retrieval more efficient. In this paper, we demonstrate quantitatively how syntactic information is useful in filtering out irrelevant documents. We also compare two different syntactic labelings – simple Part-of-speech (POS) labeling and Supertag labeling – and show how the richer (more fine-grained) representation of supertags leads to more efficient and effective document filtering. We have implemented a system which exploits syntactic information in a flexible manner to filter documents. The system has been tested on a large collection of news sentences, and achieves an F-score of 89 for filtering out irrelevant sentences. Its performance and modularity makes it a promising postprocessing addition to any Information Retrieval system.

## 1 Enhancing Information Retrieval

The availability of vast amounts of useful textual information in machine-readable form has led to a resurgence of interest in Information Retrieval (IR). There is considerable academic and business interest now in analyzing unrestricted text to extract information of relevance, and in the development of information retrieval and information extraction systems.

Although the ultimate goal in Information Retrieval is to have a system which 'understands' all the text that it processes, and responds to users' queries 'intelligently', there are many open problems in syntactic processing, semantic analysis and discourse processing that need to be solved before tools that are required for 'understanding' texts could be developed.

As a result of the limitation of not being able to automatically 'understand' texts, most IR systems approximate linguistic information by using keywords and features such as proximity and adjacency operators. But the standard problems of synonymy and polysemy adversely affect recall and precision of information retrieval. Users typically have to scan a large number of potentially relevant items to get to the information that they are looking for. (See (Salton and McGill, 1983), (Frakes and Baeza-Yates, 1992) for details on work in information retrieval.)

Clearly, it is inadequate to just retrieve documents which contain keywords of interest. Since any coherent text contains significant latent information, such as syntactic structure and patterns of language use, this can be exploited to make information retrieval more efficient.

In this paper, we demonstrate quantitatively how syntactic information is useful in filtering out irrelevant documents. In contrast to earlier approaches which used syntactic information during information retrieval stage, for example (Croft *et al., 1991*), we use it in a filtering stage, after basic information retrieval. We also compare two different syntactic labelings – simple Part-of-speech (POS) labeling and Supertag labeling and show how the richer (more fine-grained) representation of Supertags leads to more efficient and effective document filtering.

The layout of the paper is as follows. The next section, Section 2, sets the context for the work presented in this paper. Section 3 describes the basic ideas behind POS tagging and supertagging. These

*On leave from the National Centre for Software Technology, Gulmohar Cross Road No. 9, Juhu, Bombay 400 049, India

two syntactic labeling schemes are used to identify sentential patterns of relevance. These patterns can then be applied to filter out irrelevant documents. In Section 4, we describe the identification of the training set, the induction of patterns of relevance and their application for filtering information. In Section 5, we describe our experiments in information filtering using POS tagging and supertagging. The results of our experiments and their implications are discussed in Section 6.

## 2 Using Patterns for Document Filtering

There has been considerable interest in specific aspects of retrieval on news data worldwide as indicated by the Message Understanding Conferences, TIPSTER and TREC Conferences and SIGIR Conferences. Since September 1994, we have been experimenting with retrieving information about *official appointments*, treating it as a sample domain. We are interested in retrieving sentences where the main event is an *appointment* event[1], such as:

> *Telecom Holding Corp., Tampa, Fla., appointed William Mercurio president and chief executive.* *[NYT]*

To detect such sentences, one could simply identify sentences with the string *appoint*. However, this is too promiscuous, since sentences which comment on appointments, sentences which include information about appointments in adjunct clauses, sentences which mention *well-appointed apartments*, etc., are all likely to be retrieved by such simple patterns, as would, for example, the sentence:

> *But, ultimately forced to choose between the commission he appointed and a police chief popular in the community, Riordan chose something of a political middle ground. [NYT]*

It is clear that there are syntactic cues which could be used to filter out some of these irrelevant sentences. But the task of identifying a relevant sentence with respect to official appointments is not simple. While all the following sentences contain the word *appoint*, we may (subjectively) consider only the first of the following sentences relevant:

> *a. The Philadelphia Flyers 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 retrieve documents containing sentences which conform to such patterns, or reject documents that do not conform to the patterns of interest. Hand-crafting such patterns is time-intensive and expensive. The alternative is to develop some semi-automated method of identifying and using patterns of relevance. We are developing such a system, named Glean.

### 2.1 Glean: A Tool for Information Filtering

Glean, a tool for information filtering, is being developed in a research collaboration. Glean seeks to innovatively overcome some of the problems in IR mentioned above, and in particular, uses the notion of 'agents', a combination of augmented textual-patterns and code, to identify and use specific structural features in the text.

Conceptually, Glean consists of two main phases, as illustrated in Figure 1: the pattern training phase and the pattern application phase. In the pattern training phase, we induce a set of patterns of relevance (which we call *augmented-patterns*) for the domain of interest. This is done in a series of steps. We first manually select a training set of sentences relevant to our domain of interest (say, news about appointments) from a corpus of news text and obtain syntactic descriptions of these sentences. From these descriptions, we identify syntactic regularities in the training set, and generalize them to get augmented-patterns of relevance for that domain.

In the application phase, we use these patterns as filters after retrieval. We use a standard IR system to retrieve documents which are potentially relevant. Sentences which refer to the domain of interest are selected from these documents and syntactically analyzed. These tagged sentences are compared against the patterns of relevance collected in the training phase, to determine if the documents containing these sentences should be deemed relevant, or filtered out.

This paper describes experiments concentrating on one specific component of Glean, namely, the syntactic analysis stage. We compare two different syntactic labeling schemes: POS tagging and supertagging (Joshi and Srinivas, 1994).[2]

[1] This is different from the problem of extracting information pertaining to appointments even when this information is not the main focus of the sentence.

[2] For convenience, we will use terms such as *tags*

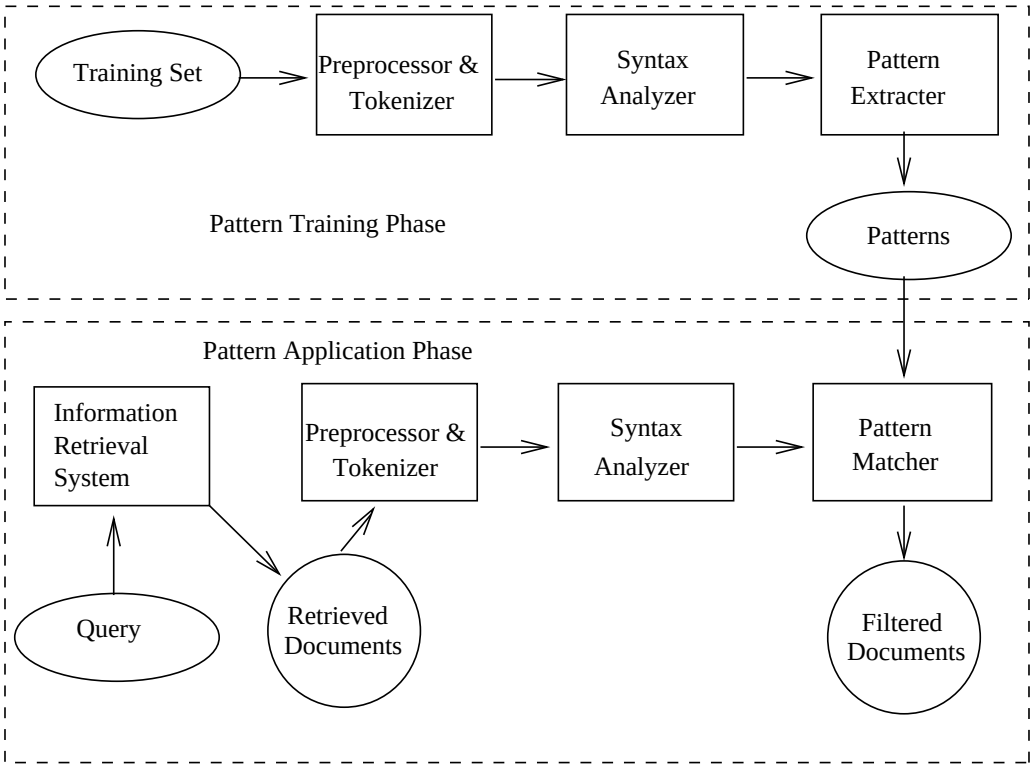


Figure 1: Overview of the Glean filtering scheme

## 3 POS Tagging and Supertagging: Extracting Syntactic Information

### 3.1 POS Tagging

Part-of-speech disambiguation techniques (*taggers*) have been used in several NL applications. These typically use information about the possible POS tags associated with each word, and local constraints on co-occurrence of POS tags. These taggers are local in the sense that they 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 successful.

Tagging helps in disambiguating words, and in associating each word with a unique tag; hence tagging can provide information on ways in which each word is used.

The tagger that we use is a N-gram tagger (similar to (Church, 1988)), and uses the tagset (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, there is no discrimination, 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 found to be about 95% correct on Wall Street Journal data.

Figure 2(a) depicts the POS tags assigned to each word of the phrase *well appointed apartment* and the sentence *She was appointed by the Governor in 1996.*

### 3.2 Supertagging

The other approach to structural analysis is based on Lexicalized Tree Adjoining Grammar (LTAG) and uses the "supertagging" technique (Joshi and Srinivas, 1994) which is described in this section.

#### 3.2.1 Brief Overview of LTAGs

The primitive elements of LTAG formalism are **elementary trees**. Elementary trees are of two types: *initial trees* and *auxiliary trees*. Initial trees are minimal linguistic structures that contain no recursion, such as simple sentences, NPs, PPs etc. Auxiliary trees are recursive structures which represent constituents that are adjuncts to basic

and *tagged sentences* to refer to concepts in both these schemes, and expect them to be distinguished by context.

```
(a) well/RB appointed/VBN apartment/NN
    She/PRP was/VBD appointed/VBN by/IN the/DT Governor/NNP in/IN 1996/CD
(b) well/B_ARBvx appointed/B_Vn apartment/A_NXN
    She/A_NXN was/B_Vvx appointed/A_nx1V by/B_vxPnx the/B_Dnx Governor/A_NXN
       in/B_vxPnx 1996/A_NXN
```

Figure 2: (a) POS tags and (b) Supertags assigned to the phrase *well appointed apartment* and the sentence *She was appointed by the Governor in 1996.*

structure (e.g. relative clauses, sentential adjuncts, adverbials). Each elementary structure is associated with at least one lexical item. Elementary trees are combined by two operations, *substitution* and *adjunction*. A parse in an LTAG yields two structures: *the derived tree* which is the result of combining the elementary trees anchored by the words of the input; *the derivation tree* which provides the history of the parsing process is similar to the dependency structure for an input. For a more formal and detailed description of LTAGs see (Schabes *et al., 1988*). 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*).

### 3.2.2 Supertagging

The elementary trees of LTAG localize dependencies, including long distance dependencies, by requiring that all and only the dependent elements be present within the same tree. As a result of this localization, a lexical item may be (and almost always is) associated with more than one elementary tree. We call these elementary trees *supertags*, since they contain more information (such as subcategorization and agreement information) than standard part-of-speech tags. Hence, each word is associated with more than one supertag. For instance, the word *appointed* would have different supertags corresponding to its use as a transitive verb, in a relativized form, in a passive form etc. In the process of parsing, each word is associated with just one supertag (assuming there is no global ambiguity), and the supertags of all the words in a sentence are combined by substitution and adjunction.

Instead of relying on parsing to disambiguate the supertags, we can use local statistical information (as in standard part-of-speech disambiguation) in the form of N-gram models based on the distribution of supertags in a LTAG parsed corpus. We use a trigram model (Church, 1988) to disambiguate the supertags so as to assign one supertag to each word – a process termed supertagging. The trigram model is trained on a corpus of sentences where each word is annotated with the supertag that would be associated with the word in the correct parse of the sentence. The trigram model of supertagging is very efficient (linear time) and robust. The performance of the supertagger trained on 180,000 words of Wall Street Journal text and tested on 20,000 words of Wall Street Journal text is summarized in Table 1.

| Number of words | Number of words correctly supertagged | % correct |
|---|---|---|
| 22,000 | 19,668 | 89.4% |

Table 1: The performance of the supertagger on the Wall Street Journal Corpus

There are 300 supertags used by the supertagger. However, supertagging does not code for morphological information about words. (In LTAG, this distinction is maintained using features.) Thus we cannot distinguish between different numbers or tenses of a word. However, supertags distinguish between the two *to*'s in, for example, *I have to go to New York*.

Figure 2(b) depicts the supertags assigned to each word of the phrase *well appointed apartment* and the sentence *She was appointed by the Governor in 1996*. We can interpret the process of supertagging as disambiguating words, and associating each word with a unique supertag. We use this discrimination between supertags to provide us information about how different supertags of each word are used, and to distinguish between relevant and irrelevant uses of the word, wherever possible.

## 4 Inducing and Using Patterns

In this section, we describe how augmented-patterns for the domain of appointments (of people to posts) are extracted from a corpus of news text, and applied to filter out irrelevant information. In this description, all training and testing is done with sentences as the units of information. However, the methodology applies to larger units of text, where

the relevant sentences are extracted using simple keyword matching.

### 4.1 Identifying the Training Set

A large textual corpus is first segmented into sentences. All sentences related to the word(s) of interest (in this case, *appoint* or a morphological variant) are extracted using some simple tool. These sentences are examined to see which of them are relevant to the domain of interest. Some decisions about the relevance of sentences may not be very easy. These decisions determine the scope of the filtering that is achieved using this system. 'Augmented-patterns' are then induced from the relevant sentences.

### 4.2 Inducing Patterns from Training Data

The training data is first tokenized into sentences and the relevant sentences are processed to identify phrases that denote names of people, names of places or designations. These phrases are converted effectively to one lexical item. The chunked relevant sentences are then tagged (with POS tags or supertags) and the tags associated with the words in the sentences are used to create noun-groups (involving prenominal modifiers) and verb-groups (involving auxiliaries, modals, verbs and adverbs). At this stage, we have an abstract view of the structure of each sentence.

We look at a small window around the word(s) of our interest (in this case, one chunk on either side of the word *appoint* or its morphological variant), skipping 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 words. The patterns for all sentences are then sorted, and duplicates removed. The resulting patterns can be used for filtering. Once a set of patterns is identified for a particular class of query, it can be saved in a library for later use.

Generalization brings out the syntactic commonality between sentences, and permits an economical description of most members of a set of sentences. We expect that a few patterns will suffice to describe the majority of sentences, while several patterns may be necessary to describe the remaining sentences. We could also sort patterns by the number of sentences that each of them describes, and ignore all patterns below some reasonable threshold. Note that generalization could increase recall while reducing precision, while thresholding decreases recall.

### 4.3 Pattern Application

The task in the pattern application phase is to employ the patterns induced in the pattern training phase to classify new sentences into relevant and irrelevant ones. The new sentences could be part of documents retrieved from news-wire texts, from the World Wide Web (WWW), etc. The relevance of a document is decided on the basis of the relevance of the sentences contained in it.

In this phase, the sentences of each document are subjected to similar stages of processing as were the training sentences. Each sentence in each document is chunked based on simple named-entity recognition and then labeled with syntactic information (POS tags or supertags). The syntactic labels for the words in each sentence are used to identify noun and verb chunks. At this stage, the sentence is ready to be matched against the patterns obtained from the training phase. A sentence is deemed to be relevant if it matches one or more of these patterns. Since the pattern matching is based on simple regular expressions specified over words and syntactic labels, it is extremely fast and robust. A document is deemed relevant if it contains at least one relevant sentence.

## 5 The Experiment: POS Tagging *vs* Supertagging

This section describes an experiment where we use techniques discussed in the previous sections to retrieve relevant documents about appointments. The objective here is to quantitatively measure the performance improvement achieved by richer syntactic information for filtering out irrelevant documents.

### 5.1 The experiment: Training Phase

The text corpus constituted of approximately 52 MB of New York Times (NYT) data comprising of the August 1995 wire service output.[3] 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`. We plan to handle other equivalent words and phrases later.

The 494 sentences containing *appoint** were examined manually, and a subset of them (56 sentences) which were actually relevant to appointments being announced were identified. This constituted the training corpus. This included sentences about the appointments of groups of people such as panel,

[3] NYT text was chosen since it was felt that it would have more variety than, for instance, the Wall Street Journal.

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 3: Sample patterns involving POS tags and Supertags

| Domain | Total Sents | Relevant Sents | Classified as relevant | | | Classified as irrelevant | | |
|---|---|---|---|---|---|---|---|---|
| | | | Total | Correct | Incorrect | Total | Correct | Incorrect |
| NYT July95 (Supertags) | 529 | 95 | 168 | 77 | 91 | 361 | 343 | 18 |
| NYT July95 (Part of Speech) | 529 | 95 | 73 | 42 | 31 | 456 | 392 | 64 |
| Base Case (all appoint*) | 529 | 95 | 529 | 95 | 434 | 0 | 0 | 0 |

Table 2: Classification of appoint* sentences

commissions etc. We rejected sentences where *appointed* was used as an adjectival or in a relative clause, and where a variant of *appoint* was used in the noun sense (eg. *appointee/appointment*). Most of the 56 acceptable sentences came from sentences with the word *appointed*; a few came from *appoint* and *appointment*.

#### 5.1.1 Patterns obtained using POS tags

The 56 relevant sentences were preprocessed to normalize punctuation. Named-entities were identified and grouped into single tokens. These sentences were then POS tagged and the tags were used to chunk verb groups and noun phrases. The chunked sentences were then processed to obtain 20 generalized patterns.

#### 5.1.2 Patterns obtained using supertags

In a similar manner, the training sentences were processed to obtain 21 distinct patterns using supertags instead of POS tags.

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 3. Using such patterns, sentences from a variety of domains were categorized into relevant and irrelevant sentences.

### 5.2 The experiment: Testing Phase

From the July 1995 NYT wire service text, all sentences (a total of 529 sentences) containing the word *appoint* or its variant were extracted using `grep`. This constituted the base case, where sentences are retrieved using a simple retrieval mechanism (such as `grep`), with no filtering applied.

These 529 sentences also constituted the test set. 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).

The patterns obtained from the training phase were applied to the 529 sentences. As before, these sentences were processed in a manner similar to the training data. All sentences which matched the augmented-patterns were deemed relevant by the program. The relevant and irrelevant sets for each method (POS tags and Supertags) were compared to the standard relevant and irrelevant sets.

The result of these experiments are summarized in Table 2, for the supertagging 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 standard set.

We have computed the recall, precision and F-score measure for the three methods shown in Table 2. F-score (Hobbs et al, 1992) is defined as

| Domain | Recall | Precision | F-score ($\beta = 1.0$) | F-score ($\beta = 0.5$) | F-score ($\beta = 1.5$) |
|---|---|---|---|---|---|
| NYT July95 (Supertagging) | (77/95) = 81% | (77/168) = 49% | 61 | 72 | 56 |
| NYT July95 (Part of Speech) | (42/95) = 44% | (42/73) = 58% | 50 | 46 | 53 |
| Base Case (all appoint*) | (95/95) = 100% | (95/529)= 18% | 31 | 52 | 24 |

Table 3: Precision and Recall of different filters, for **relevant sentences**

| Domain | Recall | Precision | F-score ($\beta = 1.0$) | F-score ($\beta = 0.5$) | F-score ($\beta = 1.5$) |
|---|---|---|---|---|---|
| NYT July95 (Supertagging) | (348/434) = 80% | (348/373) = 94% | 86 | 82 | 89 |
| NYT July95 uniq (Part of Speech) | (380/434) = 88% | (380/452) = 84% | 88 | 89 | 87 |
| Base Case (all appoint*) | (0/434) = 0% | (0/0) – | – | – | – |

Table 4: Precision and Recall of different filters, for **irrelevant sentences**

follows:

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

F-score provides a method of combining recall and precision. It provides a parameter $\beta$ that can be set to measure the performance of a system depending on the relative importance of recall to precision.

Table 3 shows the recall, precision for relevant sentences, for each of the three methods shown in Table 2. It also shows F-scores for the three cases: precision is as important as recall ($\beta$=1), precision is less important than recall ($\beta$=0.5) and precision is more important than recall ($\beta$=1.5). Similar results for irrelevant sentences are summarized in Table 4.

# 6 Discussion

In this section, we discuss the novel aspects of our approach, and the relative merits and demerits of the two tagging schemes used. We also provide a detailed discussion of the performance of POS tagging and supertagging.

## 6.1 Syntactic Filtering works!

Tables 2 to 4 show clearly that syntactic information (supertagging, in particular) can be used to reduce the amount of information to be perused by upwards of 70%. The loss in recall is about 20%.

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, syntactic information can be exploited in the form of a filter in any IR system. In fact, the Glean system uses a conventional IR system named Khoj to do the first level retrieval. In another experiment, we used our approach to filter documents retrieved from the WWW using a popular search engine.

## 6.2 POS Tagging & Supertagging: Merits and Demerits

**Granularity of description:**

The tags (POS and supertag labels) employed in our approach serve to categorize different syntactic contexts of a word. In general, a richer set of tags leads to a better discrimination. The supertags provide an optimal descriptive granularity, which is neither at the level of complete parses (which may often be hard or impossible to obtain), nor at the simpler level of parts of speech. Since supertags are richer in representation when compared to POS tags, it is expected that supertags would provide better discrimination.

**Sources of Error:**

Both POS tagging and supertagging use statistical methods, and their accuracy and completeness depend on the material that was used to train them. The genre of the training material also introduces lexical 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 and pattern application phases can cause erroneous patterns to be created and lead to wrong categorization of documents respectively.

**Processing Speed:**

POS tagging is simpler, and is almost twice as fast as supertagging. Filtering using supertagging takes 1.33 seconds per sentence, which corresponds to a speed of about 22 words of the test set per second. Filtering using POS tags takes 0.68 second per sentence, corresponding to 44 words per second. These figures are on a Sun Ultra E4000 with two 167Mhz Ultrasparcs, with 320MB memory. The system is implemented as a series of programs in interpreted PERL.

### 6.3 Performance and Error Analysis

Filtering with either POS tagging or supertagging is better at reducing information overload, than retrieval without filtering. The filtering mechanism used is much better at weeding out irrelevant material than in identifying relevant material. This was also noticed in our experiments with data obtained off WWW.

On error analysis, we discovered that one of the reasons for the low precision of supertagging for retrieving relevant sentences was due to the fact that auxiliary verbs and infinitives were not being distinguished by the supertags.[4] As a result, verb groups, for instance, are over-generalized. Retaining function words without generalizing them should lead to better performance using supertags.

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. We believe that increasing the window size to include two chunks on the left would improve the performance of the system further.

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, but it is not simple to filter such sentences out, without additional information.

We are addressing some of these problems in order to reduce the overall error rate. We are also testing the system with other domains, and with multiword concepts (such as *take charge*), with very encouraging results. We have plans to test cascaded filters, using patterns got from both relevant and irrelevant training sets. We hope to present the results of these experiments in the final version of this paper.

A prototype of the complete Glean system is expected to be operational within a short period.

[4] The POS tagset retains this distinction.

## References


Eric Brill. Some Advances in Transformation-Based Part of Speech Tagging In *Proceedings of AAAI, 1994.*

W. Bruce Croft, Howard R. Turtle, and David D. Lewis. 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)*, Chicago, USA, October 1991, pages 32-45.

Kenneth Ward Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas.

Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. XTAG System - A Wide Coverage Grammar for English. In *Proceedings of the* $17^{th}$ *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan, August 1994.

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

Jerry Hobbs, Doug Appelt, John Bear, David Israel and W. Mabry Tyson (1992) *FASTUS: A system for extracting information from natural language text*, SRI Technical Report No. 519

Aravind K. Joshi and B. Srinivas. 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, August 1994.

Mitchell M. Marcus and Beatrice Santorini and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. In *Computational Linguistics*, Vol. 19 No.2, pages 313–330, June 1993.

Gerald Salton and Michael J. McGill. *Introduction to modern information retrieval.* McGraw-Hill, New York, 1983.

Yves Schabes, Anne Abeillé, and Aravind K. Joshi. 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, August 1988.