Institute for Research in Cognitive Science

Gleaning Information from the Web: Using Syntax to Filter Out Irrelevant Information

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--31

Gleaning information from the Web: Using Syntax to Filter out Irrelevant Information

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

In this paper, we describe a system called Glean, which is predicated on the idea that any 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 an approach to information retrieval that makes use of syntactic information obtained using a tool called a supertagger. A supertagger is used on a corpus of training material to semi-automatically induce patterns that we call augmented-patterns. We show how these augmented patterns may be used along with a standard Web search engine or an IR system to retrieve information, and to identify relevant information and filter out irrelevant items. We describe an experiment in the domain of official appointments, where such patterns are shown to reduce the number of potentially irrelevant documents by upwards of 80%.

Introduction: IR and WWW

Vast amounts of textual information are now available in machine-readable form, and a significant proportion of this is available over the World Wide Web (WWW). However, any particular user would typically be interested only in a fraction of the information available. The goal addressed by Information Retrieval (IR) systems and services in general and by search engines on the Web in particular is to retrieve all and only the information that is relevant to the query posed by a user.

Early information retrieval systems treated stored text as arbitrary streams of characters. Retrieval was usually based on exact word matching, and it did not matter if the stored text was in English, Hindi, Spanish, etc. Later IR systems treated text as a collection of words, and hence several new features were made possible, including the use of term expansion, morphological analysis, and phrase-indexing. However, all these methods have their limitations, and there have been several attempts to go beyond these methods. See (Salton & McGill 1983), (Frakes & Baeza-Yates 1992) for further details on work in information retrieval.

With the recent growth in activity on the Web, much more information has become accessible online. Several search engines have been developed to handle this explosion of information. These search engines typically explore hyperlinks on the Web, and index information that they encounter. All the information that they index thus becomes available to users' searches. As with most IR systems, these search engines use inverted indexes to ensure speed of retrieval, and the user is thus able to get pointers to potentially relevant information very fast. However, these systems usually offer only keyword-based searches. Some offer boolean searches, and features such as proximity and adjacency operators. Since the retrieval engines are geared to maximizing recall, there is little or no attempt to intelligently filter the information spewed out at the user. The user has to scan a large number of potentially relevant items to get to the information that she is actually looking for. Thus, even among experienced users of IR systems, there is a high degree of frustration experienced in searching for information on the Web.

Many of the (non-image) documents available on the Web are natural language (NL) texts. Since they are available in machine-readable form, there is a lot of scope for trying out different NL techniques on these texts. However, there has not been much work in applying these techniques to tasks such as information retrieval. In this paper, we describe an application which uses NL techniques to enhance retrieval. The system we describe is predicated on the fact that any coherent text contains significant latent information, such as syntactic structure and patterns of language

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

use, which can be used to reduce an IR or Web users' information load.

Task Definition

There has been considerable interest in specific aspects of retrieval on news data in the Research Centers we are associated with. 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 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*. But this is not enough. Sentences which comment on appointments, sentences which talk of *well-appointed apartments*, sentences which include information about appointments in a relative clause 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]*

The task of identifying a relevant sentence with respect to official appointments is not simple. While the following sentences contain the word *appoint*, we may (subjectively) consider only the last of the following sentences relevant:

> *The US trustee shall appoint any such committee.*
> *The President appoints judges of the Supreme Court.*
> *The Philadelphia Flyers will meet today to appoint a new manager ....*

It is clear that there are syntactic clues which may be used to filter out some of these irrelevant sentences. If we can use such information, we can identify syntactic patterns of interest, and retrieve documents containing sentences which conform to such patterns, or reject documents that do not conform to the patterns of interest. However, the task of identifying patterns can be very difficult. Hand-crafting such patterns is time-intensive and expensive. The alternative is to develop some semi-automated method of identifying patterns of relevance or irrelevance.

Our task, thus, is to develop a system which uses syntactic information inherent in text to improve the efficiency of retrieval (given any basic IR tool) by automatically or semi-automatically identifying patterns of relevance.

Methodology

Our approach to this problem consists of two phases, a pattern training phase and a pattern application phase, as illustrated in Figure 1. In the pattern training phase, we manually select a set of sentences relevant to our domain of interest (say, news about appointments) from a corpus of news text, and call it our training set. We use a tool called a *supertagger* (Joshi & Srinivas 1994) to obtain a syntactic description of these sentences. The supertagger gives us a view of the syntactic structure of a sentence at an optimal level of granularity, which is neither at the level of complete parses (which may often be hard or impossible to obtain), nor at the overly simplified level of parts of speech. From the supertagged sentences, we identify syntactic regularities in the training set, which gives us a set of patterns (which we call *augmented-patterns*) of (ir)relevance for the domain of interest.

These patterns can easily be used as filters in retrieval. We first use an IR system or a search engine to retrieve documents which are potentially relevant. Sentences which refer to the domain of interest are selected from these documents and supertagged in the syntactic analysis phase. These supertagged sentences are compared against the patterns of (ir)relevance to determine if the documents containing these sentences should be deemed relevant, or filtered out. Figure 1 provides an overview of the whole process.

Such a tool for information filtering, named Glean, is being developed in a research collaboration between the National Centre for Software Technology (NCST), Bombay, the Institute for Research in Cognitive Science (IRCS) and the Center for the Advanced Study of India (CASI), University of Pennsylvania. Glean seeks to innovatively overcome some of the problems in IR mentioned above, and is predicated on the idea that any coherent text contains significant latent information, such as syntactic structure and patterns of language use, which can be used to enhance retrieval efficiency. In particular, Glean uses the notion of 'agents', a combination of augmented textual-patterns and code, to identify specific structural features in the text.

This paper describes one aspect of Glean, which permits us to filter upwards of 80% of irrelevant documents in experiments that we have conducted on a prototype. The layout of the paper is as follows. In the next section, we describe the basic ideas behind the syntactic formalism we use. In the section on extracting patterns, we describe our method of identifying the training set and the automatic induction of patterns of relevance from the training set. We then describe our experiments and performance results of

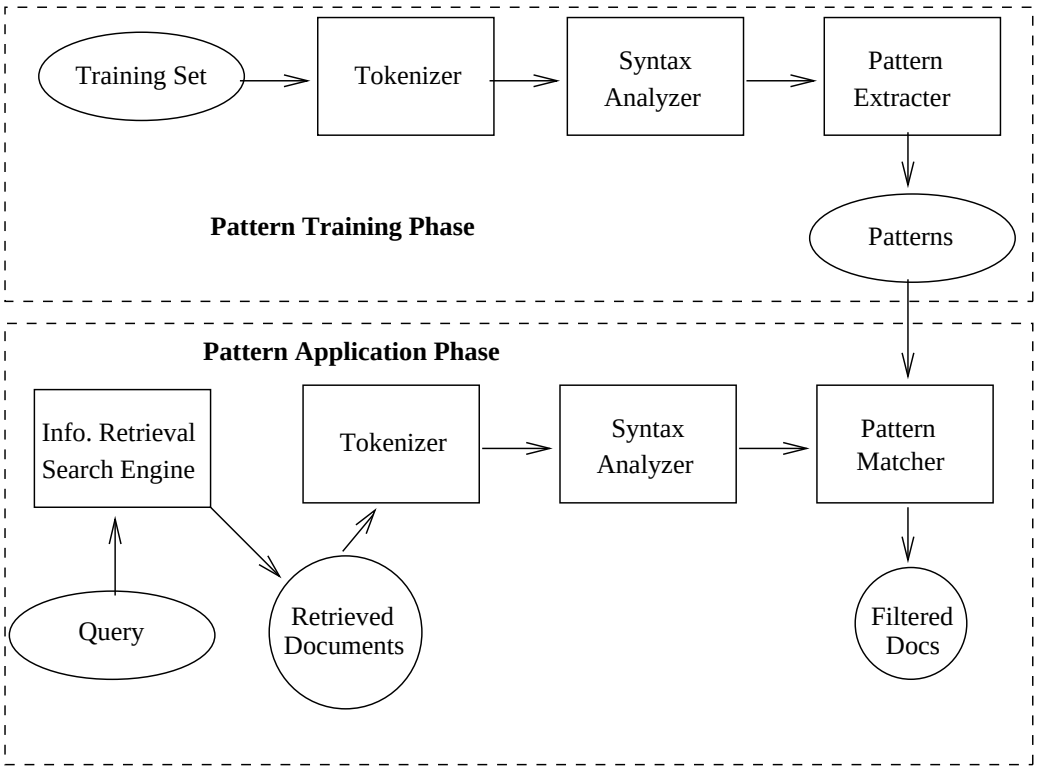

Figure 1: Overview of the Glean filtering scheme
For efficiency, the operations following the tokenizer are applied only to sentences which contain the words of interest, and not on the entire document.

using this methodology to filter information retrieved from the World Wide Web. In the last section, we discuss some of the issues we have encountered in the process of developing this system.

Supertagging: Extracting Syntactic Information

Our approach to filtering uses a rich syntactic representation based on Lexicalized Tree Adjoining Grammar (LTAG) and uses the "supertagging" technique described in (Joshi & Srinivas 1994). These are briefly described in this section.

Brief Overview of LTAGs

The primitive elements of the 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 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. The derivation tree is similar to the dependency structure for an input. For a more formal and detailed description of LTAGs see (Schabes, Abeillé, & Joshi 1988). A wide-coverage English grammar (named the XTAG 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).

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 in-

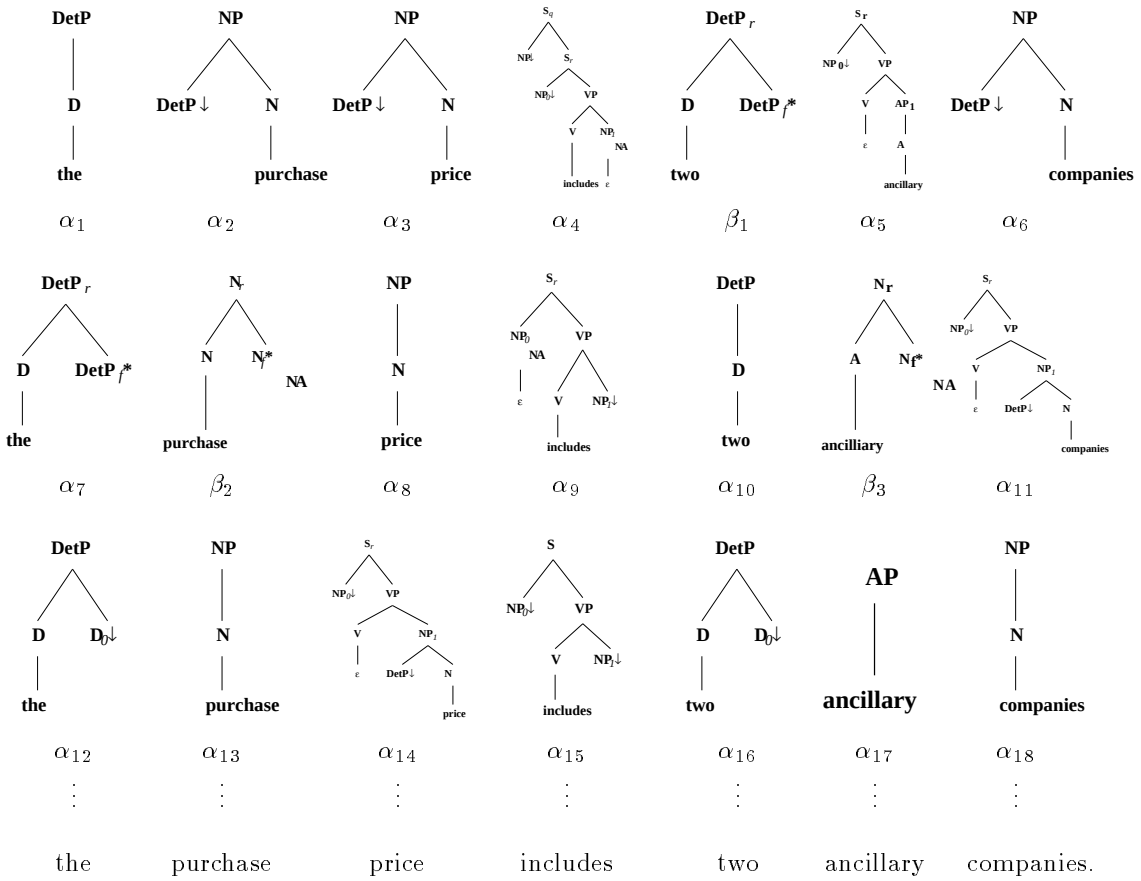

Figure 2: A sample set of supertags with each word of the sentence *the purchase price includes two ancillary companies*. For example, the three supertags shown for the word *includes* represent the supertag for object extraction (α_4), the supertag for imperative construction (α_9) and the supertag for the canonical indicative construction (α_{15}), in that order.

formation) than standard part-of-speech tags. Hence, each word is associated with more than one supertag. A word like *appointed* would have different supertags corresponding to its use as a transitive verb, in a relativized form, in a passive form etc. Supertags for recursive and non-recursive constructs are labeled with βs and αs respectively. Figure 2 depicts the set of elementary trees assigned to each word of the sentence *the purchase price includes two ancillary companies*, where each α_n and β_n denotes a different supertag. 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 an LTAG parsed corpus. We use a trigram model (Church 1988) to disambiguate the supertags so as to assign the most appropriate supertag to each word, given the context of the sentence. This process is termed *supertagging*. The trigram model has been trained on a corpus of Wall Street Journal sentences that were parsed using the wide coverage XTAG grammar. This model of supertagging is very efficient (linear time) and robust. The trigram model of supertagging achieves an accuracy of 92.2% on Wall Street Journal data. Performance details of this and other models of supertagging are presented in (Joshi & Srinivas 1994; Srinivas 1997).

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 syntactic variants of each word are used, and to distinguish between relevant and irrelevant uses of the word, wherever possible. This would not be possible if we used only part-of-speech (POS) tags: the discrimination we get with POS tags is poorer, and is not as

effective as supertags. A comparison of POS against supertags in terms of their ability to discriminate between relevant and irrelevant documents is presented in (Chandrasekar & Srinivas 1996).

Extracting Relevant Patterns from a News Corpus

In this section, we describe how we extract augmented-patterns for the domain of appointments (of people to posts) from a corpus of news text.

Identifying the Training Set

The training corpus constituted of a corpus of approximately 52 MB of New York Times (NYT) text data comprising of the August 1995 output.[1] 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`. Other words and phrases such as *became, has become, took charge, took over, given charge, has been elected, has been named, has named, promoted to* and *the new ...* are also used in news text to indicate appointments. However, some of these words often occur in other completely different contexts. These words were not analyzed in this experiment, but we intend to handle them in a later experiment.

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. We decided that we would deem relevant only those sentences where the appointee is named (or is referred to by a pronoun), and where the designation, or the name of the appointer is mentioned. This included sentences about the appointments of groups of people such as panels, commissions etc., which we decided to accept. We rejected sentences where *appointed* was used as an adjective or in a relative clause, and where a variant of *appoint* was used as a noun (eg. *appointee/appointment*). Most of the 56 acceptable sentences contained the word *appointed*; one contained *appoint*. 'Augmented-patterns' were then automatically induced from these relevant sentences, as described in the next section.

There is some manual one-time effort involved in selecting 'relevant' sentences. We are exploring methods to reduce the effort involved, and providing tools to quickly partition sample sentences into relevant and irrelevant sets. Note that once such a partitioning is done, the extraction of patterns is completely automatic.

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

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

Figure 3: A Sample Pattern for *appointed*
Key: \S* refers to any word; 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.

Extracting Patterns from Training Data

The relevant sentences are processed to identify phrases that denote names, names of places or designations. These phrases are converted effectively to one lexical item. The chunked relevant sentences are then supertagged and the supertags 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 supertagged sentences with noun-group and verb-group chunks identified, giving us an abstract view of the structure of each sentence.

We look at a small window around the word(s) of interest to us (in this case, one chunk on either side of the word *appoint* or its morphological variant), skipping punctuation marks. The word and supertag groups in this window are then generalized to a small set of augmented patterns, where each augmented pattern is a description involving supertags, punctuation symbols and some words. The patterns for all sentences are then sorted, and duplicates removed.

A sample pattern, which matches 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 is shown in Figure 3.

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 limit the number of patterns by sorting them according to the number of sentences that each of them describes, and ignoring patterns below some reasonable threshold. Note that generalization (as well as underspecification of patterns) could increase recall while reducing precision, while thresholding decreases recall.

Once a set of patterns is identified for a particular class of query, it can be saved in a library for later use. In this model, we can save augmented-patterns along with other information (to be detailed later) as

System	Total Docs	Relevant Docs	Classified as relevant			Classified as irrelevant		
			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	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%

Table 2: Precision and Recall of Glean for retrieving relevant documents.

an 'agent'. We discuss ideas of using such a library of predefined agents, as well extending this approach to *ad hoc* queries, in the discussion section.

A total of 56 selected relevant sentences were processed and 20 distinct augmented-patterns were obtained. Using these patterns, sentences can be categorized into relevant and irrelevant sentences.

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 the World Wide Web, from news-wire texts 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 supertagged using the supertagger. The supertags 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 at least one of these patterns. Since the pattern matching is based on simple regular expressions specified over words and supertags, it is extremely fast and robust. A document is deemed relevant if it contains at least one relevant sentence.

In the next section, we describe an experiment of classifying documents retrieved from the Web into relevant and irrelevant categories, given a search query about appointments.

Gleaning Information from the Web

This section describes an experiment where we use techniques discussed in the previous sections on documents about appointments retrieved from the World Wide Web. The objective here is to quantitatively measure the performance improvement in terms of filtering out irrelevant documents. Note that these results are applicable to any context where documents are available in a machine readable form.

Design of the Experiment

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 (see `http://altavista.digital.com/`) to retrieve the URLs matching a search expression, using the WWW::Search and WWW::Search::AltaVista Perl modules distributed from ISI (`http://www.isi.edu/lsam/tools/WWW_SEARCH/`). The document corresponding to each matching URL

System	Recall	Precision
Plain Web Search (Without Glean)	–	–
With Glean filtering	(50/56) = 89.3%	(50/55) = 90.9%

Table 3: Precision and Recall of Glean for filtering out irrelevant documents

was downloaded using a simple socket application program, with timeouts to account for busy networks or failed connections.

There are several hundred documents that mention events about appointments on the Web. To restrict the test set retrieved to a manageable number, we searched the Web using the Alta Vista search expression shown below, where we require that the document retrieved contains the words/expressions *Fortune 500*, *company* and *CEO*, as well as a form of the word *appoint*:

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

Retrieval and Filtering Performance

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 to 3 show the performance of the system in terms of Recall and Precision for relevant and irrelevant documents. The first row shows the performance of the Web search engine by itself, while the second row shows the performance of Glean's filtering on the output of the Web search engine. It is interesting to note that this method performs better in filtering out irrelevant documents than in identifying relevant documents. 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.

We also examined the false positives and false negatives and found that four of the six false positives were due to sentences dealing with generic ideas about appointments. For example, a sentence such as *The equally competent women in business are not being appointed to boards at an equivalent rate* was considered irrelevant (to the concept of appointment announcements) in the training, and during hand-classification of the test data (gold standard), while the program accepted such sentences as relevant.

Discussion

There are special problems and possibilities with retrieval on the Web. Web retrieval may suffer from several problems such as failed, aborted or incomplete document retrievals, and inconsistencies between indexes and documents because documents changed or moved after indexing. IR systems for the Web have to keep these problems in mind, and include techniques such as timeouts and error-checking methods to get around such problems. On the positive side, the amount of online material available should be positive inducement for trying out example-based or statistical techniques, especially in new domains, languages and scripts.

One major improvement we are considering is to move the filtering to the host supplying the document ('supply-side checking'!). That is, instead of obtaining a huge document and then checking to see if it is likely to be relevant, it may be simpler to send a program (coded in a language such as Java (Arnold & Gosling 1996)) to the site from which the document is being retrieved, to check for its relevance to the query. We have described a mechanism to implement such server-side checking in (Chandrasekar & Sarkar 1997). Advanced NL processing is resource-intensive, and mechanisms such as these may be required to make applications similar to the one presented in this paper practical.

The rest of the discussion here is applicable to IR systems and to Web search engines. As is seen from Tables 1 to 3, 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. The number of documents in this experiment is small, but we intend to collect similar figures for other experiments involving larger numbers of documents.

As briefly noted above, the performance of this mechanism is much better for filtering out irrelevant

material than it is for identifying relevant material. This was also noticed in our experiments with New York Times data, as described in (Chandrasekar & Srinivas 1996). There are many possible reasons for this, including extra-syntactic phenomena which we have not accounted for, inadequate size of window used in creating patterns, unusual patterns of word use, etc.

Errors in supertagging could also lead to wrong categorization of documents, as could errors in spelling. However, the 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.

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 test areas, with very encouraging results.

We have considered the use of this system for applications involving selective dissemination of information (SDI). SDI is used to route selected items from a flow of documents to users with pre-specified profiles of interest. Mechanisms such as the ones we have discussed may be used to implement filters for SDI applications.

The other mode of operation is to define libraries of agents for pre-defined concepts. Of course, definitions of a concept may differ from user to user, and customizability of agents would be a definite requirement. We have considered the provision of tools (including the use of relevance feedback methods) for users to create their own (local) definitions of patterns for concepts of interest to them.

A Web interface to this system is available and is being enhanced. A version of the complete Glean is under development, and a prototype is expected to be operational within a short period.

Acknowledgments

It is a pleasure to thank the people behind tools such as Alta Vista as well as the Perl community at large, for creating such exciting tools and systems. We thank the Linguistic Data Consortium, University of Pennsylvania, for providing us access to New York Times and Wall Street Journal data for our experiments. Two anonymous referees provided us with insightful comments which have improved the quality of this paper. RC also thanks NCST, IRCS and CASI for the opportunity to work in this area. This work is partially supported by NSF grant NSF-STC SBR 8920230, ARPA grant N00014-94 and ARO grant DAAH04-94-G0426.

References

Arnold, K., and Gosling, J. 1996. *The Java Programming Language.* Addison-Wesley.

Chandrasekar, R., and Sarkar, A. 1997. Searching the web with server-side filtering of irrelevant information. Submitted for publication.

Chandrasekar, R., and Srinivas, B. 1996. Using syntactic information in document filtering: A comparative study of part-of-speech tagging and supertagging. Technical Report IRCS 96–29, University of Pennsylvania.

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

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).*

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

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).*

Salton, G., and McGill, M. J. 1983. *Introduction to modern information retrieval.* New York: McGraw-Hill.

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).*

Srinivas, B. 1997. Performance evaluation of supertagging for partial parsing. Submitted for publication.