Using Information Extraction to Improve Document Retrieval

John Bear, David Israel, Jeff Petit, and David Martin
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

January 9, 1998

1 Abstract

We describe an approach to applying a particular kind of Natural Language Processing (NLP) system to the TREC routing task in Information Retrieval (IR). Rather than attempting to use NLP techniques in indexing documents in a corpus, we adapted an information extraction (IE) system to act as a post-filter on the output of an IR system. The IE system was configured to score each of the top 2000 documents as determined by an IR system and on the basis of that score to rerank those 2000 documents. One aim was to improve precision on routing tasks. Another was to make it easier to write IE grammars for multiple topics.

2 Introduction

Researchers have pursued a variety of approaches to integrating natural language processing with document retrieval systems. The central idea in the literature is that some, perhaps shallow variant of the kind of syntactic and semantic analysis performed by general-purpose natural language processing systems can provide information useful for improving the indexing, and thus the retrieval, of documents. [SparckJones1992, Lewis1992, Hearst1992] The work in this area has seen some success, but significant performance improvements have yet to be demonstrated. [Faloutsos and Oard1996] We have pursued a different hypothesis, that an information extraction (IE) system can be pipelined with a document retrieval system in such a way as to improve performance on routing tasks.

The goal of a document retrieval system, as embodied in the routing task of TRECs [Harman1996], is to consult a large database of documents and return a subset of documents ordered by decreasing likelihood of being relevant to a particular topic. In the TREC6 routing task, a document retrieval system returns the 1000 documents it judges most likely to be relevant to a query out of a database of roughly one million documents. A system performs well if a high proportion of the articles returned, high relative to the ratio of relevant articles in the corpus, are relevant to the topic, and if the relevant articles are ranked earlier in its ordering than the irrelevant ones.

The goal of an IE system, as embodied in the scenario template task of MUCs [Grishman and Sundheim1995], is to consult a corpus of documents, usually smaller than those involved in document retrieval tasks, and extract prespecified items of information. (In MUC-6, for instance, the test corpus consisted of 100 newspaper articles.) Such a task might be defined, for instance, by specifying a template schema instances of which are to be filled automatically on the basis of a linguistic analysis of the texts in the corpus.

A system performs well to the extent that the material it extracts captures the relevant information in the documents. Note that if one were to apply the distinction between ad hoc and routing queries to the MUC scenario template task, it would be classified as a routing query; the task is known in advance and it is assumed that IE systems will have been especially tuned to the task.

Our approach to using NLP techniques for IR was to adapt an IE system, SRI's FASTUS system [Appelt et al.1995], to enable us to write small grammars for many topics and to use those grammars as queries to be run against the top 2000 documents for those topics, as determined by an IR system—in our case GE's version of SMART. In the end we were able to produce grammars for 23 topics.

As noted above, the output of an IR system for a given topic on the routing task is a list of the documents ordered by decreasing likelihood of relevance. Our adaptation of FASTUS involved having each grammar rule that matched some segment of an article assign a score to that segment. We then summed the scores to get a total for the article.

For each of the 23 topics for which we had written grammars, we had FASTUS process each article in GE's 2000 top articles for that topic and rank them by score. The highest-scoring articles were ranked first, and importantly, in the case of ties we used GE's order. For the other 24 topics, we submitted GE's top 1000 articles.

In the following sections, we will describe, first, the FASTUS Information Extraction System and then the main features of the adaptation of FASTUS to the current IR effort. We end with a brief summary and some tentative conclusions.

3 Background

3.1 FASTUS

SRI's FASTUS system is based on a cascade of finite-state transducers that compute the transformation of text from sequences of characters to domain templates. Each transducer (or "phase") in FASTUS takes the output of the previous phase and maps it into structures that constitute the input to the next phase, or in the case of the final phase, that contain the domain template information that is the output of the extraction process. A typical FASTUS application might employ the following sequence of phases, although the number of transducers in different applications may vary.

1. *Tokenizer.* This phase accepts a stream of characters as input, and transforms it into a sequence of tokens.

2. *Multiword Analyzer.* This phase is generated automatically by the lexicon to recognize token sequences (like "because of") that are combined to form single lexical items.

3. *Name Recognizer.* This phase recognizes word sequences that can be unambiguously identified as names from their internal structure (like "ABC Corp." and "John Smith").

4. *Parser.* This phase constructs basic syntactic constituents of the language, consisting only of those that can be nearly unambiguously constructed from the input using finite-state rules (i.e., noun groups, verb groups, and particles).

5. *Combiner.* This phase produces larger constituents from the output of the parser when it can be done fairly reliably on the basis of local information. Examples are possessives, appositives, "of" prepositional phrases ("John Smith, 56, president of IBM's subsidiary"), coordination of same-type entities, and locative and temporal prepositional phrases.

6. *Domain or Clause-Level Phase.* The final phase recognizes the particular combinations of subjects, verbs, objects, prepositional phrases, and adjuncts that are necessary for correctly filling the templates for a given IE task.

The rules for each phase are specified in SRI's pattern language, called FASTSPEC. The rules take the form of regular productions that are translated automatically into finite-state machines by an optimizing compiler.

3.2 Adapting FASTUS for IR

The design of FASTUS was motivated by the design of MUC style scenario template tasks: a fairly narrowly defined prespecified information requirement was posed and up to a month's effort was devoted to writing application grammars to answer that requirement. In writing the grammars for MUCs 4 and 5 very little thought was given to the various ways in which greater generality of application might be built into the system. This changed with MUC6; we began work aimed toward making it easier to apply FASTUS to new topics.

For MUC6 we developed an approach that involved writing general, application-independent, clause-level patterns for which we would then write application-specific instances; typically, these instances were tied to the argument structure of the topic-relevant verbs. (For MUC6, where the task involved recognizing high-level management changes, these verbs included "resign", "succeed", "replace".) Given that we already had good reasons for extending this separation between application-independent rules and application-specific instances to earlier phases of FASTUS, in particular to the Parser and Combiner, the TREC routing task represented an extremely useful testbed for these adaptations.

Consider topic #12, for example. We want to recognize the various ways in which the simple predication *pollute(x, body-of-water)* might be expressed and then to automatically generate patterns to parse:

- full clauses in the Domain phase
  - "they polluted the stream"
  - "the reservoir has been contaminated"
- complex noun phrases in the Combiner phase
  - "the contamination of the creek"
  - "the bay's pollution"
- compound nouns in the Parser phase
  - "the water pollution"
  - "the polluted lake"

We give as an example the general pattern for the first of the two complex noun phrases. In such phrases, the object of the "of" phrase is the object of the event expressed by the head of the noun phrase ("contamination"):

```
ComplexNP  -->  ({NP[??subj] | NP[??obj]}   P[subcat=gen])
                {V-ING[TRANS,??head] | NP[TRANS,??head]}
                { P["of"]   NP[??obj] |
                  P["by"]  NP[??subj] |
                  P[??prep1] NP[??pobj1] |
                  P[??prep2] NP[??pobj2] }*   ;
  ??semantics  ;;
```

The topic-specific instance is as follows:

```
Instantiate
OfNP
??label = combiner-1-pollute
??subj = chemical
??head = pollute
??obj = body-of-water
??semantics =  weight = (assign-weight ((subj && obj) 10000)| (obj 1000)) ;;
```

The topic-specific instance can be thought of as a collection of macro definitions. During grammar-compilation, the "macro calls" in the patterns are expanded. In the example above, the string "??subj" is replaced by "chemical"; "??head", by "pollute", and so on. The resulting instantiated pattern is shown below:

```
ComplexNP  -->  ({NP[chemical] | NP[body-of-water]}   P[subcat=gen])
                {V-ING[TRANS, pollute] | NP[TRANS, pollute]}
                { P["of"]   NP[body-of-water] |
                  P["by"]  NP[chemical] }* ;
weight = (assign-weight ((subj && obj) 10000)| (obj 1000))
                ;;
```

Items in square brackets represent constraints on the phrase. For instance, "stream", "river" and "reservoir" are all nouns with the lexical feature *body-of-water* and only noun phrases with such nouns as heads satisfy the constraints on NP's in the rule instance.

3.3 An Initial Experiment

Having extended the method of general rules and application-specific instances to the Parser and Combiner, we were in a position to write grammars for multiple topics. We modeled our approach on an experiment we had performed running output from the INQUERY IR system through the MUC6 version of FASTUS.

The MUC-6 scenario template task is quite similar to TREC topic #15: "Document will announce the appointment of a new CEO and/or the resignation of a CEO of a company." In essence, the only difference between the MUC6 task and TREC topic #15 is that the latter is limited to the position of CEO. INQUERY was run with TREC topic #15 as an ad hoc query, producing a set of 1000 text documents it deemed most likely to be relevant, and ranking them in order from most likely relevant to least likely. Both the document set and the ordering served as inputs to FASTUS.

We tried two different schemes for using the information from FASTUS to reorder the input list. Both involved configuring the MUC6 grammar to assign scores to phrases based on correlation of phrase type with relevance. In one scheme, we assigned scores to patterns manually, based on intuitions as to differential contributions to relevance judgments; in the second, a probabilistic model for the relevance of a document was inferred from a set of training data

As a basis for the first experiment, we picked 100 articles from the middle of the ordered set that INQUERY produced (in particular, articles ranked 401 through 500). The templates that the FASTUS MUC-6 system produced from those articles were examined to identify criteria for assigning a relevance rank to an article. We then had FASTUS assign a numerical score from 0.1 to 1000 to the templates that it produced for a phrase as follows:

1. CEO + person name + company name $\mapsto$ 1000
2. CEO + company name $\mapsto$ 100
3. CEO + person name $\mapsto$ 10
4. CEO + transition verb $\mapsto$ 1
5. CEO + BE verb $\mapsto$ 0.1

The score of a phrase was taken to be the sum of the scores of the templates created from that phrase; the scores from the phrases were summed to yield an article's score.

For the second experiment, we asked how a system for automatically identifying features concerning the output of FASTUS and determining the relative strengths of these features, would compare with the results obtained by the manually tuned system. A probabilistic model for the relevance of a document was inferred from a set of training data.

The results of these initial experiments [Kehler1996] were encouraging enough to motivate us to try both a larger and a more realistic experiment: one involving routing queries for many topics, none of which could have as much effort put into developing queries/grammars for it as was involved in producing the MUC6 application.

4 TREC6

For TREC6 we teamed with GE. They provided us with a ranked list of 2,000 documents for each query (using their version of SMART). We developed grammars for 23 of the 47 topics. For these 23 topics, FASTUS ran over the 2,000 articles, reordered them, and truncated to 1000. For the other 24 topics, we simply truncated GE's ordering at 1000 documents.

As in our first experiments, the reordering is achieved by having patterns—that is, instances (see example above)—assign a score to the segment (phrase) of an article successfully matched against. An article's total score is the sum of the scores of all the patterns that matched against phrases in that article.

As before, we broke ties by maintaining GE's relative order within a class of articles with identical scores.

For each topic we read a small number of relevant articles (10-15), constructed a topic-specific grammar by writing instances of the kind exemplified above, and then ran the grammar over some portion of the training data. Whenever a pattern matched a phrase, the phrase was recorded as being either a correct match or a false positive. We would review both sets of phrases, look at some of the relevant articles that were missed, and revise the grammar. After a small number of iterations of this kind, we would declare the grammar done, and move on to the next topic.

For training, we used a subset of the TREC6 training data, but we did not run over any of the results of GE's output on that corpus. We return to this point in our concluding section.

The scores were assigned with a threshold score in mind. An article had to contain at least one pattern that had a score of 1000 or above to be moved toward the beginning of the ordering; scores below 1000 had no effect on the order and were used solely for diagnostics. There was one exception to this general rule. For topic #11 (the space program), we tried the following mini-experiment: phrases were assigned maximum scores of 250, so that at least four matching phrases were needed to move an article to the front of the ranking. This was intended to handle cases where we could find no especially reliable phrasal indicators of relevance. Another way to put this is that this method is a crude approximation to a statistical approach based on co-occurrence data.

When writing the grammars our approach was to aim for high precision and to sacrifice recall when we were in a position to make a precision/recall tradeoff.

5 Results

As noted above, we were able to write grammars for 23 of the topics. Most of these grammars were written by a Stanford undergraduate who was, at the outset, completely unfamiliar with FASTUS. He spent about 5 to 6 hours per topic.

Overall we improved the average precision very slightly over our input, from 27% to 27.3%. We have included a graph, Figure 1, of precision versus recall for the 23 topics. The overall results of the combined system for average precision are: 12 topics above the median; 3 at the median; and 8 below the median.

FASTUS improved the average precision (non-interpolated) compared to the GE input on 17 of the 23 topics. On 12 of these the resulting average precision was above the median; in seven of these cases, we transformed above median input into an even better ordering. On one of these 7, topic #10001 (soil pollution), FASTUS had the best average precision (.6322). There are several possible reasons for this success. One is that there was less training data for this topic than for any of the others: 100 articles instead of (app.) 1000. Our

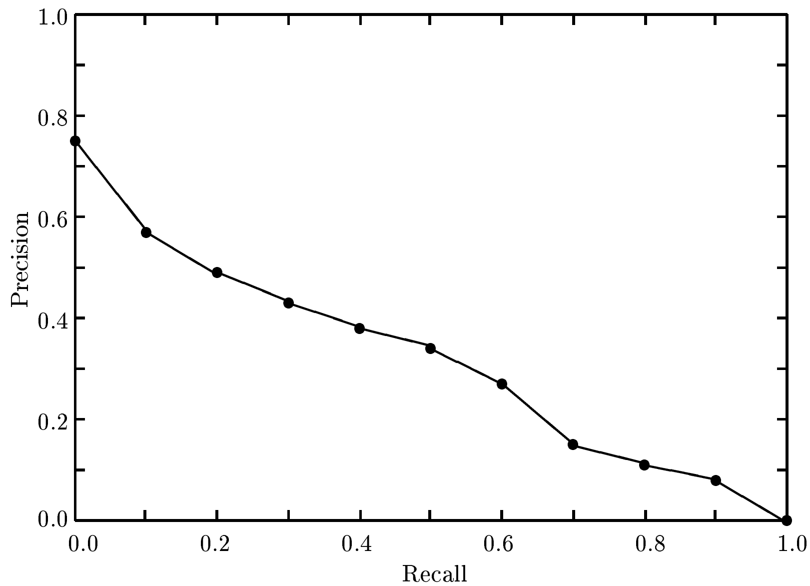

Figure 1: Recall vs. Precision for the 23 Topics

approach may suffer less from this relative scarcity of training data than purely statistical approaches. We only wrote grammars for two of the topics that had 100 or fewer training articles, so this is still conjecture. The other topic of this kind was #282 (violent juvenile crime), on which we very slightly improved above-median input. Second, we were able to reuse parts of the grammar from topic #12 (water pollution) and we benefitted in much the same way as if we had had more relevant articles. Finally, the topic may just be one where the information tends to be expressed in ways that our patterns can recognize.

On six of the topics, FASTUS lowered the average precision of the input order. A characterization of these cases is instructive. Two of these, topics #23 (legal repercussions of agrochemical use), and #194 (writer's earnings) had 7 and 8 relevant documents in the training sets, respectively. When faced with that little data, we could only guess at the various ways in which relevant information might be expressed and at which patterns would recognize them. Obviously we did not make very good guesses.

One of the six topics on which we degraded input performance was topic #11. As mentioned above, we departed from our normal method on this topic, assigning maximum scores of 250 to patterns in order to require that at least four phrases match. We did quite poorly on this topic as well.

Obviously, our approach is sensitive to the way information is expressed. Topic #228 (environmental cleanup success stories) represents an example of a topic to which our approach is not well suited. The relevant articles were not characterized by a relatively small number of highly indicative phrase or event types—in this respect this topic was like #11—and our approach did poorly.

We have not been able to characterize our performance on the two remaining topics on which we degraded performance, beyond being convinced that we could have and should have done better.

6 Discussion and Conclusion

We have described an experiment in the use of a particular kind of Natural Language Processing technology within an Information Retrieval application. The experiment involved adapting FASTUS, an Information Extraction system, for use as a post-filter to be run over the output an IR system, GE's version of SMART. The results of the experiment are of two different kinds: First, the experiment motivated significant changes to the architecture of FASTUS, changes aimed at making it easier to develop application-specific grammars. The resultant grammars can be used for typical IE tasks, as well as for IR tasks. Second, the results on the TREC6 routing task, while certainly not impressive, just as certainly do not foreclose the possibility of using IE technology in this way in IR applications. Rather they suggest that some care must be exercised in determining the proper range of application of this mixed-technology approach to IR, for there is little reason to think it is appropriate everywhere. At least two simple guidelines can already be induced; one purely quantitative, the other, not:

- There must be sufficient data, in particular, enough relevant articles, (i) to accumulate patterns for the initial grammar-writing exercise and (ii) to use as a training corpus for adding to and "debugging" those grammars.
- There must be fairly reliable indicators of relevance that are fully phrasal in structure.

We have also learned some other more engineering-oriented lessons:

- If relevant data is scarce in a given corpus, it is worth it to go out and look for more.
- In following a hybrid approach such as ours, it is important to use the output of the IR system in training.

Perhaps these latter lessons should have been obvious from the beginning. In any event, we hope to make good use of them, and others, as we pursue this approach to applying Information Extraction technology for Information Retrieval.

7 Acknowledgments

We would like to thank Matt Caywood and Mabry Tyson for their efforts in making FASTUS run over much larger amounts of text than ever before. We would also like to thank Tomek Strzalkowski and his group at GE for providing us with their system's output. Without it we would not have been able to participate. We are happy to acknowledge that this work was funded under Contract No. N66001-94-C-6044 from the Naval Command, Control, and Ocean Surveillance Center. We also received internal research and development funding from SRI International.

References

[Appelt et al.1995] Doug Appelt, John Bear, Jerry Hobbs, David Israel, Megumi Kameyama, Andrew Kehler, Mark Stickel, and Mabry Tyson. 1995. SRI International's FASTUS System MUC-6 Test Results and Analysis. In *Proceedings of the 6th Message Understanding Conference*, ARPA, Columbia, MD.

[Appelt et al.1996] Doug Appelt, John Bear, Jerry Hobbs, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1996. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Emmanuel Roche and Yves Schabes (eds.) *Finite State Devices for Natural Language Processing*. MIT Press, Cambridge, MA.

[Berger, Pietra, and Pietra1996] Adam Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

[Croft et al.1996] Bruce Croft, James Allan, Lisa Ballestros, James Callan, and Zhihong Lu. 1996. Recent Experiments with INQUERY. In *Proceedings of TREC-4*, to appear.

[Faloutsos and Oard1996] Christos Faloutsos and Douglas Oard. 1996. A Survey of Information Retrieval and Filtering Methods. Technical Report, Information Filtering Project, University of Maryland, College Park, MD.

[Grishman and Sundheim1995] Ralph Grishman and Beth Sundheim. 1995. Design of the MUC-6 Evaluation. In *Proceedings of the 6th Message Understanding Conference*, ARPA, Columbia, MD.

[Harman1996] Donna Harman. 1996. Overview of the Fourth Text REtrieval Conference (TREC-4). In *Proceedings of TREC-4*, to appear.

[Hearst1992] Marti Hearst. 1992. Direction-Based Text Interpretation as an Information Access Refinement. In [Jacobs1992].

[Jacobs1992] Paul Jacobs (ed.) 1992. *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* Lawrence Erlbaum Associates, Hillsdale, NJ. ew Jersey.

[Lewis1992] David Lewis. 1992. Text Representation for Intelligent Text Retrieval: A Classification-Oriented View. In [Jacobs1992].

[Sparck Jones1992] Karen Sparck Jones. 1992. Assumptions and Issues in Text-Based Retrieval. In [Jacobs1992].

[Kehler1996] John Bear, David Israel and Andy Kehler. 1996. Using Information Extraction to Improve Document Retrieval. Unpublished mss.