Natural Language Information Retrieval: TREC-7 Report

Tomek Strzalkowski, Gees Stein, G. Bowden Wise
GE Research & Development
Jose Perez-Carballo
Rutgers University
Pasi Tapanainen, Timo Jarvinen, Atro Voutilainen
University of Helsinki
Jussi Karlgren
Swedish Institute of Computer Science

Team id: GERSH

1. Summary

The GE/Rutgers/SICS/Helsinki team has performed runs in the main ad-hoc task. All submissions are NLP-assisted retrieval. We used two retrieval engines: SMART and InQuery built into the stream model architecture where each stream represents an alternative text indexing method.

The processing of TREC data was performed at Helsinki using the commercial Functional Dependency Grammar (FDG) text processing toolkit. Six linguistic streams have been produced, described below. Processed text streams were sent via ftp to Rutgers for indexing. Indexing was done using Inquery system. Additionally, 4 steams produced by GE NLToolset for TREC-6 were reused in SMART indexing.

Adhoc topics were processed at GE using both automatic and manual topic expansion. We used the interactive Query Expansion Tool to expand topics with automatically generated summaries of top 30 documents retrieved by the original topic. Manual intervention was restricted to accept/reject decisions on summaries. We observed time limit of 10 minutes per topic. Automatic topics expansion was done by replacing human summary selection by an automatic procedure, which accepted only the summaries that obtained sufficiently high scores. Two sets of expanded topics (automatic and manual) were sent to Helsinki for NL processing, and then on to Rutgers for retrieval. Rankings were obtained from each stream index and then merged using a combined strategy developed at GE and SICS.

2. Background

The work reported here was part of the Natural Language Information Retrieval project (NLIR) (Strzalkowski et al., 1997; Strzalkowski, 1995). One of the thrusts of this project has been to demonstrate that robust NLP techniques can help to derive better representation of text documents for indexing and search purposes than any simple word and string-based methods commonly used in statistical full-text retrieval. This was based on the premise that linguistic processing can uncover certain critical semantic aspects of document content, something that simple word counting cannot do, thus leading to more accurate representation. We demonstrated that NLP can be done efficiently on a very large scale, and that it can have a significant impact on IR. At the same time, it became clear that exploiting the full potential of linguistic processing is harder than originally anticipated. In particular, simple linguistically motivated indexing (LMI) techniques turned out to be no more effective than well-executed statistical approaches, while more advanced NLP techniques, such as concept extraction, remained too expensive for large-scale applications (Sparck-Jones, 1999).

Given this state of affairs, we went on to investigate specific conditions under which LMI could be more beneficial. For example, we have noticed that the amount of improvement in recall and precision which we could attribute to NLP, appeared to be related to the type and length of the initial search request. Longer, more detailed topic statements responded well to LMI, while terse one-sentence search directives showed little improvement. This is not particularly surprising considering that the shorter queries either contain a handful of highly discriminating terms or are deliberately vague. On the other hand, detailed statements are more typical for situations where the user is uncertain of how to succinctly express the query. In such cases linguistic processing helps to sharpen the query thus making it more effective.

We adopted the *topic expansion* approach in which the original topic is expanded using passages selected from sample retrieved documents. The intent was to expand the initial search specifications in order to cover their various angles, aspects and contexts. Based on the observations that NLP is more effective with highly descriptive queries, we designed an expansion method in which passages from related, though not necessarily relevant documents were imported into the user queries. This method produced a fairly dramatic improvement in the performance of several different statistical search engines that we tested boosting the average precision by anywhere from 40% to as much as 130%. Therefore, we concluded that topic expansion appears to lead to a genuine, sustainable advance in IR effectiveness. Moreover, we showed at TREC-7 that this process can be automated while maintaining at least some of performance gains.

It is not difficult to see why topic expansion, when done properly, works this well. As the expansion progresses, the expanded search topic takes the form of an extended brief on that topic, a meta-document that contains the information that the user seeks. In other words, as we keep improving the query to get more relevant documents, we are in effect forming an answer. The expanded topic is indeed close to an answer, as demonstrated by the following example:

> ORIGINAL TOPIC (Topic 362 Description):
> *Identify incidents of human smuggling.*
>
> EXPANDED TOPIC:
> *Federal immigration agents arrested 31 illegal aliens at Los Angeles International Airport overnight, bringing to more than 200 the number of people nabbed in a nationwide crackdown on high-altitude "people smuggling," authorities said today.*
>
> *A federal grand jury indicted a Carlsbad motel operator, five Los Angeles men and a Mexican national on charges of running an alien-smuggling ring that whisked about 600 people per month to Santa Ana and Los Angeles.*
>
> *INS officials described Eastern's flight, which departs daily from LAX at 10:50 p.m., as an unwitting conduit in a massive transcontinental smuggling operation that apparently moved several thousand illegals out of Los Angeles in the last month alone. Some of the aliens arrested said they paid as much as $4,000 for a package deal, transportation from home, housing, the Eastern plane ticket and a job in New York.*
>
> *Federal agents took 60 illegal Chinese aliens into custody in southern Alabama and announced the arrest of the alleged ringleader of a smuggling operation that planned to bring 35,000 more into the United States. Officials from the Immigration and Naturalization Service and the U.S. Customs Service said the aliens were caught as they arrived by plane from Panama in Fairhope, Ala. The ring planned to bring in Chinese nationals from Panama and Bolivia in an operation believed to be run by persons linked to former Panamanian dictator Manuel A. Noriega.*
>
> *U.S. Border Patrol agents intercepted a tractor-trailer rig late Tuesday packed with 105 illegal aliens on Interstate 15 near Rancho Bernardo, authorities said. Agents stopped the truck about 11:30 p.m. and arrested the driver, Frank Ellinger, 45, of National City on suspicion of smuggling aliens, patrol spokesman Michael Gregg said.*

The reader may note that this expanded topic, reads a bit like the *News from Every State* column in *USA Today*. It is in effect a (likely incomplete) brief on a single subject. It its thus more than just an expanded search topic, and represents an important step towards a new kind of information retrieval where the information, not the document containing it, becomes the target.[1]

The example shown above has been obtained through a human-assisted interactive topic expansion process explained in more details below. In a fully automated expansion, where NLP techniques replace human judgments, the results are not nearly as good as yet. Thus far we have used only very simple linguistic tools (i.e., those suitable for high-volume IR applications) to assist automatic expansion, but we see this area as ripe for more advanced processing techniques, including entity and event extraction, coreference and cross-reference techniques, etc.

3. Ad-Hoc submissions

In TREC-7 we participated in the ad-hoc track only. Below are short descriptions of official runs.

3.1. Summarization-based manually-assisted topic expansion

This was a multi-stream run using InQuery against the manually expanded topics. Summaries used in expansion were derived from top-ranked documents retrieved by SMART using the initial topics (title+description only). The key characteristics of this run is the 10 minute time limit imposed on topic expansion. All expansion has been performed via the Query Expansion Tool interface (QET) which allows the user to view only the summaries of top retrieved documents, and select or deselect them for topic expansion. By default, summaries of all top 30 documents were used for expansion unless the user manually deselected some (this was precisely the only form of manual intervention allowed.)

We observed that for many queries 2 interactions were possible within the 10 minute interval. The first interaction (submit original query, wait for result, get 30 summaries, review & deselect summaries, and commit the selections) would take typically 4-6 minutes. In the second interactions, only the new documents retrieved in top 30 ranks (if any) were considered, therefore usually 3-4 minutes were sufficient. The target of expansion was to get between 5 and 10 "relevant" summaries within the allotted time. If this was achieved within the first interaction, no further search was performed. Otherwise, the second interaction was attempted if at least 3 minutes remained. This 6-4 split was determined in dry-run trials with TREC-6 queries.

The topic expansion interaction proceeds as follows:

1. The initial natural language topic statement is submitted to a standard retrieval engine via a Query Expansion Tool (QET) interface. The statement is converted into an internal search query and run against the database.
2. The system returns topic-related summaries of top N (=30) documents that match the search query.
3. The user reviews the summaries (approx. 5-15 seconds per summary) and *de-selects* these that are *not* relevant. For TREC-7 evaluations, we set time limit of 10 minutes per query (clock time).
4. All remaining summaries are automatically attached to the search topic.
5. The expanded topic is passed through a series of natural language indexing steps and then submitted for the final retrieval.

3.2. Summarization-based automatic topic expansion with InQuery

This was a single-stream automatic run using InQuery against the automatically expanded topics. Plain stems stream and syntactic noun phrase stream were combined and converted into a single InQuery-syntax representation. Again, the expanded topics were generated using summaries obtained from SMART-retrieved documents. The original un-expanded short topics (title+description only) were submitted to

[1] This approach bears only superficial similarity to passage retrieval used in standard IR (Callan, 1994; Kwok et al., 1993). In passage retrieval fixed-size segments are weighted against the search query which is helpful in assessing relevance of longer documents. However, no attempt is made at extracting coherent "stories".

SMART (version. 11) in stems-stream mode, and the top 100 returned documents were retained. These documents were automatically summarized with GE Summarizer using topic title as to obtain a 5% topical indicative summaries.

Summaries were selected for expansion if they had a sufficient level of "overlap" with the original search topic. The "overlap" score was determined by the number of shared terms, as well as the "locality" of the summary. In this experiment we required that there was at least 60% overlap on the content terms between the summary and the original topic. In addition, multi-paragraph summaries were required for each paragraph to have at least 40% overlap with the topic, except for the blocks of consecutive paragraphs.

These selection criteria are fairly simplistic and tests performed with TREC-6 data were generally inconclusive as to their effectiveness. This is because term overlap is not a good indication of relevance (we know that!). Moreover, the goal of expansion was to add new terms to the topic, not just more of the same, thus a too-high degree of overlap would not be of much use.

3.3. Summarization-based automatic topic expansion with SMART

This was a multi-stream automatic run produced using SMART rather than InQuery. Automatically expanded queries were NL processed using GE NLToolset and run against the 4-stream index originally produced for TREC-6. Streams were merged using the same procedure that was developed for TREC-6.

4. Helsinki's NLP System overview

We used Helsinki's Functional Dependency Grammar (FDG) includes the EngCG-2 tagger and dependency syntax which links phrase heads to their modifiers and verbs to their complements and adjuncts. FDG was applied to the whole corpus, with the output passed to the stream extractor. The streams were generated as follows:

4.1. Simple Streams

0. *stem*: just stemmed words, stopwords removed.
1. *name*: all proper names
2. *aan*: simple noun phrases with attributes. Basically adjective-noun sequences minus some exceptions.

4.2. Direct Dependency Streams

3. *sv*: subject-verb pairs where the subject is a noun phrase.
4. *vo*: verb-complement pairs. The complement includes objects and some object-like adverbial classes.

4.3. Indirect Dependency Streams

5. *nofn*: "N1 ... of ... N2" pairs, where N1 and N2 are heads of simple noun phrases.
6. *sc*: subject-complement pairs where the complement modifies the subject (flowers grow wild => wild+flower).

5. Details of Helsinki's FDG System

5.1. Functional Dependency Grammar

The Functional Dependency Grammar (FDG) parser (Jarvinen and Tapanainen, 1997; 1998; Tapanainen and Jarvinen, 1997) produces surface-syntactic analyses for sentences in terms of explicit dependency structures. These structures are trees where the words correspond to the nodes.

In principle, each word in a sentence is connected to an unique head by a labeled arc, though also partial analyses for complex sentences are allowed. The labels refer to syntactic functions such as subject, object, and so on. The highest node is connected to an external root.

A simplified example in Table 1 shows the analysis of the sentence *I tamed a bird*. The arc between *I* and *tamed* denotes that *I* is the modifier of *tamed* and its syntactic function is that of subject. Similarly, *a* modifies *bird*, and it is a determiner.

The text format of the analysis in Figure 1 shows the functional labels with a numeric pointer to the head, and some additional information produced by the parser. The third column shows the base form of the word, and the last column contains the word-class information. Due to the strong correlation of the syntactic analysis produced by the FDG to the semantic relations, the output is usable to tasks where semantic rather than syntactic information is required.

We use the FDG output to collect pairs of words that were connected by certain syntactic relations. For example, if we excerpt words connected by the object relation, the analysis in Figure 1 produces the normalised string "tame bird".

Table 1. Sample analysis

1 I i subj:>2 PRON	2 tamed tame main:>0 V	3 a a det:>4 DET	4 bird bird obj:>2 N

5.2. Morphological analysis and lemmatization

The adjectives and nouns were returned to their morphological baseforms. The participial adjectives and nouns are returned to the verbal form (e.g. growing economy → grow+economy) which makes them equivalent to the verbal usage (e.g. the economy grows → grow+economy).

5.3. Names

The name recognizer, based on the Conexor Name Recognizer (at www.conexor.fi), identifies "named entities" consisting of one or more words. Typically, names are nominal heads written in the upper case, with any number of pre-modifiers. Also coordinations and certain types of post-modification (e.g. post-modifying PPs) are recognized as legitimate parts of names, e.g. "Procter & Gamble"; the "City of London". We do not regard titles as names (though they certainly are useful clues for identifying names).

In our system, names are identified on the basis of three types of information: lexical, orthographical and grammatical. This information is used on the basis of hand-written linguistic rules. The name recognizer is reasonably fast; on a mid-range Pentium PC running Linux, it processes well above 2,000 words per second. At present, our name recognizer performs no sub-classification. The ability to identify e.g. persons, organizations and locations remains to be added in the program. No rigorous evaluation of the name recognizer has been carried out. Our hunch is that well over 90% of names occurring in many types of English text (at least journalistic, fiction and scientific texts) are recognized.

5.4. Noun phrase streams

The simple noun phrases with attributives are collected to one stream. The syntactic position in the sentence was used to filter the noun phrases. Adverbs of time (e.g. "tomorrow night") and other generic adverbs (e.g. *Emissaries returned *home**) were excluded by using the syntactic function given by FDG. We did not apply stop word lists here. The *of*-genetive streams are represented through the head words of noun phrases. For instance, the noun phrase *[large burlap sacks] of [the imported material]* contains two noun phrases. The head words of both are collected into the stream: *material+sack*. Word-class information is used for filtering out undesired candidates for the streams.

5.5. Valency streams

Also verbs are excerpted together with their various dependents. There are two classes: subject-verb and verb-complement pairs. The latter include predicatives, direct objects and other object-like adverbials.

Many of the subject and object types are filtered out by using word-class labels and various heuristics to exclude non-nominal elements.

5.6. Indirect dependency

Sometimes the connecting information is obtained indirectly using the syntactic functions in the FDG output. A typical pair is a subject and its complement e.g. {\em flowers grow wild => wild+flower}.

5.7. An example: text and streams

"Gardening: The perennial pleasures of spring- Robin Lane Fox prepares to strike an economic blow for a better garden on a shoestring.
BEFORE LONG, better weather ought to have caused gardeners' sap to rise: act now while enthusiasm is fresh and strike an economic blow for a better garden on a shoestring.
Seeds are no longer as cheap as they were and, admittedly, I sometimes grow them for the hell of it, just to see if I can make them come up.
It is no longer time to postpone the plunge, but the first seeds to go in are not the most obvious.
It is still too early to be sowing tobacco plants, cosmos daisies and all the mainstays of summer bedding which grow quickly and will be too far advanced if started before March.
Perennial flowers are another matter.
Among these early sowers, I am casting my net more widely and am being prompt with less-familiar perennials which ought to flower from July onwards.
Geraniums are obvious candidates, especially now that so many colours have been selected and bred for seed-raising: even for amateurs, cuttings are almost a matter of the past.
I leave most of the geraniums to others, but carnations are another matter.
Not long ago, I was editing Vita Sackville-West's old gardening columns when I was carried away by her description of the Chabaud strain of carnation.
Their colours, she felt, had the quality of a Van Gogh painting- I remember that she described some of them as bistre.
On the spur of a good read, I tried to grow my own, but started too late.
From a sowing in mid-March, I had none of her fancies, no bistred beauties or blooms of old blood-red.
The wretched plants never flowered at all.
Once bitten, never shy: you know the gardening instinct.
So, this year I am starting Chabaud carnations from seed in the first week of February.
Somewhere in Britain, people must still grow them happily because garden centres stock them on open shelves in their standard ranges of seed from Suttons or Thompson and Morgan.
The seed will germinate in the usual amateur's pot, filled with a standard seed compost and covered with a tight stretch of cling-film to retain the heat and sweat.
Chabaud carnations like heat in order to spring into growth.
They will germinate in a warm cupboard, below the spare bath towels, if you remember to retrieve them and roll back the cling-film at the first signs of emergent shoots……"

5.7.1. Noun phrase stream

perennial+pleasure spring robin+lane+fox economic+blow good+garden shoestring good+weather gardener+sap enthusiasm economic+blow good+garden shoestring seed hell time plunge seed tobacco+plant cosmos+daisy mainstay summer+bed

5.7.2. Of-genitive stream

spring+pleasure bed+mainstay past+matter geranium+many strain+description carnation+strain paint+quality read+spur fancy+none blood-red+bloom February+week seed+range cling-film+stretch shoot+sign seed+list success+three Shirley+butcher stockist+list flower+mass pink+touch flower+variety

5.7.3. Name stream

robin+lane+fox Vita+SackvilleWest chabaud van+Gogh midMarch chabaud Britain Sutton Thompson Morgan Chiltern+seed Cumbria northwest+England gaura+lindheimeri Chiltern+seed butcher+of+Shirley Croydon south+London snowcloud gaura

5.7.4. Verb-object stream

strike+blow strike+blow postpone+plunge sow+plant cast+net edit+column have+quality grow+own start+late know+instinct start+carnation retain+heat have+day sell+mixture join+list reach+ft equal+gaura give+mass have+habit have+hybrid catch+mood

5.7.5. Subject-verb stream

fox+prepare sap+rise colour+select plant+flower people+grow centre+stock seed+germinate sowing+owe seedsman+sell seed+join border+need dozen+come nursery+charge balloon+show

5.7.6. Subject-complement stream

fresh+enthusiasm flower+matter geranium+candidate carnation+matter available+catalogue white+valerian valerian+plant confuse+white variety+valerian wild+flower good+success name+platycodon good+form name+blue

6. Stream Merging

Our goal was to merge results from searches over several stream indexes built over the same data set. The merging program takes the ranked lists of documents obtained from each stream, and produces a single unified ranking. As in the past, we aimed at obtaining a better ranking than any single stream search could produce.

6.1. What we know from past experience

We can draw on experience from past searches to estimate behavior in future ones. The parameters we have recourse to are

- past average precision for the respective streams;
- how these measures vary at different combinations of ranges of ranks and scores; and
- consistency in behavior - how much the precision measures and the overlap fluctuate from query to query.

The observable measures we can make use of at merge time are for each document its rank and relative score for each stream. The parameters we have chosen to disregard the consistency measure after some not very systematic measurements indicating the fluctuation is small for TREC data, and fold in the total average precision with the others in a matrix of estimated precision.

6.2. Example data

Here is an example: the table shows the average precision at various ranks for thirty TREC 6 queries designated as training material. This tells us that documents that are ranked between 0 and 10 for both streams (stems on the y axis and pairs on the x axis) have an average precision of 0.482; documents that are ranked between 26 and 100 in the stems stream and top 10 by the pairs stream have an average precision of 0.289; documents that are ranked between 26 and 100 in the pairs stream but not ranked at all by the stems stream have an average precision of 0.046.

	0	10	25	100	200	1000
0	0.000	0.077	0.066	0.046	0.022	0.008
10	0.241	0.482	0.262	0.400	0.250	0.270
25	0.228	0.303	0.318	0.226	0.094	0.183
100	0.105	0.289	0.303	0.192	0.206	0.146
200	0.066	0.100	0.211	0.163	0.173	0.130
1000	0.026	0.124	0.158	0.150	0.157	0.095

6.3. Theoretical issues - distribution

This distribution ideally should be modeled by some useful function of two or three parameters which could be used for an estimate of future behavior: something like

prec(rankA,rankB,overlap(A,B))

which could be trained by using data such as past overlaps - relevant and non-relevant - at various ranks. Until we come up with something like it, we can use the matrix directly as an estimate.

6.4. Merging - using the information

The simplest merging approach would be to use the precision estimates as probability estimates directly. Thus, in the example, first pick all documents that are shared in the top ten, thereafter the documents that are in the top ten for the stems stream but rank 26-100 in the pairs stream, thereafter documents that have ranks 11-25 in both streams, and so forth. As is obvious, the training data are insufficient: the matrix cell values should be expected to grow monotonically from the 10-10 origin outwards, or possibly to retreat eventually, not to waver like the ones in the example.

6.5. Experiment

The approach is easy enough to test. First, however, the order of documents within cells must be determined. Below are a couple of experiments. The result hinges on how the cell ranges are chosen. Choose the intervals too small, and overfitting will occur; too large, and no learning will take place.

Also, we must determine if the rank or the relative score is a better measure of document relevance. Relative scores gave somewhat better performance in our experiments, but are dependent on the implementation of the stream, and thus somewhat less convincing in the general case. We find that a slight improvement indeed occurs in the test data. However, the approach is vulnerable to overfitting, and needs to be tested on more material to be useful.

Also, an noticeable problem was the number of non-judged documents in the collection. The standard trec_eval script judges non-judged documents to be one hundred per cent non-relevant, which seems overly pessimistic.

6.6. Some further observations

In our experiments we choose to model past relative performance by the TREC 11-point average precision, but take into account the precision distribution over retrieval scores. Average precision and recall is arguably the most important measure. If we know from historical data that a certain stream produces consistently excellent results and another consistently low-grade results, we should weight them in proportion to that performance.

We ran the results stream by stream, and built seven-way relative average precision matrices, where each cell represented a score range within the seven participating streams. We then merged the streams, picking the cells with highest average precision in the training data first. For example, if the stream aan scores a document more than 0.926 and it is not ranked in any other stream it has only 0.101 likelihood of being relevant; if it is scored more than 0.942 by stream aan and more than 0.935 by the plain stems stream it has 0.318 likelihood of being relevant. Not a surprising result.

The experiment went well, and we tuned a number of parameters - score ranges, cell sizes, etc. - to produce usefully improved results, but for the main run, we found that reworking the entire processing chain gave us too little training data for the algorithm to produce stable results. In the end we went back to the tried hand-worked scheme of previous years.

7. Stylistics experiments

This year, as previous years, we ran several experiments to predict relevance of an item from non-topical features: stylistic features which in other experiments have predicted the genre of texts with reasonable accuracy, and which seem to give significant correlation with relevance using standard statistical measures (e.g. Karlgren, 1996). However, the predictive power is too weak for rule generation, and several important assumptions - such as requirements of normal distribution of variables - underlying standard multivariate categorization metrics are not met. We are currently performing a series of experiments using machine learning techniques.

8. Preliminary Results Analysis

We continue to analyze the results. The preliminary examination indicates that the merging system did not work as we anticipated. The problem may not be with the merging itself, rather with the way streams were defined in TREC-7. In contrast with TREC-6, we decided to make more fine-grained distinctions between various text representations, resulting in many "thin streams", i.e., streams that retrieve few documents based on very limited information, although highly specialized features. This, we believe, created ranked lists that were far less reliable than with "fatter" streams, i.e., the score differentials due to content were too small to be reliably distinguished from noise. Therefore, any merging system using these ranks would produce unreliable results. This is indeed our experience this year. While the main unmerged stream runs performed quite well, all merged runs did poorly.

The table below summarizes the "unofficial" results obtained with the expanded topics, before any NLP indexing and stream merging took place. The results correspond to NLIR "stems stream", the basic word-based stream. The reader should note the performance increase. These results are significantly better than any merged runs officially submitted.

queries	*original T+D*	*long T+D+N*	*expanded automatic*	*expanded interactive*
SYSTEM	RU-INQUERY	RU-INQUERY	RU-INQUERY	RU-INQUERY
PRECISION AVERAGES				
11pt Average	0.1692	0.2036	0.2019	0.2932
%change		+20.0	+20.0	+73.0
At 10 docs	0.4620	0.5000	0.4120	0.6140
%change		+8.0	-11.0	+33.0
At 30 docs	0.3153	0.3587	0.3013	0.4327
%change		+14.0	-3.0	+37.0
At 100 doc	0.1756	0.2068	0.1922	0.2668
%change		+18.0	+9.0	+52.0
Recall	0.44	0.51	0.46	0.69
%change		+16.0	+5.0	+57.0

In automatic expansion we observed a good precision increase (about 20%) over the unexpanded topics. This is encouraging, but not nearly as effective as in manual expansion where we noted 73% increase.

Still, however unsophisticated, the automated expansion did produce an increase nearly identical to what is attributed to the *narrative* field in the topics.

Acknowledgments

The work reported here was supported in part by the Defense Advanced Research Projects Agency under the Tipster III contract 97-F157200-000 through the Office of Research and Development.

References

Callan, Jamie. 1994. “Passage-Level Evidence in Document Retrieval.” Proceedings of ACM SIGIR'94. pp. 302-310.

Jarvinen, T., and Tapanainen, P. 1997. “A dependency parser for English.” Tech. Rep. TR-1, Department of General Linguistics, University of Helsinki, Finland.

Jarvinen, T., and Tapanainen, P. 1998. “Towards an implementable dependency grammar.” In Processing of Dependency-Based Grammars. Montreal, Canada. S. Kahane and A. Polguere, Eds., COLING-ACL'98, Association for Computational Linguistics, Universite de Montreal, pp. 1-10.

Kwok, K.L., L. Papadopoulos and Kathy Y.Y. Kwan. 1993. “Retrieval Experiments with a Large Collection using PIRCS.” Proceedings of TREC-1 conference, NIST special publication 500-207, pp. 153-172.

Sparck-Jones, Karen. 1999. “What Is The Role for NLP in Text Retrieval”. In T. Strzalkowski (ed.) Natural Language Information Retrieval. Kluwer. pp. 1-25.

Strzalkowski, Tomek, Louise Guthrie, Jussi Karlgren, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, and Jon Wilding. 1997. “Natural Language Information Retrieval: TREC-5 Report.” Proceedings of TREC-5 conference.

Strzalkowski, Tomek. 1995. “Natural Language Information Retrieval.” Information Processing and Management, Vol. 31, No. 3, pp. 397-417. Pergamon/Elsevier.

Strzalkowski, Tomek, Fang Lin, Jose Perez-Carballo, and Jin Wang. 1997. “Natural Language Information Retrieval: TREC-6 Report.” Proceedings of TREC-6 conference.

Tapanainen, P., and Jarvinen, T. 1997. “A non-projective dependency parser.” In Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, D.C. Association for Computational Linguistics, pp. 64-71.