Natural Language Processing in Information Retrieval

Thorsten Brants

Google Inc.

Abstract

Many Natural Language Processing (NLP) techniques have been used in Information Retrieval. The results are not encouraging. Simple methods (stopwording, porter-style stemming, etc.) usually yield significant improvements, while higher-level processing (chunking, parsing, word sense disambiguation, etc.) only yield very small improvements or even a decrease in accuracy. At the same time, higher-level methods increase the processing and storage cost dramatically. This makes them hard to use on large collections. We review NLP techniques and come to the conclusion that (a) NLP needs to be optimized for IR in order to be effective and (b) document retrieval is not an ideal application for NLP, at least given the current state-of-the-art in NLP. Other IR-related tasks, e.g., question answering and information extraction, seem to be better suited.

1 Introduction

Many Natural Language Processing (NLP) techniques, including stemming, part-of-speech tagging, compound recognition, de-compounding, chunking, word sense disambiguation and others, have been used in Information Retrieval (IR). The core IR task we are investigating here is document retrieval. Several other IR tasks use very similar techniques, e.g. document clustering, filtering, new event detection, and link detection, and they can be combined with NLP in a way similar to document retrieval.

NLP and IR are very different areas of research, and recent major conferences only have a small number of papers investigating the use of NLP techniques for information retrieval. The three conferences listed in table 1 had 411 full papers in total. Only 6 of them (1.5%) explicitly dealt with NLP for retrieval. The percentage is slightly higher for conferences with a main focus on IR (SIGIR, ECIR: 2.0%) than for conferences with a main focus on NLP (ACL: 1.0%). In most cases, researchers work on using existing NLP components (stemmers, taggers, ...), apply them to an IR data set and queries, and then use standard IR techniques. This out-of-the-box use of NLP components that are not geared towards IR might be one reason why NLP techniques are only moderately successful when compared to state-of-the art non-NLP retrieval techniques.

The moderate success contradicts the intuition that NLP should help IR, which is shared by a large number of researchers. This article reviews the research on combining the two areas and attempts to identify reasons for why NLP has not brought a breakthrough to IR.

Table 1: Publications (full papers) explicitly investigating the use of NLP techniques for document retrieval at recent major conferences (#full is the total number of full papers at that conference).

Conf.	**#full**	**NLP for document retrieval**
SIGIR'01	46	-none-
SIGIR'02	44	Improving Stemming for Arabic … (Larkey et al. 2002)
		Part-of-Speech patterns … (Allan and Raghavan 2002)
SIGIR'03	44	Word Sense Disambiguation in IR … (Stokoe et al. 2003)
ECIR'01	17	-none-
ECIR'02	23	-none-
ECIR'03	31	Stemming and Decompounding …
		(Braschler and Ripplinger 2003)
ACL'01	70	… Retrieval: Dumber is Better (Baldwin 2001)
ACL'02	65	-none-
ACL'03	71	Optimizing Story Link Detection … (Farahat et al. 2003)

2 Retrieval Evaluation Metrics

There are many ways of evaluating document retrieval. Most commonly used, and therefore used throughout this paper, are the following three metrics. They assume that the system emits a ranked list of relevant documents.

11pt precision. This is the average precision measured at recall levels of 0%, 10%, 20%, …, 100%. For each recall level, one goes down the ranked list of results until the recall level is reached and then determines the fraction of relevant documents so far. One interpolates between points to reach the particular recall levels: $Prec(Level = r) = \max_{s \geq r} Prec(Level = s)$. Sometimes, a smaller number of recall levels is used, e.g., 3pt precision averaging precision at recall levels of 25%, 50%, and 75%.

Average precision. This is the average precision when measured at all relevant document positions in the ranking. Not retrieved relevant documents are counted with a precision of 0. As an example: the system returns 5 documents. There are 3 relevant documents: at positions 2 and 3, the third one is not retrieved. The average precision is $(1/2 + 2/3 + 0)/3 = 39\%$.

R-precision. This is the precision after R documents are retrieved, where R is the number of relevant documents for the current query.

All three metrics combine precision and recall in one value. They only reach a perfect score of 100% if all relevant documents are at the top of the ranked list. Average precision additionally requires to return only relevant documents.

3 Natural Language Processing in Information Retrieval

3.1 Stopwords

Almost all IR applications remove stopwords (function words, low-content words, very high frequency words) before processing documents and queries. This usually increases system performance. But there are many counter-examples that are handled poorly after stopword removal, e.g.:

1. *To be or not to be*
2. *New Year* celebrations
3. *Will and* Grace
4. *On the* road *again*

(Words in italics are considered stopwords). Adjusting the stopword list to the given task can significantly improve results (Farahat et al. 2003). Creating stopword lists is not generally considered to be NLP, but NLP techniques can help to create specific lists and to deal with examples 1 – 4 above.

3.2 Stemming

Stemming is the task of mapping words to some base form. The two main methods are (1) linguistic/dictionary-based stemming, and (2) Porter-style stemming (Porter 1980). (1) has higher stemming accuracy, but also higher implementation and processing costs and lower coverage. (2) has lower accuracy, but also lower implementation and processing costs and is usually sufficient for IR.

Stemming maps several terms onto one base form, which is then used as a term in the vector space model. This means that, on average, it increases similarities between documents or documents and queries because they have an additional common term after stemming, but not before. This results in an increase in recall, but sacrifices precision.

Stemming has a relatively low processing cost, especially when using Porter-style stemming. It reduces the index size, and it usually slightly improves results, e.g. (Strzalkowski and Vauthey 1992): 0.328 average precision without stemming, 0.356 with stemming. This makes it very attractive for use in IR. However, measuring the effect of stemming on retrieval is not trivial. (Harman 1991) and (Krovetz 1993) both used the CACM collection for their research. But they do not reach the same conclusion regarding Porter stemming: Harman did not find any significant effect, while Krovetz found an improvement from 0.324 to 0.368 (avg. 3pt precision) when using stemming. The effect depends on the particular system investigated, and the queries used for the evaluation.

The positive net benefit of stemming that is found in most investigations is likely to be a superposition of positive and negative cases. Inflectional stemming is mostly beneficial, but there are ambiguous cases in which stemming is questionable (at best). E.g., a user is probably not likely to be looking for “Window” when

entering the query term "Windows" (house part vs. operating system). Other examples of poor inflectional stemming are Doors/Door (music band vs. house part), and Utilities/Utility (energy supply vs. usefulness).

Derivational stemming has mixed effects. It is most likely ok to map *resignation* to *resign*, and *assassination* to *assassin*. But many mappings generated by a simple stemmer are wrong or introduce ambiguities: *expedition* → *expedite*; *importance* → *import*; *organization* → *organ*; etc.

An interesting research question is the automatic detection of when to use stemming in order to avoid overstemming and understemming. Previous research suggests that the effect depends on the baseline system and the data used. Therefore, stemming should be learned and optimized jointly with the IR system.

Character n-grams can be used as a non-NLP alternative to stemming. The character n-grams may span across word boundaries. This makes preprocessing documents simple and language independent, at the cost of increasing the index size.

(Kamps et al. 2003) compared a system using stemming and compound splitting with a system using n-grams. Average precisions for Dutch were 0.4984 (stem/split) vs. 0.4996 (n-grams). For German, they found 0.4840 (stem/split) vs. 0.5005 (n-grams). (Mayfield et al. 2000) even found a more dramatic effect for German: 0.161 (stem) vs. 0.283 (n-grams). However, they did not use compound splitting, which probably explains the difference. They did not find a significant difference for English, French, and Italian. Overall, results of stemming/compound splitting seem to be comparable to using character n-grams. Stemming/splitting is usually preferred since it comes with much smaller memory requirements.

3.3 Part-of-Speech Tagging

Part-of-speech tagging is the task of assigning a syntactic category to each word in a text, thereby resolving some ambiguities. E.g., the tagger decides whether the word *ships* is used as a plural noun or a 3rd person singular present tense verb. A variety of techniques have been used, e.g. statistical (Ratnaparkhi 1996, Brants 2000), memory-based (Daelemans et al. 1996), rule-based (Brill 1992) and many more. The accuracies for small and medium sized tagsets are usually in the middle or high 90s.

(Kraaij and Pohlmann 1996) investigate the "success" of different parts-of-speech for retrieval. They define a "successful term" as a query term that appears in a relevant document. For Dutch, they find that 58% of the successful terms are nouns (including nominal compounds and proper names), 29% are verbs, 13% are adjectives. When looking at the query terms present in the highest number of relevant documents, they find that 84% of these terms are nouns. This shows the higher importance of nouns. And indeed, (Arampatzis et al. 1990) found an improvement when using nouns only for retrieval, compared to using all stemmed words: 0.537 avg. precision (all stemmed words) vs. 0.559 (nouns only). However, the improvement is small (only 4% relative), and it is unclear whether a similar

improvement would be found when using a state-of-the-art system as a baseline.

Instead of making hard decisions and selecting particular parts-of-speech for indexing, one could assign weights depending on the part-of-speech. But we are not aware of a study that used this technique for retrieval.

Another way of using part-of-speech information is separating terms by part-of-speech. Each pair of (stemmed) term and part-of-speech forms one dimension in the vector space model, instead of just the term in the original model. This technique was used in (Farahat et al. 2003), and yielded a 10% improvement for new event detection, but a 4% decrease for link detection. The mixed success is partially due to the fact that sometimes we actually do want different part-of-speech to match. While it is good to differentiate between *building*/Noun and *building*/Verb, it is likely that *finding*/Noun and *finding*/Verb should match.

3.4 Compounds and Statistical Phrases

Compounds and statistical phrases index multitoken units instead of single tokens. The technique used in SMART (Buckley et al. 1993) is to collect pairs of adjacent non-stopwords and then use all pairs with a frequency above some threshold. It is possible to use longer n-grams, but this is expensive because of the large number of longer n-grams. Bigrams already significantly increase the index size, even when pruning by frequency. But they improve avg. precision by around 10% relative, so are usually worth the effort (Salton et al. 1975, Fagan 1997).

In practice, a mix of single-token units and multi-token units is used. Single tokens alone match documents that should not match (e.g. matching *New* in *New York*). Using multi-token units alone adds a very high penalty for slight variations, e.g. documents containing *James T. Kirk* suddenly would not match anymore when the query is *James Kirk*. Adding both single-token units and multi-token units to the document vector alleviates these problems.

It is a research question to detect whether a query is for

- single tokens, not a compound (*York* should not return *New York*)
- single tokens, alone and as a compound (*Nobel* may return *Nobel Prize*)
- a compound, parts may be found separately (*natural language* may return *language*, not *natural*; *wine stores* may return *wine*, not *stores*)
- a compound, not single tokens (*New York* should not return *New* or *York*)

The net benefit of using compounds is positive, and it is likely to be further improved if the cases above can be separated automatically. Making these distinctions is related to determining compositionality of compounds, but it is not the same. Processing needs to be tailored to the retrieval task in order to identify those compounds that are improve retrieval accuracy.

Similar work is done on Chinese, Japanese, and Korean word segmentation for information retrieval. Results are not entirely conclusive but simple bigrams or statistically determined segments seem to be slightly better than dictionary-based segments (Wilkinson 1997, Chen et al. 1997).

Table 2: German compound splitting examples

Compound	**Partition**	**Status**	**(translation)**
Sonnenenergie	Sonne+Energie	ok	(solar energy)
Bauernhaus	Buaer+Haus	ok	(farm house)
Frühstück	Früh+Stück	error	(breakfast)
Niederschlag	Nieder+Schlag	error	(precipitation)
Kernkraftwerk	Kern+Kraft+Werk	?	(nuclear power plant)
Flugzeug	Flug+Zeug	?	(airplane)

3.5 Compound Splitting

Many languages, e.g., Dutch, Finnish, German, Swedish, and many more, form words by concatenating other words in a productive process. Being able to separate compounds should improve retrieval quality. A simple algorithm for compound splitting is to consider all other words found in the lexicon as possible parts. Optionally, one can require a minimum length of parts (e.g. length ≥ 4), allow linking elements (e.g. *-e-*, *-en-*, *-n-*, *-s-* in German), and require that the frequency of each part is larger than the frequency of the compound.

The net benefit of compound splitting is usually positive. (Chen 2002) found 4 – 13% relative improvement for Dutch. However, it is a research question to automatically determine which split is beneficial for retrieval. Table 2 shows a few German examples. Even if a particular compound split is justified from a linguistic perspective, it does not necessarily help in retrieval, e.g., separating *Kinderarbeit* (child labor) into *Kind+Arbeit* retrieves many documents about working parents. As with many of the other NLP techniques, compound splitting needs to be adapted to retrieval.

3.6 Chunking and Shallow Parsing

Chunking and Shallow Parsing aim at separating words in a sentence into basic phrases, e.g. noun phrases or simple verb phrases. A large number of techniques has been tried. The best system in the CoNLL 2000 shared task evaluation for chunking is based on Support Vector Machines and achieves an F-Score of 93.48% (Kudoh and Matsumoto 2000). Chunks are used in the vector space model the same way as n-grams or compounds: both the individual terms as well as the whole chunk are added as separate dimensions to the vector.

Even though state-of-the-art chunkers achieve high accuracies, we are not aware of any investigation that showed improvements over using n-grams when using chunking.

3.7 Head-Modifier Pairs

Head-modifier pairs are based on dependencies between words that can either be derived from standard phrase-based parsing or by using a dependency parser, e.g.

(Tapanainen and Järvinen 2000). This technique has been used by (Strzalkowski et al. 1999a) and (Hull et al. 1996). Word pairs consisting of heads and modifiers are added as new dimensions to the vector space model. While improvements over simple baselines can be achieved, we are not aware of an investigation that shows improvements over the use of simple word-n-grams that are derived without parsing.

Part of the reason for the limited success is the large number of spurious pairs, e.g., the pair *Soviet+president* will also match *former Soviet president*, and ambiguities that are hard to resolve. Usually, only a subset of pairs in three-word phrases is useful for retrieval:

- *natural language processing*
  → *nat+lang* (ok); *lang+proc* (ok); *nat+proc*(error)
- *incremental information processing*
  → *incr+info* (error); *info+proc* (ok); *incr+proc* (ok)
- *executive vice president*
  → *exec+vice* (error); *vice+pres* (ok); *exec+pres* (?)
- *insider trading case*
  → *ins+trad* (ok); *trad+case* (error); *ins+case* (?)

Automatically identifying the correct (or useful) pairs is a hard task. Pair frequency is used, but the usefulness for retrieval is limited.

3.8 Word Sense Disambiguation

Word sense disambiguation is the task of distinguishing the correct sense of a word in context. When used for information retrieval, terms are replaced by their senses in the document vector. (Voorhees 1993) found that 3pt precision decreases by 5 – 17% on English data using this method. (Volk et al. 2002) performed similar experiments on English and German data and report mixed results. Using EuroWordnet (Vossen 1998), average precision decreases by 23% for English, and 9% for German. However, using MeSH, they found a 7% improvement for English and 12% improvement for German.

A major difference between EuroWordnet and MeSH (Medical Subject Heading) is that MeSH is specialized to the medical domain, which also was the domain of the data (Volk et al. 2002) were working on. These results suggest that it pays off to adapt the ontology the targeted domain.

A factor for the negative results of using a general-purpose ontology is that word sense disambiguation for short queries is hard because of the missing context, and it is not necessary for long queries because the other terms narrow down the search anyways.

Table 3: Effect of query length (Strzalkowski *et al.*, 1999)

data set	**TREC-2**		**TREC-3**		**TREC-4**	
query length	**115 terms**		**70 terms**		**10 terms**	
Runs	base	NL	base	NL	base	NL
Avg. Prec.	0.2224	0.3111	0.2271	0.2735	0.2082	0.2272
		+40%		+20%		+9%

Table 4: Effect of query length comparing indexing stems with indexing stems and noun phrases (Strzalkowski *et al.*, 1999)

query length	**long**		**short**	
Runs	stems	stems+phrases	stems	stems+phrases
Avg. Prec.	0.2626	+25%	0.1682	+7%

4 The Query Length Effect

Query length has an effect on the impact of natural language processing. (Strzalkowski et al. 1999b) found the correlation shown in table 3, which compares a baseline system with a system that uses various NLP techniques. In general, the impact of NLP is larger for longer queries. This seems to have a simple explanation: short queries lack a lot of the context information that is used in NLP. The same authors also confirmed this effect when focusing more on the individual components. Table 4 compares results on long queries (the "narrative" part of TREC queries) and short queries (the "description" part) when using stems only for indexing with using stems plus noun phrases.

5 The TREC NLP Track

In 1996, TREC organized a separate Natural Language Processing track. The goals were to see if available NLP techniques have an impact on IR, and if NLP has value in specific situations even if it is not advantageous in general situations. The chosen task was ad hoc retrieval with queries that a researcher might give in a library environment. There were two run conditions: one was completely automatic, the other one allowed the manual modification of queries.

Table 5: Best retrieval results using NLP at the TREC-5 NLP track

	baseline	**System A**	**System B**	**System C**
avg. prec.	0.1771	0.2280	0.2220	0.2010
% change	–	+29	+25	+13

Table 5 summarizes the best results of the three systems using NLP in the automatic run. The SMART system (Salton 1988) was used as a baseline. It used a TFIDF weighted vector space model, and all adjacent non-stopword bigrams with frequencies above some threshold as phrases. All systems achieved an improvement over the baseline.

The systems used a variety of methods in addition to their own baseline (non-NLP) methods:

System A (Hull et al. 1996): Head/modifier recognition with a finite state parser (Grefenstette 1996), using the following relations: subject–verb, verb–direct object, verb–adjunct, noun–noun, adjective–noun, adverb–verb; words in pairs were stemmed, used independent of the order in which they occurred, no stopwords were allowed in pairs; additionally, bigrams extracted by SMART were used.

System B (Strzalkowski et al. 1996): morphological stemming; shallow parsing for extracting head/modifier pairs using TTP (Strzalkowski and Vauthey 1992); normalization and order-independent representation of pairs; recognition of proper names.

System C (Tong et al. 1996): noun-noun and adjective-noun bigrams, selected based on bigram and single-term frequencies; the pairs depend on the order they occur in the text; single terms are not used if a bigram is recognized as a unit; for detected NPs, the individual terms, head-modifier pairs, as well as the full NP were added to the document vectors.

Comparison of the different NLP techniques comes with two grains of salt. The underlying base systems are too different to allow conclusions about the NLP methods used. Furthermore, all the improvements over the baseline system are significant, but other non-NLP techniques are much more successful: full-text query expansion achieves more than 40% improvement over the baseline. It is unknown yet whether the techniques can be combined to yield an even bigger improvement.

NLP techniques used at the TREC-5 NLP track improved retrieval, but they were not a breakthrough and came with very high processing costs. The evaluation used relatively long queries (a sentence or a paragraph). The benefit for shorter queries is expected to be even smaller.

6 Conclusions

Overall, we see a modest benefit of NLP techniques in IR. However, this benefit comes with large computational costs, and non-NLP techniques tend to yield greater improvements.

Small positive effects often seem to be a superposition of positive and negative effects. Automatically separating positive and negative instances would help a lot. Such a separation would require a joint focus on NLP and retrieval, not to build an NLP system and then apply it to retrieval more or less as a black box.

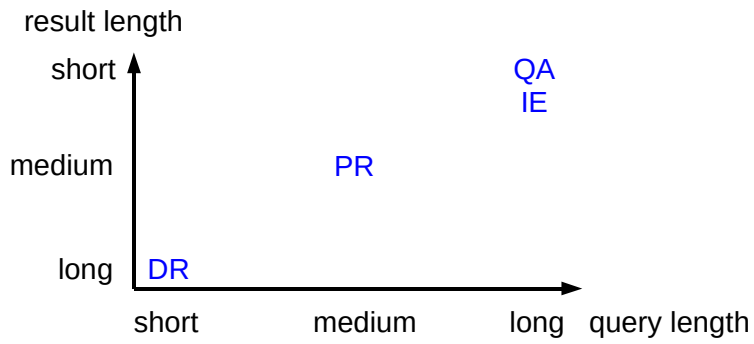

Figure 1: Classification of **D**ocument **R**etrieval, **P**assage **R**etrieval, **Q**uestion **A**nswering, and **I**nformation **R**etrieval according to query length and result length

Processing techniques that were developed directly for information retrieval tend to be more successful than techniques that were developed independently based on linguistic. The Porter stemming algorithm is very fast and tailored for normalization in retrieval systems. It is successful despite its linguistic flaws, and variants of it can be found for many languages (Porter 2002). Similarly, statistical "phrases" as investigated in the retrieval community collide with linguistic knowledge. But they are optimized for the retrieval task and are therefore successful. Word sense disambiguation wasn't designed with retrieval in mind. And in general it does not help retrieval (it even decreases quality). However, we see improvements when the set of senses if optimized for the domain, like MeSH for medical texts.

An interesting research question is whether other NLP techniques like part-of-speech tagging, chunking, or parsing can be tailored to the retrieval task or particular domains. They should do enough processing to benefit retrieval, not more and not less.

We noted that the success of NLP techniques depends on the length of the queries: the longer the queries, the bigger the benefit of NLP. If we look at tasks other than document retrieval we see that their query lengths vary, suggesting that the tasks with longer query lengths are better suited for NLP. Similarly, different tasks produce different result lengths. Shorter result lengths mean that the system has fewer opportunities to deliver the right information: if the system can return a whole document then there is a higher chance that the requested information is somewhere in there than when it only is allowed to return one sentence. Therefore, we hypothesize that shorter results require better processing in order to detect syntactic and semantic variants. Taking together the observation about query lengths and the hypothesis about result length, applications other than document retrieval are much better suited for NLP. Figure 1 classifies document retrieval, passage retrieval, question answering, and information extraction along the two dimensions, showing that question answering and information extraction are very good candi-

dates for NLP techniques.

Acknowledgments

I would like to thank Hiyan Alshawi, Francine Chen, Ayman Farahat, Alex Franz, Marius Pasca, Jay Ponte, and Amit Singhal for valuable discussions on the topics covered and for help in preparing this paper.

References

Allan, J. and Raghavan, H.(2002), Using part-of-speech patterns to reduce query ambiguity, In *Proceedings of SIGIR-02*, Tampere, Finland.

Arampatzis, A., van der Weide, Th.P., Koster, C.H.A. and van Bommel, P.(1990), *Text Filtering using Linguistically-motivated Indexing Terms*, Technical Report CSI-R9901, Computing Science Institute, University of Nijmegen, Nijmegen, The Netherlands.

Baldwin, T.(2001), Low-cost, High-performance Translation Retrieval: Dumber is Better, In *Proceedings of ACL-01*, Toulouse, France.

Brants, T.(2000), TnT – A Statistical Part-of-Speech Tagger, In *Proceedings of the Sixth Conference on Applied Natural Language Processing ANLP-2000*, Seattle, WA.

Braschler, M. and Ripplinger, B.(2003), Stemming and Decompounding for German Text Retrieval, In *Proceedings of ECIR-03*, Pisa, Italy.

Brill, E.(1992), A simple rule-based part of speech tagger, In *Proceedings of ANLP-92*, pages 152–155, Trento, Italy.

Buckley, C., Allan, J. and Salton, G.(1993), Automatic Routing and Ad-hoc Retrieval Using SMART: TREC 2, In *Proceedings of TREC-2*, Gaithersburg, Maryland, USA.

Chen, A., He, J., Xu, L., Gey, F.C. and Meggs J.(1997), Chinese text retrieval without using a dictionary, In *Proceedings of SIGIR-97*, pages 42–49, Philadelphia, PA, USA.

Chen, A.(2002), Cross-Language Retrieval Experiments at CLEF 2002, In *Proceedings of CLEF-2002*, Rome, Italy.

Daelemans, W., Zavrel, J., Berck, J. and Gillis S.(1996), MBT: A Memory-Based Part of Speech Tagger-Generator, In *Proceedings of the Workshop on Very Large Corpora*, Copenhagen, Denmark.

Fagan, J.L.(1997), An Examination of Syntactic and Non-Syntactic Methods, In *Proceedings of SIGIR-97*, pages 91–111, Philadelphia, PA, USA.

Farahat, A., Chen, F. and Brants, T.(2003), Optimizing Story Link Detection is not Equivalent to Optimizing New Event Detection, In *Proceedings of ACL-03*, Sapporo, Japan.

Grefenstette, G.(1996), Light Parsing as Finite State Filtering, In *Proceedings of the Workshop on Extended finite state models of language, ECAI'96*, Budapest, Hungary.

Harman, D.(1991), How effective is suffixing, *JASIS*, **42**(1):7–15.

Hull, D.A., Grefenstette, G., Schulze, B.M., Gaussier, E., Schütze, H. and Pedersen, J.O.(1996), Xerox TREC-5 Site Report: Routing, Filtering, NLP, and Spanish Tracks, In *Proceedings of TREC-5*, Gaithersburg, Maryland, USA.

Kamps, J., Monz, C., de Rijke, M. and Sigurbjörnsson, B.(2003) The University of Amsterdam at CLEF-2003, In *Results of the CLEF 2003 Cross-Language System Evaluation Campaign*, pages 71–78, Trondheim, Norway.

Kraaij, W. and Pohlmann, R.(1996), Viewing stemming as recall enhancement, In *Proceedings of SIGIR-96*, pages 40–48, Zürich, Switzerland.

Krovetz, R.(1993), Viewing Morphology as an Inference Process, In *Proceedings of SIGIR-93*, pages 191–202, Pittsburgh, PA, USA.

Kudoh, T. and Matsumoto, Y.(2000), Use of Support Vector Learning for Chunk Identification, In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

Larkey, L.S., Ballesteros, L. and Connell, M.E.(2002), Improving stemming for Arabic information retrieval, In *Proceedings of SIGIR-02*, Tampere, Finland.

Mayfield, J., McNamee, P. and Piatko, C.(2000), The JHU/APL HAIRCUT System at TREC-8, In *Proceedings of TREC-8*, Gaithersburg, Maryland, USA.

Porter, M.(1980), An algorithm for suffix stripping, *Program*, **14**(3):130–137.

Porter, M.(2002), *Snowball*, `http://snowball.tartarus.org/`.

Ratnaparkhi, A.(1996), A Maximum Entropy Model for Part-of-Speech Tagging, In *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP-96*, Philadelphia, PA.

Salton, G., Yang, C.S. and Yu, C.T.(1975), A Theory of Term Importance in Automatic Indexing, *Journal of the American Society for Information Science*, **26**(1):33–44.

Salton, G.(1988), *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley.

Stokoe, C., Oakes, M.P. and Tait, J.(2003), Word sense disambiguation in information retrieval revisited, In *Proceedings of SIGIR-03*, Toronto, Canada.

Strzalkowski, T. and Vauthey, B.(1992), Information Retrieval Using Robust Natural Language Processing, In *Proceedings of ACL-92*, pages 104–111, Newark, Deleware, USA.

Strzalkowski, T., Guthrie, L., Karlgren, J., Leistensnider, J., Lin, F., Perez-Carballo, J., Straszheim, T., Wang, J. and Wilding, J.(1996), Natural Language Information Retrieval: TREC-5 Report, In *Proceedings of TREC-5*, Gaithersburg, Maryland, USA.

Strzalkowski, T., Perez-Carballo, J., Karlgren, J., Hulth, A., Tapanainen, P. and Lahtinen, T.(1999a), Natural Language Information Retrieval: TREC-8 Report, In *Proceedings of TREC-8*, Gaithersburg, Maryland, USA.

Strzalkowski, T., Lin, F., Wang, J. and Perez-Carballo, J.(1999b), Evaluating Natural Language Processing Techniques in Information Retrieval, In Tomek Strzalkowski (ed.), *Natural Language Information Retrieval*. Kluwer Academic Publishers, Dordrecht.

Tapanainen, P. and Järvinen, T.(2000), A non-projective dependency parser, In

Proceedings of ANLP-97, pages 64–71, Washington, DC, USA.

Tong, X., Zhai, C., Milic-Frayling, N. and Evans, D.A.(1996), Evaluation of Syntactic Phrase Indexing – CLARIT NLP Track Report, In *Proceedings of TREC-5*, Gaithersburg, Maryland, USA.

Volk, M., Ripplinger, B., Vintar, Š, Buitelaar, P., Raileanu, D. and Sacaleanu B.(2002), Semantic Annotation for Concept-Based Cross-Language Medical Information Retrieval, *International Journal of Medical Informatics*, **67**(1-3).

Voorhees, E.M.(1993), Using WordNet to Disambiguate Word Senses for Text Retrieval, In *Proceedings of SIGIR-93*, pages 171–180, Pittsburgh, PA, USA.

Vossen, P.(1998), *EuroWordNet: a multilingual database with lexical semantic networks*, Kluwer Academic Publishers, Norwell, MA, USA.

Wilkinson, R.(1997), Chinese Document Retrieval at TREC-6, In *Proceedings of TREC-6*, Gaithersburg, Maryland, USA.