Document Processing with LinkIT

David K. Evans, Judith L. Klavans and Nina Wacholder
Columbia University
Department of Computer Science
and
Center for Research on Information Acess
500 W. 120th Street
New York, NY, 10027, USA
{devans, klavans, nina}@cs.columbia.edu

Abstract

We present a linguistically-motivated technique for the recognition and grouping of simplex noun phrases (SNPs) called LinkIT. Our system has two key features: (1) we efficiently gather minimal NPs, i.e. SNPs, as precisely and linguistically defined and motivated in our paper ; (2) we apply a refined set of post-processing rules to these SNPs to link them within a document. The identification of SNPs is performed using a finite state machine compiled from a regular expression grammar, and the process of ranking the candidate significant topics uses frequency information that is gathered in a single pass through the document. We evaluated the NP identification component of LinkIT and found that it outperformed other NP chunkers in precision and recall. The system is currently used in several applications which are described, such as web page characterization and multi-document summarization.

1 Introduction

We present a linguistically-motivated technique for the recognition and grouping of simplex noun phrases (SNPs) called LinkIT[1]; our tool has been used in a variety of text analysis tasks described in the paper. Like other NP identifiers, we use a part of speech (POS) tagger and a regular expression grammar. Our system differs from other approaches in two respects: (1) we focus on the efficient gathering of minimal NPs, i.e. SNPs, as precisely and linguistically defined and motivated in our paper, (2) we apply a refined set of post-processing rules to these SNPs to rank and link them within a document.

An NP is a maximal NP with a common or proper noun as its head, where the SNP may include premodifiers such as determiners and possessives but not post-nominal constituents such as prepositions or relativizers. Examples of SNPs are **asbestos fiber** and **9.8 billion Kent cigarettes**. SNPs can be contrasted with complex NPs such as **9.8 billion Kent cigarettes with the filters** where the head of the NP is followed by a preposition, or **9.8 billion Kent cigarettes sold by the company,** where the head is followed by a participial verb (Wacholder, 1998).

With LinkIT, we produce a representation of the document that goes beyond just looking at the lexical forms of the words in the document. By identifying and linking SNPs in the document and doing some simple analysis on the verbs in the document, we can identify the major entities and concepts in the document, and can ignore other entities in the document which are simply low frequency references (Klavans & Wacholder, 1998). We hypothesize that the SNPs in a document provide a good representation of the content of the document.

---

[1] LinkIT may be freely licensed for research purposes. Information can be found at http://www.columbia.edu/cu/cria/LinkIT/ or contact the authors for more information.

1.1 System Description

The identification of SNPs is performed quickly using a finite state machine compiled from a regular expression grammar, and the process of ranking the candidate significant topics uses frequency information that can be gathered in one pass through the document. LinkIT can process approximately 4.11 MB tagged text/sec. LinkIT uses a part of speech tagger available from MITRE in the Alembic Utilities, a freely available set of NLP tools (Aberdeen *et al.* 1995) for tokenization and tagging. The POS tagged text is input to LinkIT and is parsed sequentially by a finite state machine that extracts SNPs and other syntactic elements. If the extracted element is an SNP, it is compared to previously parsed SNPs with respect to modifiers, heads, and other properties. If the element is not an SNP, LinkIT records it and performs element-specific processing. After all of the SNPs in the document have been extracted, the SNPs are sorted by similarity of the lexical form of the head. The groups of SNPs are then ranked using the frequency of the head as an approximation of their relative significance within the document (Wacholder 1998).

1.2 Overview of processing

The main module has access to a list of text units, identified by type and identified by the rule used for identification of the unit. If the unit is an SNP, information about the SNP is extracted from the marked-up text, such as part of speech and role information. An entry is created for the SNP in a list of SNPs for the entire document, and the SNP is checked for links to previous NPs in the document. If the unit is not an SNP, LinkIT performs processing appropriate to that type of unit.

To determine NP boundaries, LinkIT uses a finite-state lexer built from a small hand-crafted regular expression grammar. The input to the lexer is part of speech tagged text. The lexer contains regular expressions to identify SNPs, sentence boundaries, paragraph boundaries, dates, and simple verb phrases.

The lexer takes the input text and matches it to one of the input patterns, returning the text of the largest match found. When matching to the set of regular expressions, preference is given to expressions that minimize the amount of input that is unable to match to the regular expression before the start of the matched text. For those expressions that skip the same amount of text between the previous and current match, longer matches are preferred. The text that matched the final regular expressions, as well as the text between the last matched text and the current matched text is returned to the LinkIT main module. The lexer also sets variables that indicate which regular expression was used, what sentence and paragraph the match was in, and the number of the first and last tokens in the matched text.

Once all of the SNPs for the document have been extracted, they are grouped based on the similarity of the lexical form of the head. Two SNPs are placed in the same group if they have the same head, ignoring differences in plurality or case. These SNP groups are then ranked in order of their relative significance as estimated by the frequency of the number of SNPs in the group. The resulting list can be sorted and output in a variety of ways. Optionally, for each word that is in the document, if it is part of an SNP, LinkIT can output a list of the SNPs that the word is in, broken down by occurrence of the word as the head of an SNP, and as a modifier in an SNP.

1.3 SNP Processing

LinkIT creates a data structure to store information associated with each SNP returned by the lexer. A list of the words in the SNP is created, and for each word in the SNP, LinkIT extracts the part of speech tag, and any other special feature that might be associated with that word,

based either on information provided by Alembic or based on LinkIT's own processing. For named entities, Alembic may assign the feature POST or a TITLE feature: POST is assigned to words that indicates a job position, such as general or secretary. TITLE is assigned to human titles, such as Dr. or Mr. A named entity is a sequence of words that refer to a location, place, or organization, as tagged by the Alembic Utilities. The list of words and their associated information are stored in the SNP structure.

In order to recognize expressions such as "fast and cheap," if the previous unit returned by the lexer consisted of an adjective followed by a coordinating conjunction, LinkIT checks for intervening text between the previous unit and the current SNP. If there is no intervening text, the adjective and coordinating conjunction are attached to the beginning of the current SNP, and processing continues as normal. If there is some intervening text, the adjective and coordinating conjunction variable is cleared, and the current SNP is not modified.

If the head of the current SNP is an empty head (i.e., a noun whose head makes a relatively small contribution to the semantics of the SNP (Klavans et al. 1992)), and the only text between the current SNP and the previous SNP is the word "of", the data associated with the previous and current SNP is adjusted to indicate that the SNPs may be part of a larger NP that includes a prepositional phrase headed by "of". To support identification of empty head nouns, we have implemented a dictionary module for LinkIT.

1.4 Special Processing

As mentioned previously, LinkIT performs some special processing for certain units returned from the lexer. Specific action is taken for each of the following cases: possessive "'s", title, sentence boundary, comma, new paragraph, and the sequence of an adjective followed by a coordinating conjunction. In each of these cases, LinkIT updates state information pertinent to those returned units. There are six different cases in which LinkIT performs some special processing, two of which – sentence boundaries and new paragraphs – are related to the form of the document:

- Sentence boundary. The Alembic utilities detect sentence boundaries using a statistical method. The lexer returns a sentence boundary that has been tagged in the input file, after making corrections in a few cases where the tagger makes consistent errors. LinkIT updates its count of the number of sentences it has seen on receipt of a sentence boundary unit. The sentence count is used to determine which sentence an SNP is in when it is returned by the lexer.
- New paragraph. When the lexer detects two or more carriage returns in a row, it returns a new paragraph unit. LinkIT simply updates its count of the number of paragraphs in the document, similar to recognition of a new sentence unit.

The other four cases – titles, commas, adjective followed by coordinating conjunction, and the possesive "'s" – are more closely related to the content of the document:

- Titles (e.g. Mr., Dr., etc.) Alembic Utility marks titles, which are returned by the lexer to the main module as independent units. When the LinkIT main module receives a title, it requests the next SNP from the lexer, attaches the title to the beginning of the next NP, and marks that NP as likely to be a human entity. It would also have been possible to include the title words in the NP rules; however, by creating rules that allow a special title tag in the phrase, the size of the resulting finite state machine would have been increased.

- Comma. When the lexer returns a comma, LinkIT checks to see if the previous two SNPs are potentially in apposition. For example, in “Kim Smith, the first-prize winner, congratulated her competitors.”, “Kim Smith” and “the first-prize winner” are in apposition. To check for appositives, LinkIT keeps a stack of the past three units. If units in the stack are an SNP, a comma, and an SNP, in that order, and, if the current unit is a comma, the two previous SNPs might be in apposition. A comma is placed on the stack only if there are less than three units on the stack, and there is no intervening text between the previous SNP and the current comma. If there is text between the current comma and the previous NP, the entire stack is cleared. If a possible apposition is found, that relation is made between the two SNPs, and the stack is reset to contain just one NP and one comma, which represent the two previous appositive SNPs.
- Adjective followed by coordinating conjunction. Another case that LinkIT handles is coordination of adjectives, as in "fast and cheap machines". An adjective followed by a coordinating conjunction is returned as an adjective-coordinating-conjunction unit. A variable is set that retains the information for the returned unit, and if the next unit is an NP with no interceding words, the adjective and coordinating conjunction are added to the beginning of the next SNP. Similar to possessive "'s" modification, this is done with a variable that is set, and a check in the main LinkIT module.
- Possessive "'s". LinkIT treats phrases with a possessive "'s", as in "Boston's Dana Farber Cancer Institute", as three separate units. The first is “Boston”, the second is a possessive "'s", and the third is “Dana Farber Cancer Institute”. LinkIT considers this relationship to be similar to "The Dana Farber Cancer Institute of Boston." When the LinkIT main module receives a possessive "'s" from the lexer, it sets the first NP as a possible head of the second NP, and the second NP as a possible modifier of the first NP. At the point where a possessive "'s" is returned from the lexer, LinkIT does not know what the second NP will be, so a variable is set in the lexer and the main module checks for that variable.

1.5 Noun Phrase Linking

Finally, lexical relations are made between the words in the current NP to the words previously seen in the document. For each modifier in the current NP, we check for other occurrences of that word within the document. Efficient search is supported using a hash table. Each word is reduced to its singular form; irregular words are reduced to their correct form using a dictionary. Case is ignored in the comparison. If there has been a previous occurrence of the word, a link is added from the word to the previous word. For the head of the NP, LinkIT searches for similar words, but also assigns a group number to the NP based on what is matched. If no previous occurrences of the word exist, then a new group is formed, and the NP is assigned the next sequential number for a group. When a match to a head of another NP is found, the NP is assigned the group number of the matching head, and a previous occurrence relation is made from the head of the current NP to the matched head. If the matched word was not the head of its NP, then a new group is created as in the case above when a match is not found.

2 Applications

With the proliferation of information available via the Internet, it has become increasingly common for natural language processing techniques to augment statistical based methods for information retrieval, document processing, and document browsing. Advanced search engines now use phrases and simple noun phrase identification to help improve the quality of searches (Evans, DA & Zhang 1996). Efficient natural language analysis applications, such as LinkIT, make it possible to apply NL techniques in areas that have traditionally eschewed such approaches due to processing constraints.

There are many possible applications of having such a rich representation of the "aboutness" of the document. The LinkIT system is currently used by three projects at Columbia University. Using the LinkIT output over a collection of documents, a topic detection and tracking system has been built (Negrilla 1998). The system works by looking at the LinkIT output for each document, detecting similarities and differences, and tracking how that topic, as represented by the SNPs, changes over time. LinkIT has also been used in a paragraph level similarity detection component of a multiple document summarization system (Hatzivassiloglou *et al.,* 1999; McKeown *et al.* 1999). The output from LinkIT could also be used as the input for a term variant finder, such as FASTR (Jacquemin 1999). It would be possible to use LinkIT on a selection of documents that has been shown likely to be relevant by some other method, in order to make more fine distinctions between the documents. This could be used as a second stage to information retrieval to help a user visualize the content of the returned documents, or as a browsing tool for a static collection of documents in a digital library.

In our current research, we are exploring the hypothesis that, compared to just looking at the words in the document without regard to their syntactic role, we should be able to more accurately match documents to user queries. We believe that we will not be misled by spurious hits caused by a document that mentions, but does not actually focus on, a certain topic. We have done a pilot study where we used LinkIT output as the basis for an index of a document collection and have shown that retrieval performance using the LinkIT output files is comparable to retrieval performance when using the entire text of the document, even though the base document length has been reduced by approximately 30%. We believe this is due to the information bearing content of the SNPs (Wacholder *et al*., in progress.).

3 Evaluation

3.1 Experimental Design

We designed an experiment to test LinkIT's performance at NP identification as compared to other NP identifiers. The task consists of identifying the NPs in a test collection of 14 documents, with an evaluation of the results. In this experiment, LinkIT's additional capabilities of lexical chain identification and noun group ranking are not evaluated.

3.2 The Data Set

The data set consisted of NPs from documents wsj_0300 - wsj_0314 of the Penn Wall Street Journal Treebank (Marcus, Santorini, & Marcinkiewicz 1993). The noun phrases were extracted from the parsed data files of the Treebank. An automatic process was used to extract the smallest unit marked as an NP in the Treebank, and each resulting file was then examined to verify the correctness of the NPs extracted. In certain cases, complex noun phrases were manually split into smaller units; for example, NPs that contained a conjunction were split if we judged that there was ambiguity regarding the applicability of the head of the NP to each constituent of the phrase.

doc	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
ins	5	21	3	3	0	1	16	2	8	21	2	2	3	2	28
tot	264	358	54	38	142	344	63	43	114	168	46	55	20	46	282

Table 1: Number of manual corrections and total number of NPs per file

In **Table 1**, the **doc** row indicates the document number, while the **ins** row indicates how many NPs needed to be inserted manually. The **tot** row indicates the total number of NPs in that document. There was a total of 2037 NPs in the hand-corrected test set which we treated as the gold standard for this evaluation.

3.3 Noun Phrase judgments

Each system was tested over the plain text files corresponding to the parsed data files for the test noun phrases. For the initial evaluation, we compared output of the LinkIT system to output from the text chunking tool of (Ramshaw & Marcus, 1995). The Penn chunker applies the transformation-based learning technique (Brill 1993) to the chunking task.

A human judge rated the acceptability of each NP in the system's output by assigning it to one of six categories representing the relationship between the NP in the gold standard set and the NP in the system output. The following judgments were assigned:

1. **Correct -** A perfect match of the two NPs, i.e., both are exactly the same without respect to punctuation or other artifacts of the specific mark-up process. For example, for the gold standard NP "Battle-tested Japanese industrial managers" the identical NP "Battle-tested Japanese industrial managers" would be labeled correct, while the NP "Japanese industrial managers" would not.
2. **Missing -** A NP in the gold standard is completely missing from the test set. For example, if the NP "loose work habits" is in the gold standard but does not exist at all in the set of NPs output by the test system, the NP is labeled as missing.
3. **Under-generated -** A NP in the system output partially matches a NP in the gold standard set, but the NP output by the system is under-generated, i.e., the words in the NP in the test set are a proper subset of the words in the gold standard NP. For example, for the gold standard NP "congressional elections" the NP "elections" would be labeled as under-generated.
4. **Over-generated -** The test set NP contains more words than the gold standard NP, i.e. the words in the gold standard NP are a proper subset of the words in the NP in the test set. For the gold standard NP "a presumption" the NP "a presumption some" would be labeled as over-generated.
5. **Mismatch -** There is some overlap between the two NPs but neither is a proper subset of the other. In this case the test set NP contains some word(s) not in the gold standard NP and the gold standard NP contains some word(s) not in the test set NP.
6. **False positive -** A NP is not in the gold standard set at all - it is a false positive. For example, the NP "egregiously" was not in the gold standard, and was judged to be a false positive.

The number of NP judgements for each category per system over the transformed evaluations can be seen in **Table 2** below.

System	Correct	Mismatch	Missing	Under-generated	Over-generated	False Positive
LinkIT	1689	6	45	329	94	12
Chunker	1368	8	245	69	339	72

Table 2: Individual category results per system

Table 2 shows that the distribution of the system's errors is different across judgement categories. LinkIT tended to produce NPs that were under-generated, while UPenn's Chunker tended to over-generate NPs. This is probably indicative of the different underlying approach and methodology of the two systems. These differences may be the responsible for some perplexing results in the evaluation, as discussed in Section 3.5.

3.4 NP Evaluation Results

Two forms of results are reported. First, the raw results that come from a straightforward analysis of the human judgment evaluations are collected. Due to differences in what the programs identify as NPs in the most simple case, the raw results were transformed to try to normalize performance on simple NPs. For example, some systems might not report pronouns as NPs since they are high-frequency, low-content words. By changing evaluation labels, we aimed to reduce the effect on evaluation results of the types of NPs identified by each system. Transformations were performed to: (1) change NPs that had been judged as under-generated to a complete match if they were only missing certain words in the first position (specified in the first column of **Table 3**), and (2) change a judgement that a NP was completely missing from the system's output to a complete match if the missing phrase was one of the ones listed in the second column of **Table 3**. The effect these transformations had on the results can be seen in **Table 3**, which tabulates the results over both the raw and transformed evaluations.

Allowable missing words in first position	Allowable omissions
its, the, a, an, this, some, their, his, that, these, $	itself, it, he, we, there, they, I, this, some, that, them, those, us, she, you

Table 3: Transformations made to raw results

The results are summarized for evaluations using the raw results and the transformed results in **Table 4** below. LinkIT appears to perform better over this data set than the UPenn Chunker. However, we are not fully confident that the comparisons are precise.

System	Raw Results		Transformed Results	
	Precision	Recall	Precision	Recall
LinkIT	76%	78%	79%	83%
UPenn Chunker	72%	65%	74%	67%

Table 4: Recall and Precision per system for NP identification

3.5 NP Identification—Comparison of LinkIT and the UPenn Chunker

The evaluation of NP identification is a difficult task since definitions of NPs vary. In this particular evaluation we defined six different classes for characterizing the relationship between an NP in the test set and an NP in the evaluation set. However, because we are forced to assign relationships between NPs to one of these six categories, we lose information.

The UPenn Chunker did not appear to perform as well as LinkIT in the test reported in this paper. LinkIT's precision was 79% and the recall 83%, in comparison to 67% recall and 74% precision for the UPenn Chunker. However, Ramshaw and Marcus report a recall and precision of 93% for base NP chunks trained on a much larger test set (950K words). We can only conclude that the discrepancy is due to the difference in what counts as an NP; we plan to investigate this problem further.

For this initial evaluation, we used the default bigram setting for the UPenn Chunker, which may have implications for Table 4. We believe that the settings would obtain optimal output for the Wall Street Journal data set, of which a subset was used for testing in this experiment, under the assumption that the data files trained over the Wall Street Journal corpus will give better results than files trained over the Brown or other corpora.

The UPenn Chunker was the best at recognizing long NPs. This resulted in some problems though; looking at Table 4 we see that the UPenn Chunker over-generated 339 NPs, 245 more than LinkIT. Due to the particular methodology of this implementation, this resulted in penalizing the UPenn Chunker. However, if NPs were judged solely on their grammaticality, many of the NPs that were categorized as over-generated would be acceptable, since the two sequential SNPs are actually part of a larger grammatical NP. It is not the case, however, that only two parts of a grammatical NP were joined; there were also many cases where a larger NP was identified that was nonsensical. For example, phrases like "Mexico's restrictive investment regulations" were identified as NPs when they occurred as the two sequential NPs "Mexico" and "restrictive investment regulations" in the test set. On the other hand, interesting cases were found such as examples of a noun followed by punctuation and then a word from a new sentence, such as the two word phrase 'unfamiliarity. "Because'.

LinkIT consistently made corresponding mistakes in the under-generation category. This is due to the design of LinkIT, where we intentionally decided to focus on Simple Noun Phrases in a document. In many of the test NPs, LinkIT identified two SNPs that together comprised the entire NP. It should be noted, however, that LinkIT does retain information on the links between SNPs, and in cases such as possessive modification and apposition those links are recorded. For example, a noun phrase like "the Secretary of the Health Department" would be split into "the Secretary" and "Health Department" but there would be a link relating the two. While we could have generated a different form of the output to join these sorts of noun phrases, we did not want inherently bias the results and so ran LinkIT under its default settings. Unlike with UPenn's Chunker, there are very few cases when LinkIT will split a larger NP into two smaller SNPs of which one of them, or both, is ungrammatical. This is again due to the linguistic decisions underlying the LinkIT system.

3.6 NP Identification—Comparison of LinkIT and Arizona Noun Phraser (AZNP)

Different tasks call for different approaches to natural language processing. We were interested at looking at the performance of tools targeted for precision tasks such indexing. At the same time, we also are interested in tasks such as Information Retrieval (IR) where words that are deemed to be "low content" are often ignored in favor of more higher content words deemed more discriminating. In IR systems that integrate some natural language properties, a combined approach may be needed. For example, in search engines the differences between the phrases "a penny" and "the penny" is likely to be insignificant. In contrast, for systems that require language understanding, the distinction between the phrases "two apples" and "no apples" could well be important. To look at how one system targeted for an IR application performed over this task, we

performed an evaluation on the Arizona Noun Phraser (Tolle & Chen 2000; Tolle 1997). It must be stressed that the Arizona Noun Phraser is targeted for an IR task, and as such employs a definition of NPs that is more suited to that domain. However, bearing this and the stringent nature of our evaluation in mind, the Arizona Noun Phraser was able to achieve an impressive recall of 61% and precision of 66%.

In the case of the Arizona Noun Phraser, many NPs tested fell into the mismatched NP category, when a more expressive set of relationships might not have penalized it. For example, for the two sequential NPs "a man" and "extraordinary qualities", the Arizona Noun Phraser generated the NP "man with extraordinary qualities". Had it generated the NP "a man with extraordinary qualities" it could be assigned to the over-generation category twice. Since the Arizona Noun Phraser did not include the "a", we were forced to assign the NP "man" to the mismatch category since it contained the "a" from the NP "a man" and the "extraordinary qualities" NP from the following noun phrase.

3.7 Further Evalution

The evaluation performed in this paper only targeted one aspect of the LinkIT system: NP identification. While that is a central aspect of the system, we did not perform an evaluation of the lexical linking and noun phrase group ranking features of the system. While these features are integral to the usage of LinkIT for certain projects, it is difficult to design an evaluation due to the complexity of creating an evaluation metric for these tasks. In the future we would like to evaluate these components of the system in a task-based evaluation.

4 Conclusion

In this paper we have shown that LinkIT outperforms other tools at the task of NP identification. The LinkIT system was presented and described, along with a sample of applications that have used LinkIT as a component.

5 Acknowledgements

This work was supported by NSF Grant IRI-97-12069, as part of the Information and Data Management Workshop (http://www.cs.pitt.edu/~panos/idm98/) and also by NSF Grant CDA-97-53054.

We would like to thank Kristin M. Tolle and Dr. Hsinchun Chen for their help and for the use of the Arizona Noun Phraser, and for important discussion of the results. We would also like to thank University of Pennsylvania for making their tagger and chunker publicly available.

6 References

Aberdeen, John, John Burger, David Day, Lynette Hirschman, Patricia Robinson and Marc Vilain (1995). MITRE: Description of the Alembic System as Used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

Brill, Eric (1993). Automatic grammar induction and parsing free text: A transformation-based approach. In *Proceedings of the DARPA Speech and Natural Language Workshop* pp:237-242.

Evans, David A., Chengxiang Zhai (1996). Noun-Phrase Analysis in Unrestricted Text for Information Retrieval. In *Association for Computational Linguistics* (pp17-24).

Hatzivassiloglou, Vasileios, Judith L. Klavans and Eleazar Eskin (1999). Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. In *EMNLP/VLC-99 Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*. University Of Maryland, College Park, MD, USA

Jacquemin, Christian (1999). Syntagmatic and Paradigmatic Representations of Term Variation. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics* (pp.341-348). University of Maryland, College Park, MD, USA.

Klavans, Judith L., Nina Wacholder, (1998). Automatic Identification of Significant Topics in Domain-Independent Full Text Documents. In *Proceedings of the Information and Data Management Workshop*. Available at http://www.cs.pitt.edu/~panos/idm98/Imported/nina.html

Klavans, Judith L., Martin Chodorow, and Nina Wacholder, (1992). Building a Knowledge Base from Parsed Definitions. In Karen Jensen, Goerge Heidorn, Steve Richardson (Eds.) *Natural Language Processing: The PLNLP Approach* (Chapter 11) Kluwer.

Marcus M. P., B. Santorini, and M. A. Marcinkiewicz, (1993). Building a Large Annotated Corpus of English: The Penn Treebank. In *Computational Linguistics* (19).

McKeown, Kathleen R., Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay and Eleazar Eskin, (1999). Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence AAAI-1999.* Orlando, Florida.

Negrilla, Stefan (1998). Clustering Algorithms Summer Project. Computer Science Report, Columbia University.

Ramshaw, Lance A. and Mitchell P. Marcus (1995). Text Chunking Using Transformation-Based Learning. In *Proceedings of the Third Association for Computational Linguistics Workshop on Very Large Corpora*.

Tolle, Kristin M. and Hsinchun Chen (2000). Comparing Noun Phrasing Techniques for Use with Medical Digital Library Tools. In *Journal of the American Society for Information Science Association* 51(4):352-370.

Tolle, Kristin M. (1997). Improving Concept Extraction from Text Using Noun Phrasing Tools: An Experiment in Medical Information Retrieval. Master Thesis. University of Arizona, Department of Management Information Systems.

Wacholder, Nina (1998). Simplex NPs clustered by head: a method for identifying significant topics in a document. In *Proceedings of Workshop on the Computational Treatment of Nominals* COLING-ACL, pp70-79. Montreal.

Wacholder, Nina, Judith L. Klavans and David Kirk Evans (in progress). An Analysis of the Role of Grammatical Categories in a Statistical Information Retrieval System. Columbia University, Department of Computer Science.