Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation

Kuang-hua Chen and Hsin-Hsi Chen

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.
Internet: hh_chen@csie.ntu.edu.tw

Abstract
To acquire noun phrases from running texts is useful for many applications, such as word grouping, terminology indexing, etc. The reported literatures adopt pure probabilistic approach, or pure rule-based noun phrases grammar to tackle this problem. In this paper, we apply a probabilistic chunker to deciding the implicit boundaries of constituents and utilize the linguistic knowledge to extract the noun phrases by a finite state mechanism. The test texts are SUSANNE Corpus and the results are evaluated by comparing the parse field of SUSANNE Corpus automatically. The results of this preliminary experiment are encouraging.

1. Introduction
From the cognitive point of view, human being must recognize, learn and understand the entities or concepts (concrete or abstract) in the texts for natural language comprehension. These entities or concepts are usually described by noun phrases. The evidences from the language learning of children also show the belief (Snow and Ferguson, 1977). Therefore, if we can grasp the noun phases of the texts, we will understand the texts to some extent. This consideration is also captured by theories of discourse analysis, such as Discourse Representation Theory (Kamp, 1981).

Traditionally, to make out the noun phrases in a text means to parse the text and to resolve the attachment relations among the constituents. However, parsing the text completely is very difficult, since various ambiguities cannot be resolved solely by syntactic or semantic information. Do we really need to fully parse the texts in every application? Some researchers apply shallow or partial parsers (Smadja, 1991; Hindle, 1990) to acquiring specific patterns from texts. These tell us that it is not necessary to completely parse the texts for some applications.

This paper will propose a probabilistic partial parser and incorporate linguistic knowledge to extract noun phrases. The partial parser is motivated by an intuition (Abney, 1991):

1. When we read a sentence, we read it chunk by chunk.

Abney uses two level grammar rules to implement the parser through pure LR parsing technique. The first level grammar rule takes care of the chunking process. The second level grammar rule tackles the attachment problems among chunks. Historically, our statistics-based partial parser is called chunker. The chunker receives tagged texts and outputs a linear chunk sequences. We assign a syntactic head and a semantic head to each chunk. Then, we extract the plausible maximal noun phrases according to the information of syntactic head and semantic head, and a finite state mechanism with only 8 states.

2. Previous Works
Church (1988) proposes a part of speech tagger and a simple noun phrase extractor. His noun phrase extractor brackets the noun phrases of input tagged texts according to two probability matrices: one is starting noun phrase matrix; the other is ending noun phrase matrix. The methodology is a simple version of Garside and Leech's probabilistic parser (1985). Church lists a sample text in the Appendix of his paper to show the performance of his work. It demonstrates only 5 out of 248 noun phrases are omitted. Because the tested text is too small to assess the results, the experiment for large volume of texts is needed.
Bourigault (1992) reports a tool, LEXTER, for extracting terminologies from texts. LEXTER triggers two-stage processing: 1) analysis (by identification of frontiers), which extracts the maximal-length noun phrase; 2) parsing (the maximal-length noun phrases), which, furthermore, acquires the terminology embedded in the noun phrases. Bourigault declares the LEXTER extracts 95% maximal-length noun phrases, that is, 43500 out of 46000 from test corpus. The result is validated by an expert. However, the precision is not reported in the Bourigault's paper.

Voutilainen (1993) announces NPtool for acquisition of maximal-length noun phrases. NPtool applies two finite state mechanisms (one is NP-hostile; the other is NP-friendly) to the task. The two mechanisms produce two NP sets and any NP candidate with at least one occurrence in both sets will be labeled as the "ok" NP. The reported recall is 98.5-100% and the precision is 95-98% validated manually by some 20000 words. But from the sample text listed in Appendix of his paper, the recall is about 85% and we can find some inconsistencies among these extracted noun phrases.

3. Language Model

Parsing can be viewed as optimizing. Suppose an n-word sentence, \( w_1, w_2, ..., w_n \) (including punctuation marks), the parsing task is to find a parsing tree \( T \), such that \( P(T \mid w_1, w_2, ..., w_n) \) has the maximal probability. We define \( T \) here to be a sequence of chunks, \( c_1, c_2, ..., c_m \), and each \( c_i \) \((0 < i \leq m)\) contains one or more words \( w_j \) \((0 < j \leq n)\). For example, the sentence "parsing can be viewed as optimization." consists of 7 words. Its one possible parsing result under our demand is:

\[
(2) \quad [\text{parsing}] \quad \text{[can be viewed]} \quad \text{[as optimization]} \quad [.] \quad c_1 \quad c_2 \quad c_3 \quad c_4
\]

Now, the parsing task is to find the best chunk sequence, \( C^* \), such that

\[
(3) \quad C^* = \text{argmax}_{C_i} P(C_i \mid w^n)
\]

The \( C_i \) is one possible chunk sequence, \( c_1, c_2, ..., c_m \), where \( m_i \) is the number of chunks of the possible chunk sequence. To chunk raw text without other information is very difficult, since the word patterns are many millions. Therefore, we apply a tagger to preprocessing the raw texts and give each word a unique part of speech. That is, for an n-word sentence, \( w_1, w_2, ..., w_n \) (including punctuation marks), we assign part of speeches \( t_1, t_2, ..., t_n \) to the respective words. Now the real working model is:

\[
(4) \quad C^* = \text{argmax}_{C_i} P(C_i \mid t^n)
\]

Using bi-gram language model, we then reduce \( P(C_i \mid t_1, t_2, ..., t_n) \) as (5),

\[
(5) \quad P(C_i \mid t^n) = P(c_i \mid t^n)
\]

\[
\equiv \prod_{j=1}^{n} P(c_i \mid c_{i-1}, t^n) \times P(c_i \mid t^n)
\]

\[
\equiv \prod_{j=1}^{n} P(c_i \mid c_{i-1}) \times P(c_i)
\]

where \( P(\cdot) \) denotes the probability for the \( i \)th chunk sequence and \( c_0 \) denotes the beginning mark of a sentence. Following (5), formula (4) becomes

\[
(6) \quad \text{argmax}_{C_i} P(C_i \mid t^n)
\]

\[
\equiv \text{argmax}_{C_i} \prod_{j=1}^{n} P(c_i \mid c_{i-1}) \times P(c_i)
\]

\[
= \text{argmax}_{C_i} \sum_{i=1}^{n} [\log(P(c_i \mid c_{i-1})) + \log(P(c_i))]
\]

In order to make the expression (6) match the intuition of human being, namely, 1) the scoring metrics are all positive, 2) large value means high score, and 3) the scores are between 0 and 1, we define a score function \( S(P(\cdot)) \) shown as (7).

\[
(7) \quad S(P(\cdot)) = 0 \quad \text{when} \quad P(\cdot) = 0;
\]

\[
S(P(\cdot)) = 1.0/(1.0+\text{ABS}(\log(P(\cdot)))) \quad \text{o/w.}
\]

We then rewrite (6) as (8).

\[
(8) \quad C^* = \text{argmax}_{C_i} P(C_i \mid t^n)
\]

\[
\equiv \text{argmax}_{C_i} \prod_{j=1}^{n} P(c_i \mid c_{i-1}) \times P(c_i)
\]

\[
= \text{argmax}_{C_i} \sum_{i=1}^{n} [\log(P(c_i \mid c_{i-1})) + \log(P(c_i))]
\]

The final language model is to find a chunk sequence \( C^* \), which satisfies the expression (8).

Dynamic programming shown in (9) is used to find the best chunk sequence. The \( \text{score}(i) \) denotes the score of position \( i \). The words between position \( \text{pre}[i] \) and position \( i \) form the best chunk from the viewpoint of position \( i \). The \( \text{dscore}(c_i) \) is the score of the probability
For example, LOB corpus defines four kinds of verbal words under the coarse POS V: VB*, DO*, BE* and HV*2. The secondary priority within the coarse POS V is:

(12) VB* > HV* > DO* > BE*

Furthermore, we define the semantic head and the syntactic head (Abney, 1991).

(13) Semantic head is the head of a phrase according to the semantic usage; but syntactic head is the head based on the grammatical relations.

Both the syntactic head and the semantic head are useful in extracting noun phrases. For example, if the semantic head of a chunk is the noun and the syntactic one is the preposition, it would be a prepositional phrase. In some case, we will find some chunks contain only one word, called one-word chunks. They maybe contain a conjunction, e.g., that. Therefore, the syntactic head and the semantic head of one-word chunks are the word itself.

Following these definitions, we extract the noun phrases by procedure (14):
a chunker. The tag sets of LOB and SUSANNE are different. Since the tag set of SUSANNE corpus is subsumed by the tag set of LOB corpus, a TAG-MAPPER is used to map tags of SUSANNE corpus to those of LOB corpus. The chunker will output a sequence of chunks. Finally, a finite state NP-TRACTOR will extract NPs. Figure 2 shows the finite state mechanism used in our work.

![Figure 2. The Finite State Machine for Noun Phrases](image)

The symbols in Figure 2 are tags of LOB corpus. N* denotes nouns; P* denotes pronouns; J* denotes adjectives; A* denotes quantifiers, qualifiers and determiners; IN denotes prepositions; CD* denotes cardinals; OD* denotes ordinals, and NR* denotes adverbial nouns. Asterisk * denotes a wildcard. For convenience, some constraints, such as syntactic and semantic head checking, are not shown in Figure 2.

5. First Stage of Experiments
Following the procedures depicted in Figure 1, we should train a chunker firstly. This is done by using the SUSANNE Corpus (Sampson, 1993; Sampson, 1994) as the training texts. The SUSANNE Corpus is a modified and condensed version of Brown Corpus (Francis and Kucera, 1979). It only contains the 1/10 of Brown Corpus, but involves more information than Brown Corpus. The Corpus consists of four kinds of texts: 1) A: press reportage; 2) G: belles letters, biography, memoirs; 3) J: learned writing; and 4) N: adventure and Western fiction. The Categories of A, G, J and N are named from respective categories of the Brown Corpus. Each Category consists of 16 files and each file contains about 2000 words.

The following shows a snapshot of SUSANNE Corpus.

| Category | Files | Paragraphs | Sentences | Words |
|----------|-------|-----------|-----------|-------|
| A        | 16    | 767       | 1445      | 37180 |
| G        | 16    | 280       | 1554      | 37583 |
| J        | 16    | 197       | 1353      | 36554 |
| N        | 16    | 723       | 2568      | 38736 |
| Total    | 64    | 1967      | 6920      | 150053|

In order to avoid the errors introduced by tagger, the SUSANNE corpus is used as the training and testing texts. Note the tags of SUSANNE corpus are mapped to LOB corpus. The 3/4 of texts of each categories of SUSANNE Corpus are both for training the chunker and testing the chunker (inside test). The rest texts are only for testing (outside test). Every tree structure contained in the parse field is extracted to form a potential chunk grammar and the adjacent tree structures are also extracted to form a potential context chunk grammar. After the training process, total 10937 chunk grammar rules associated with different scores and 37198 context chunk grammar rules are extracted. These chunk grammar rules are used in the chunking process.

Table 3 lists the time taken for processing SUSANNE corpus. This experiment is executed on the Sun Sparc 10, model 30 workstation. T denotes time, W word, C chunk, and S sentence. Therefore, T/W means the time taken to process a word on average.

| Category | T/W | T/C | T/S |
|----------|-----|-----|-----|
| A        | 0.00295 | 0.0071 | 0.0758 |
| G        | 0.00283 | 0.0069 | 0.0685 |
| J        | 0.00275 | 0.0073 | 0.0743 |
| N        | 0.00309 | 0.0066 | 0.0467 |
| Av.      | 0.00291 | 0.0070 | 0.0663 |

According to Table 3, to process a word needs 0.00291 seconds on average. To process all SUSANNE corpus needs about 436 seconds, or 7.27 minutes.

In order to evaluate the performance of our chunker, we compare the results of our chunker with the denotation made by the SUSANNE Corpus. This comparison is based on the following criterion:

The content of each chunk should be dominated by one non-terminal node in SUSANNE parse field.
This criterion is based on an observation that each non-terminal node has a chance to dominate a chunk.

Table 4 is the experimental results of testing the SUSANNE Corpus according to the specified criterion. As usual, the symbol C denotes chunk and S denotes sentence.

| Cat. | TEST | OUTSIDE TEST | INSIDE TEST |
|------|------|--------------|-------------|
|     | # of correct | # of incorrect | total #     | correct rate | # of correct | # of incorrect | total #     | correct rate | # of correct | # of incorrect | total #     | correct rate |
| A   | 4866   | 40            | 4906        | 0.99        | 4748         | 153          | 4901        | 0.97        | 5163         | 79           | 5242        | 0.98        |
| G   | 3983   | 15            | 4138        | 0.99        | 37583        | 170          | 4505        | 0.96        | 5163         | 79           | 5242        | 0.98        |
| J   | 1213   | 2            | 1215        | 0.99        | 36554        | 170          | 4505        | 0.96        | 5163         | 79           | 5242        | 0.98        |
| N   | 4801   | 1            | 4802        | 0.99        | 38736        | 170          | 4505        | 0.96        | 5163         | 79           | 5242        | 0.98        |

Table 4 shows the chunker has more than 98% chunk correct rate and 94% sentence correct rate in outside test, and 99% chunk correct rate and 97% sentence correct rate in inside test. Note that once a chunk is mischopped, the sentence is also mischopped. Therefore, sentence correct rate is always less than chunk correct rate. Figure 3 gives a direct view of the correct rate of this chunker.

6. Acquisition of Noun Phrases
We employ the SUSANNE Corpus as test corpus. Since the SUSANNE Corpus is a parsed corpus, we may use it as criteria for evaluation. The volume of test texts is around 150,000 words including punctuation marks. The time needed from inputting texts of SUSANNE Corpus to outputting the extracted noun phrases is listed in Table 5. Comparing with Table 3, the time of combining chunks to form the candidate noun phrases is not significant.

Table 5. Time for Acquisition of Noun Phrases

| Words | Time (sec.) | Time/Word |
|-------|-------------|-----------|
| A     | 37180       | 112.32    | 0.00302  |
| G     | 37583       | 108.80    | 0.00289  |
| J     | 36554       | 103.04    | 0.00282  |
| N     | 38736       | 122.72    | 0.00317  |
| Total | 150053      | 446.88    | 0.00298  |

The evaluation is based on two metrics: precision and recall. Precision means the correct rate of what the system gets. Recall indicates the extent to which the real noun phrases retrieved from texts against the real noun phrases contained in the texts. Table 6 describes how to calculate these metrics.

Table 6. Contingency Table for Evaluation

| System | NP | non-NP |
|--------|----|--------|
| a      | b  |        |
| c      |    |        |

The rows of "System" indicate our NP-TRACTOR thinks the candidate as an NP or not an NP; the columns of "SUSANNE" indicate SUSANNE Corpus takes the candidate as an NP or not an NP. Following Table 6, we will calculate precision and recall shown as (16).

\[
\text{Precision} = \frac{a}{a+b} \times 100%
\]

\[
\text{Recall} = \frac{a}{a+c} \times 100%
\]

To calculate the precision and the recall based on the parse field of SUSANNE Corpus is not so straightforward at the first glance. For example, (17) itself is a noun phrase but it contains four noun phrases. A tool for extracting noun phrases should output what kind of and how many noun phrases, when it processes the texts like (17). Three kinds of noun phrases (maximal noun phrases, minimal noun phrases and ordinary noun phrases) are defined first. Maximal noun phrases are those noun phrases which are not contained in other noun phrases. In contrast, minimal noun phrases do not contain any other noun phrases.
Apparently, a noun phrase may be both a maximal noun phrase and a minimal noun phrase. Ordinary noun phrases are noun phrases with no restrictions. Take (17) as an example. It has three minimal noun phrases, one maximal noun phrases and five ordinary noun phrases.

In general, a noun-phrase extractor forms the front end of other applications, e.g., acquisition of verb subcategorization frames. Under this consideration, it is not appropriate to taking (17) as a whole to form a noun phrase. Our system will extract two noun phrases from (17), "a black badge of frayed respectability" and "his neck".

(17) [[[a black badge] of [frayed respectability]] that ought never to have left [his neck]]

We calculate the numbers of maximal noun phrases, minimal noun phrases and ordinary noun phrases denoted in SUSANNE Corpus, respectively and compare these numbers with the number of noun phrases extracted by our system.

Table 7 lists the number of ordinary noun phrases (NP), maximal noun phrases (MNP), minimal noun phrases (mNP) in SUSANNE Corpus. MmNP denotes the maximal noun phrases which are also the minimal noun phrases. On average, a maximal noun phrase subsumes 1.61 ordinary noun phrases and 1.09 minimal noun phrases.

To calculate the precision, we examine the extracted noun phrases (ENP) and judge the correctness by the SUSANNE Corpus. The CNP denotes the correct ordinary noun phrases, CMNP the correct maximal noun phrases, CmNP correct minimal noun phrases and CMmNP the correct maximal noun phrases which are also the minimal noun phrases. The results are itemized in Table 8. The average precision is 95%.

The automatic validation of the experimental results gives us an estimated recall. Appendix provides a sample text and the extracted noun phrases. Interested readers could examine the sample text and calculate recall and precision for a comparison.

7. Applications

Identification of noun phrases in texts is useful for many applications. Anaphora resolution (Hirst, 1981) is to resolve the relationship of the noun phrases, namely, what the antecedent of a noun phrase is. The extracted noun phrases can form the set of possible candidates (or universal in the terminology of discourse representation theory). For acquisition of verb subcategorization frames, to bracket the noun phrases in the texts is indispensable. It can help us to find the boundary of the subject, the object and the prepositional phrase. We would use the acquired noun phrases for an application of adjective grouping. The extracted noun phrases may contain adjectives which pre-modify the head noun. We then utilize the similarity of head nouns to group the adjectives. In addition, we may give the head noun a semantic tag, such as Roget's Thesaurus provides, and then analyze the adjectives. To automatically produce the index of a book,
we would extract the noun phrases contained in the book, calculate the inverse document frequency (IDF) and their term frequency (TF) (Salton, 1991), and screen out the implausible terms.

These applications also have impacts on identifying noun phrases. For applications like anaphora resolution and acquisition of verb subcategorization frames, the maximal noun phrases are not suitable. For applications like grouping adjectives and automatic book indexing, some kinds of maximal noun phrases, such as noun phrases postmodified by "of" prepositional phrases, are suitable; but some are not, e.g., noun phrases modified by relative clauses.

8. Concluding Remarks

The difficulty of this work is how to extract the real maximal noun phrases. If we cannot decide the prepositional phrase "over a husband eyes" is licensed by the verb "pull", we will not know "the wool" and "a husband eyes" are two noun phrases or form a noun phrase combined by the preposition "over".

(18) to pull the wool over a husband eyes
to sell the books of my uncle

In contrast, the noun phrase "the books of my uncle" is so called maximal noun phrase in current context. As the result, we conclude that if we do not resolve PP-attachment problem (Hindle and Rooth, 1993), to the expected extent, we will not extract the maximal noun phrases. In our work, the probabilistic chunker decides the implicit boundaries between words and the NPTTRACTOR connects the adjacent noun chunks. When a noun chunk is followed by a preposition chunk, we do not connect the two chunks except the preposition chunk is led by "of" preposition.

Comparing with other works, our results are evaluated by a parsed corpus automatically and show the high precision. Although we do not point out the exact recall, we provide estimated values. The testing scale is large enough (about 150,000 words). In contrast, Church (1988) tests a text and extracts the simple noun phrases only. Bourigault's work (1992) is evaluated manually, and dose not report the precision. Hence, the real performance is not known. The work executed by Voutilainen (1993) is more complex than our work. The input text first is morphologized, then parsed by constraint grammar, analyzed by two different noun phrases grammar and finally extracted by the occurrences. Like other works, Voutilainen's work is also evaluated manually.

In this paper, we propose a language model to chunk texts. The simple but effective chunker could be seen as a linear structure parser, and could be applied to many applications. A method is presented to extract the noun phrases. Most importantly, the relations of maximal noun phrases, minimal noun phrases, ordinary noun phrases and applicable noun phrases are distinguished in this work. Their impacts on the subsequent applications are also addressed. In addition, automatic evaluation provides a fair basis and does not involve human costs. The experimental results show that this parser is a useful tool for further research on large volume of real texts.

Acknowledgements

We are grateful to Dr. Geoffrey Sampson for his kindly providing SUSANNE Corpus and the details of tag set to us.

References

Abney, Steven (1991), "Parsing by Chunks," in Principle-Based Parsing, Berwick, Abney and Tenny (Eds.), Kluwer Academic Publishers, pp. 257-278.

Bourigault, Didier (1992), "Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases," Proceedings of the 15th International Conference on Computational Linguistics, COLING-92, Vol. III, Nantes, France, pp. 977-981.

Church, Kenneth (1988), "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," Proceedings of Second Conference on Applied Natural Language Processing, pp. 136-143.

Francis, N. and Kucera, H. (1979), Manual of Information to Accompany a Standard Sample of Presentday Edited American English, for Use with Digital Computers, Department of Linguistics, Brown University, Providence, R. I., U.S.A., original ed. 1964, revised 1971, revised and augmented 1979.

Garside, Roger and Leech, Geoffrey (1985), "A Probabilistic Parser," Proceedings of Second Conference of the European Chapter of the ACL, pp. 166-170.

Hindle, Donald (1990), "Noun Classification from Predicate-Argument Structures," Proceedings of 28th Annual Meeting of ACL, pp. 268-275.

Hindle, Donald and Rooth, Mats (1993), "Structural Ambiguity and Lexical Relations," Computational Linguistics, 19(1), pp. 103-120.

Hirst, G. (1981), Anaphora in Natural Language Understanding: a Survey, Lecture Notes 119, Springer-Verlag.

Johansson, Stig (1986), The Tagged LOB Corpus: Users' Manual, Bergen: Norwegian Computing Centre for the Humanities.
Appendix
For demonstration, we list a sample text quoted from N18:0010a-N18:0250e, SUSANNE Corpus. The extracted noun phrases are bracketed. We could compute the precision and the recall from the text as a reference and compare the gap with the experimental results itemized in Section 6. In actual, the result shows that the system has high precision and recall for the text.

interesting JJ ._. [ I_PP1A ] had_HVD [ a_AT one-room_JJ studio_NN ]  which_WDTR overlooked_VBD [ an_AT ancient_JJ courtyard_NN ] filled_VBN with_IN [ flowers_NNS and_CC plants_NNS ] +,_, blooming_VBG everlastingly_RB in_IN [ the_ATI southern_JJ sun_NN ] +,_. [ I_PP1A ] had_HVD come_VBN to_IN [ New_JJ Orleans_NP ] + [ two_CD years_NRS ] earlier_RBR after_IN [ graduating_VBG college_NN ] +,_, partly_RB because_CS [ I_PP1A ] loved_VBD [ the_ATI city_NPL ] and_CC partly_RB because_CS there_EX was_BEDZ quite(QL [ a_AT noted_JJ art_NN colony_NN ] there_RN +,_. When_CS [ my_PP$ Uncle_NPT ] offered_VBD [ me_PP10 ] [ a_AT part-time_JJ job_NN ] which_WDTR would_MD take_VB [ care_NN ] of_IN [ my_PP$ normal_JJ expenses_NNS ] and_CC give_VB [ me_PP1O ] [ time_NR ] to_TO paint_VB [ I_PP1A ] accepted_VBD +,_. [ The_ATI arrangement_NN ] turned_VBD out_RP to_TO be_BE excellent_JJ +,_. [ I_PP1A ] loved_VBD [ the_ATI city_NPL ] and_CC [ I_PP1A ] particularly_RB loved_VBD [ the_ATI city_NPL ] and_CC [ I_PP1A ] particularly_RB loved_VBD [ the_ATI city_NPL ] and_CC [ I_PP1A ] particularly_RB loved_VBD [ the_ATI city_NPL ] and_CC [ I_PP1A ] particularly_RB loved_VBD [ the_ATI city_NPL ] and_CC [ I_PP1A ] particularly_RB loved_VBD [ the_ATI gaiety_NN and_CC spirit_NN ] of_IN [ an_ATI New_JJ Orleans_NP ] +,_, [ Mardi_NR Gras_NR ] +,_. [ I_PP1A ] had_HVD seen_VBN [ two_CD of_IN them_PP3OS ] and_CC [ we_PP1AS ] would_MD soon_RB be_BE in_IN another_DT city-wide_JJ +,_, [ joyous_JJ celebration_NN with_IN romance_NN ] in_IN [ the_ATI air_NN ] +,_, and_CC +,_, when_CS [ you_PP2 ] took_VBD [ a_AT walk_NPL ] [ you_PP2 ] never_RB knew_VBD what_WDT [ adventure_NN or_CC pair_NN of_IN sparkling_JJ eyes_NNS ] were_BED waiting_VBG around_IN [ the_ATI next_OD corner_NPL ] +,_.

Salton, G. (1991), "Developments in Automatic Text Retrieval," Science, Vol. 253, pp. 974-979.

Sampson, Geoffrey (1993), "The SUSANNE Corpus," ICAME Journal, No. 17, pp. 125-127.

Sampson, Geoffrey (1994), English for the Computer, Oxford University Press.

Sells, Peter (1985), Lectures on Contemporary Syntactic Theories, Lecture Notes, No. 3, CSLI.

Smadja, Frank (1991), Extracting Collocations from Text, An Application: Language Generation, Ph.D. Dissertation, Columbia University, 1991.

Snow, C.E. and Ferguson, C.A. (Eds.) (1977), Talking to Children: Language Input and Acquisition, Cambridge, Cambridge University Press.

Voutilainen, Atro (1993), "NPtool, a Detector of English Noun Phrases," Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives, Ohio State University, Columbus, Ohio, USA, pp. 48-57.