Automatic Refinement of a POS Tagger Using a Reliable Parser and Plain Text Corpora

Hideki Hirakawa, Kenji Ono, Yumiko Yoshimura
Human Interface Laboratory
Corporate Research & Development Center
Toshiba Corporation

Komukai-Toshiba-cho 1, Saiwai-ku, Kawasaki, 212-8582, Japan
{hideki.hirakawa, kenji2.ono, yumiko.yoshimura}@toshiba.co.jp

Abstract

This paper proposes a new unsupervised learning method for obtaining English part-of-speech(POS) disambiguation rules which would improve the accuracy of a POS tagger. This method has been implemented in the experimental system APRAS (Automatic POS Rule Acquisition System), which extracts POS disambiguation rules from plain text corpora by utilizing different types of coded linguistic knowledge, i.e., POS tagging rules and syntactic parsing rules, which are already stored in a fully implemented MT system.

In our experiment, the obtained rules were applied to 1.7% of the sentences in a non-training corpus. For this group of sentences, 78.4% of the changes made in tagging results were an improvement. We also saw a 15.5 % improvement in tagging and parsing speed and an 8.0 % increase of parsable sentences.

1 Introduction

Much research has been done on knowledge acquisition from large-scale annotated corpora as a rich source of linguistic knowledge. Major works done to create English POS taggers (henceforth, "taggers"), for example, include (Church 1988), (Kupiec 1992), (Brill 1992) and (Voutilainen et al. 1992). The problem with this framework, however, is that such reliable corpora are hardly available due to a huge amount of the labor-intensive work required. In case of the acquisition of non-core knowledge, such as specific, lexically or domain dependent knowledge, preparation of annotated corpora becomes more serious problem.

One viable approach then is to utilize plain text corpora instead, as in (Mikheev 1996). But The method proposed by (Mikheev 1996) has its own weaknesses, in that it is restricted in scope. That is, it aims to acquire rules for unknown words in corpora from their ending characters without looking at the context. In the meantime, (Brill 1995a) (Brill 1995b) proposed a method to acquire context-dependent POS disambiguation rules and created an accurate tagger, even from a very small annotated text by combining supervised and unsupervised learning. The weakness of his method is that the effect of unsupervised learning decreases as the training corpus size increases.

The problem in using plain text corpora for knowledge acquisition is that we need a human supervisor who can evaluate and sift the obtained knowledge. An alternative to this would be to use a number of modules of a well-developed NLP system which stores most of the highly reliable general rules. Here, one module functions as a supervisor for other modules, since all these modules are designed to work cooperatively and the knowledges stored in each module are correlated.

Keeping this idea in mind, we propose a new unsupervised learning method for obtaining linguistic rules from plain text corpora using the existing linguistic knowledge. This method has been implemented in the rule extraction system APRAS (Automatic POS Rule Acquisition

System), which automatically acquires rules for refining the morphological analyzer (tagger) in our English-Japanese MT system ASTRANSAC (Hirakawa et al. 1991) through the interaction between the system's tagger and parser on the assumption that they are considerably accurate.

This paper is organized as follows: Section 2 illustrates the basic idea of our method; Section 3 gives the outline of APRAS; Sections 4 and 5 describe our experiments.

2 Basic Idea

Our MT system has a tagger which can generate ranked POS sequences of input sentences according to their plausibility and also a parser which judges the parsability of the derived POS sequences one by one until a parsable one is found[1] . In our framework, this tagger can be viewed as a POS candidate generator, and the parser as a sifter.

Now sentences can be categorized into the following three:

(P) a balanced sentence, whose top ranked sequence, or initial POS sequence, is parsable,

(Q) a conflicting sentence, in which the top ranked sequence is unparsable, but there are parsable ones in the rest of the sequences; and

(R) an unparsable sentence, in which all the POS sequences are unparsable.

Before going on to our main discussion, we will briefly explain the terminology used in this paper. Here we call a highest-ranking parsable POS sequence as the "Most Preferable Parsable POS sequence," or simply "MPP POS sequence." For our purposes, we will make use of balanced sentences and conflicting sentences.

We call the POS of a word in the initial POS sequence as its "initially tagged POS" and that in the MPP POS sequence as its "parsable POS." We call the word whose initially tagged POS and parsable POS differ as a "focus word." Since the tagger is accurate, we can expect only a few POS differences between the initial and MPP POS sequences for a sentence. Finally, let us call the POS's of the preceding and succeeding words as the "POS context of the focus word."

Conflicting sentences, and their initial POS sequences, parsable POS sequences, and focus words can be automatically extracted. Through extraction out of a large amount of plain text corpora combined with statistical filtering, it would be possible to automatically select the proper POS conditions that could determine POS's of focus words. Then, we extract "POS Adjusting rules" or "PA rules" defined as below:

PA rule: W(IPOS) $\rightarrow$ W(PPOS): C

C: Context
W :Word
IPOS: Initially tagged POS
PPOS: Parsable POS

Means "Give priority to the parsable POS over the initially tagged POS in a particular context shown as 'C'."

PA rules do not determine POS's of words from their context, but change the judgement made by the tagger in a particular context. Extracted PA rules are independent rules to the tagger and the parser used in the extraction. At the same time, these rules are optimized for the tagger and the parser, since they are derived only from conflicting sentences, not from balanced sentences. Hence, the knowledge already coded in the system will not be extracted.

In the following section, we give the outline of APRAS focusing on its two modules.

3 Outline of APRAS

Fig. 1 shows the application of APRAS to an MT system. APRAS works in two phases, a rule extraction phase and a rule application phase. Note that the same tagger and the parser of the MT system are used throughout.

[1] Here only top-N POS sequences are tried, where N is a pre-defined constant to limit parsing time.

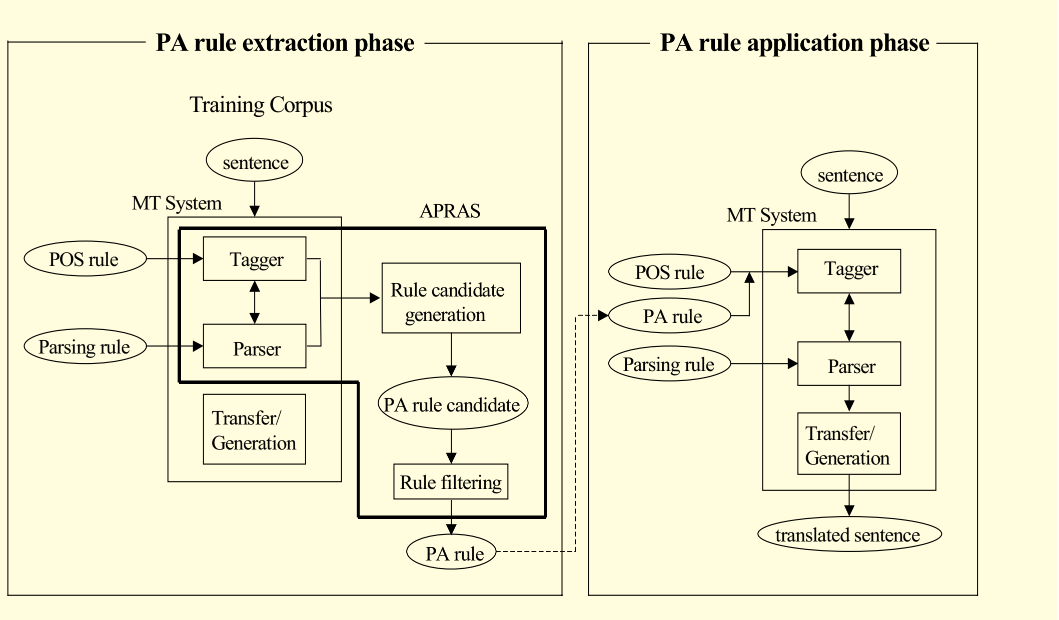


Figure 1: Application of APRAS to an MT System

In the rule extraction phase, the tagger analyzes each sentence in a training corpus and produces plausible POS sequences. The parser then judges the parsability of each POS sequence. Whenever a conflicting sentence appears, the rule generation module outputs the candidates of PA rules.

After all PA rule candidates for this training corpus are generated, the rule filtering module statistically weighs the validity of obtained PA rule candidates, and filters out unreliable rules. Sentences in the training corpus are not translated in this phase.

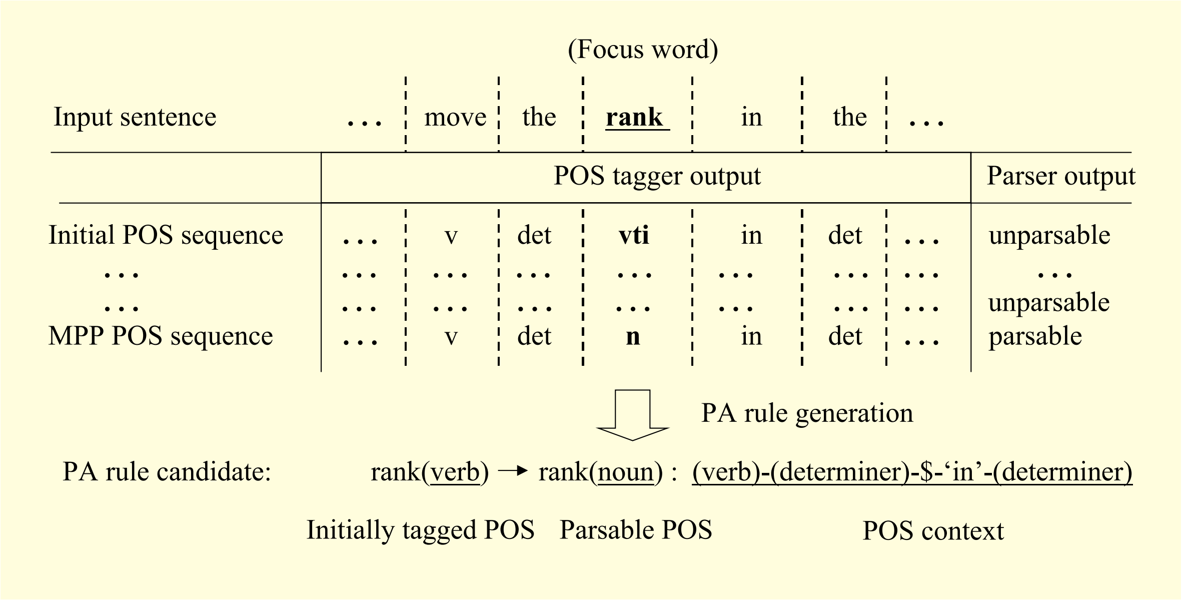
In the rule application phase, both the already installed POS rules and the acquired PA rules are used for tagging. A sentence is parsed and then translated into target language. PA rules basically act to avoid the tagger's wasteful generation of POS sequences. This would improve the ranking of POS sequences the tagger outputs and also increase the chances that the parser will find a parsable or better POS sequence in the improved ranking.

3.1 Rule Generation Module

PA rule candidates are generated from conflicting sentences. Balanced and unparsable sentences generate no PA rule candidate. The words in balanced sentences are recorded along with their POS's and POS contexts to be used in the rule filtering module. Whenever the system encounters a conflicting sentence in a training corpus, the system compares the initial POS sequence with the MPP POS sequence of the sentence and picks up focus words. Then, for every focus word, the system generates a PA rule candidate which consists of a focus word, its initially tagged POS, parsable POS, and the POS context, i.e., the preceding POS's and the succeeding POS's.

Fig. 2 illustrates how a PA rule candidate is generated. The focus word is `rank', its initially tagged POS is `(verb)', its parsable POS is `(noun)', and the POS context is "(verb)-(determiner)-$-`in'-(determiner)", where `$' denotes the focus word. The POS context is composed of preceding two POS's and succeeding two POS's. Here surface words can be used instead of POS, like `in' in the example. The generated PA rule candidate can be read as: If the word `rank' appears in a POS context "(verb)-(determiner)-$-`in'-(determiner)", then give priority to `(noun)' over `(verb)'.

In this rule generation module, two important factors should be taken into account: namely, context size and levels of abstraction. If we expand the context of a focus word, the PA rule should gain accuracy. But its frequency in the training corpus would drop, thereby making it difficult to perform statistical filtering. To ensure statistical reliability, we need a large-



(Focus word)

Input sentence	. . .	move	the	**rank**	in	the	. . .	
	POS tagger output							Parser output
Initial POS sequence	. . .	v	det	**vti**	in	det	. . .	unparsable
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	unparsable
MPP POS sequence	. . .	v	det	**n**	in	det	. . .	parsable

PA rule generation

PA rule candidate: rank(verb) → rank(noun) : (verb)-(determiner)-$-‘in’-(determiner)

Initially tagged POS Parsable POS POS context

Figure 2: PA Rule Candidate Generation

sized training corpus. At present we set the context size to be two words.

In choosing adequate levels of abstraction or specification of POS in the context, we grouped together those POS tags which influence the choice of POS of a focus word in a similar manner as one super-POS tag, as in (Haruno & Matsumoto 1997). We also changed some POS tags for functional words like prepositions and words such as `be' and `have' to tags which denote their literal forms, because the choice of POS of a focus word is highly dependent on the word itself. As a result, we obtained 513 POS tags including 16 POS tags for nouns, 17 for verbs, 410 for prepositions and phrasal prepositions, and 70 for adjectives and adverbs.

3.2 Rule Filtering Module

This section deals with how to statistically filter out inappropriate rules from the generated PA rule candidates. For this purpose, we introduce what we call "adjustment ratios."

Table 1 shows the parsing process of a sentence in which word W appears in POS context *C*: P1-P2-$-P3-P4. In this context, the word W has two possible POS's, X and Y. Case A shows the case of balanced sentences where the tagger first tagged W with X and the parser found it parsable. Case B shows the case of conflicting sentences where the tagger first tagged W with unparsable X and then with Y which proved to be parsable[2].

Let N_a and N_b be the number of sentences in cases A and B, respectively. Assume the parser is accurate enough to be able to judge a majority of sentences with correct POS contexts to be parsable[3], and those with incorrect POS unparsable.

Table 1 : Transition of POS of W in Parsing Process for Context C

$\mathrm{POS}_{W,C}\mathrm{X}$	A
$\mathrm{POS}_{W,C}\mathrm{X} \rightarrow \mathrm{POS}_{W,C}\mathrm{Y}$	B

Then, adjustment ratios can be formulated as follows :

[2] Here only two possibilities, namely X and Y, are considered. However it is easy to generalize the transition process for cases where focus words have more than two POS candidates.

[3] The accuracy of POS sequences accepted by our parser is more than 99% (Yoshimura 1995).

[4] Financial Times(1992-1994, approx. 210,000 documents) in NIST Standard Reference Data TREC Document Database: Disk4 (Special No. 22), National Institute of Standards and Technology, U.S. Department of Commerce (http://www.nist.gov/srd).

$$\textit{adjustment ratio}_{W;E}(\mathrm{X} \rightarrow \mathrm{Y}) = \frac{N_b}{N_a + N_b}$$

When the value is high, the tagger should change the POS from X to Y, whereas when the value is low, the tagger should not change the POS in the given context. Thus, based on the statistics of an accurate parser's judgement, adjustment ratios can be a criterion for the validity of PA rules. The rules whose adjustment ratios are above the threshold are extracted and output as PA rules. The threshold is fixed by examining PA rule candidates as will be mentioned in the next section. More importantly, PA rules are considered to be `optimized' to the parser. First, the selection and application of inappropriate PA rules do not immediately deteriorate the parser output, since PA rules only serve to eliminate wasteful generation of POS sentences. Second, the existence of inappropriate PA rules eventually shortens the processing time for those sentences for which the parser produces an errorneous syntactic structure due to a lack of syntactic knowledge.

4 Rule Extraction Experiment

We applied the method described in Section 3.2 to English news articles (6,684,848 sentences, 530MB)[4] as a training corpus and obtained 300,438 different PA rule candidates. Since rules with low frequencies do not have reliable adjustment ratios, we omitted rules with a frequency below 6 and thus obtained 17,731 rules.

Table 2: Adjustment Ratios and the Validity of Extracted Rules

Adjustment ratio(%)	Total	Valid	(%)	Invalid	(%)	Undecidable
0-9	20	0	(0)	19	(95)	1
10-19	25	3	(12)	17	(68)	5
20-29	24	4	(17)	10	(42)	10
30-39	16	8	(50)	2	(13)	6
40-49	15	10	(67)	1	(7)	4
50-59	15	7	(47)	4	(27)	4
60-69	15	15	(100)	0	(0)	0
70-79	17	15	(88)	1	(6)	1
80-89	16	15	(94)	1	(6)	0
90-99	18	14	(78)	3	(17)	1
100	29	23	(79)	2	(7)	4
total	210	114		60		36

To verify the validity of adjustment ratio-based rule selection method described in Section 3.2, we examined some of the obtained PA rules whose frequencies are 10, 20, and 30, referring to the original sentences from which they were generated, and classified the rules into the following three categories.

(P) Valid: applicable to any sentence.

(Q) Invalid: inapplicable to every sentence. This type of rule is derived when an incorrect POS sequence was judged to be parsable, due to a lack of coverage of parsing rules in the parser.

(R) Undecidable: The derived rule is neither valid nor invalid, either because the POS context or POS specifications are insufficient to uniquely determine the POS of the focus word, or because both the initially tagged POS and the parsable POS are inadequate for the POS context.

An example of (3) is:
trading(present particle) → trading(noun):
(noun)-'of'-$-(determiner)-(noun)

The word "trading" is a present particle in sentences like ".. index features represent a more convenient and liquid way of trading an index basket than ...," while it is a noun in sentences like "By the close of trading the deal was quoted at 99.82 bid."

Table 2 shows the result of the classification. As is clear in the table, for adjustment ratios below 30 %, there are more invalid rules than valid rules, and for adjustment ratios above 30 %, the converse is true. The percentage of invalid rules is small above 60 %.

These results prove the validity of our adjustment ratio-based rule selection framework. By setting the threshold to 60%, we can extract in an unsupervised manner PA rules of which 86% are valid and 7% invalid, but the presence of such invalid PA rules are unlikely to cause a serious deterioration, as mentioned previously. Incidentally, rules whose adjustment ratio is below 30 % could be used as prohibition rules to be applied in the given POS contexts. These rules are not used in the next experiment.

Thus, we eliminated the extracted 17,731 rules whose adjustment ratio are below 60% and obtained 4,494 rules such as :

group(V) → group(N) :
(ADV)-(N)-$-(NAME)-(CC)
report(N) → report(V) :
(ADV)-','-$-(NAME)-(NAME)
related(VP) → related(PP) :
(NAME)-(CC)-$-(N)-(PNC)
open(V) → open(ADJ) :
(N)-','-$-(N)-','
further(V) → further(ADV) :
'to'-(V)-$-(NU)-(DEM)

where ADJ=adjective, ADV=adverb, CC=coordinate conjuction, DET=determiner, DEM=demonstrative, NAMEP=place name, N=noun, NAME=proper noun, NT=noun meaning "time", NU=noun meaning "unit", PP=past particle, PNC=punctuation mark other than commas, V=verb (other than past form), VP=verb (past form).

5 Rule Application Experiment

By using PA rules, we can expect that:

(P) the process time would be reduced by obtaining a parsable POS sequence at an earlier stage, and
(Q) both tagging precision and parsing accuracy would improve.

To prove the above statements, we applied the 3,921 PA rules[5] extracted in the previous experiment for tagging entirely different English news articles (146,229 sentences; 2.26M Words) from the training corpus. Among them, 2,421 sentences (1.7%) or 2,476 words (0.11%) satisfied the conditions of these PA rules, which were then tagged and parsed with and without the PA rules. We measured the difference in the elapsed time[6] and the number counted of successfully parsed sentences. The result is shown in Table 3. The tagging time was extended by 11.5%, but the parsing time and the total processing time were reduced by 24% and 15.5%, respectively, while the ratio of successfully parsed sentences improved by 8.0%.

We also examined 524 POS differences out of all the resulting differences in the tagger's outputs made by the PA rules, and obtained the following figures.

- Improved:	411 (78.4%)
- Worse:	84 (16.0%)
- Neither improved nor worse	29 (5.5%)

Out of the 84 worsened cases, 43 were due to invalid rules acquired through wrong parsing because of a lack of sufficient parsing rules. There are highly frequent expressions characteristic of financial reports which our parser cannot parse. However, again, this kind of invalid rules would not make a significant difference in the final output of the parser. The remaining 32 cases were due to learning from wrongly segmented sets of words and also from distinct header expressions like "FT 14 MAY 91 / World News in Brief ". These errors can be easily eliminated by not learning from these data. Adopting the rule accuracy obtained from the above examination, we can expect 62.4% (78.4% − 16.0%) improvement for words with PA-rule applied. Since PA-rules are applied to 0.11% of the words in corpus, 0.07% improvement of POS tagging is expected. We measured the tagging precision with and without the acquired PA rules for a test corpus containging 5,630 words, and observed that the precision rose to 98.65% from the initial 98.60%, i.e. 0.05% improvement. Since PA rules are lexically based rules, the ratio of sentences which satisfied the rule conditions is rather low, but the number of those sentences would increase in proportion to the number of PA rules acquired.

If we expand the size of a training corpus, we could obtain much more PA rules. In fact, we observed many valid rules in the eliminated PA rule candidates whose frequency is immediately

[5] Out of 4,494, 573 rules have been eliminated in this experiment. These cases involved distinction between compound words (ex. `that is'(adverb)) and non-compound words (ex. `that(pronoun)+is(verb)'). This accompanies changes in the window of context, which requires further research.

[6] The elapsed time is measured on WorkStation SUN Ultra U1E/200.

Table 3 : Processing Time and Parsable Sentence Ratio

	Without PA rules	With PA rules
Tagging time(sec.)	79.40	88.49(+9.09, +11.5%)
Parsing time(sec.)	252.17	191.73(-60.44, -24.0%)
Total processing time(sec.)	331.57	280.22(-51.35, -15.5%)
Parsable sentence ratio	64.0%	72.0%

below the threshold. Since the observed frequency distribution of PA rules was exponential, we can expect PA rules would increase exponentially by expanding the size of a training corpus.

This expansion also enables us to specify POS context in detail, like widening the context window, subcategorizing POS tags employed in context, assigning one surface functional word to a lexical tag, etc. To make detailed classification fully effective, we will need to generalize specific rules to the level that reflects the maximum distinction of individual examples.

6 Conclusion

In this paper we presented a new approach to acquiring linguistic knowledge automatically from plain text corpora, and proved its feasibility by the experiment. Our method utilizes well-developed modules in a NLP system, including a tagger and a parser, and enables us to extract valid rules with high accuracy. It is robust in that the application of the extracted incorrect knowledge does not cause a serious performance deterioration.

As a first step to obtaining lexically dependent knowledge, we examined the validity of obtained POS rules to measure the viability of our unsupervised learning method from plain text corpora. In the future we will expand the size of training corpora and make use of invalid PA rules with a low adjustment ratio.

References

Brill, Eric. 1992: *A Simple Rule-Based Part of Speech Tagger*, in Proceedings of the Third Conference on Applied Natural Language Processing, pp. 152-155.

Brill, Eric. 1995a: *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging*, in Computational Linguistics, Volume 21, Number 4.

Brill, Eric. 1995b: *Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging*, Workshop on Very Large Corpora.

Church, Kenneth. 1988: *A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text*, in Proceedings of the Second Conference on Applied Natural Language Processing, Austin, Texas, pp.126-143.

Haruno, Masahiko and Yuji Matsumoto. 1997: *Mistake-Driven Mixture of Hierarchical Tag Context Trees*, in Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, Madrid, Spain.

Hirakawa, Hideki, Hiroyasu Nogami and Shin'ya Amano. 1991: *EJ/JE Machine Translation System ASTRANSAC-Extensions toward Personalization*, in Proceedings of MT SUMMIT-III, Washington, D.C., 1991, pp.73-80.

Kupiec, Julian. 1992: *Robust Part-of-Speech Tagging Using a Hidden Markov Model*, Computer Speech & Language, 6(3), pp.225-242.

Mikheev, Andrei. 1996: *Unsupervised Learning of Word-Category Guessing Rules*, in Proceedings of the 34th Annual Meeting of the Association of Computational Linguistics, Santa Cruz, California.

Voutilainen, Atro, Juha Heikkilä and Arto Anttila. 1992: *CONSTRAINT GRAMMAR OF ENGLISH - A Performance-Oriented Introduction*, Publications of the Department of General Linguistics, University of Helsinki, No.21.

Yoshimura, Yumiko. 1995: *Selection of English Part-of-Speech Strings Using Syntactic Analysis Information*, in Proceedings of the 50th Annual Convention of IPS Japan, 3-65, March (in Japanese).