A Unified Statistical Model for the Identification of English BaseNP

Endong Xun
Microsoft Research China
No. 49 Zhichun Road Haidian District
100080, China,
i-edxun@microsoft.com

Changning Huang
Microsoft Research China
No. 49 Zhichun Road Haidian District
100080, China,
cnhuang@microsoft.com

Ming Zhou
Microsoft Research China
No. 49 Zhichun Road Haidian District
100080, China,
Mingzhou@microsoft.com

Abstract

This paper presents a novel statistical model for automatic identification of English baseNP. It uses two steps: the N-best Part-Of-Speech (POS) tagging and baseNP identification given the N-best POS-sequences. Unlike the other approaches where the two steps are separated, we integrate them into a unified statistical framework. Our model also integrates lexical information. Finally, Viterbi algorithm is applied to make global search in the entire sentence, allowing us to obtain linear complexity for the entire process. Compared with other methods using the same testing set, our approach achieves 92.3% in precision and 93.2% in recall. The result is comparable with or better than the previously reported results.

1 Introduction

Finding simple and non-recursive base Noun Phrase (baseNP) is an important subtask for many natural language processing applications, such as partial parsing, information retrieval and machine translation. A baseNP is a simple noun phrase that does not contain other noun phrase recursively, for example, the elements within [...] in the following example are baseNPs, where NNS, IN VBG etc are part-of-speech tags [as defined in M. Marcus 1993].

[Measures/NNS] of/IN [manufacturing/VBG activity/NN] fell/VBD more/RBR than/IN [the/DT overall/JJ measures/NNS] ./.

Figure 1: An example sentence with baseNP brackets

A number of researchers have dealt with the problem of baseNP identification (Church 1988; Bourigault 1992; Voutilainen 1993; Justeson & Katz 1995). Recently some researchers have made experiments with the same test corpus extracted from the 20th section of the Penn Treebank Wall Street Journal (Penn Treebank). Ramshaw & Markus (1998) applied transform-based error-driven algorithm (Brill 1995) to learn a set of transformation rules, and using those rules to locally updates the bracket positions. Argamon, Dagan & Krymolowski (1998) introduced a memory-based sequences learning method, the training examples are stored and generalization is performed at application time by comparing subsequence of the new text to positive and negative evidence. Cardie & Pierce (1998 1999) devised error driven pruning approach trained on Penn Treebank. It extracts baseNP rules from the training corpus and prune some bad baseNP by incremental training, and then apply the pruned rules to identify baseNP through maximum length matching (or dynamic program algorithm).

Most of the prior work treats POS tagging and baseNP identification as two separate procedures. However, uncertainty is involved in both steps. Using the result of the first step as if they are certain will lead to more errors in the second step. A better approach is to consider the

two steps together such that the final output takes the uncertainty in both steps together. The approaches proposed by Ramshaw & Markus and Cardie&Pierce are deterministic and local, while Argamon, Dagan & Krymolowski consider the problem globally and assigned a score to each possible baseNP structures. However, they did not consider any lexical information.

This paper presents a novel statistical approach to baseNP identification, which considers both steps together within a unified statistical framework. It also takes lexical information into account. In addition, in order to make the best choice for the entire sentence, Viterbi algorithm is applied. Our tests with the Penn Treebank showed that our integrated approach achieves 92.3% in precision and 93.2% in recall. The result is comparable or better that the current state of the art.

In the following sections, we will describe the detail for the algorithm, parameter estimation and search algorithms in section 2. The experiment results are given in section 3. In section 4 we make further analysis and comparison. In the final section we give some conclusions.

2 The statistical approach

In this section, we will describe the two-pass statistical model, parameters training and Viterbi algorithm for the search of the best sequences of POS tagging and baseNP identification. Before describing our algorithm, we introduce some notations we will use

2.1 Notation

Let us express an input sentence E as a word sequence and a sequence of POS respectively as follows:

$$E = w_1 \quad w_2 \quad \ldots \quad w_{n-1} \quad w_n$$

$$T = t_1 \quad t_2 \quad \ldots \quad t_{n-1} \quad t_n$$

Where n is the number of words in the sentence, t_i is the POS tag of the word w_i.

Given E, the result of the baseNP identification is assumed to be a sequence, in which some words are grouped into baseNP as follows

$$\ldots w_{i-1} \quad [w_i \quad w_{i+1} \quad \ldots w_j] \quad w_{j+1} \ldots$$

The corresponding tag sequence is as follows:

(a)

$$B = t_{i-1} \quad [t_i \quad t_{i+1} \ldots t_j] \quad t_{j+1} \ldots = t_{i-1} \quad b_{i,j} \quad t_{j+1} \ldots = \eta_1 \quad \eta_2 \quad \ldots \eta_m$$

In which $b_{i,j}$ corresponds to the tag sequence of a baseNP: $[t_i \quad t_{i+1} \ldots \quad t_j]$. $b_{i,j}$ may also be thought of as a baseNP rule. Therefore B is a sequence of both POS tags and baseNP rules. Thus $1 \le m \le n$, $\eta_i \in$ (POS tag set $\cup$ baseNP rules set), This is the first expression of a sentence with baseNP annotated. Sometime, we also use the following equivalent form:

(b)

$$Q = \ldots (t_{i-1}, bm_{i-1}) \quad (t_i, bm_i) \quad (t_{i+1}, bm_{i+1}) \ldots (t_j, bm_j) \quad (t_{j+1}, bm_{j+1}) \ldots = q_1 \quad q_2 \ldots q_n$$

Where each POS tag t_i is associated with its positional information bm_i with respect to baseNPs. The positional information is one of $\{F, I, E, O, S\}$. F, E and I mean respectively that the word is the left boundary, right boundary of a baseNP, or at another position inside a baseNP. O means that the word is outside a baseNP. S marks a single word baseNP. This second expression is similar to that used in [Marcus 1995].

For example, the two expressions of the example given in Figure 1 are as follows:

(a) B= [NNS] IN [VBG NN] VBD RBR IN [DT JJ NNS]

(b) Q=(NNS S) (IN O) (VBG F) (NN E) (VBD O) (RBR O) (IN O) (DT F) (JJ I) (NNS E) (. O)

2.2 An 'integrated' two-pass procedure

The principle of our approach is as follows. The most probable baseNP sequence B^* may be expressed generally as follows:

$$B^* = \arg\max_B (p(B \mid E))$$

We separate the whole procedure into two passes, i.e.:

$$B^* \approx \arg\max_B (P(T \mid E) \times P(B \mid T, E)) \qquad (1)$$

In order to reduce the search space and computational complexity, we only consider the N best POS tagging of E, i.e.

$$T(N-best) = \arg\max_{T=T_1,\ldots,T_N} (P(T \mid E)) \qquad (2)$$

Therefore, we have:

$$B^* \approx \arg\max_{B, T=T_1,\ldots T_N} (P(T \mid E) \times P(B \mid T, E)) \qquad (3)$$

Correspondingly, the algorithm is composed of two steps: determining the N-best POS tagging using Equation (2). And then determining the best baseNP sequence from those POS sequences using Equation (3). One can see that the two steps are integrated together, rather that separated as in the other approaches. Let us now examine the two steps more closely.

2.3 Determining the N best POS sequences

The goal of the algorithm in the 1st pass is to search for the N-best POS-sequences within the search space (POS lattice). According to Bayes' Rule, we have

$$P(T \mid E) = \frac{P(E \mid T) \times P(T)}{P(E)}$$

Since $P(E)$ does not affect the maximizing procedure of $P(T \mid E)$, equation (2) becomes

$$T(N-best) = \arg\max_{T=T_1,\ldots,T_N}(P(T \mid E)) = \arg\max_{T=T_1,\ldots,T_N}(P(E \mid T) \times P(T)) \quad (4)$$

We now assume that the words in E are independent. Thus

$$P(E \mid T) \approx \prod_{i=1}^{n} P(w_i \mid t_i) \quad (5)$$

We then use a trigram model as an approximation of $P(T)$, i.e.:

$$P(T) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-2}, t_{i-1}) \quad (6)$$

Finally we have

$$T(N-best) = \arg\max_{T=T_1,\ldots,T_N}(P(T \mid E))$$

$$= \arg\max_{T=T_1,\ldots,T_N}\left(\prod_{i=1}^{n} P(w_i \mid t_i) \times P(t_i \mid t_{i-2}, t_{i-1})\right) \quad (7)$$

In Viterbi algorithm of N best search, $P(w_i \mid t_i)$ is called lexical generation (or output) probability, and $P(t_i \mid t_{i-2}, t_{i-1})$ is called transition probability in Hidden Markov Model.

2.3.1 Determining the baseNPs

As mentioned before, the goal of the 2nd pass is to search the best baseNP-sequence given the N-best POS-sequences.

Considering E, T and B as random variables, according to Bayes' Rule, we have

$$P(B \mid T, E) = \frac{P(B \mid T) \times P(E \mid B, T)}{P(E \mid T)}$$

Since $P(B \mid T) = \frac{P(T \mid B) \times P(B)}{P(T)}$ we have,

$$P(B \mid T, E) = \frac{P(E \mid B, T) \times P(T \mid B) \times P(B)}{P(E \mid T) \times P(T)} \quad (8)$$

Because we search for the best baseNP sequence for each possible POS-sequence of the given sentence E, so $P(E \mid T) \times P(T) = P(E \cap T) = const$,

Furthermore from the definition of B, during each search procedure, we have $P(T \mid B) = \prod_{i=1}^{n} P(t_i, \ldots t_j \mid b_{i,j}) = 1$. Therefore, equation (3) becomes

$$B^* = \arg\max_{B,T=T_1,\ldots,T_N}(P(T \mid E) \times P(B \mid T, E))$$

$$= \arg\max_{B,T=T_1,\ldots,T_N}(P(T \mid E) \times P(E \mid B, T) \times P(B)) \quad (9)$$

using the independence assumption, we have

$$P(E \mid B, T) \approx \prod_{i=1}^{n} P(w_i \mid t_i, bm_i) \quad (10)$$

With trigram approximation of $P(B)$, we have:

$$P(B) \approx \prod_{i=1}^{m} P(n_i \mid n_{i-2}, n_{i-1}) \quad (11)$$

Finally, we obtain

$$B^* = \arg\max_{B,T=T_1,\ldots T_N}(P(T \mid E) \times \prod_{i=1}^{n} P(w_i \mid bm_i, t_i) \times \prod_{i=1,m} P(n_i \mid n_{i-2}, n_{i-1}))$$

□12□

To summarize, In the first step, Viterbi N-best searching algorithm is applied in the POS tagging procedure, It determines a path probability f_t for each POS sequence calculated as follows:

$$f_t = \prod_{i=1,n} p(w_i \mid t_i) \times p(t_i \mid t_{i-2}, t_{i-1}).$$

In the second step, for each possible POS tagging result, Viterbi algorithm is applied again to search for the best baseNP sequence. Every baseNP sequence found in this pass is also asssociated with a path probability

$$f_b = \prod_{i=1}^{n} p(w_i \mid t_i, bm_i) \times \prod_{i=1,m} p(n_i \mid n_{i-2}, n_{i-1}).$$

The integrated probability of a baseNP sequence is determined by $f_t^{\alpha} \times f_b$, where α is a

normalization coefficient ($\alpha = 2.4$ in our experiments). When we determine the best baseNP sequence for the given sentence E, we also determine the best POS sequence of E, which corresponds to the best baseNP of E.

Now let us illustrate the whole process through an example: "stock was down 9.1 points yesterday morning.". In the first pass, one of the N-best POS tagging result of the sentence is: T = NN VBD RB CD NNS NN NN. For this POS sequence, the 2nd pass will try to determine the baseNPs as shown in Figure 2. The details of the path in the dash line are given in Figure 3, Its probability calculated in the second pass is as follows (Φ is pseudo variable):

$$P(B|T,E) = p(stock|NN,S)\times p(was|VBD,O)\times p(down|RB,O)\times p(NUMBER|CD,B)$$
$$\times p(points|NNS,E)\times p(yesterday|NN,B)\times p(morning|NN,E)\times p(.|.,O)$$
$$\times p([NN]|\Phi,\Phi)\times p(VBD|\Phi,[NN])\times p(RB|[NN],VBD)\times p([CD\quad NNS]|VBD,RB)$$
$$\times p([NN\quad NN]|RB,[CD\quad NNS])\times p(.|[CD\quad NNS],[NN\quad NN])$$

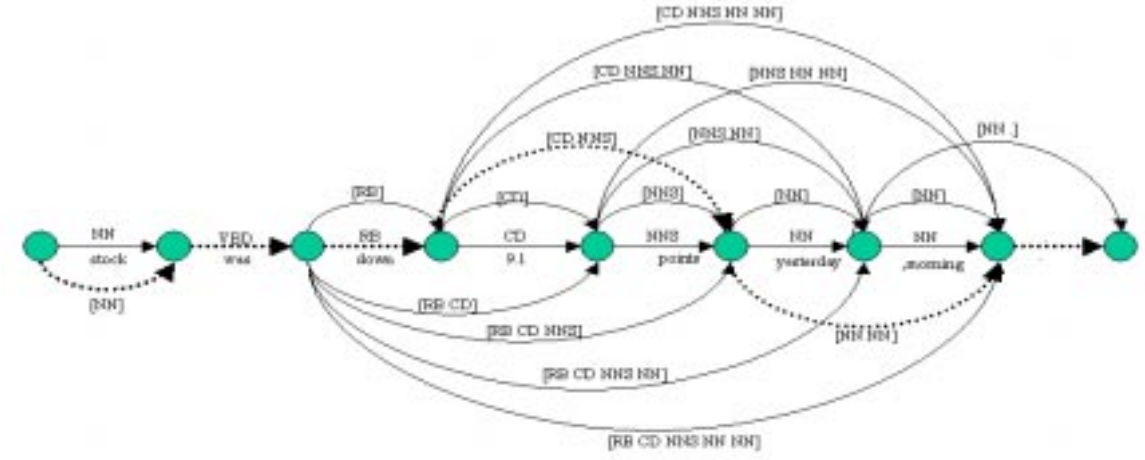

Figure 2: All possible brackets of "stock was down 9.1 points yesterday morning"

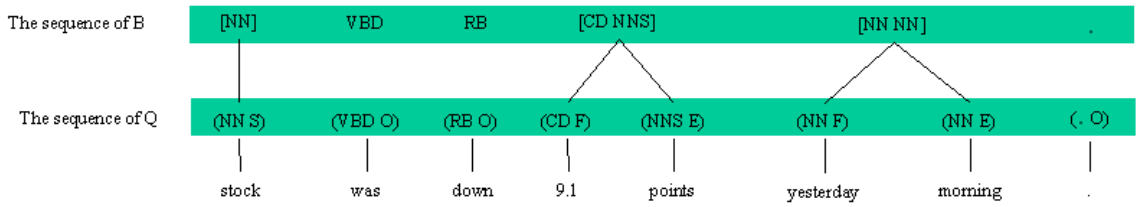

Figure 3: the transformed form of the path with dash line for the second pass processing

2.4 The statistical parameter training

In this work, the training and testing data were derived from the 25 sections of Penn Treebank. We divided the whole Penn Treebank data into two sections, one for training and the other for testing.

As required in our statistical model, we have to calculate the following four probabilities: (1) $P(t_i|t_{i-2},t_{i-1})$, (2) $P(w_i|t_i)$, (3) $P(n_i|n_{i-2}n_{i-1})$ and (4) $P(w_i|t_i,bm_i)$. The first and the third parameters are trigrams of T and B respectively. The second and the fourth are lexical generation probabilities. Probabilities

(1) and (2) can be calculated from POS tagged data with following formulae:

$$p(t_i \mid t_{i-2}, t_{i-1}) = \frac{count(t_{i-2} t_{i-1} t_i)}{\sum_j count(t_{i-2} t_{i-1} t_j)} \quad (13)$$

$$p(w_i \mid t_i) = \frac{count(w_i \; with \quad tag \quad t_i)}{count(t_i)} \quad (14)$$

As each sentence in the training set has both POS tags and baseNP boundary tags, it can be converted to the two sequences as B (a) and Q (b) described in the last section. Using these sequences, parameters (3) and (4) can be calculated, The calculation formulas are similar with equations (13) and (14) respectively.

Before training trigram model (3), all possible baseNP rules should be extracted from the training corpus. For instance, the following three sequences are among the baseNP rules extracted.

(1) *DT CD CD NNPS*
(2) *RB JJ NNS NNS*
(3) *NN NN POS NN*

\- - - - - -

There are more than 6,000 baseNP rules in the Penn Treebank. When training trigram model (3), we treat those baseNP rules in two ways. (1) Each baseNP rule is assigned a unique identifier (UID). This means that the algorithm considers the corresponding structure of each baseNP rule. (2) All of those rules are assigned to the same identifier (SID). In this case, those rules are grouped into the same class. Nevertheless, the identifiers of baseNP rules are still different from the identifiers assigned to POS tags.

We used the approach of Katz (Katz.1987) for parameter smoothing, and build a trigram model to predict the probabilities of parameter (1) and (3). In the case that unknown words are encountered during baseNP identification, we calculate parameter (2) and (4) in the following way:

In the experiments, the training and testing sets are derived from the 25 sections of Wall Street Journal distributed with the Penn Treebank II, and the definition of baseNP is the same as Ramshaw's, Table 1 summarizes the average performance on both baseNP tagging and POS tagging, each section of the whole Penn Treebank was used as the testing data and the other 24 sections as the training data, in this way

$$p(w_i \mid bm_i, t_i) = \frac{count(bm_i, t_i)}{\max_j (count(bm_j, t_i))^2} \quad (15)$$

$$p(w_i \mid t_i) = \frac{count(t_i)}{\max_j (count(t_j))^2} \quad (16)$$

Here, bm_j indicates all possible baseNP labels attached to t_i, and t_j is a POS tag guessed for the unknown word w_i.

3 Experiment result

We designed five experiments as shown in Table 1. "UID" and "SID" mean respectively that an identifier is assigned to each baseNP rule or the same identifier is assigned to all the baseNP rules. "+1" and "+4" denote the number of beat POS sequences retained in the first step. And "UID+R" means the POS tagging result of the given sentence is totally correct for the 2nd step. This provides an ideal upper bound for the system. The reason why we choose *N*=4 for the N-best POS tagging can be explained in Figure 4, which shows how the precision of POS tagging changes with the number *N*.

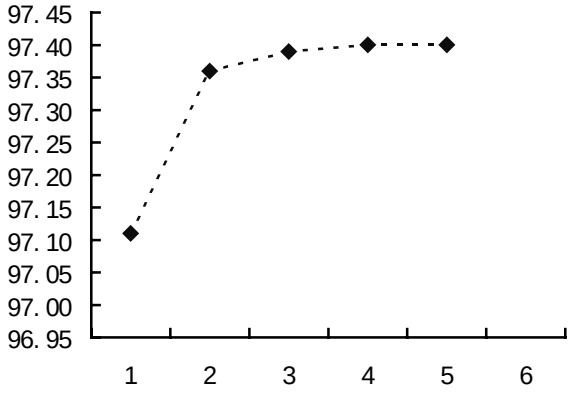

Figure 4: POS tagging precision with respect to different number of N-best

we have done the cross validation experiments 25 times.

	Precision (baseNP %)	Recall (baseNP %)	F-Measure (baseNP %)	$\frac{P+R}{2}$ (baseNP %)	Precision (POS %)
UID+1	92.75	93.30	93.02	93.02	97.06
UID+4	92.80	93.33	93.07	93.06	97.02
SID+1	86.99	90.14	88.54	88.56	97.06
SID+4	86.99	90.16	88.55	88.58	97.13
UID+R	93.44	93.95	93.69	93.70	100

Table 1 The average performance of the five experiments

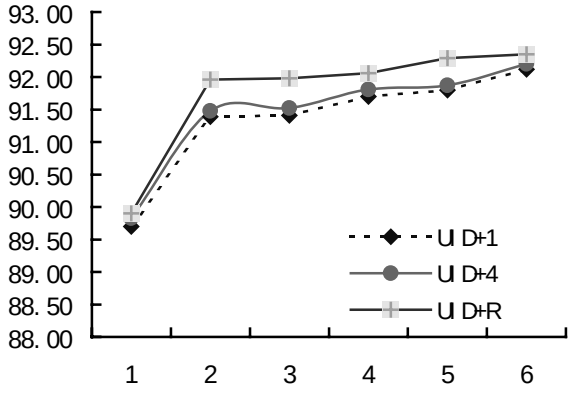

Figure 5: Precision under different training sets and different POS tagging results

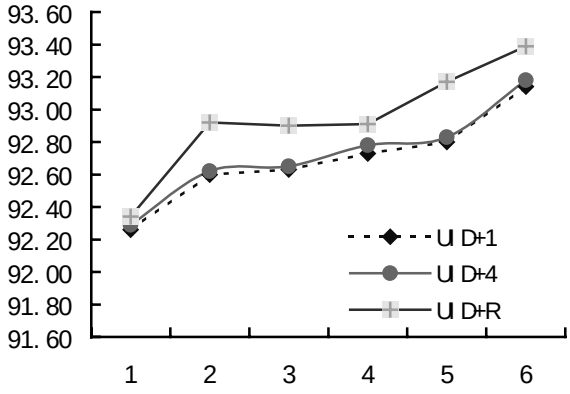

Figure 6: Recall under different training sets and different POS tagging results

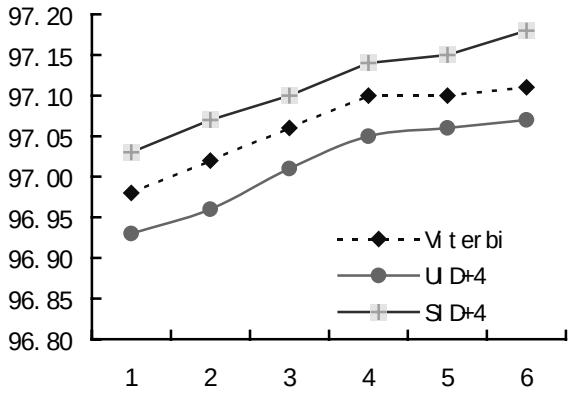

Figure 7: POS tagging precision under different training sets

Figure 5 -7 summarize the outcomes of our statistical model on various size of the training data, x-coordinate denotes the size of the training set, where "1" indicates that the training set is from section 0-8th of Penn Treebank, "2" corresponds to the corpus that add additional three sections 9-11th into "1" and so on. In this way the size of the training data becomes larger and larger. In those cases the testing data is always section 20 (which is excluded from the training data).

From Figure 7, we learned that the POS tagging and baseNP identification are influenced each other. We conducted two experiments to study whether the POS tagging process can make use of baseNP information. One is UID+4, in which the precision of POS tagging dropped slightly with respect to the standard POS tagging with Trigram Viterbi search. In the second experiment SID+4, the precision of POS tagging has increase slightly. This result shows that POS tagging can benefit from baseNP information. Whether or not the baseNP information can improve the precision of POS tagging in our approach is determined by the identifier assignment of the baseNP rules when training trigram model of $P(n_i \mid n_{i-2}, n_{i-1})$. In the future, we will further study optimal baseNP rules clustering to further improve the performances of both baseNP identification and POS tagging.

4 Comparison with other approaches

To our knowledge, three other approaches to baseNP identification have been evaluated using Penn Treebank-Ramshaw & Marcus's transformation-based chunker, Argamon et al.'s MBSL, and Cardie's Treebank_lex in Table 2, we give a comparison of our method with other these three. In this experiment, we use the testing data prepared by Ramshaw (available at http://www.cs.biu.ac.il/~yuvalk/MBSL), the training data is selected from the 24 sections of Penn Treebank (excluding the section 20). We can see that our method achieves better result than the others

.

	Transformation-Based (Training data: 200k)	Treebank_Lex	MBSL	Unified Statistical
Precision (%)	91.8	89.0	91.6	92.3
Recall (%)	92.3	90.9	91.6	93.2
F-Measure (%)	92.0	89.9	91.6	92.7
$\frac{P+R}{2}$	92.1	90.0	91.6	92.8

Table 2: The comparison of our statistical method with three other approaches

	Transforamtion-Based	Treebank_Lex	MBSL	Unified Statistical
Unifying POS & baseNP	NO	NO	NO	YES
Lexical Information	YES	YES	NO	YES
Global Searching	NO	NO	YES	YES
Context	YES	NO	YES	YES

Table 3: The comparison of some characteristics of our statistical method with three other approaches

Table 3 summarizes some interesting aspects of our approach and the three other methods. Our statistical model unifies baseNP identification and POS tagging through tracing N-best sequences of POS tagging in the pass of baseNP recognition, while other methods use POS tagging as a pre-processing procedure. From Table 1, if we reviewed 4 best output of POS tagging, rather that only one, the F-measure of baseNP identification is improved from 93.02 % to 93.07%. After considering baseNP information, the error ratio of POS tagging is reduced by 2.4% (comparing SID+4 with SID+1).

The transformation-based method (R&M 95) identifies baseNP within a local windows of sentence by matching transformation rules. Similarly to MBSL, the 2nd pass of our algorithm traces all possible baseNP brackets, and makes global decision through Viterbi searching. On the other hand, unlike MSBL we take lexical information into account. The experiments show that lexical information is very helpful to improve both precision and recall of baseNP recognition. If we neglect the probability of $\prod_{i=1}^{n} P(w_i \mid t_i, bm_i)$ in the 2nd pass of our model, the precision/recall ratios are reduced to 90.0/92.4% from 92.3/93.2%. Cardie's approach to Treebank rule pruning may be regarded as the special case of our statistical model, since the maximum-matching algorithm of baseNP rules is only a simplified processing version of our statistical model. Compared with this rule pruning method, all baseNP rules are kept in our model. Therefore in principle we have less likelihood of failing to recognize baseNP types

As to the complexity of algorithm, our approach is determined by the Viterbi algorithm approach, or $O(n)$, linear with the length.

5 Conclusions

This paper presented a unified statistical model to identify baseNP in English text. Compared with other methods, our approach has following characteristics:

(1) baseNP identification is implemented in two related stages: N-best POS taggings are first determined, then baseNPs are identified given the N best POS-sequences. Unlike other approaches that use POS tagging as pre-processing, our approach is not dependant on perfect POS-tagging, Moreover, we can apply baseNP information to further increase the precision of POS tagging can be improved. These experiments triggered an interesting future research challenge: how to cluster certain baseNP rules into certain identifiers so as to improve the precision of both baseNP and POS tagging. This is one of our further research topics.
(2) Our statistical model makes use of more lexical information than other approaches. Every word in the sentence is taken into account during baseNP identification.
(3) Viterbi algorithm is applied to make global search at the sentence level.
Experiment with the same testing data used by the other methods showed that the precision is 92.3% and the recall is 93.2%. To our knowledge, these results are comparable with or better than all previously reported results.

References

Eric Brill and Grace Ngai. (1999) *Man vs. machine: A case study in baseNP learning.* In Proceedings of the 18th International Conference on Computational Linguistics, pp.65-72. ACL'99

S. Argamon, I. Dagan, and Y. Krymolowski (1998) *A memory-based approach to learning shallow language patterns.* In Proceedings of the 17th International Conference on Computational Linguistics, pp.67-73. COLING-ACL'98

Cardie and D. Pierce (1998) *Error-driven pruning of treebank grammas for baseNP identification.* In Proceedings of the 36th International Conference on Computational Linguistics, pp.218-224. COLING-ACL'98

Lance A. Ramshaw and Michael P. Marcus (In Press). *Text chunking using transformation-based learning.* In Natural Language Processing Using Very large Corpora. Kluwer. Originally appeared in The second workshop on very large corpora WVLC'95, pp.82-94.

Viterbi, A.J. (1967) *Error bounds for convolution codes and asymptotically optimum decoding algorithm.* IEEE Transactions on Information Theory IT-13(2): pp.260-269, April, 1967

S.M. Katz.(1987) *Estimation of probabilities from sparse data for the language model component of speech recognize.* IEEE Transactions on Acoustics, Speech and Signal Processing. Volume ASSP-35, pp.400-401, March 1987

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, pages 136-143. Association of Computational Linguistics.

M. Marcus, M. Marcinkiewicx, and B. Santorini (1993) *Building a large annotated corpus of English: the Penn Treebank.* Computational Linguistics, 19(2): 313-330