Reducing Parsing Complexity by Intra–Sentence Segmentation based on Maximum Entropy Model

Sung Dong Kim, Byoung-Tak Zhang, Yung Taek Kim
School of Computer Science and Engineering,
Seoul National University, Korea
{sdkim,btzhang}@scai.snu.ac.kr, ytkim@cse.snu.ac.kr

Abstract

Long sentence analysis has been a critical problem because of high complexity. This paper addresses the reduction of parsing complexity by intra–sentence segmentation, and presents maximum entropy model for determining segmentation positions. The model features lexical contexts of segmentation positions, giving a probability to each potential position. Segmentation coverage and accuracy of the proposed method are 96% and 88% respectively. The parsing efficiency is improved by 77% in time and 71% in space.

1 Introduction

Long sentence analysis has been a critical problem in machine translation because of high complexity. In EBMT (example–based machine translation), the longer a sentence is, the less possible it is that the sentence has an exact match in the translation archive, and the less flexible an EBMT system will be (Cranias et al., 1994). In idiom–based machine translation (Lee, 1993), long sentence parsing is difficult because more resources are spent during idiom recognition phase as sentence length increases. A parser is often unable to analyze long sentences owing to their complexity, though they have no grammatical errors (Nasukawa, 1995).

In English–Korean machine translation, idiom–based approach is adopted to overcome the structural differences between two languages and to get more accurate translation. The parser is a chart parser with a capability of idiom recognition and translation, which is adapted to English–Korean machine translation. Idioms are recognized prior to syntactic analysis and the part of a sentence for an idiom takes an edge in a chart (Winograd, 1983). When parsing long sentences, an ambiguity of an idiom's range may cause more edges than the number of words included in the idiom (Yoon, 1994), which increases parsing complexity much. A parser of practical machine translation system should be able to analyze long sentences in a reasonable time.

Most context–free parsing algorithms have $O(n^3)$ parsing complexities in terms of time and space, where n is the length of a sentence (Tomita, 1986). Our work is motivated by the fact that parsing becomes more efficient, if n becomes shorter. This paper deals with the problem of parsing complexity by way of reducing the length of sentence to be analyzed. This reduction is achieved by **intra–sentence segmentation**, which is distinguished from **inter–sentence segmentation** that is used for text categorization (Beeferman et al., 1997) or sentence boundary identification (Palmer and Hearst, 1997)(Reynar and Ratnaparkhi, 1997). Intra–sentence segmentation plays a role as a preliminary step to a chart–based, context–free parser in English–Korean machine translation.

There have been several methods for reducing parsing complexities by intra–sentence segmentation. In (Lyon and Frank, 1995)(Lyon and Dickerson, 1997), they took advantage of the fact that the declarative sentences almost always consist of three segments: [*pre–subject* : *subject* : *predicate*]. The complexity could be reduced by decomposing a sentence into three sections. *Pattern rules* (Li et al., 1990) and *sentence patterns* (Kim and Kim, 1995) were used to segment long English sentences. They showed low segmentation coverage, which means that many of long sentences are not segmented by the pattern rules or sentence patterns. And they require much human efforts to construct pattern rules or collect sentence patterns. These factors may prevent them being applicable to practical machine translation systems.

This paper presents a trainable model for identifying potential segmentation positions

in a sentence and determining appropriate segmentation positions. Given a corpus annotated with segmentation positions, our model automatically learns the contextual evidences about segmentation positions, which relieves human of burden to construct pattern rules or sentence patterns. These evidences are combined under the maximum entropy framework (Jaynes, 1957) to estimate the probability for each position. By intra–sentence segmentation based on the proposed model, we achieve more improved parsing efficiency by 77% in time and 71% in space.

In Section 2 we introduce the maximum entropy model. Section 3 describes features incorporated into the model and the process of identifying potential segmentation positions. The determination schemes of segmentation positions are described in Section 4. Segmentation performance of the model is presented with the degree of contribution to efficient parsing by the segmentation in Section 5. We also compare our approach with other intra–sentence segmentation approaches. Section 6 draws conclusions and presents some further works.

2 Maximum Entropy Modeling

Sentence patterns or pattern ruels specify the sub–structures of the sentences. That is, segmentation positions are determined in view of the global sentence structure. If there is no matched rules or patterns with a given sentence, the sentence could not be segmented. We assume that whether a word is a segmentation position depends on its surrounding context. We try to find factors that affect the determination of segmentation positions. Maximum entropy is a technique for automatically acquiring knowledge from incomplete information, without making any unsubstantiated assumptions. It masters subtle effects so that we may accurately model subtle dependencies. It does not make any unwarranted assumptions, which means that maximum entropy learns exactly what the data says. Therefore it can perform well on unseen data.

The idea is to construct a model that assigns a probability to each potential segmentation position in a sentence. We build a probability distribution $p(y|x)$, where $y \in \{0, 1\}$ is a random variable specifying the potential segmentation position in a context x. A **feature** of a context is a binary–valued indicator function f expressing the information about a specific context.

Given a training sample of size N, $(x_1, y_1), \ldots, (x_N, y_N)$, an **empirical probability distribution** can be defined as

$$\tilde{p}(x, y) = \frac{\#(x, y)}{N},$$

where $\#(x, y)$ is the number of occurrences of (x, y). The expected value of feature f_i with respect to the empirical distribution $\tilde{p}(x, y)$ is expressed as

$$\tilde{p}(f_i) \equiv \sum_{x,y} \tilde{p}(x, y) f_i(x, y)$$

and the expected value of f_i with respect to the probability distribution $p(y|x)$ is

$$p(f_i) \equiv \sum_{x,y} \tilde{p}(x) p(y|x) f_i(x, y),$$

where $\tilde{p}(x)$ is the empirical distribution of x in the corpus. We want to build probability distribution $p(y|x)$ that is required to accord to the feature f_i useful in selecting segmentation positions: $p(f_i) = \tilde{p}(f_i)$ for all $f_i \in \mathcal{F}$, where $\mathcal{F}$ is the set of candidate features. This makes the probability distribution be built on only training data.

Given a feature set $\mathcal{F}$, let $\mathcal{C}$ be the subset of all distributions $\mathcal{P}$ that satisfies the requirement $p(f_i) = \tilde{p}(f_i)$:

$$\mathcal{C} \equiv \{p \in \mathcal{P} \mid p(f_i) = \tilde{p}(f_i), \text{ for all } f_i \in \mathcal{F}\}. \quad (1)$$

We choose a probability distribution consistent with all the facts, but otherwise as uniform as possible. The uniformity of the probability distribution $p(y|x)$ is measured by the conditional entropy:

$$H(p) = -\sum_{x,y} p(x, y) \log p(y|x)$$

$$\equiv -\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x).$$

Thus, the probability distribution with maximum entropy is the most uniform distribution.

In building a model, we consider the linear exponential family $\mathcal{Q}$ given as

$$\mathcal{Q}(f) = \{p(y|x) = \frac{1}{Z_\lambda(x)} \exp(\sum_i \lambda_i f_i(x, y))\}, \quad (2)$$

where λ_i are real–valued parameters and $Z_\lambda(x)$ is a normalizing constant:

$$Z_\lambda(x) = \sum_y \exp(\sum_i \lambda_i f_i(x,y)).$$

An intersection of the class $\mathcal{Q}$ of exponential models with the class of desired distribution (1) is nonempty, and the intersection contains the maximum entropy distribution and furthermore it is unique (Ratnaparkhi, 1994).

Finding $p_* \in \mathcal{C}$ that maximizes $H(p)$ is a problem in constrained optimization, which cannot be explicitly written in general. Therefore, we take advantage of the fact that the models in $\mathcal{Q}$ that satisfy $p(f_i) = \tilde{p}(f_i)$ can be explained under the maximum likelihood framework (Ratnaparkhi, 1994). Maximum likelihood principle also gives the unique distribution p_*, the intersection of the class $\mathcal{Q}$ with $\mathcal{C}$.

We assume each occurrence of (x, y) is sampled independently. Thus, log–likelihood $L_{\tilde{p}}(p)$ of the empirical distribution $\tilde{p}$ as predicted by a model p can be defined as

$$L_{\tilde{p}}(p) \equiv \log \prod_{x,y} p(y|x)^{\tilde{p}(x,y)} = \sum_{x,y} \tilde{p}(x,y) \log p(y|x).$$

That is, the model we want to build is

$$p_* = \arg\max_{p \in \mathcal{C}} H(p) = \arg\max_{q \in \mathcal{Q}} L_{\tilde{p}}(q).$$

The parameters λ_i of exponential model (2) are obtained by the *Generalized Iterative Scaling* algorithm (Darroch and Ratcliff, 1972).

3 Construction of Features

This section describes the features. From a corpus, contextual evidences of segmentation positions are collected and combined, resulting in features. The features are used in identifying potential segmentation positions and included in the model.

3.1 Segmentable Positions and Safe Segmentation

A sentence is constructed by the combination of words, phrases, and clauses under the well–defined grammar. A sentence can be segmented into shorter segments that correspond to the constituents of the sentence. That is, segments correspond to the nonterminal symbols of the context–free grammar[1]. The position of a word is called **segmentable position** that can be a starting position of a specific segment.

Though the analysis complexity can be reduced by segmenting a sentence, there is a mis–segmentation risk that causes parsing failures. A segmentation can be called **safe segmentation** that results in a coherent blocks of words. In English–Korean translation, safe segmentation is defined as the one which generates safe segments. A segment is safe, when there is a syntactic category symbol N^P dominating the segment and the segment can be combined with adjacent segments under a given grammar. In Figure 1, (a) is an unsafe segmentation because the second segment cannot be analyzed into one syntactic category, resulting in parsing failure. By the safe segmentation (b), the first segment corresponds to a noun phrase and the second to a verb phrase, so that we can get a correct analysis result.

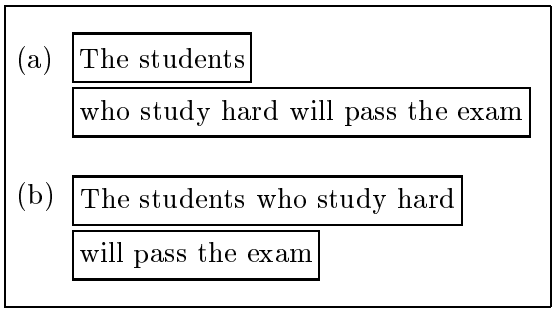

Figure 1: Examples of unsafe/safe segmentation in English–Korean translation.

3.2 Lexical Contextual Constraints

A **lexical context** of a *word* includes seven–word window: three to the left of a *word* and three to the right of a *word* and a *word* itself. It also includes the part–of–speeches of these words, subcategorization information for two words to the left, and position value. The position value $posi\_v$ of the ith word w_i is calculated as

$$posi\_v = \lceil \frac{i}{n} \times R \rceil,$$

where n is the number of words and R[2] represents the number of regions in the sentence. Region is the sequentially ordered block of

[1] Nonterminal symbols include the ones for phrases, such as NP (noun phrase) and VP (verb phrase), and the ones for clauses like RLCL (relative clause), SUBCL (subordinate clause).

[2] It is a heuristically set value, and we set R as 4.

words in a sentence, and *posi_v* represents the region in which a word lies. It is included to reflect the influence of the position of a word on being a segmentation position. Thus, the lexical context of a word is represented by 17 attributes as shown in Figure 2.

$s\_position?$
$word_i$
$w_{i-3}, \ldots, w_{i+3}$
$p_{i-3}, \ldots, p_{i+3}$
$s\_cat_{i-2},\ s\_cat_{i-1}$
$posi\_v$

Figure 2: The structure of lexical context.

An example of a training data and a resulting lexical context is shown in Figure 3. A symbol '#' represents a segmentation position marked by human annotators. Therefore, the lexical context of word *when* includes the value 1 for attribute *s_position*? and followings: three words to the left of *when* (*became*, *terribly*, and *worried*) and part–of–speeches of each word (VERB ADV ADJ), three words to the right (*they*, *saw*, and *what*) and part–of–speeches (PRON VERB PRON), subcategorization information for two words to the left (0 1), and position value (2).

Of course his parents became terribly worried **#when** they saw what was happening to Atzel.

(1 **when** became terribly worried they saw what VERB ADV ADJ PRON VERB PRON 0 1 2)

Figure 3: An example of a training data and a lexical context.

To get reliable statistics, much training data is required. To alleviate this problem, we generate **lexical contextual constraints** by combining lexical contexts and collect statistics for them. To generate lexical contextual constraints and to identify segmentable positions, we define two operations *join* ($\oplus$) and *consistency* ($\equiv$). Let $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ be lexical contexts and $(C_1, \ldots, C_n)$ be lexical contextual constraint. The operation *join* is defined as

$$(a_1, \ldots, a_n) \oplus (b_1, \ldots, b_n) = (C_1, \ldots, C_n),$$

$$C_i = \begin{cases} '*' & \text{if} \quad a_i \neq b_i \\ a_i & \text{if} \quad a_i = b_i \end{cases},$$

where $'*'$ is *don't–care term* accepting any value. A lexical contextual constraint is generated as a result of *join* operation. The *consistency* is defined as

$$\Big((a_1, \ldots, a_n) \equiv (C_1, \ldots, C_n)\Big) = k,$$

$$k = \begin{cases} 1 & \text{if} \quad (C_i = a_i \text{ or } C_i =' *') \text{ for all } 1 \leq i \leq n \\ 0 & \text{otherwise} \end{cases}$$

The algorithm for generating lexical contextual constraints is shown in Figure 4.

- Input: a set of active lexical contexts
  $LC_w = \{lc_1 \ldots lc_n\}$ for word w,
  where $lc_i = (a_1, \ldots, a_n)$.
- Output: a set of lexical contextual constraints $LCC_w = \{lcc_1 \ldots lcc_k\}$,
  where $lcc_i = (C_1, \ldots, C_n)$.

1. Initialize $LCC_w = \emptyset$
2. Do the followings for each $lc_i \in LC_w$

(a) For all $lc_j (j \neq i)$, $Count(lc_j)$ = # of matched attributes with lc_i

(b) $max\_cnt = \arg\max_{lc_j \in LC_w} Count(lc_j)$

(c) For all lc_j, where $Count(lc_j) = max\_cnt$,
$lcc = lc_i \oplus lc_j, \quad LCC_w \leftarrow LCC_w \cup \{lcc\}$

Figure 4: Algorithm for generating lexical contextual constraints.

A *lcc* plays the role of a feature. Following is an example of a feature.

$$f(x, y) = \begin{cases} 1 & \text{if } x_{word} = \text{“}that\text{”} \text{ and} \\ & x_{i-1} = \text{“}say\text{”} \text{ and } y = 1 \\ 0 & \text{otherwise} \end{cases}$$

We collect the statistics for each *lcc*. The frequency of each *lcc* is counted as the number of lexical contexts that satisfy the *consistency* operation with the *lcc*.

$$\#(lcc) = \sum_{i=1}^{n} \Big(lc_i \equiv lcc\Big).$$

Identifying segmentable positions is performed with the *consistency* operation with the lexical context of word w and $lcc \in LCC_w$. The word whose lexical context is consistent with lcc is identified as a segmentable position.

4 Determination Schemes of Segmentation Positions

Segmentation positions are determined through two steps: identifying segmentable positions and selecting the most appropriate position among them. Segmentable positions are identified using the *consistency* operation. Maximum entropy model in Section 2 gives a probability to each position.

Segmentation performance is measured in terms of coverage and accuracy. Coverage is the ratio of the number of actually segmented sentences to the number of segmentation target sentences that are longer than α words, where α is a fixed constant distinguishing long sentences from short ones. Accuracy is evaluated in terms of the safe segmentation ratio. They are defined as follows:

$$coverage = \frac{\text{\# of actually segmented Sent.}}{\text{\# of Sent. to be segmented}} \quad (3)$$

$$accuracy = \frac{\text{\# of Sent. with safe segmentation}}{\text{\# of actually segmented Sent.}} \quad (4)$$

4.1 Baseline Scheme

No contextual information is used in identifying segmentable positions. They are empirically identified. A word that is tagged as a segmentation position more than 5 times is identified as a segmentable position. A set of segmentable positions, $\mathcal{D}$, is as follows.

$$\mathcal{D} = \{w_i \mid w_i \text{ is tagged as segmentation position and } \#(\text{tagged } w_i) \geq 5\}$$

In order to select the most appropriate position, the segmentation appropriateness of each position is evaluated by the probability of word w_i:

$$p(w_i) = \frac{\text{\# of tagged } w_i}{\text{\# of } w_i \text{ in the corpus}}$$

$p(w_i)$ represents the tendency that word w_i will be used as a segmentation position. A segmentation position w_* is selected as the one that has highest $p(w_i)$ value:

$$w_* = \arg\max_{w_i \in \mathcal{D}} p(w_i).$$

This scheme serves as a baseline for comparing the segmentation performance of the models.

4.2 A Scheme using Lexical Contextual Constraints

Lexical contextual constraints are used in identifying segmentable positions. Compared with the baseline scheme, this scheme considers contextual information of a word. All consistent words with the defined lexical contextual constraints form a set of segmentable positions $\mathcal{D}$.

$$\mathcal{D} = \{w_i \mid (lc_{w_i} \equiv lcc_{w_i}) = 1\}.$$

The maximum likelihood principle gives a probability distribution for $p(y \mid lcc_{w_i})$, where $y \in \{0, 1\}$. Segmentation appropriateness is evaluated by $p(1 \mid lcc_{w_i})$. A position with the highest $p(1 \mid lcc_{w_i})$ becomes a segmentation position:

$$w_* = \arg\max_{w_i \in \mathcal{D}} p(1 \mid lcc_{w_i}).$$

4.3 A Scheme using Lexical Contextual Constraints with Word Sets

Due to insufficient training samples for constructing lexical contextual constraints, some segmentable positions may not be identified. To alleviate this problem we introduce **word sets** whose elements have linguistically similar features. We define four word sets: *coordinate conjunction set, subordinate conjunction set, interogative set, auxiliary verb set*. The categories of word sets and the examples of their members are shown in Table 1.

Table 1: The word sets and examples.

Word Set	Examples
Coordinate Conjunctions	*and, or, but*
Subordinate Conjunctions	*if, when, ...*
Interogatives	*how, what, ...*
Auxiliary Verbs	*can, should, ...*

Coordinate conjunctions have only 3 members, but they frequently apprear in long sentences. Subordinate conjunctions have 25

members, interogatives 5 members, and auxiliary verbs have 12 members now. The words belonging to each word set are treated equally. Lexical contextual constraints are constructed for words and word sets, so the statistics is collected for both of them. The set of segmentable positions $\mathcal{D}$ is defined somewhat differently as:

$$\mathcal{D} = \{w_i, ws_j \mid (lc_{w_i} \equiv lcc_{w_i}) = 1$$

$$\text{or} \quad (lc_{ws_j} \equiv lcc_{ws_j}) = 1\},$$

where ws_j denotes a word set to which the jth word in a sentence belongs.

In this scheme, $p(1 \mid lcc_{w_i})$ or $p(1 \mid lcc_{ws_i})$ expresses the segmentation appropriateness of the position. Therefore, a segmentation position is determined by

$$w_* = \arg \max_{\{w_i, ws_j\} \in \mathcal{D}} \{p(1 \mid lcc_{w_i}),\ p(1 \mid lcc_{ws_j})\}.$$

5 Experiments

5.1 Corpus and Construction of the Maximum Entropy Model

We construct the corpus from two different domains, where the sentences longer than 15 words are extracted[3]. The training portion is used to generate lexical contextual constraints and to collect statistics for maximum entropy model construction. From high school English texts, 1500 sentences are tagged with segmentation positions by human. Two people who have some knowledge about English syntactic structures read sentences, and marked words as segmentation positions where they paused.

After generating lexical contextual constraints, we constructed the maximum entropy model $p(y|x)$, where x is a lexical contextual constraint and $y \in \{0, 1\}$. The model incorporates features that occur more than 5 times in the training data. 3626 candidate features were generated without word sets and 3878 features with word sets. In Table 2, training time and the number of active features of the model are shown.

Segmentation performance is evaluated using test portion that consists of 1800 sentences from two domains: high school English texts and the Byte Magazine.

[3]The sentences with commas are excluded because comma is an explicit segmentation position. Segments resulting from a segmentation at commas may be the manageable–sized ones. Our work is to segment long sentences without explicit segmentation positions.

Table 2: Construction of models.

	Training Time	# of Active Features
Without Word Sets	10 min	2720
With Word Sets	12 min	2910

5.2 Segmentation Performance

In addition to coverage and accuracy, SC value is also defined to express the degree of contribution to efficient parsing by segmentation. It is the ratio of the sentences that can benefit from intra–sentence segmentation. If a long sentence is not segmented or is segmented at unsafe segmentation positions, the sentence is called a **segmentation error sentence**. SC value is calculated as

$$SC = 1 - \frac{\#\text{ of segmentation error sentences}}{\#\text{ of segmentation target sentences}}.$$

A sentence longer than α words is considered as the segmentation target sentence, where α is set to 12. Table 3 compares segmentation performance for each determination scheme.

Table 3: Segmentation performance of the determination schemes of segmentation position.

Determination Schemes	Coverage/ Accuracy (%)	SC
Baseline	100/77.6	0.776
LCC	90.7/89	0.808
LCC with Word Sets	95.8/87.9	0.865

By the comparison of the baseline scheme with others, the accuracy is observed to depend on the context information. Word sets are helpful for increasing coverage with less degradation of accuracy. Each scheme has superiority in terms of the different measures. But in terms of applicability to practical systems, the third scheme is best for our purpose. Table 4 shows the segmentation performance of the scheme using LCC with word sets.

SC value for the sentences from the same domain as training data is about 0.88, and

Table 4: Segmentation performance of LCC with word sets.

Domain	Sent. Length	Coverage/ Accuracy(%)	SC
High–School English Text	15∼19	99.0/95.9	0.95
	20∼24	100/94.0	0.94
	25∼29	96.0/81.3	0.78
	30∼	100/67.5	0.68
Byte Magazine	15∼19	94.0/92.6	0.87
	20∼24	91.0/91.2	0.83
	25∼29	92.5/94.6	0.88
	30∼	93.5/86.1	0.81
Total	1800	95.8/87.9	0.87

about 0.85 for the sentenes from the Byte Magazine. Though they slightly differ between test domains, about 87% of long sentences can be parsed with less complexity and without causing parsing failures. It suggests that the intra–sentence segmentation method can be utilized for efficient parsing of the long sentences.

5.3 Parsing Efficiency

Parsing efficiency is generally measured by the required time and memory for parsing. In most cases, parsing sentences longer than 30 words could not complete without intra–sentence segmentation. Therefore, the parsing is performed for the sentences longer than 15 and less than 30 words. Ultra–Sparc 30 machine is used for experiments. The efficiency improvement was measured by

$$EI_{time} = \frac{t_{unseg} - t_{seg}}{t_{unseg}} \times 100,$$

$$EI_{memory} = \frac{m_{unseg} - m_{seg}}{m_{unseg}} \times 100,$$

where t_{unseg} and m_{unseg} are time and memory during parsing without segmentation and t_{seg}, m_{seg} are for the parsing with segmentation. Table 5 summarizes the results.

By segmenting long sentences into several manageable–sized segments, we can parse long sentences with much less time and space.

5.4 Comparison with Related Works

The intra–sentence segmentation method based on the maximum entropy model is compared with other approaches in terms of the segmentation coverage and the improvement of parsing efficiency.

Table 5: Comparison of parsing efficiency with/without segmentation.

	High-School English Text	Byte Magazine
With Segmentation	4.6 sec	5.4 sec
	0.9 MB	1.1 MB
Without Segmentation	19.6 sec	25.1 sec
	3.4 MB	3.7 MB
Improvement	76.5%	78.5%
	73.5%	70.3%

In (Lyon and Frank, 1995)(Lyon and Dickerson, 1997), a sentence is segmented into three segments. Though parsing efficiency can be improved by segmenting a sentence, this method may be applied to only simple sentences[4]. Long sentences are generally coordinate sentences[5] or complex sentences[6]. They have more than two subjects, so applying this method to such sentences seems to be inappropriate.

In (Kim and Kim, 1995), sentence patterns are used to segment long sentences. This method improve parsing efficiency by 30% in time and 58% in space. However collecting sentence patterns requires much human efforts and segmentation coverage is only about 36%.

Li's method (Li et al., 1990) for sentence segmentation also depends upon manual–intensive pattern rules. Segmentation coverage seems to be unsatisfactory for practical machine translation system.

The proposed method can be applied to coordinate and complex sentences as well as simple sentences. It shows segmentation coverage of about 96%. In addition, it needs no other human efforts except for constructing training data. Human annotators have only to read sentences and mark segmentation positions, which is more simple than collecting pattern rules or sentence patterns. We can also get much improved parsing efficiency: about 77% in time and about 71% in space.

[4] A simple sentence has one subject and one predicate.

[5] A coordinate sentence results from the combination of several simple sentences by the coordinate conjunctions.

[6] A complex sentence consists of a main clause and several subordinate clauses.

6 Conclusion and Future Work

Practical machine translation systems should be able to accommodate long sentences. Thus intra–sentence segmentation is required as a means for reducing parsing complexity. This paper presents a method for intra–sentence segmentation based on the maximum entropy model. The method builds statistical models automatically from a text corpus to provide the segmentation appropriateness for safe segmentation.

In the experiments with 1800 test sentences, about 87% of them were benefited from segmentation. The statistical intra–sentence segmentation method can also relieve human of the burden of constructing information, such as segmentation rules or sentence patterns. Experiments suggest that the proposed maximum entropy models can be incorporated into the parser for practical machine translation systems.

Further works can be done in two directions. First, studies on recovery mechanisms for unsafe segmentation before parsing seem necessary since unsafe segmentation may cause parsing failures. Second, parsing control mechanisms should be studied that exploit the characteristics of segmentation positions and the parallelism among segments. This will enhance parsing efficiency further.

References

D. Beeferman, A. Berger, and J. Lafferty. 1997. Text Segmentation using Exponential Models. In *Second Conference on Empirical Methods in Natural Language Processing*. Providence, RI.

Lambros Cranias, Harris Papageorgiou, and Stelios Piperdis. 1994. A Matching Technique in Example-Based Machine Translati on. In *Proceedings of 1994 COLING*, pages 100–104.

J.N. Darroch and D. Ratcliff. 1972. Generalized Iterative Scaling for Log-linear Models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

E.T. Jaynes. 1957. Information Theory and Statistical Mechanics. *Physical Review*, 106:620–630.

Sung Dong Kim and Yung Taek Kim. 1995. Sentence Analysis using Pattern Matching in English-Korean Machine Translation. In *Proceedings of the 1995 ICCPOL*, Oct. 25-28.

Ho Suk Lee. 1993. *Automatic Construction of Transfer Dictionary based on the C orpus for English-Korean Machine Translation*. Ph.D. thesis, Seoul National University. In Korean.

Wei-Chuan Li, Tzusheng Pei, Bing-Huang Lee, and Chuei-Feng Chiou. 1990. Parsing Long English Sentences with Pattern Rules. In *Proceedings of 25th Conference of COLING*, pages 410–412.

Caroline Lyon and Bob Dickerson. 1997. Reducing the Complexity of Parsing by a Method of Decomposition. In *International Workshop on Parsing Technology*, September.

Caroline Lyon and Ray Frank. 1995. Neural Network Design for a Natural Language Parser. In *International Conference on Artificial Neural Networks*.

Tetsura Nasukawa. 1995. Robust Parsing Based on Discourse Information. In *33rd Annual Meeting of the ACL*, pages 33–46.

David D. Palmer and Marti A. Hearst. 1997. Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics*, 23(2):241–265.

A. Ratnaparkhi. 1994. A Simple Introduction to Maximum Entropy Models for Natural Language Processing. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania 3401 Walnut Street, Suite 400A Philadelphia, PA 19104-6228, May. IRCS Report 97–08.

J.C. Reynar and A. Ratnaparkhi. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19. Washington D.C.

Masaru Tomita. 1986. *Efficient Parsing for Natural Language*. Kluwer Academic Publishers.

T. Winograd. 1983. *Language as a Cognitive Process: Syntax*, volume 1. Addison-Wesley.

Sung Hee Yoon. 1994. Efficient Parser to Find Bilingual Idiomatic Expressions for English-Korean Machine Translation. In *Proceedings of the 1994 ICCPOL*, pages 455–460.