A Syntax-Free Approach to Japanese Sentence Compression

Tsutomu HIRAO, Jun SUZUKI and **Hideki ISOZAKI**
NTT Communication Science Laboratories, NTT Corp.
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{hirao,jun,isozaki}@cslab.kecl.ntt.co.jp

Abstract

Conventional sentence compression methods employ a syntactic parser to compress a sentence without changing its meaning. However, the reference compressions made by humans do not always retain the syntactic structures of the original sentences. Moreover, for the goal of on-demand sentence compression, the time spent in the parsing stage is not negligible. As an alternative to syntactic parsing, we propose a novel term weighting technique based on the positional information within the original sentence and a novel language model that combines statistics from the original sentence and a general corpus. Experiments that involve both human subjective evaluations and automatic evaluations show that our method outperforms Hori's method, a state-of-the-art conventional technique. Because our method does not use a syntactic parser, it is 4.3 times faster than Hori's method.

1 Introduction

In order to compress a sentence while retaining its original meaning, the subject-predicate relationship of the original sentence should be preserved after compression. In accordance with this idea, conventional sentence compression methods employ syntactic parsers. English sentences are usually analyzed by a full parser to make parse trees, and the trees are then trimmed (Knight and Marcu, 2002; Turner and Charniak, 2005; Unno et al., 2006). For Japanese, dependency trees are trimmed instead of full parse trees (Takeuchi and Matsumoto, 2001; Oguro et al., 2002; Nomoto, 2008)[1] This parsing approach is reasonable because the compressed output is grammatical if the input is grammatical, but it offers only moderate compression rates.

An alternative to the tree trimming approach is the sequence-oriented approach (McDonald, 2006; Nomoto, 2007; Clarke and Lapata, 2006; Hori and Furui, 2003). It treats a sentence as a sequence of words and structural information, such as a syntactic or dependency tree, is encoded in the sequence as features. Their methods have the potential to drop arbitrary words from the original sentence without considering the boundary determined by the tree structures. However, they still rely on syntactic information derived from fully parsed syntactic or dependency trees.

We found that humans usually ignored the syntactic structures when compressing sentences. For example, in many cases, they compressed the sentence by dropping intermediate nodes of the syntactic tree derived from the source sentence. We believe that making compression strongly dependent on syntax is not appropriate for reproducing reference compressions. Moreover, on-demand sentence compression is made problematic by the time spent in the parsing stage.

This paper proposes a syntax-free sequence-oriented sentence compression method. To maintain the subject-predicate relationship in the compressed sentence and retain fluency without using syntactic parsers, we propose two novel features: intra-sentence positional term weighting (IPTW) and the patched language model (PLM). IPTW is defined by the term's positional information in the original sentence. PLM is a form of summarization-oriented fluency statistics derived from the original sentence and the general language model. The weight parameters for these features are optimized within the Minimum Classification Error (MCE) (Juang and Katagiri, 1992) learning framework.

Experiments that utilize both human subjective and automatic evaluations show that our method is

[1] Hereafter, we refer these compression processes as "tree trimming."

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 826–833,
Suntec, Singapore, 2-7 August 2009.

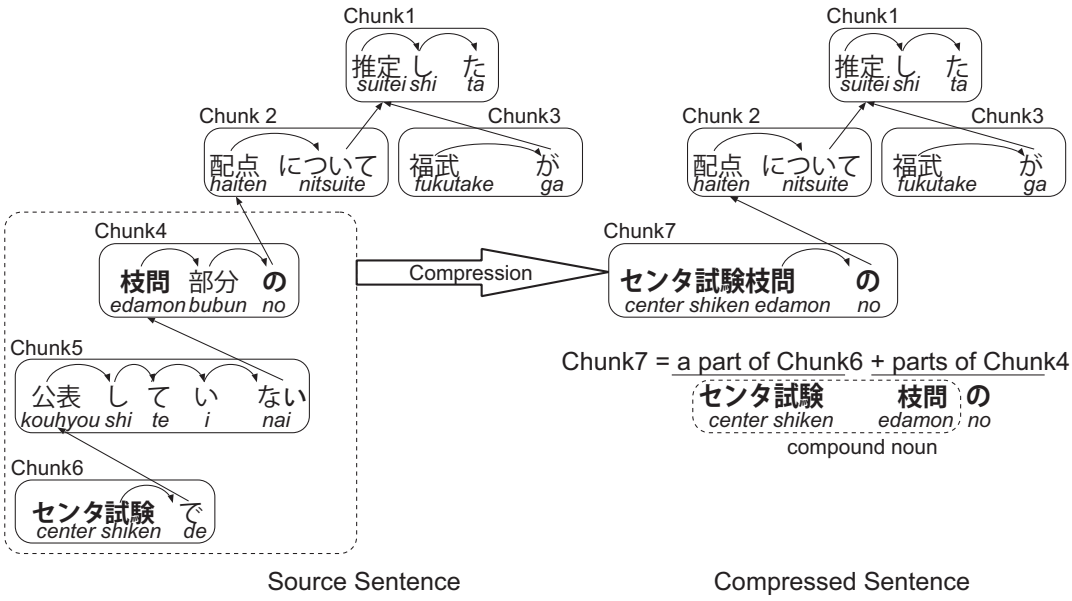

Figure 1: An example of the dependency relation between an original sentence and its compressed variant.

superior to conventional sequence-oriented methods that employ syntactic parsers while being about 4.3 times faster.

2 Analysis of reference compressions

Syntactic information does not always yield improved compression performance because humans usually ignore the syntactic structures when they compress sentences. Figure 1 shows an example. English translation of the source sentence is "Fukutake Publishing Co., Ltd. presumed preferential treatment with regard to its assessed scores for a part of the questions for a series of Center Examinations." and its compression is "Fukutake presumed preferential scores for questions for a series of Center Examinations."

In the figure, each box indicates a syntactic chunk, *bunsetsu*. The solid arrows indicate dependency relations between words[2]. We observe that the dependency relations are changed by compression; humans create compound nouns using the components derived from different portions of the original sentence without regard to syntactic constraints. 'Chunk 7' in the compressed sentence was constructed by dropping both content and functional words and joining other content words contained in 'Chunk 4' and 'Chunk 6' of the original sentence. 'Chunk 5' is dropped completely. This compression cannot be achieved by tree trimming.

According to an investigation in our corpus of manually compressed Japanese sentences, which we used in the experimental evaluation, 98.7% of them contain at least one segment that does not retain the original tree structure. Human usually compress sentences by dropping the intermediate nodes in the dependency tree. However, the resulting compressions retain both adequacy and fluency. This statistic supports the view that sentence compression that strongly depends on syntax is not useful in reproducing reference compressions. We need a sentence compression method that can drop intermediate nodes in the syntactic tree aggressively beyond the tree-scoped boundary.

In addition, sentence compression methods that strongly depend on syntactic parsers have two problems: 'parse error' and 'decoding speed.' 44% of sentences output by a state-of-the-art Japanese dependency parser contain at least one error (Kudo and Matsumoto, 2005). Even more, it is well known that if we parse a sentence whose source is different from the training data of the parser, the performance could be much worse. This critically degrades the overall performance of sentence compression. Moreover, summarization systems often have to process megabytes of documents. Parsers are still slow and users of on-

[2]Generally, a dependency relation is defined between *bunsetsu*. Therefore, in order to identify word dependencies, we followed Kudo's rule (Kudo and Matsumoto, 2004)

demand summarization systems are not prepared to wait for parsing to finish.

3 A Syntax Free Sequence-oriented Sentence Compression Method

As an alternative to syntactic parsing, we propose two novel features, intra-sentence positional term weighting (IPTW) and the patched language model (PLM) for our syntax-free sentence compressor.

3.1 Sentence Compression as a Combinatorial Optimization Problem

Suppose that a compression system reads sentence $\boldsymbol{x}= x_1\ , x_2,\ \ldots,\ x_j,\ \ldots,\ x_N$, where x_j is the j-th word in the input sentence. The system then outputs the compressed sentence $\boldsymbol{y}$ $=y_1,\ y_2,\ \ldots, y_i,\ \ldots,\ y_M$, where y_i is the i-th word in the output sentence. Here, $y_i \in \{x_1, \ldots, x_N\}$. We assume y_0=x_0=<s> (BOS) and y_{M+1}=x_{N+1}=</s> (EOS). We define function $I(\cdot)$, which maps word y_i to the index of the word in the original sentence. For example, if source sentence is $\boldsymbol{x} = x_1, x_2, \ldots, x_5$ and its compressed variant is $\boldsymbol{y} = x_1, x_3, x_4$, $I(y_1) = 1$, $I(y_2) = 3$, $I(y_3) = 4$.

We define a significance score $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\Lambda})$ for compressed sentence $\boldsymbol{y}$ based on Hori's method (Hori and Furui, 2003). $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_h\}$ is a parameter vector.

$$f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Lambda}) = \sum_{i=1}^{M+1} \{g(\boldsymbol{x}, I(y_i); \boldsymbol{\lambda}_g) \quad + \quad h(\boldsymbol{x}, I(y_i), I(y_{i-1}); \boldsymbol{\lambda}_h)\} \tag{1}$$

The first term of equation (1) ($g(\cdot)$) is the importance of each word in the output sentence, and the second term ($h(\cdot)$) is the the linguistic likelihood between adjacent words in the output sentence.

The best subsequence $\hat{\boldsymbol{y}}= \underset{\boldsymbol{y}}{\operatorname{argmax}}\, f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Lambda})$ is identified by dynamic programming (DP) (Hori and Furui, 2003).

3.2 Features

We use IPTW to define the significance score $g(\boldsymbol{x}, I(y_i); \boldsymbol{\lambda_g})$. Moreover, we use PLM to define the linguistic likelihood $h(\boldsymbol{x}, I(y_{i+1}), I(y_i); \boldsymbol{\lambda}_h)$.

3.2.1 Intra-sentence Positional Term Weighting (IPTW)

IDF is a global term weighting scheme in that it measures the significance score of a word in a text corpus, which could be extremely large. By contrast, this paper proposes another type of term weighting; it measures the positional significance score of a word within its sentence. Here, we assume the following hypothesis:

- The "significance" of a word depends on its position within its sentence.

In Japanese, the main subject of a sentence usually appears at the beginning of the sentence (BOS) and the main verb phrase almost always appears at the end of the sentence (EOS). These words or phrases are usually more important than the other words in the sentence. In order to add this knowledge to the scoring function, term weight is modeled by the following Gaussian mixture.

$$N(\mathrm{psn}(\boldsymbol{x}, I(y_i)); \boldsymbol{\lambda}_g) \quad = \quad m_1 \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2}\left(\frac{\mathrm{psn}(\boldsymbol{x}, I(y_i)) - \mu_1}{\sigma_1}\right)^2\right) + m_2 \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{1}{2}\left(\frac{\mathrm{psn}(\boldsymbol{x}, I(y_i)) - \mu_2}{\sigma_2}\right)^2\right) \tag{2}$$

Here, $\boldsymbol{\lambda}_g = \{\mu_k, \sigma_k, m_k\}_{k=1,2}$. $\mathrm{psn}(\boldsymbol{x}, I(y_i))$ returns the relative position of y_i in the original sentence $\boldsymbol{x}$ which is defined as follows:

$$\mathrm{psn}(\boldsymbol{x}, I(y_i)) = \frac{\mathrm{start}(\boldsymbol{x}, I(y_i))}{\mathrm{length}(\boldsymbol{x})} \tag{3}$$

'length($\boldsymbol{x}$)' denotes the number of characters in the source sentence and 'start($\boldsymbol{x}, I(y_i)$)' denotes the accumulated run of characters from BOS to ($\boldsymbol{x}, I(y_i)$). In equation (2), μ_k,σ_k indicates the mean and the standard deviation for the normal distribution, respectively. m_k is a mixture parameter.

We use the distribution (2) in defining $g(\boldsymbol{x}, I(y_i); \boldsymbol{\lambda}_g)$ as follows:

$$g(\boldsymbol{x}, I(y_i); \boldsymbol{\lambda_g}) \quad = \quad \begin{cases} \mathrm{IDF}(\boldsymbol{x}, I(y_i)) \times N(\mathrm{psn}(\boldsymbol{x}, I(y_i)); \boldsymbol{\lambda}_g) \\ \qquad \text{if pos}(\boldsymbol{x}, I(y_i)) = \text{noun, verb, adjective} \\ \mathrm{Constant} \times N(\mathrm{psn}(\boldsymbol{x}, I(y_i)); \boldsymbol{\lambda}_g) \\ \qquad \text{otherwise} \end{cases} \tag{4}$$

Here, $\mathrm{pos}(\boldsymbol{x}, I(y_i))$ denotes the part-of-speech tag for y_i. $\boldsymbol{\lambda}_g$ is optimized by using the MCE learning framework.

3.2.2 Patched Language Model

Many studies on sentence compression employ the n-gram language model to evaluate the linguistic likelihood of a compressed sentence. However, this model is usually computed by using a huge volume of text data that contains both short and long sentences. N-gram distribution of short sentences may different from that of long sentences. Therefore, the n-gram probability sometimes disagrees with our intuition in terms of sentence compression. Moreover, we cannot obtain a huge corpus consisting solely of compressed sentences. Even if we collect headlines as a kind of compressed sentence from newspaper articles, corpus size is still too small. Therefore, we propose the following novel linguistic likelihood based on statistics derived from the original sentences and a huge corpus:

$$\mathrm{PLM}(\boldsymbol{x}, I(y_j), I(y_{j-1})) = \begin{cases} 1 & \text{if } I(y_j) = I(y_{j-1}) + 1 \\ \lambda_{\mathrm{PLM}}\, \mathrm{Bigram}(\boldsymbol{x}, I(y_j), I(y_{j-1})) & \\ & \text{otherwise} \end{cases} \quad (5)$$

PLM stands for Patched Language Model. Here, $0 \leq \lambda_{\mathrm{PLM}} \leq 1$, Bigram$(\cdot)$ indicates word bigram probability. The first line of equation (5) agrees with Jing's observation on sentence alignment tasks (Jing and McKeown, 1999); that is, most (or almost all) bigrams in a compressed sentence appear in the original sentence as they are.

3.2.3 POS bigram

Since POS bigrams are useful for rejecting ungrammatical sentences, we adopt them as follows:

$$P_{\mathrm{pos}}(\boldsymbol{x}, I(y_{i+1}) | I(y_i)) = P(\mathrm{pos}(\boldsymbol{x}, I(y_{i+1})) | \mathrm{pos}(\boldsymbol{x}, I(y_i))). \quad (6)$$

Finally, the linguistic likelihood between adjacent words within $\boldsymbol{y}$ is defined as follows:

$$h(\boldsymbol{x}, I(y_{i+1}), I(y_i); \boldsymbol{\lambda}_h) = \mathrm{PLM}(\boldsymbol{x}, I(y_{i+1}), I(y_i)) + \lambda_{(\mathrm{pos}(\boldsymbol{x}, I(y_{i+1}))|\mathrm{pos}(\boldsymbol{x}, I(y_i)))} P_{\mathrm{pos}}(\boldsymbol{x}, I(y_{i+1}) | I(y_i))$$

3.3 Parameter Optimization

We can regard sentence compression as a two class problem: we give a word in the original sentence class label $+1$ (the word is used in the compressed output) or -1 (the word is not used). In order to consider the interdependence of words, we employ the Minimum Classification Error (MCE) learning framework (Juang and Katagiri, 1992), which was proposed for learning the goodness of a sequence. $\boldsymbol{x}_t$ denotes the t-th original *sentence* in the training data set T. $\boldsymbol{y}_t^*$ denotes the reference compression that is made by humans and $\hat{\boldsymbol{y}}_t$ is a compressed sentence output by a system.

When using the MCE framework, the misclassification measure is defined as the difference between the score of the reference sentence and that of the best non-reference output and we optimize the parameters by minimizing the measure.

$$d(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{\Lambda}) = \{\sum_{t=1}^{|T|} f(\boldsymbol{x}_t, \boldsymbol{y}_t^*; \boldsymbol{\Lambda}) - \max_{\hat{\boldsymbol{y}}_t \neq \boldsymbol{y}_t^*} f(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t; \boldsymbol{\Lambda})\} \quad (7)$$

It is impossible to minimize equation (7) because we cannot derive the gradient of the function. Therefore, we employ the following sigmoid function to smooth this measure.

$$L(d(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Lambda})) = \sum_{t=1}^{|T|} \frac{1}{1 + \exp(-c \times d(\boldsymbol{x}_t, \boldsymbol{y}_t; \boldsymbol{\Lambda}))} \quad (8)$$

Here, c is a constant parameter. To minimize equation (8), we use the following equation.

$$\nabla L = \frac{\partial L}{\partial d} \left(\frac{\partial d}{\partial \lambda_1}, \frac{\partial d}{\partial \lambda_2}, \ldots \right) = 0 \quad (9)$$

Here, $\frac{\partial L}{\partial d}$ is given by:

$$\frac{\partial L}{\partial d} = \frac{c}{1 + \exp(-c \times d)} \left(1 - \frac{1}{1 + \exp(-c \times d)} \right) \quad (10)$$

Finally, the parameters are optimized by using the iterative form. For example, λ_w is optimized as follows:

$$\lambda_{w(\mathrm{new})} = \lambda_{w(\mathrm{old})} - \epsilon \frac{\partial L}{\partial \lambda_{w(\mathrm{old})}} \quad (11)$$

Our parameter optimization procedure can be replaced by another one such as MIRA (McDonald et al., 2005) or CRFs (Lafferty et al., 2001). The reason why we employed MCE is that it is very easy to implement.

4 Experimental Evaluation

4.1 Corpus and Evaluation Measures

We randomly selected 1,000 lead sentences (a lead sentence is the first sentence of an article excluding the headline.) whose length (number of words) was greater than 30 words from the Mainichi Newspaper from 1994 to 2002. There were five different ideal compressions (reference compressions produced by human) for each sentence; all had a 0.6 compression rate. The average length of the input sentences was about 42 words and that of the reference compressions was about 24 words.

For MCE learning, we selected the reference compression that maximize the BLEU score (Papineni et al., 2002) ($= \mathrm{argmax}_{r \in R}\mathrm{BLEU}(r, R \backslash r)$) from the set of reference compressions and used it as correct data for training. Note that r is a reference compression and R is the set of reference compressions.

We employed both automatic evaluation and human subjective evaluation. For automatic evaluation, we employed BLEU (Papineni et al., 2002) by following (Unno et al., 2006). We utilized 5-fold cross validation, *i.e.*, we broke the whole data set into five blocks and used four of them for training and the remainder for testing and repeated the evaluation on the test data five times changing the test block each time.

We also employed human subjective evaluation, *i.e.*, we presented the compressed sentences to six human subjects and asked them to evaluate the sentence for fluency and importance on a scale 1 (worst) to 5 (best). For each source sentence, the order in which the compressed sentences were presented was random.

4.2 Comparison of Sentence Compression Methods

In order to investigate the effectiveness of the proposed features, we compared our method against Hori's model (Hori and Furui, 2003), which is a state-of-the-art Japanese sentence compressor based on the sequence-oriented approach.

Table 1 shows the feature set used in our experiment. Note that 'Hori−' indicates the earlier version of Hori's method which does not require the dependency parser. For example, label 'w/o IPTW + Dep' employs IDF term weighting as function $g(\cdot)$ and word bigram, part-of-speech bigram and dependency probability between words as function $h(\cdot)$ in equation (1).

Table 1: Configuration setup

Label	$g()$	$h()$
Proposed	IPTW	PLM + POS
w/o PLM	IPTW	Bigram+POS
w/o IPTW	IDF	PLM+POS
Hori−	IDF	Trigram
Proposed+Dep	IPTW	PLM + POS +Dep
w/o PLM+Dep	IPTW	Bigram+POS+Dep
w/o IPTW+Dep	IDF	PLM+POS+Dep
Hori	IDF	Trigram+Dep

Table 2: Results: automatic evaluation

Label	BLEU
Proposed	**.679**
w/o PLM	.617
w/o IPTW	.635
Hori−	.493
Proposed+Dep	.632
w/o PLM+Dep	.669
w/o IPTW+Dep	.656
Hori	.600

To obtain the word dependency probability, we use Kudo's relative-CaboCha (Kudo and Matsumoto, 2005). We developed the n-gram language model from a 9 year set of Mainichi Newspaper articles. We optimized the parameters by using the MCE learning framework.

5 Results and Discussion

5.1 Results: automatic evaluation

Table 2 shows the evaluation results yielded by BLUE at the compression rate of 0.60.

Without introducing dependency probability, both IPTW and PLM worked well. Our method achieved the highest BLEU score. Compared to 'Proposed', 'w/o IPTW' offers significantly worse performance. The results support the view that our hypothesis, namely that the significance score of a word depends on its position within a sentence, is effective for sentence compression. Figure 2 shows an example of Gaussian mixture with pre-

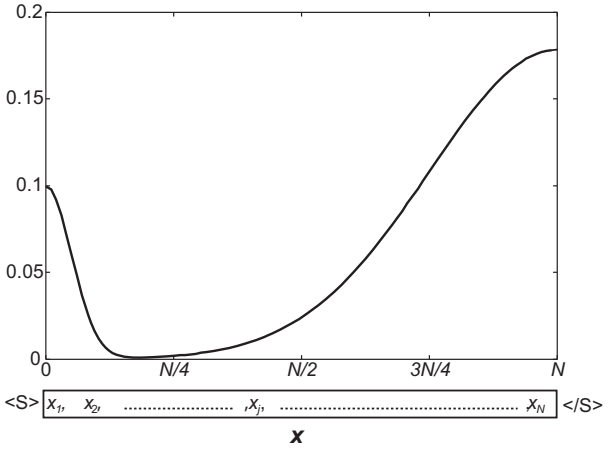

Figure 2: An example of Gaussian mixture with predicted parameters

dicted parameters. From the figure, we can see that the positional weights for words have peaks at BOS and EOS. This is because, in many cases, the subject appears at the beginning of Japanese sentences and the predicate at the end.

Replacing PLM with the bigram language model (w/o PLM) degrades the performance significantly. This result shows that the n-gram language model is improper for sentence compression because the n-gram probability is computed by using a corpus that includes both short and long sentences. Most bigrams in a compressed sentence followed those in the source sentence.

The dependency probability is very helpful provided either IPTW or PLM is employed. For example, 'w/o PLM + Dep' achieved the second highest BLEU score. The difference of the score between 'Proposed' and 'w/o PLM + Dep' is only 0.01 but there were significant differences as determined by Wilcoxon signed rank test. Compared to 'Hori−', 'Hori' achieved a significantly higher BLEU score.

The introduction of both IPTW and PLM makes the use of dependency probability unnecessary. In fact, the score of 'Proposed + Dep' is not good. We believe that this is due to overfitting. PLM is similar to dependency probability in that both features emphasize word pairs that occurred as bigrams in the source sentence. Therefore, by introducing dependency probability, the information within the feature vector is not increased even though the number of features is increased.

Table 3: Results: human subjective evaluations

Label	Fluency	Importance
Proposed	4.05 ($\pm$0.846)	3.33 ($\pm$0.854)
w/o PLM + Dep	3.91 ($\pm$0.759)	3.24 ($\pm$0.753)
Hori−	3.09 ($\pm$0.899)	2.34 ($\pm$0.696)
Hori	3.28 ($\pm$0.924)	2.64 ($\pm$0.819)
Human	4.86 ($\pm$0.268)	4.66 ($\pm$0.317)

5.2 Results: human subjective evaluation

We used human subjective evaluations to compare our method to human compression, 'w/o PLM + Dep' which achieved the second highest performance in the automatic evaluation, 'Hori−' and 'Hori'. We randomly selected 100 sentences from the test corpus and evaluated their compressed variants in terms of 'fluency' and 'importance.'

Table 3 shows the results, mean score of all judgements as well as the standard deviation.

The results indicate that human compression achieved the best score in both fluency and importance. Human compression significantly outperformed other compression methods. This results supports the idea that humans can easily compress sentences with the compression rate of 0.6. Of the automatic methods, our method achieved the best score in both fluency and importance while 'Hori−' was the worst performer. Our method significantly outperformed both 'Hori' and 'Hori−' on both metrics. Moreover, our method outperformed 'w/o PLM + Dep' again. However, the differences in the scores are not significant. We believe that this is due to a lack of data. If we use more data for the significant test, significant differences will be found. Although our method does not employ any explicit syntactic information, its fluency and importance are extremely good. This confirms the effectiveness of the new features of IPTW and PLM.

5.3 Comparison of decoding speed

We compare the decoding speed of our method against that of Hori's method.

We measured the decoding time for all 1,000 test sentences on a standard Linux Box (CPU: Intel© Core™ 2 Extreme QX9650 (3.00GHz), Memory: 8G Bytes). The results were as follows:

Proposed: 22.14 seconds
(45.2 sentences / sec),

Hori: 95.34 seconds
(10.5 sentences / sec).

Our method was about 4.3 times faster than Hori's method due to the latter's use of dependency parser. This speed advantage is significant when on-demand sentence compression is needed.

6 Related work

Conventional sentence compression methods employ the tree trimming approach to compress a sentence without changing its meaning. For instance, most English sentence compression methods make full parse trees and trim them by applying the generative model (Knight and Marcu, 2002; Turner and Charniak, 2005), discriminative model (Knight and Marcu, 2002; Unno et al., 2006). For Japanese sentences, instead of using full parse trees, existing sentence compression methods trim dependency trees by the discriminative model (Takeuchi and Matsumoto, 2001; Nomoto, 2008) through the use of simple linear combined features (Oguro et al., 2002). The tree trimming approach guarantees that the compressed sentence is grammatical if the source sentence does not trigger parsing error. However, as we mentioned in Section 2, the tree trimming approach is not suitable for Japanese sentence compression because in many cases it cannot reproduce human-produced compressions.

As an alternative to these tree trimming approaches, sequence-oriented approaches have been proposed (McDonald, 2006; Nomoto, 2007; Hori and Furui, 2003; Clarke and Lapata, 2006). Nomoto (2007) and McDonald (2006) employed the random field based approach. Hori et al. (2003) and Clarke et al. (2006) employed the linear model with simple combined features. They simply regard a sentence as a word sequence and structural information, such as full parse tree or dependency trees, are encoded in the sequence as features. The advantage of these methods over the tree trimming approach is that they have the potential to drop arbitrary words from the original sentence without the need to consider the boundaries determined by the tree structures. This approach is more suitable for Japanese compression than tree trimming. However, they still rely on syntactic information derived from full parsed trees or dependency trees. Moreover, their use of syntactic parsers seriously degrades the decoding speed.

7 Conclusions

We proposed a syntax free sequence-oriented Japanese sentence compression method with two novel features: IPTW and PLM. Our method needs only a POS tagger. It is significantly superior to the methods that employ syntactic parsers. An experiment on a Japanese news corpus revealed the effectiveness of the new features. Although the proposed method does not employ any explicit syntactic information, it outperformed, with statistical significance, Hori's method a state-of-the-art Japanese sentence compression method based on the sequence-oriented approach.

The contributions of this paper are as follows:

- We revealed that in compressing Japanese sentences, humans usually ignore syntactic structures; they drop intermediate nodes of the dependency tree and drop words within *bunsetsu*,
- As an alternative to the syntactic parser, we proposed two novel features, Intra-sentence positional term weighting (IPTW) and the Patched language model (PLM), and showed their effectiveness by conducting automatic and human evaluations,
- We showed that our method is about 4.3 times faster than Hori's method which employs a dependency parser.

References

J. Clarke and M. Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proc. of the 21st COLING and 44th ACL*, pages 377–384.

C. Hori and S. Furui. 2003. A new approach to automatic speech summarization. *IEEE trans. on Multimedia*, 5(3):368–378.

H. Jing and K. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. In *Proc. of the 22nd SIGIR*, pages 129–136.

B. H. Juang and S. Katagiri. 1992. Discriminative Learning for Minimum Error Classification. *IEEE Trans. on Signal Processing*, 40(12):3043–3053.

K. Knight and D. Marcu. 2002. Summarization beyond sentence extraction. *Artificial Intelligence*, 139(1):91–107.

T. Kudo and Y. Matsumoto. 2004. A Boosting Algorithm for Classification of Semi-Structured Text. In *Proc. of the EMNLP*, pages 301–308.

T. Kudo and Y. Matsumoto. 2005. Japanese Dependency Parsing Using Relative Preference of Dependency (in japanese). *IPSJ Journal*, 46(4):1082–1092.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of the 18th ICML*, pages 282–289.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online Large Margrin Training of Dependency Parser. In *Proc. of the 43rd ACL*, pages 91–98.

R. McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proc. of the 11th EACL*, pages 297–304.

T. Nomoto. 2007. Discriminative sentence compression with conditional random fields. *Information Processing and Management*, 43(6):1571–1587.

T. Nomoto. 2008. A generic sentence trimmer with crfs. In *Proc. of the ACL-08: HLT*, pages 299–307.

R. Oguro, H. Sekiya, Y. Morooka, K. Takagi, and K. Ozeki. 2002. Evaluation of a japanese sentence compression method based on phrase significance and inter-phrase dependency. In *Proc. of the TSD 2002*, pages 27–32.

K. Papineni, S. Roukos, T. Ward, and W-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistic (ACL)*, pages 311–318.

K. Takeuchi and Y. Matsumoto. 2001. Acquisition of sentence reduction rules for improving quality of text summaries. In *Proc. of the 6th NLPRS*, pages 447–452.

J. Turner and E. Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proc. of the 43rd ACL*, pages 290–297.

Y. Unno, T. Ninomiya, Y. Miyao, and J. Tsujii. 2006. Trimming cfg parse trees for sentence compression using machine learning approach. In *Proc. of the 21st COLING and 44th ACL*, pages 850–857.