Example-Based Sentence Reduction Using the Hidden Markov Model

MINH LE NGUYEN
Japan Advanced Institute of Science and Technology
SUSUMU HORIGUCHI
Tohoku University
AKIRA SHIMAZU and BAO TU HO
Japan Advanced Institute of Science and Technology

Sentence reduction is the removal of redundant words or phrases from an input sentence by creating a new sentence in which the gist of the original meaning of the sentence remains unchanged. All previous methods required a syntax parser before sentences could be reduced; hence it was difficult to apply them to a language with no reliable parser. In this article we propose two new sentence-reduction algorithms that do not use syntactic parsing for the input sentence. The first algorithm, based on the template-translation learning algorithm, one of example-based machine-translation methods, works quite well in reducing sentences, but its computational complexity can be exponential in certain cases. The second algorithm, an extension of the template-translation algorithm via innovative employment of the Hidden Markov model, which uses the set of template rules learned from examples, can overcome this computation problem. Experiments show that the proposed algorithms achieve acceptable results in comparison to sentence reduction done by humans.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural language Processing – *Text summarization, Sentence reduction*; G.3 [**Mathematics of Computing**]: Probability and Statistics - *Markov-processes*; *Probabilistic algorithms*;

General Terms: Algorithms, Experimentation, Languages

Additional Key Words and Phrases: Sentence reduction, example-based sentence reduction, HMM-based sentence reduction

---

1. INTRODUCTION

In recent years, automatic text-summarization has attracted a great deal of attention due to its utility in dealing with the rapidly increasing amount of text information available on the Internet. There have been various applications of text-summarization techniques such as search engine hits (summarizing information on a hit list retrieved by search engines), hand-held devices (creating a screen-size version of a book), and headline-generation on television [Mani and Maybury 1999].

Research in automatic text-summarization has focused on extracting or identifying important clauses, sentences, and paragraphs in the given texts. The essence of this research is sentence-reduction, i.e., reducing long sentences into short ones, so that the gist of the meaning of the short sentence remains the same as the original one.

Various methods for sentence-reduction have been used in many tasks, such as providing audio-scanning services for the blind [Grefenstette 1998]; removing clauses

---

This research was supported in part by a JAIST international research project grant and JSPS Grant-in-Aid for Scientific Research.
Authors' address: Graduate Information School, Japan Advanced Institute of Science and Technology, 923-1292, Ishikawa, Japan.

from sentences before indexing documents for information retrieval [Olivers and Dolan 1999]; and enhancing the performance of text-summarization [Mani and Maybury 1999].

Reducing sentences for cut-and-paste summarization was studied by Jing [2000]. The method requires that a sentence be parsed into high-level descriptions by using multiple sources in order to locate extraneous phrases for removal. The multiple sources include syntactic knowledge, context information, and statistics computed from a corpus of examples written by a human professional. Knight and Marcu [2002] developed two methods for the sentence- compression problem, which are similar to those for sentence reduction. They devised both noisy-channel and decision-tree approaches in which the Collins' parser [Collins 1999], is used to parse input sentences. An alternative use of syntactic parsing was proposed by Riezler et al. [2003], in which they use the ambiguity-packing and lexical-functional grammar to enhance the performance of their sentence-reduction method**.** In summary, the sentence-reduction methods mentioned above require a syntax parser before sentences can be reduced. Although these methods work efficiently in languages such as English, it is difficult to apply them to languages that have no method for reliable syntactic parsing. Similar work on sentence reduction that does not use parsers is proposed by Withbrock and Mittal [1999]. This method is the only one that applies a probabilistic model that is trained directly on <Headline, Document> pairs; it is mainly applied to headline-generation tasks.

Motivated by this limitation, it would be useful to develop sentence-reduction methods without parsing that can produce similar or better results than methods with parsing. In addition, it is worth recalling that generating multiple best-reduction outputs is essential in sentence reduction [Lin 2003].

Example-based machine translation (EBMT), originally proposed by Nagao [1984], is one of the main approaches to corpus-based machine translation. In the EBMT framework, the translation-template learning (TTL) method [Cickli and Günvenir 1998, 2001] has been successfully applied to translations from English to Turkish. In this article we focus on investigating the use of translation-template learning method to solve the sentence-reduction problem. Intuitively, when considering long sentences as a source language and reduced sentences as a target language, the problem of sentence reduction is equivalent to the translation problem.

When using the template-learning algorithm to reduce sentences, syntactic parsing for representing sentences is not required. TTL uses exemplars of long sentences and their reduced versions to automatically generate template rules with all the advantages of EBMT. However, one drawback of applying the TTL algorithm to sentence reduction is the exponential calculation problem, which comes to the fore when the original TTL is applied to a long sentence and the number of template rules is large. To solve this problem, we propose a novel method that uses dynamic programming on a hidden Markov model (HMM) that is built by the set of learned template rules. The proposed algorithm does not only avoid the exponential calculation problem, but also outperforms the TTL algorithm in the sentence-reduction task.

The rest of this article is organized as follows: Section 2 presents a TTL-based algorithm for sentence reduction; Section 3 demonstrates the reduction process using template rules obtained by TTL to build the HMM and the HMM-based template-reduction algorithm; Section 4 shows experiment results when applying the proposed algorithms on the Vietnamese language; and Section 5 concludes and proposes problems for future solution.

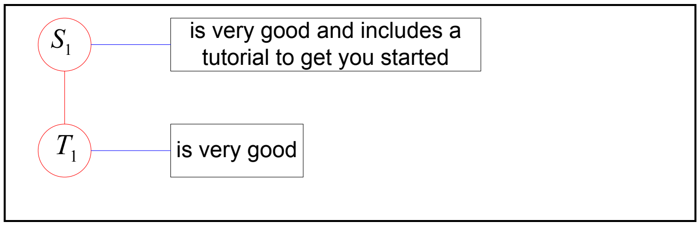

Fig. 1. Template reduction rule example.

2. TEMPLATE LEARNING FOR SENTENCE REDUCTION

The template-learning algorithm has been applied to machine translation [Cickli and Günvenir 1998]. But in order to apply the algorithm to sentence reduction, some definitions are necessary, as follows.

2.1 Rules for Template Reduction

Although in this article we consider the sentence-reduction problem on one language, here we discuss the problem in the general case, that of reducing a sentence from a source language (SL) to a reduced sentence in a target language (TL). A template- reduction rule is defined in the form $S_1S_2...S_i...S_N \leftrightarrow T_1T_2...T_j...T_K$ in which S_i are either constants or variables in SL, and T_j are either constants or variables in TL. A constant can be a phrase or a word, while a variable can be substituted for by constants. Each variable S_i on the left side of the rule is aligned with a variable T_j on the right side of the rule. Figure 1 depicts an example of a template-reduction rule where S_1 are variables, and the phrase “is very good and includes a tutorial to get you started" is reduced to the phrase “is very good".

We call a template-reduction rule with no variable a lexical rule. A lexical rule can be used as a value of a variable in a template-reduction rule to reduce a long sentence to a shorter one. For example, if the lexical rule “The document" $\leftrightarrow$ “Document" is in the set of template-reduction rules, then the input sentence “The document is very good and includes a tutorial to get you started" can be reduced to the sentence “Document is very good" by using the template-reduction rule in Figure 1.

2.1 Learning Template-Reduction Rules

We apply the TTL algorithm [Cickli and Günvenir 1996] to infer template-reduction rules using the similarities and differences in two examples taken from a corpus of pairs of long sentences and their corresponding reduced sentences.

Formally, a reduction example $E_a : E_a^1 \leftrightarrow E_a^2$ is composed of a pair of sentences, E_a^1 and E_a^2, where E_a^1 is an original sentence in SL and E_a^2 is a reduced sentence in TL. A similarity between two sentences of a language is a non-empty sequence of common items (root words or morphemes) in both sentences. A difference between two sentences in a language is a pair of subsequences having no common items, one is subsequence of the first sentence and the other of the second sentence. Given two reduction examples (E_a, E_b), our problem is to find similarities between the constituents of E_a and E_b. A sentence is considered a sequence of lexical items. If no similar constituents (viewed as subsequences of lexical items) can be found, then no template reduction rule is learned from these examples. If there are similar constituents, then *a match sequence* $M_{a,b}$ is generated in the following form

$$M_{a,b} = S_0^1 D_0^1 ... D_{n-1}^1 S_n^1 D_n^1 \leftrightarrow S_0^2 D_0^2 S_1^2 ... D_{m-1}^2 S_m^2 D_m^2 \quad (1)$$

Here $n, m \geq 1$, S_k^i represents a similarity between E_a^i and E_b^i, and $D_k^i = (D_{k,a}^i, D_{k,b}^i)$ represents a difference between E_a^i and E_b^i, where $D_{k,a}^i$ and $D_{k,b}^i$ are non-empty

(1)
E_a = "The document is very good and includes a tutorial to get you started" $\leftrightarrow$ "Document is very good".
E_a^1 ="The document is very good and includes a tutorial to get you started" and E_a^2 ="Document is very good".
(2)
E_b = "This paper is very good and includes a tutorial to get you started" $\leftrightarrow$ "Paper is very good". Where E_b^1 = "This paper is very good and includes a tutorial to get you started" and E_b^2 = "Paper is very good".

subsequences of items between two similar constituents. For instance, consider the following reduction examples:

For these reduction examples, the matching algorithm obtains the following match sequence.

$M_{a,b}$ = (The document, This paper) "is very good and includes a tutorial to get you started" $\leftrightarrow$ (Document, Paper) "is very good".

That is,

S_0^1 = "", D_0^1 = (The document, This paper), $D_{0,a}^1$ = (The document), $D_{0,b}^1$ = (This paper)
S_1^1 = "is very good and includes a tutorial to get you started",
S_0^2 = "", D_0^2 =(Document, Paper), $D_{0,a}^2$ =(Document),
$D_{0,b}^2$ =(Paper), S_1^2 = "is very good"

Intuitively, in the above example the similar elements and the different ones on the left-hand side should be aligned with similar elements and the different elements in the right-hand side. Thus, in this case "(The document, This paper)" is aligned with "(Document, Paper)", and "is very good and includes a tutorial to get you started" is aligned with "is very good". We consider "(The document, This paper)" and "(Document, Paper)" as variables, and we can generate the template-reduction rule in Figure 1.

We also obtain two lexical rules:

$D^1_{0,a} \leftrightarrow D^2_{0,a}$, or "The document" $\leftrightarrow$ "Document"
$D^1_{0,b} \leftrightarrow D^2_{0,b}$, or "This paper" $\leftrightarrow$ "Paper".

The main idea of translation-template learning algorithm can be summarized as follows: After finding a match sequence, the TTL will define alignments between the similar elements on the left-hand side and the similar elements on the right-hand side of the match sequence. The TTL also defines alignments between the different elements on the left-hand side and the different elements on the right-hand side of the match sequence. These processes are based on the template rules previously learned from examples.

Assume that the match sequence has n different elements and m different elements on its left-hand side and its right-hand side, respectively. If $n = m$ and there are $n-1$ different elements on the left-hand side aligned with the elements on the right-hand side,

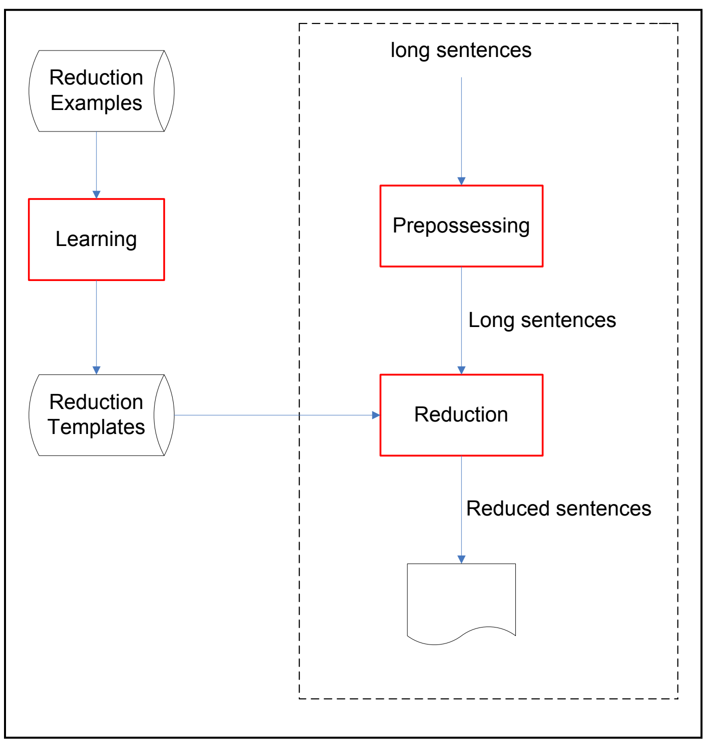

Fig. 2. Framework for sentence reduction using template learning.

we can define those elements on the right-hand side that will be aligned with the remaining different elements on the left-hand side. All the different elements were replaced by variables, and we then obtained a template-reduction rule. This idea is similar to that for similar elements in the match sequence. The details of this algorithm are presented in Cickli and Günvenir [1998; 2001].

The translation-template learning method has the advantage that it does not need to parse the input sentence. Obviously, we can use some preprocessing techniques like morphological analysis and shallow parsing to enrich linguistic information for the given input sentence. For simplicity, in this article we only use the morphological analysis to represent an input sentence.

Figure 2 depicts the framework for sentence reduction via template learning. The corpus of reduced-sentence examples is used to generate template rules in the translation-template learning algorithm. In the reduction process, a given long sentence is represented on the surface level by processes like pos-tagging, morphological analysis, or chunking. Sentence reduction using template rules will then be performed in order to generate the reduced sentence; this process is presented in the following sections.

3. SENTENCE REDUCTION USING A TEMPLATE RULE

This section presents two algorithms for sentence reduction by using template-reduction rules. In the first one, the original method of translation-template learning is applied to problem-reduction problems. In the second algorithm, we propose a novel method that uses the hidden Markov model, which works efficiently when the input sentence is long and the number of template-reduction rules is big.

3.1 Sentence Reduction Using Template Rules

To illustrate the behavior of sentence reduction using template rules, consider the sentence "It is likely that two companies will work on integrating multimedia with database technology" using the template rule[1] in Figure 3.

Two phrases "It is likely that" and "will work on" in the input sentence are matched with the template rule. By using lexical rules, the reduction algorithm then tries to find all possible choices to replace variables S_2 and S_4. It finds all lexical rules whose left side is matched with "two companies" and "integrating multimedia with database technology" for variables S_2 and S_4, respectively. Figure 3 shows three choices for S_2 and S_4, from which we have six reduction results. Intuitively, the best reduction output is "Two companies will work on integrating multimedia with database technology".

There are two obstacles to the use of the original template-reduction method:

- How do we determine the best outputs when using template reductions?
- Suppose a template rule has t variables and each variable has l matched lexical rules, so we have l^t choices for reduction. How can we deal with this exponential calculation?

To solve these problems, we developed an HMM-based method described in the next sections.

---

[1]The template rule is learned via the two following examples:

It is likely that he will work on through storm ↔ He will work on through the storm.

It is likely that she will work on this book ↔ She will work on this book.

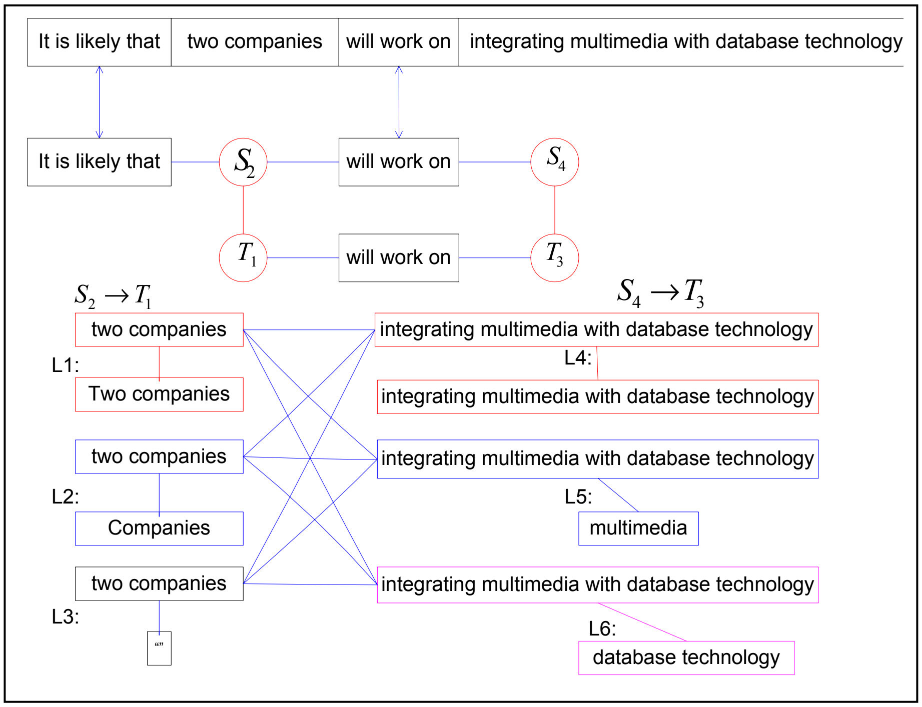

Fig. 3. Example of reduction-based HMM.

3.2 HMM-Based Reduction

3.2.1. *The HMM Model for Sentence Reduction.* The main idea here is that instead of considering all lexical rules, we use a dynamic programming algorithm to find a likely sequence of lexical rules for a given input sentence. The probabilities $P(L_i \mid L_j)$ between two lexical rules L_i and L_j are given in Table I, from which the likely sequence of lexical rules is (L_1, L_4) , and we obtain the following best reduction output: "Two companies will work on integrating multimedia with database technology".

Given an input sentence $e_1e_2...e_m$ (e_i is a token) and a set of template rules $r_1, r_2,..., r_d$, our problem is to find a sequence of lexical rules that will best explain the reduction results in a given sentence. This problem is equivalent to finding all likely reduction results for each rule r_i (*i=1,...,d*).

For the rule $r_i : S_1S_2...S_N \leftrightarrow T_1T_2...T_K$ in the above example, each constant S_j (*j=1,...,N*) can be associated to a phrase on the right-hand side of the rule r_i, and each variable S_j (*j=1,...,N*) in the rule can be associated with set of lexical rules whose left-hand side is a substring that starts from a possible position within the input sentence.

Table I. Probability Table

	L_4	L_5	L_6
L_1	0.5	0.3	0.2
L_2	0.4	0.3	0.3
L_3	0.4	0.3	0.3

Consider a lexical rule as a *hidden state* and a substring in the input sentence as an *observed symbol*. The problem of reduction is then equivalent to finding a lexical rule for each variable. To find the most likely sequence, we must find a sequence of lexical rules that maximize the probability $P(e_1, e_2, ...e_m, r_i)$.

Since $r_i : S_1S_2...S_N \leftrightarrow T_1T_2...T_K$ and the map between $S_1S_2...S_N$ and r_i is one by one, we obtain

$$P(r_i \mid e_1, e_2, ..., e_m) = P(S_1S_2...S_N \mid e_1e_2...e_m) \tag{2}$$

Applying Bayes rule, we have

$$P(S_1S_2...S_N \mid e_1e_2...e_m) = \frac{P(e_1e_2...e_m \mid S_1S_2...S_N)}{P(e_1e_2...e_m)} \times P(S_1S_2...S_N) \tag{3}$$

Since $e_1e_2...e_m$ is a sequence of input words, we have the probability $P(e_1e_2...e_m)$, we need to maximize the following:

$$P(e_1e_2...e_m \mid S_1S_2...S_N) \times P(S_1S_2...S_N) \tag{4}$$

Using the bigram model, $P(e_1e_2...e_m \mid S_1S_2...S_N)$ can be approximated as

$$P(e_1e_2...e_m \mid S_1S_2...S_N) = \prod_{i=1}^{N} P(e_{j_i}...e_{j_i+l} \mid S_i) \tag{5}$$

where $e_{j_i}...e_{j_i+l}$ matches the left side of a lexical rule matching S_i, and $j_i, j_{i+1}, ..., j_{i+l}$ is a sequence of word positions within the input sentence e. $P(S_1S_2...S_N)$ can also be approximated as

$$P(S_1S_2...S_N) = \prod_{i=1}^{N-1} P(S_{i+1} \mid S_i) \tag{6}$$

Thus, we get

$$P(e_1e_2...e_m \mid S_1S_2...S_N) \times P(S_1S_2...S_N) = \prod_{i=1}^{N} P(e_{j_i}...e_{j_i+l} \mid S_i) \times \prod_{i=1}^{N-1} P(S_{i+1} \mid S_i) \tag{7}$$

To find the sequence of lexical rules that maximizes formula (7), we can use the Viterbi algorithm [Viterbi 1967], a kind of dynamic programming. If the rule has *t* variables and each variable consists of *l* elements, then the complexity is $l^2 \times t$ while the recursive may be l^t. Thus, each rule can define a reduction score, and reductions in output for the input sentence can be sorted according to their score values. Therefore, our HMM-based method can avoid the exponential calculation problem by using dynamic programming. In addition, it can sort reduction results and achieve better accuracy without using any complex processes in a set of template rules. Interestingly, it provides a new perspective for applying statistical machine-learning theory to sentence reduction.

3.2.2. *Estimates of the HMM Model for Sentence Reduction.* We now describe our HMM-based method for sentence reduction. An HMM is specified by a five-tuple $(O, L, A, B, \prod)$, where L is the set of lexical rules and O is the output alphabet. $\prod, A, B$ are, respectively, the probabilities for the initial state, state transition, and symbol emission.

The HMM-based method is estimated by using the Baum-Welch learning method [Baum and Eagon 1967], described as follows:

The corpora of long and reduced sentences are used to generate observed symbol sequences. Each long sentence is reduced by using lexical rules if its reduced output is the same as a reduced sentence in the corpus. After obtaining a sequence of lexical rules, a sequence of observed symbols is generated because each observed symbol is on the left side of a lexical rule. Therefore, using a set of template rules and the corpus, we can generate a training data O_{train} in the form

$$
\begin{aligned}
&O_{t_1} O_{t_1+1} \ldots O_{m_1} \leftrightarrow l_{t_1} l_{t_1+1} \ldots l_{m_1} \\
&O_{t_2} O_{t_2+1} \ldots O_{m_2} \leftrightarrow l_{t_2} l_{t_2+1} \ldots l_{m_2} \\
&\ldots \\
&O_{t_k} O_{t_k+1} \ldots O_{m_k} \leftrightarrow l_{t_k} l_{t_k+1} \ldots l_{m_k}
\end{aligned}
$$

Here $O_{t_k} O_{t_k+1} \ldots O_{m_k}$ is a sequence of observed symbols associated with a sequence of lexical rules $l_{t_k} l_{t_k+1} \ldots l_{m_k}$.

$C(l^j)$, $C(l^j, l^k)$, and $C(O^j, l^k)$ denote, respectively, the number of occurrences of the lexical rule l^j, the number of occurrence of the lexical rule l^j following the lexical rule l^k, and the number of occurrences of the observed symbol O^j corresponding to the lexical rule l^k. With this notation, the initialization algorithm for estimating an HMM model on the training data above is described in Algorithm 1.

For all lexical rule l^j do
  For all lexical rule l^k do
    $P(l^j \mid l^k) = \frac{C(l^j, l^k)}{C(l^j)}$
For all lexical rule l^j do
  For all observed symbols O^l do
    $P(O^l \mid l^j) = \frac{C(O^l, l^j)}{C(l^j)}$

Algorithm 1. An initialization algorithm.

To avoid the data-sparseness problem, we used an additive smoothing technique [Chen and Goodman 1998] for probabilities in Algorithm 1. Assume that we are given O'_{train} ,an unlabeled training data which only consists of a set of observed sequences. After initializing the probabilities of observed symbols and lexical rules using Algorithm 1 on the training data O_{train}, the Baum-Welch learning approach is used to estimate the HMM for sentence reduction by maximizing $P(O'_{train} \mid A, B, \prod)$.

4. EXPERIMENT AND DISCUSSION

The corpus for sentence reduction is collected from the Vietnam agency web-site (http://www.vnagency.com.vn) by performing the decomposition program [Nguyen and Horiguchi 2002] using the news and its summaries. After correcting them manually, we obtained more than 1,500 pairs of long and reduced sentences, which were then used to generate template rules for our reduction methods. The number of template rules and the number of lexical rules using translation-template learning are 11,034 rules and 2,287 rules, respectively. Using template rules and the data corpus, we obtained the training data for estimating the HMM model described in Section 3.2.2, in which the initialization parameters for the HMM model are estimated by using Algorithm 1. The training data for estimating the HMM model consists of 1,500 observed sequences, in which each sequence corresponds to a sequence of lexical rules. We obtained another1,200 sentences from the same web-site, in which the number of sentences that cannot be recognized by the template rules is 10%. We randomly selected 32 from 1,200 sentences for testing; the remaining sentences were used to extract observed sequences for training the HMM model by means of the Forward-Backward algorithm.

It is difficult to compare our methods to previous ones by using a parsing approach, because there was no reliable syntax parser for the Vietnamese language. However, we manually parsed all sentences in our corpus in order to use the decision-tree-based reduction approach described in Knight and Marcu [2002]. After performing the C4.5 training program [Quilan 1993] on the corpus above, we were able to test the reduction-based decision tree model.

We implemented five sentence-reduction methods as follows:

– the baseline method is the one that obtains a reduced sentence with the highest word-bigram score;
– the sentence reduction-based decision tree model (decision-based);
– the proposed reduction method using the TTL algorithm (EBSR);
– the reduction method using the HMM-based reduction algorithm (EBSR-HMM); and
– the EBSR-HMM algorithm using the n-best of the Viterbi algorithm.

Table II. Experimental Results

Method	Compression	Grammaticality	Importance
Baseline	57.19	4.78 ±1.19	4.34 ±1.92
EBRS	65.20	6.80 ±1.30	6.49 ±1.80
Decision-Based	60.25	7.40 ±1.32	7.12 ±1.73
EBRS-HMM	65.15	7.70 ±1.20	7.30 ±1.60
EBRS-HMM(n-best)	68.40	8.20 ±1.32	7.90 ±1.45
Human	53.33	9.05 ±0.30	8.50 ±0.80

Template Rule		Reduction Example
<Left> (Kuala_Lumpur_hôm_qua_bác_bỏ_đề_nghị_của_Washington_là_) X1 (Đông_Nam_Á._) **</Left>** **<Right>** (Kuala_Lumpur_bác_bỏ_đề_nghị_của_Washington_) Y1 (Đông_Nam_Á._) **</Right>**	Original	Kuala Lumpur hôm qua bác bỏ đề nghị của Washington là hỗ trợ tuần tra đoạn đường biển huyết mạch ở Đông Nam Á. Kuala Lumpur yesterday refused the Washington's proposal for supporting to go on a patrol in the life-line seaway in South-east Asia.
	EBSR	Kuala Lumpur bác bỏ đề nghị của Washington hỗ trợ tuần tra đường biển huyết mạch ở Đông Nam Á. Kuala Lumpur refused the Washington's proposal for supporting to go on a patrol in the life-line seaway in South-east Asia.
	EBSR-HMM	Kuala Lumpur bác bỏ đề nghị của Washington hỗ trợ tuần tra đường biển huyết mạch ở Đông Nam Á. Kuala Lumpur refused the Washington's proposal for supporting to go on a patrol in the life-line seaway in South-east Asia.
<Left> (Trong_thời_gian_gần_đây_số_lượng_người_Việt_Nam_) X1 (ngày_càng_đông_) X3 **</Left>** **<Right>** (Số_lượng_người_Việt_Nam_) Y1 [#](ngày_càng_) Y3 **</Right>**	Original	Trong thời gian gần đây số lượng người Việt Nam đi lao động và học tập ở nước ngoài ngày càng đông. Recently, the number of Vietnamese employees and students has been increasing rapidly.
	EBSR	Số lượng người Việt Nam ở nước ngoài ngày càng. Vietnamese employees and students in aboard rapidly
	EBSR-HMM	Số lượng người Việt Nam lao động và học tập ở nước ngoài ngày càng đông. Vietnamese employees and students in aboard increase rapidly
<Left> (Công_ty_Cao_su_Sài_Gòn_Kim_Đan_) X1 (doanh_nghiệp_xuất_sắc_nhất_vì_đã_áp_dụng_sáng_tạo_các_quyền_sở_hữu_trí_tuệ_về_thương_hiệu_và_kiểu_dáng_công_nghiệp_trong_lĩnh_vực_sản_xuất_và_kinh_doanh_) X3 **</Left>** **<Right>** (Kim_Đan_) Y1 (doanh_nghiệp_xuất_sắc_nhất_) Y3 **</Right>**	Original	Công ty Cao su Sài Gòn Kim Đan nhận cúp WIPO về doanh nghiệp xuất sắc nhất vì đã áp dụng sáng tạo các quyền sở hữu trí tuệ về thương hiệu và kiểu dáng công nghiệp trong lĩnh vực sản xuất và kinh doanh. Sai Gon Kim Dan rubber Company is awarded WIPO cup about the excellent company for applying innovative intellectual ownership, commercial firm, and industrial pattern in productive and business.
	EBSR	Kim Đan nhận cúp WIPO về doanh nghiệp xuất sắc nhất . Kim Dan is awarded WIPO cup about the excellent company
	EBSR-HMM	Kim Đan nhận cúp doanh nghiệp xuất sắc nhất . Kim Dan is awarded the excellent company.

Fig.4. Examples of reduction using an example-based approach; the template rule generated by TTL algorithms. Reduction results were obtained using EBSR and EBSR-HMM.

We used the same evaluation as Knight and Marcu [2002] by showing each original sentence along with its sentence reduction to four Vietnamese judges, in order to compare them with reductions done by humans. The judges were told that all output was generated automatically. The order output was scrambled randomly across test cases. The judges participated in two experiments: In the first, they were asked to determine on a scale from 1 to 10 how well the systems did with respect to selecting the most important words in the original sentence.

In the second, they were asked to determine on a scale from 1 to 10 how grammatical the output was. Table II shows compression ratios in the first column, which indicates that the lower the compression ratio the shorter the reduced sentence. Table II also shows mean and *standard deviation* results across all judges for each algorithm and human.

The results show that the reduced sentences produced by both algorithms are more grammatical and contain more important words than the sentences produced by the baseline. T-test experiments indicate that these differences are statistically significant, at a 95% confidence interval for average scores across all judges. The performance of the proposed algorithms is much closer to human performance than the baseline algorithm.

The results in Table II also indicate that the proposed algorithms are closer to and better than the decision-based algorithm on the grammaticality and importance scales. This is especially true for EBSR-HMM using n-best of the Viterbi algorithm, which outperforms the other algorithms.

Figure 4 shows three examples of our reduction methods in testing on the Vietnamese language. Each reduction example is attached to an English translation. The left-hand side shows the template rules; the right-hand shows the reduction results using the template rules. The results of EBSR and EBSR-HMM in the first example are identical, and are close to human reduction. The result for EBSR in the second example is wrong because it did not use a correct lexical rule. Reduction results for EBSR-HMM are good for both examples 2 and 3.

5. CONCLUSIONS

A novel application of example-based machine translation using template learning for sentence reduction is discussed. Moreover, an extension of the template-learning algorithm using an HMM-based method on set of template rules for reducing sentences is proposed to overcome the limitations of translation-template learning. The two algorithms can be implemented for any language without using parsing, and achieved.acceptable results compared to human reduction. Our methods, along with the reduction method based on the noisy-channel approach, are able to generate multiple best-reduced outputs. This function is very essential for improving text-summarization performance.

We believe that with a larger corpus, our method can be used in real applications such as automatic text-summarization

REFERENCES

BAUM, L. E. AND EAGON, J. A. 1967. An inequality with application to statistical probabilistic functions of Markov processes and to a model of ecology. *Bull. Am. Math. Soc. 73* (1967), 360-363.

CHEN, S. F. AND GOODMAN, J. 1998. An empirical study of smoothing techniques for language modeling. Tech. Rep. TR-10-98, Center for Research in Computing Technology, Harvard Univ., Cambridge, MA.

CICEKLI, I. AND GUVENIR, H. A. 2001. Learning translation rules from a bilingual translation examples. *Appl. Intell.15*, 1 (2001), 57-76.

CICEKLI, I. AND GUVENIR, H. A. 1998. Learning translation templates from examples. *Inf. Syst. 23*, 6 (1998), 353-363.

COLLINS, M. 1999. Head-driven statistical model for natural language parsing. Ph.D. dissertation, Univ. of Pennsylvania.

GREFENSTETTE, G. 1998. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Working Notes of the AAAI Spring Symposium on Intelligent Text summarization*, 111-118.

JING, H. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics* (NAACL-2000).

KNIGHT, K. AND MARCU, D. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artif. Intell. 139*, 1 (2002), 91-107.

LIN, C.Y. 2003. Improving summarization performance by sentence compression: A pilot study. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, 1-9.

MANI, I. AND MAYBURY, M. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

NAGAO, M. 1984. Framework of a mechanical translation between Japanese and English by analogy principle. *Artif. Human Intell*. A. Elithorn and R Banerji (eds). NATO publication: North-Holland, Edinburgh, 173-180.

NGUYEN, M. L. AND HORIGUCHI, S. 2002. An efficient decomposition of human-written summary sentence. In *Proceedings of the 9th International Conference on Neural Information Processing* (Singapore, Nov.18-22, 2002), 705-710.

OLIVERS, H. S. AND DOLAN, W. B. 1999. Less is more; eliminating index terms from subordinate clauses. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistic*, 349-356.

QUINLAN, R. 1993. *Program for Machine learning*. Morgan Kaufman, San Mateo, CA.

VITERBI, A. J. 1967. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Inf. Theory 13* (1967), 260-269.

REZLER, S. ET AL. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation method for lexical-functional grammar. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistic* (HTL-NAACL'03, Edmonton, Canada).

WITHBROCK, J. M. AND MITTAL, O.V. 1999. Ultra-summarization: A statistical approach to generating highly-condensed non-extractive summaries. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval* (SIGR'99, Berkeley, CA). Poster session, 315-316,

Received October 2003; revised April 2004; accepted June 2004