Abbreviation Generation for Japanese Multi-Word Expressions

Hiromi Wakaki† Hiroko Fujii† Masaru Suzuki†
Mika Fukui† Kazuo Sumita†
†Toshiba Corporation
1 Komukai-Toshiba, Saiwai-ku, Kawasaki, 212-8582, Japan
{hiromi.wakaki, hiroko.fujii, masaru1.suzuki,
mika.fukui, kazuo.sumita}@toshiba.co.jp

Abstract

This paper proposes a novel method for generating Japanese abbreviations from their full forms with the Log-Linear Model (LLM) in order to take advantage of characteristic patterns of Japanese abbreviation. Our experimental results show that the method is effective for TV program titles that contain colloquial expressions. The proposed method achieved 78.8% recall for the top 30 candidates, whereas a baseline method using Conditional Random Fields (CRFs) achieved 68.3% recall. Moreover, from the results of experiments using six data sets classified according to types of character and semantic categories, we show that each performance of the above two methods depends on the types of the full forms.

1 Introduction

Much research has been done on abbreviation extraction to detect terms having the same meaning. However, most previous studies (Hisamitsu and Niwa, 2001; Park and Byrd, 2001; Schwartz and Hearst, 2003; Adar, 2004; Sakai and Masuyama, 2005; Nadeau and Turney, 2005; Okazaki and Ananiadou, 2006; Okazaki et al., 2008(1); Okazaki et al., 2008(2)) aimed at extracting abbreviations of organization names and technical terms from well-written documents such as news articles and techincal papers.

Many Japanese terms indicating individual TV programs, songs, comics, novels, and so on, are multi-word expressions and have the characteristics distinct from terms treated in most previous studies on abbreviation extraction. These terms can take several grammatical forms: a noun phrase, a sentence fragment, and even a sentence. Also, many of these expressions contain a variety of types of characters: kanji, hiragana, katakana, alphabet, digit, and symbol, and some of them contain colloquial expressions[1]. Abbreviations of these expressions are often used in colloquial text such as chat or blog, and spoken sentences. To treat an abbreviation as a term having the same meaning as the original expression for NLP applications such as keyboard-based and speech-based information retrieval, an abbreviation generation method effective for this type of multi-word expressions is needed. However, it is not easy to ascertain abbreviations associated with their full forms. This is because although these terms become widely used in speech, they do not appear in well-written documents, such as newspaper articles or research papers, in which the abbreviations are clearly defined for use in the subsequent texts with certain lexical patterns, such as parenthesis. Therefore this paper describes an approach to generate abbreviation candidates from an original term and to rank them according to their probabilities of abbreviation. We assume that top-ranked abbreviations will be narrowed down by using Web search results in the future.

2 Japanese Abbreviation

2.1 Data Sets

Transformations into abbreviations are strongly dependent on languages. For instance, the term "ファミリーレストラン (family restaurant)" is abbreviated as "ファミレス (famires)" in Japanese, whereas English speakers do not abbreviate it in the same way as Japanese do. To investigate Japanese abbreviations, we collected them from different perspectives, that is, types of character and semantic categories. Table 1 shows abbreviation data types, their word counts, and so on. Ex-

[1] TV program titles contain colloquial expressions such as slang, pun, coined words, and dialect. For example, in well-written documents, we do not see such a expression as "I'm Not An Errand Boy!" showed in Figure 2.

Proceedings of the 2009 Workshop on Multiword Expressions, ACL-IJCNLP 2009, pages 63–70,
Suntec, Singapore, 6 August 2009. ©2009 ACL and AFNLP

amples are given in Figure 2 at the end of this paper.

We extracted abbreviations listed and described on the Japanese Wikipedia site [2], which is a multilingual project to create a complete and accurate open content encyclopedia. First, we collected lists of abbreviations classified according to types of Japanese character. Japanese has three original types of character: kanji, katakana, and hiragana. Other types of character are used, such as alphabets, numbers, and symbols. However, hiragana is mainly used with kanji, and numbers and symbols are used with other characters. Therefore, we used three abbreviation lists classified according to alphabetical words [3], katakana words [4], and kanji words with hiragana[5](Figure 2) on Wikipedia. We extracted pairs of abbreviations and their full forms from each list and obtained 928, 245, and 399 abbreviations, respectively.

Also, we extracted pairs of university names and their abbreviations from a list of university abbreviations on Wikipedia [6]. In Japanese, many names of organizations have a noun phrase structure combining several nouns, such as names of places ("日本 (Japan)", "東京 (Tokyo)"), names of fields ("医科 (medical)", "科学 (science)"), for whom ("女子 (female)"), and the type of organization("大学 (university)", "研究所"(research laboratory)). Therefore, we used names of universities and extracted 523 abbreviations. Almost all of the nouns are kanji.

Additionally, we extracted abbreviations of TV program titles from descriptions on each page of Wikipedia. This is because many TV program titles contain various types of characters or colloquial expressions different from the others we extracted. However, there are no lists of TV program titles in Wikipedia. Therefore, we gathered TV program titles satisfying the following criterion: the first sentence of the description of the Wikipedia page of the TV program title indicates that the page is about the TV program. And, in the same paragraph, if abbreviations are introduced by using key phrases such as "略語は A"(it means "it is abbreviated as A"), we extracted bold or parenthetical words in the key phrases. There were 326 abbreviations.

Finally, we gathered abbreviations of TV program titles in TV schedules written in short form because of space limitations. In this process, we used program titles in TV schedules in newspapers as short forms and EPG [7] data as long forms. When a title in the schedule is written with short form of the title with the same date, time, and channel as EPG data, we recognized that it is an abbreviation and the other is its full form. We extracted 603 abbreviations.

2.2 Characteristics

In this paper, we focus on abbreviations that lack some characters compared with the full forms. The followings are well-known characteristics of Japanese abbreviations (Sakai and Masuyama, 2005; Enoki et al., 2007; Murayama and Okumura, 2008). Abbreviations are created according to rules: (1) retain the beginning of a word and omit the rest (truncation); (2) divide an original term into base words, retaining several substrings from some of them, and combine them (contraction). In particular, four-mora[8] katakana abbreviations are often created by combining two-mora as in the case of the katakana words in Figure 2. Also, the length of an abbreviation in kanji tends to be two or three letters as in the case of the kanji words in Figure 2. Moreover, if an original term consists of katakana with the specific characters such as sokuon [9] and chōn[10] in the middle, these characters tend to be dropped in abbreviations. The second and third of katakana terms in Figure 2 are an example of this.

3 Proposed Method

In this section, we propose a new method to generate Japanese abbreviations by using the Log-Linear Model to rank abbreviation candidates. As mentioned in Section 2.2, Japanese abbreviation characteristics are evident in the composition of abbreviations, not in generation rules from their full forms. Therefore, we first generate possible abbreviations from an original term and rank them in descending order of probability of abbreviations. Our method uses a three-step process as

[2] http://ja.wikipedia.org/wiki/

[3] http://ja.wikipedia.org/wiki/欧文略語一覧

[4] http://ja.wikipedia.org/wiki/カタカナ略語一覧

[5] http://ja.wikipedia.org/wiki/漢字略語一覧

[6] http://ja.wikipedia.org/wiki/大学の略称

[7] EPG (Electronic Program Guide) broadcast on some multiplexes that provide detailed information about programs in an upcoming week on some stations.

[8] The minimal unit of a syllable.

[9] Sokuon is written as "ッ" in katakana and "っ" in hiragana to show a geminate consonant

[10] Chōn, is written as "—" to show a long vowel.

Class	Type	NT	Average number of characters								SC(%)
		(#)	Av	SD	(a)	(b)	(c)	(d)	(e)	(f)	
Character type	Alpha.	928	23.5	9.0	0.0	0.0	0.0	**21.4**	0.0	2.1	100.0
	Kata.	245	8.8	3.1	0.4	**8.0**	0.3	0.0	0.0	0.2	79.2
	Kanji	399	6.3	3.8	**5.9**	0.3	0.2	0.0	0.0	0.0	91.0
Semantic category	Univ.	523	6.0	1.4	**6.0**	0.0	0.0	0.0	0.0	0.0	98.1
	TV1	326	10.5	5.6	3.1	**3.6**	2.1	1.2	0.3	0.3	28.8
	TV2	603	10.9	4.2	1.6	**4.5**	2.6	1.7	0.1	0.4	19.1

Table 1: Abbreviation data sets, their types and number of terms(NT), average number of characters with standard deviation(SD), average number of characters per term in each type of character((a)kanji, (b)katakana, (c)hiragana, (d)alphabet, (e)number, (f)space), and proportion of terms with a single type of character (SC).

follows: 1)base word division, 2)candidate generation, and 3)ranking abbreviations.

3.1 Step1: Base Word Division

In this step, we divide terms into base words for abbreviations because Japanese is an agglutinative language. In order to deal with neologisms and colloquial expressions, we divide terms by using web search results instead of morphological analyzer.

When a term t is divided into two substrings after the ith charcter t, we denote the anterior half by $s_{i,ant}$ and the posterior half by $s_{i,post}$. A link strength $D(t_i)$ between $s_{i,ant}$ and $s_{i,post}$ is defined as follows:

$$D(t_i) = \frac{hit(s)}{min(hit(s_{i,ant}),hit(s_{i,post}))}$$

Note that hit(t) is calculated as the number of search results by using the term t in double quotes as one query on the Web[11]. The formulation of $D(t_i)$ is mostly the same as Simpson's Coefficient except that the numerator is modified. We divide the term t after the kth character where $D(t_k)$ is the smallest and repeat this process by using substrings divided in the previous operation as new t_is recursively. We heuristically set the stopping conditions as two kanji characters or four characters of other types. This dividing process works well because a set of words containing a term is stylized expression that is different from a sentence.

For example, suppose that a term t is "VivaVivaV6", which is one of the TV program titles. All divisions into two of the term are "V/ivaVivaV6", "Vi/vaVivaV6", $\cdots$, and "VivaVivaV/6". Here, the symbol "/"" indicates a division point. Then, $D(t_i)$s are calculated as follows:

$$D(t_1) = \frac{hit(\text{“VivaVivaV6”})}{min(hit(\text{“V”}), hit(\text{“ivaVivaV6”}))}$$

$$\vdots$$

$$D(t_9) = \frac{hit(\text{“VivaVivaV6”})}{min(hit(\text{“VivaVivaV”}), hit(\text{“6”}))}$$

When $D(t_8)$ is the smallest of all $D(t_i)$, "VivaVivaV6" is divided into "VivaViva" and "V6". The length of "V6" is two and is satisfied with the stopping conditions. Then, we continue to calculate $D(t_i)$ of "VivaViva" because the length of the substring is not satisfied with the stopping conditions.

Finally, the divisions are fixed based on the following modifications. If a division is just before the sokuon or the chōn, we eliminate the division because these cannot appear at the beginning of a word. Also, if the division is just before "の"(no), which is a hiragana character and one of the particles used to indicate possession and so on, we insert a division after the "の"(no) to make it one word because of the stopping conditions. Additionally, we combine some segments to form one word when there is a word in a transliteration dictionary of katakana corresponding to an English word.

3.2 Step2:Candidate Generation

In this step, we generate abbreviation candidates by applying the following simple rules to all words containing a certain term. These rules are based on the Japanese abbreviation characteristics described in Section 2.2.

1) Do not use this word
2) Use this word in full
3) Use the first character of this word

[11]. We used Yahoo web search API. http://developer.yahoo.co.jp/search/web/V1/webSearch.html

4) Use the first two characters of this word
5) Use the first three characters of this word
6) Drop sokuon and chōn, and do 4)
7) Drop sokuon and chōn, and do 5)
All rules are applied to all words divided by the process in step 1. For example, in the case of "Viva/Viva/V6", all rules are used for "Viva", "Viva", and "V6". Then, if 3), 3), 2) are used for each base word, we get a candidate "VVV6". With the rules, we can get all candidates combining substrings at the beginning of each word because we used the stop conditions of character length of less than four in step 1. However, note that we use mora instead of character in the case of phonographic characters. Also, we eliminate duplicative candidates.

3.3 Step3: Ranking Abbreviations

LLM is a probabilistic model widely used as a maximum entropy model for many NLP tasks (Manning and Schutze, 1999). We use standard LLM to rank the abbreviations.

Consider a set of observations x for each sample of an object or event with y. Log-Linear Model gives a probability $p(y|x;\lambda)$ of an event by representing an event y as features $f_j(x_l, y_k)$.

$$p(y|x;\lambda) = \frac{1}{Z(x,\lambda)} \exp(\Sigma_j \lambda_j f_j(x,y)) \quad (1)$$

Here, $\lambda_j (j = 1, ..., M)$ or α_j is a model parameter, and it represents the weight of a feature $f_j(x_l, y_k)$. Also, regularization term $Z(x,\lambda)$ is calculated as follows:

$$Z(x,\lambda) = \Sigma_{y' \in Y(x)} \exp(\Sigma_j \lambda_j f_j(x,y'))$$

Note that $Y(x)$ represents a set of output y corresponding to x. The numerator of the Formula (1) is the same as the following by replacing e^{λ_j} as α_j.

$$\begin{aligned} s(x,y,\lambda) &= \exp(\Sigma_j \lambda_j f_j(x,y)) \\ &= \alpha_1^{f_1(x,y)} \alpha_2^{f_2(x,y)} \cdots \alpha_M^{f_M(x,y)} \quad (2) \end{aligned}$$

We formalize the abbreviation generation task as a ranking problem in which the probability $p(y|x;\lambda)$ of abbreviation y in a given set $Y(x)$ of abbreviation candidates is modeled when its full form x is observed. For example, assume that you assign a full form "VivavivaV6" to x. The set $Y(x)$ contains abbreviation candidates generated from the full form in Step2 such as "VVV6", "VivaV", "ViVi", and so on. We used Amis implementation [12] for Log-Linear Model.

1) **Features**
We use the features below for the Japanese abbreviation characteristics with letter length and so on as mentioned in the Section 2.2. We denote a substring of a ith base word containing an abbreviation candidate by sub_i $(i = 1, \cdots, m)$, where m is the total number of base words. Then, let $ch(sub_i)$ denote letter type of character of sub_i, and let $len(sub_i)$ denote length of sub_i. Additionally, let $sum(len(sub_i), 1, m)$ denote a summation of $len(sub_i)(i = 1, \cdots, m)$, and let $com((f_1(i), f_2(i)), 1, m)$ denote a combination of a feature $f_1(i)$ and a feature $f_2(i)$ from $i = 1$ to $i = m$. Here, we show all categories of features we used as follows:.

- $tp = com((ch(sub_i, len(sub_i))), 1, m)$
- $tl = com((ch(sub_i)), 1, m)$
- $e = com((len(sub_i)), 1, m)$
- $w = sub_i \ (i = 1, \cdots, m)$
- $ab = sum(len(sub_i), 1, m)$
- $enum = m$

A substring of a ith base word is generated by applying one of the rules from 2) to 7) in Step 2. However, when an abbreviation candidate corresponds to one substring of its full form, we set its base word to the candidate itself even if the candidate was generated by combining some substrings.

Table 2 shows features for "VVV6" whose original term is "VivaVivaV6". Its base words sub_i are "V", "V", and "V6" because of the division as "V/V/V6". When $i = 1$, $ch(sub_1)$ is equal to ALPHA, that is, an alphabetical character, and $len(sub_1)$ is equal to 1. Therefore, for "VVV6", a feature in a category tp is generated by combination of the $ch(sub_i)$ and $len(sub_i)$ from $i = 1$ to $i = m$, that is, 1ALPHA_1ALPHA_2ALPHA. Other features are also generated by calculating in the same way as tp.

We cannot list all possible features because they depend on compositions of abbreviation candidates. Therefore, we prepare a $zero$ feature for each category. If features do not appear in positive examples in a training data set, we assign them to $zero$ features. For example, because a feature "1KANJI_5KANJI" in category "tp" does not appear in positive examples of a training set, we use

[12] http://www-tsujii.is.s.u-tokyo.ac.jp/amis/

Category	Feature
tp	1ALPHA_1ALPHA_2ALPHA
tl	ALPHA_ALPHA_ALPHA
e	1_1_2
w	V, V, V6
ab	ab4
enum	enum3

Table 2: Features for abbreviation “V/V/V6” whose full form is “Viva/Viva/V6”.

“tp0” as an alternative feature. However, $w0$ is assigned when any features in category “w” do not appear in them.

We assign l_1 to a set of all features that appear in positive examples in a training data set, such as 1ALPHA_1ALPHA_2ALPHA, 1_1_2, V, V, V6, ab4. We also assign l_0 to a set of $zero$ features, i.e. tp0, tl0, e0, w0, ab0, enum0. Then, let L denote a set merged l_1 and l_0.

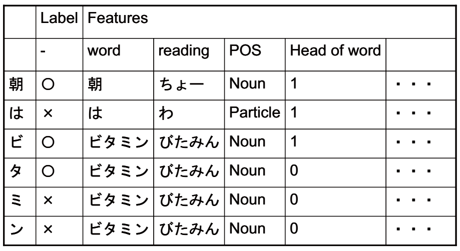
2) **Training and Test**
First, we obtain the above-mentioned feature set L with a training data set. Next, these features are assigned to all abbreviation candidates generated from the training data set in step 2. Then, a parameter α_j ($j = 1, \cdots, |L|$) of the Log-Linear Model is calculated by using Amis. Finally, the probabilities of all abbreviation candidates generated from a test data in step 2 are calculated by the Formula (2).

4 Evaluation

4.1 Baseline Method

CRFs (Lafferty et al., 2001) are Log-Linear Models, which are often used for the labeling or parsing of sequential data and are widely applied for many NLP tasks. Some researchers already used CRFs for abbreviation extraction (Okazaki et al., 2008(1)) or generation (Saikou et al., 2008). Therefore, we evaluate a method using CRFs as a baseline.

We formalize the abbreviation generation task as a sequence labeling problem in which each letter contained in an original term is to be used in its abbreviation[13] (Fig. 1). We also designed features attached to each character: morpheme word containing the letter, reading of the morpheme word, type of character, the first character or not in the morpheme word, the first character or not in the segment, and so on. We used MeCab[14] as a morphological analysis and CRF++ implementation [15] for CRFs.



	Label	Features				
	-	word	reading	POS	Head of word	
朝	○	朝	ちょー	Noun	1	・・・
は	×	は	わ	Particle	1	・・・
ビ	○	ビタミン	びたみん	Noun	1	・・・
タ	○	ビタミン	びたみん	Noun	0	・・・
ミ	×	ビタミン	びたみん	Noun	0	・・・
ン	×	ビタミン	びたみん	Noun	0	・・・

Figure 1: Feature examples of CRFs and values for the abbreviation “朝ビタ (asabita)” whose formal form is “朝はビタミン (asa wa bitamin)”.

4.2 Results and Discussion

We evaluate recall in the top 1, 5, 10, 30, and 50 abbreviation candidates generated with both the proposed method and the baseline method on the six data sets. The performance is measured under a ten-fold cross-validation where the parameters are fine-tuned in the top 30 in the training procedure.

Table 3 shows recall with the baseline method. Table 4 shows recall, and the bottom row in the table shows differences between recall with CRFs and that with proposed method in the top 30.

In the top 30, recall in Table 3 of alphabetical words, names of universities, and kanji words are 99.1%, 97.9%, and 92.5% respectively. From the point of view of types of character, most of these are composed of a single type of character as shown in column SC of Table 1. In contrast, recall in Table 3 of TV program titles 1 and 2 are 68.3% and 80.9% respectively. These results are much lower than the others. As a result of applying our method, Table 4 showed that recall of TV program titles improved 10.5% compared with the baseline method. This is because the method using CRFs cannot use the features of generated abbreviations since it is an approach to decide whether each character of an original form is to be used in its abbreviation. It seems that this leads to the disadvantages of generating abbreviations of TV program titles containing various types of character and colloquial expressions. However, there

[13] In (Saikou et al., 2008), they formalized the abbreviation generation task as a sequence labeling problem in which each mora contained in a term is to be used in its abbreviation. To avoid reading estimation, we generate abbreviations by abbreviating their original characters.

[14] http://mecab.sourceforge.net/

[15] http://crfpp.sourceforge.net/

Recall@n	Alphabet	Katakana	Kanji	Univ.	TV1	TV2
1	89.1%	29.4%	47.9%	19.9%	11.1%	9.3%
5	97.0%	67.3%	71.7%	80.9%	37.5%	45.8%
10	98.4%	77.1%	81.5%	92.9%	48.6%	62.5%
30	99.1%	89.0%	92.5%	97.9%	68.3%	80.9%
50	99.4%	93.9%	94.7%	98.9%	73.8%	86.9%

Table 3: Recall in the top 1, 5, 10, 30, and 50 abbreviation candidates generated with CRFs.

Recall@n	Alphabet	Katakana	Kanji	Univ.	TV1	TV2
1	87.4%	36.3%	39.1%	33.5%	19.9%	20.2%
5	92.2%	66.5%	65.2%	71.5%	48.2%	42.8%
10	93.0%	81.6%	73.9%	84.9%	61.3%	59.2%
30	94.1%	91.0%	85.2%	92.5%	78.8%	81.1%
50	94.4%	92.7%	86.7%	93.3%	85.3%	85.4%
all	95.6%	94.7%	90.4%	94.8%	93.9%	90.3%
Differences(Recall@30)	−5.1%	+2.0%	−7.3%	−5.4%	+10.5%	+0.2%

Table 4: Recall in the top 1, 5, 10, 30, and 50 abbreviation candidates generated with the proposed method, and differences between the recall with CRFs and that with the proposed method in the top 30.

is little difference of recall between the baseline and the proposed method for the TV program titles 2. This is because most of the TV program titles 2 were systematically created by simple rules such as getting the initial several letters that satisfy space limitations.

On the other hand, recall of the proposed method for alphabetical words, kanji words, and names of universities was −5.1%, −7.3%, −5.4% lower, respectively, than in the case of using the baseline method. This is because some abbreviations could not be generated by the given generation rules and, as can be seen in Table 4, recall of these data sets peaks. From these results, we conclude that the baseline method is suited to a term containing a single type of character such as alphabetical words and kanji words, whereas the proposed method is suited to a term containing multiple types of character.

When we used the division in step 2 as an alternative to MeCab, recall with CRFs differed approximately less than ±1% from recall in Table 3. On the other hand, when we used MeCab as an alternative to the division in step 2, recall with the proposed method was significantly lower than in Table 4.

We cannot compare our performance directly with the previous work because of the differences in data sets. For reference, Murayama et al. (2006) reported 68.4% recall in the top 30 with the Noisy-Channel Model. They used 851 abbreviations corresponding to 748 full forms extracted from Wikipedia. Saikou et al. (2008) reported 72.5% recall in the top 30 with CRFs. They used 51 abbreviations collected by WoZ[16] as test data and 781 abbreviations that appeared in Wikipedia as training data.

4.3 Combination of two methods

Table 4 shows that the baseline method is better for the alphabetical words, names of universities, and kanji words, whereas the proposed method is better for others. However the classification on Table 1 is made by hand. Here, we automatically classified them into the following case A and B based on the conditions according to types of character after merging the six data sets in Table 1. Then, we applied the method with CRFs to the case A and the proposed method to the case B.

Case A is when an original term is (1) an alphabetical term with more than two words, (2) a kanji term in which other characters do not constitute, or (3) a term of (1) or (2) with numerals or symbols. Case B is when an original term does not fulfill the conditions of the case A.

The total number of abbreviations was 3114 (1921 in the case A and 1103 in the case B). Ta-

[16] Wizard-of-Oz

ble 5 shows the number of abbreviations in each case for each data set. The total performance was measured by calculating weighted average for two recall scores, that is, in the case of A and B measured under a ten-fold cross-validation in the top 30. As a result, recall was 97.1% and 76.9% in the case A and B respectively, and the total recall was 89.4%. Additionally, we conducted an experiment in which the method with CRFs was applied to all the abbreviations as a baseline. The recall was 87.0% measured under a ten-fold cross-validation in the top 30. The results show that it is better to apply different methods according to types of character than to apply one method to the entire data set.

5 Conclusion

In this paper, we proposed a method for generating Japanese abbreviations from their full forms with LLM. As a result of experiments, the proposed method was confirmed to be effective for TV program titles. It achieved 78.8% recall in the top 30, and improved 10.5% from a baseline method using CRFs that achieved 68.3% recall. We also described difficulties in generating Japanese abbreviations by examining six data sets classified according to types of character and semantic categories. Consequently, we showed that the baseline method is suited to a term containing a single type of character such as alphabetical words and kanji words, whereas the proposed method is suited to a term containing multiple types of character. In the future, we will apply the proposed method to Japanese abbreviations generated with transliteration between English and Japanese[17]. We also plan to narrow down the top ranked abbreviation candidates by using the search results on the Web.

References

Eytan Adar. 2004. SaRAD: A Simple and Robust Abbreviation Dictionary. *Bioinformatics*, 20(4):527–533.

Masanori Enoki, Mika Koho, Kenko Ota, and Masuzo Yanagida. 2007. Automatic Generation Abbriviated Forms of Japanese Expressions and its Appričations to Speech Recognition (in Japanese). *IPSJ SIG Notes*, 313–318.

Toru Hisamitsu and Yoshiki Niwa. 2001. Extracting useful terms from parenthetical expression by combining simple rules and statistical measures: A comparative evaluation of bigram statistics. Didier Bourigault and Christian Jacquemin and Marie-Claude L'Homme editors. *Recent Advances in Computational Terminology*, 209–224.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the ICML-2001*, 282–289.

Christopher D. Manning and Hinrich Schutze. 1999. The MIT Press. *Foundations of statistical natural language processing*.

Norifumi Murayama and Manabu Okumura. 2006. Automatic Generation of Abbreviations with Noisy-channel model (in Japanese). *NLP2006*, 763–766.

Norifumi Murayama and Manabu Okumura. 2008. Statistical Model for Japanese Abbreviations. *Proceedings of the PRICAI-08*, 260–272.

David Nadeau and Peter D. Turney. 2005. A supervised learning approach to acronym identification. *Proceedings of the AI'2005*, 10 pages.

Naoaki Okazaki and Sophia Ananiadou. 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(24):3089–3095.

Naoaki Okazaki, Sophia Ananiadou, and Jun'ichi Tsujii. 2008(1). A Discriminative Alignment Model for Abbreviation Recognition. *Proceedings of the Coling 2008*, 657–664.

Naoaki Okazaki, Mitsuru Ishizuka, and Jun'ichi Tsujii. 2008(2). A Discriminative Approach to Japanese Abbreviation Extraction. *Proceedings of the IJCNLP 2008*, 889–894.

Youngja Park and Roy J. Byrd. 2001. Hybrid Text Mining for Finding Abbreviations and Their Definitions. *Proceedings of the EMNLP-2001*, 126–133.

Masahiro Saikou, Kiyokazu Miki, and Hiroaki Hattori. 2008. Automatic Generation of Abbreviations with Probabilistic Models (in Japanese). *The Acoustical Society of Japan*, 237–238.

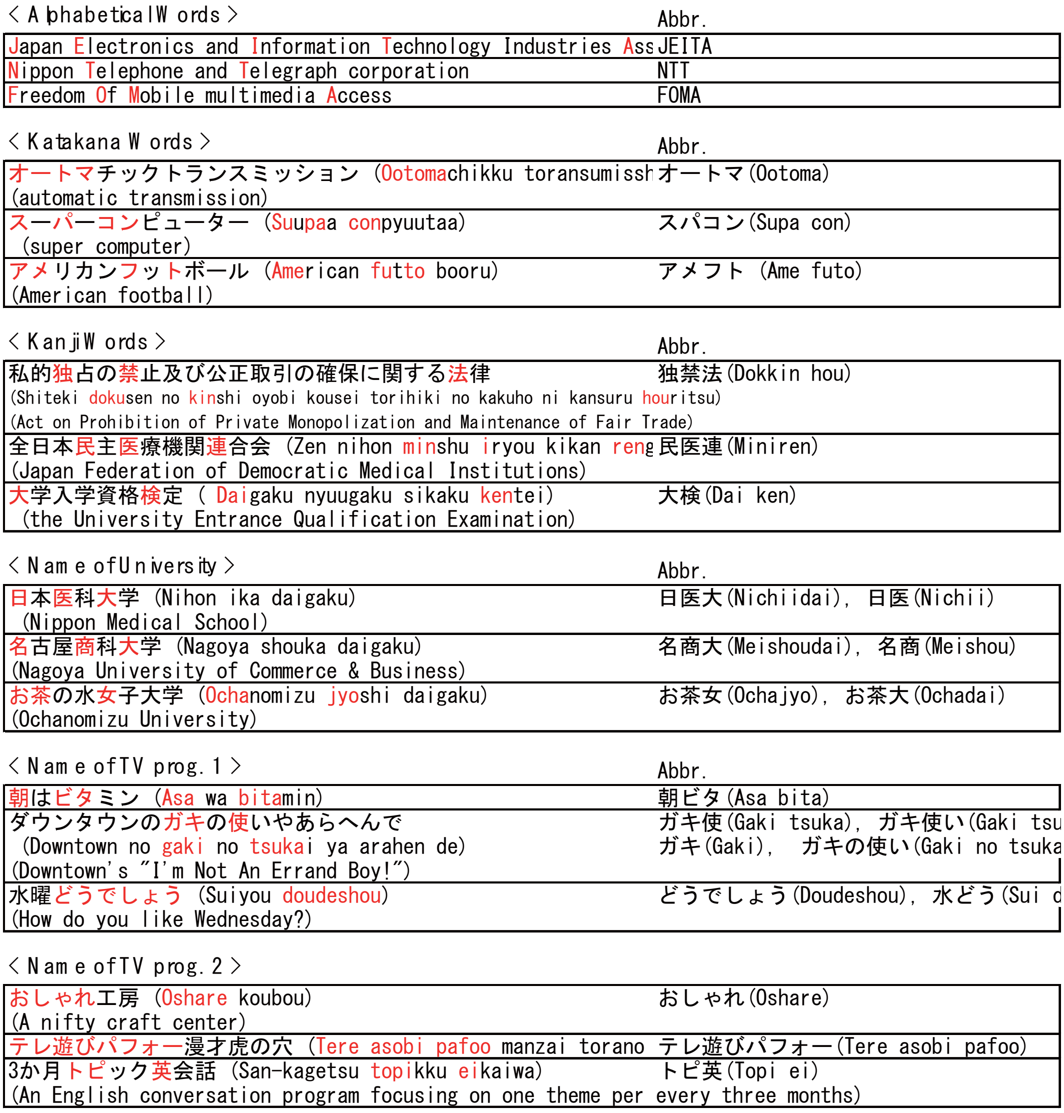
Hiroyuki Sakai and Shigeru Masuyama. 2005. Improvement of the Method for Acquiring Knowledge from a Single Corpus on Correspondences between Abbreviations and Their Original words (in Japanese). *Journal of Natural Language Processing*, 12(4):207–231.

Ariel S. Schwartz and Marti A. Hearst. 2003. A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. *Proceedings of the PSB-2003*, 451–462.

[17] For example, “ミュージックステーション (music station)” is abbreviated as “M ステ (emu sute)”. It is created by using a substring “M” of “Music” translated from “ミュージック” and a substring “ステ” of “ステーション”.

	Case A	%	Case B	%	Total
Alpha	917	(98.8)	11	(1.2)	928
Kata.	0	(0)	245	(100)	245
Kanji	363	(91.0)	36	(9.0)	399
Univ.	513	(98.1)	10	(1.9)	523
TV1	27	(8.3)	299	(91.7)	326
TV2	49	(8.1)	554	(91.9)	603
Total	1869	(61.8)	1155	(38.2)	3024
(Rec@30)					
CRF	-		-		87.0%
CRF/LLM	97.1%		76.9%		89.4%

Table 5: The number of abbreviations in case A and B for the six data sets, and recall in the top 30.



< Alphabetical Words >	Abbr.
Japan Electronics and Information Technology Industries Ass	JEITA
Nippon Telephone and Telegraph corporation	NTT
Freedom Of Mobile multimedia Access	FOMA

< Katakana Words >	Abbr.
オートマチックトランスミッション (Ootomachikku toransumissh (automatic transmission)	オートマ(Ootoma)
スーパーコンピューター (Suupaa conpyuutaa) (super computer)	スパコン(Supa con)
アメリカンフットボール (American futto booru) (American football)	アメフト (Ame futo)

< Kanji Words >	Abbr.
私的独占の禁止及び公正取引の確保に関する法律 (Shiteki dokusen no kinshi oyobi kousei torihiki no kakuho ni kansuru houritsu) (Act on Prohibition of Private Monopolization and Maintenance of Fair Trade)	独禁法(Dokkin hou)
全日本民主医療機関連合会 (Zen nihon minshu iryou kikan reng (Japan Federation of Democratic Medical Institutions)	民医連(Miniren)
大学入学資格検定 (Daigaku nyuugaku sikaku kentei) (the University Entrance Qualification Examination)	大検(Dai ken)

< Name of University >	Abbr.
日本医科大学 (Nihon ika daigaku) (Nippon Medical School)	日医大(Nichiidai), 日医(Nichii)
名古屋商科大学 (Nagoya shouka daigaku) (Nagoya University of Commerce & Business)	名商大(Meishoudai), 名商(Meishou)
お茶の水女子大学 (Ochanomizu jyoshi daigaku) (Ochanomizu University)	お茶女(Ochajyo), お茶大(Ochadai)

< Name of TV prog. 1 >	Abbr.
朝はビタミン (Asa wa bitamin)	朝ビタ(Asa bita)
ダウンタウンのガキの使いやあらへんで (Downtown no gaki no tsukai ya arahen de) (Downtown's "I'm Not An Errand Boy!")	ガキ使(Gaki tsuka), ガキ使い(Gaki tsu ガキ(Gaki), ガキの使い(Gaki no tsuka
水曜どうでしょう (Suiyou doudeshou) (How do you like Wednesday?)	どうでしょう(Doudeshou), 水どう(Sui d

< Name of TV prog. 2 >	
おしゃれ工房 (Oshare koubou) (A nifty craft center)	おしゃれ(Oshare)
テレ遊びパフォー漫才虎の穴 (Tere asobi pafoo manzai torano	テレ遊びパフォー(Tere asobi pafoo)
3か月トピック英会話 (San-kagetsu topikku eikaiwa) (An English conversation program focusing on one theme per every three months)	トピ英(Topi ei)

Figure 2: Example of data sets.