Chunking Japanese Compound Functional Expressions by Machine Learning

Masatoshi Tsuchiya[†] and **Takao Shime**[‡] and **Toshihiro Takagi**[‡]
Takehito Utsuro[††] and **Kiyotaka Uchimoto**[†‡] and **Suguru Matsuyoshi**[‡]
Satoshi Sato[‡‡] and **Seiichi Nakagawa**[††]
†Computer Center / ‡†Department of Information and Computer Sciences,
Toyohashi University of Technology, Tenpaku-cho, Toyohashi, 441–8580, JAPAN
‡Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto, 606–8501, JAPAN
††Graduate School of Systems and Information Engineering, University of Tsukuba,
1-1-1, Tennodai, Tsukuba, 305-8573, JAPAN
†‡National Institute of Information and Communications Technology,
3–5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619–0289 JAPAN
‡‡Graduate School of Engineering, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, 464–8603, JAPAN

Abstract

The Japanese language has various types of compound functional expressions, which are very important for recognizing the syntactic structures of Japanese sentences and for understanding their semantic contents. In this paper, we formalize the task of identifying Japanese compound functional expressions in a text as a chunking problem. We apply a machine learning technique to this task, where we employ that of Support Vector Machines (SVMs). We show that the proposed method significantly outperforms existing Japanese text processing tools.

1 Introduction

As in the case of other languages, the Japanese language has various types of functional words such as post-positional particles and auxiliary verbs. In addition to those functional words, the Japanese language has much more compound functional expressions which consist of more than one words including both content words and functional words. Those single functional words as well as compound functional expressions are very important for recognizing the syntactic structures of Japanese sentences and for understanding their semantic contents. Recognition and understanding of them are also very important for various kinds of NLP applications such as dialogue systems, machine translation, and question answering. However, recognition and semantic interpretation of compound functional expressions are especially difficult because it often happens that one compound expression may have both a literal (in other words, compositional) *content word* usage and a non-literal (in other words, non-compositional) *functional* usage.

For example, Table 1 shows two example sentences of a compound expression "に (ni) ついて (tsuite)", which consists of a post-positional particle "に (ni)", and a conjugated form "ついて (tsuite)" of a verb "つく (tsuku)". In the sentence (A), the compound expression functions as a case-marking particle and has a non-compositional functional meaning "*about*". On the other hand, in the sentence (B), the expression simply corresponds to a literal concatenation of the usages of the constituents: the post-positional particle "に (ni)" and the verb "ついて (tsuite)", and has a content word meaning "*follow*". Therefore, when considering machine translation of those Japanese sentences into English, it is necessary to precisely judge the usage of the compound expression "に (ni) ついて (tsuite)", as shown in the English translation of the two sentences in Table 1.

There exist widely-used Japanese text processing tools, i.e., pairs of a morphological analysis tool and a subsequent parsing tool, such as JUMAN[1]+ KNP[2] and ChaSen[3]+ CaboCha[4]. However, they process those compound expressions only partially, in that their morphological analysis dictionaries list only limited number of compound expressions. Furthermore, even if certain expressions are listed in a morphological analysis

[1] http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman-e.html
[2] http://www.kc.t.u-tokyo.ac.jp/nl-resource/knp-e.html
[3] http://chasen.naist.jp/hiki/ChaSen/
[4] http://chasen.org/~taku/software/cabocha/

Table 1: Translation Selection of a Japanese Compound Expression "に (ni) ついて (tsuite)"

(A)	私 (watashi)	は (ha)	彼 (kare)	に (ni) ついて (tsuite)		話した (hanashita)
	(*I*)	(*TOP*)	(*he*)	(*about*)		(*talked*)
	(I talked about him.)					
(B)	私 (watashi)	は (ha)	彼 (kare)	に (ni)	ついて (tsuite)	走った (hashitta)
	(*I*)	(*TOP*)	(*he*)	(*ACC*)	(*follow*)	(*ran*)
	(I ran following him.)					

Table 2: Classification of Functional Expressions based on Grammatical Function

Grammatical Function Type		# of major expressions	# of variants	Example
post-positional particle type	subsequent to predicate / modifying predicate	36	67	となると (to-naru-to)
	subsequent to nominal / modifying predicate	45	121	にかけては (ni-kakete-ha)
	subsequent to predicate, nominal / modifying nominal	2	3	という (to-iu)
auxiliary verb type		42	146	ていい (te-ii)
total		125	337	—

dictionary, those existing tools often fail in resolving the ambiguities of their usages, such as those in Table 1. This is mainly because the framework of those existing tools is not designed so as to resolve such ambiguities of compound (possibly functional) expressions by carefully considering the context of those expressions.

Considering such a situation, it is necessary to develop a tool which properly recognizes and semantically interprets Japanese compound functional expressions. In this paper, we apply a machine learning technique to the task of identifying Japanese compound functional expressions in a text. We formalize this identification task as a chunking problem. We employ the technique of Support Vector Machines (SVMs) (Vapnik, 1998) as the machine learning technique, which has been successfully applied to various natural language processing tasks including chunking tasks such as phrase chunking (Kudo and Matsumoto, 2001) and named entity chunking (Mayfield et al., 2003). In the preliminary experimental evaluation, we focus on 52 expressions that have balanced distribution of their usages in the newspaper text corpus and are among the most difficult ones in terms of their identification in a text. We show that the proposed method significantly outperforms existing Japanese text processing tools as well as another tool based on hand-crafted rules. We further show that, in the proposed SVMs based framework, it is sufficient to collect and manually annotate about 50 training examples per expression.

2 Japanese Compound Functional Expressions and their Example Database

2.1 Japanese Compound Functional Expressions

There exist several collections which list Japanese functional expressions and examine their usages. For example, (Morita and Matsuki, 1989) examine 450 functional expressions and (Group Jamashii, 1998) also lists 965 expressions and their example sentences. Compared with those two collections, *Gendaigo Hukugouji Youreishu* (National Language Research Institute, 2001) (henceforth, denoted as *GHY*) concentrates on 125 major functional expressions which have non-compositional usages, as well as their variants[5] (337 expressions in total), and collects example sentences of those expressions. As a first step of developing a tool for identifying Japanese compound functional expressions, we start with those 125 major functional expressions and their variants. In this paper, we take an approach of regarding each of those variants as a fixed expression, rather than a semi-fixed expression or a syntactically-flexible expression (Sag et al., 2002). Then, we focus on evaluating the effectiveness of straightforwardly applying a stan-

[5]For each of those 125 major expressions, the differences between it and its variants are summarized as below: i) insertion/deletion/alternation of certain particles, ii) alternation of synonymous words, iii) normal/honorific/conversational forms, iv) base/adnominal/negative forms.

Table 3: Examples of Classifying Functional/Content Usages

	Expression	Example sentence (English translation)	Usage
(1)	となると (to-naru-to)	しかしこの病気に効果がない[となると]事は重大だ。 (The situation is serious *if* it is not effective against this disease.)	functional (となると (to-naru-to) = *if*)
(2)	となると (to-naru-to)	彼が社長になるための条件の一つ[となると]考えられている。 (They think that it will *become* a requirement for him to be the president.)	content (∼ となると (to-naru-to) = *that (something) becomes* ∼)
(3)	にかけては (ni-kakete-ha)	お金を儲けること[にかけては]素晴らしい才能をもっている。 (He has a great talent *for* earning money.)	functional (∼ にかけては (ni-kakete-ha) = *for* ∼)
(4)	にかけては (ni-kakete-ha)	あまり気[にかけては]いない。 (I do not *worry* about it.)	content ((∼ を) 気にかけては ((∼)-wo-ki-ni-kakete-ha) = *worry about* ∼)
(5)	という (to-iu)	彼は生きている[という]知らせを聞いた。 (I heard *that* he is alive.)	functional (∼ という (to-iu) = *that* ∼)
(6)	という (to-iu)	「遊びに来て下さい」[という]人もいる。 (Somebody *says* “Please visit us.”.)	content (∼ という (to-iu) = *say (that)* ∼)
(7)	ていい (te-ii)	この議論が終ったら休憩し[ていい]。 (You *may* have a break after we finish this discussion.)	functional (∼ ていい (te-ii) = *may* ∼)
(8)	ていい (te-ii)	このかばんは大きく[ていい]。 (This bag is *nice because* it is big.)	content (∼ ていい (te-ii) = *nice because* ∼)

dard chunking technique to the task of identifying Japanese compound functional expressions.

As in Table 2, according to their grammatical functions, those 337 expressions in total are roughly classified into *post-positional particle* type, and *auxiliary verb* type. Functional expressions of post-positional particle type are further classified into three subtypes: i) those subsequent to a predicate and modifying a predicate, which mainly function as conjunctive particles and are used for constructing subordinate clauses, ii) those subsequent to a nominal, and modifying a predicate, which mainly function as case-marking particles, iii) those subsequent to a nominal, and modifying a nominal, which mainly function as adnominal particles and are used for constructing adnominal clauses. For each of those types, Table 2 also shows the number of major expressions as well as that of their variants listed in *GHY*, and an example expression. Furthermore, Table 3 gives example sentences of those example expressions as well as the description of their usages.

2.2 Issues on Identifying Compound Functional Expressions in a Text

The task of identifying Japanese compound functional expressions roughly consists of detecting candidates of compound functional expressions in a text and of judging the usages of those candidate expressions. The class of Japanese compound functional expressions can be regarded as closed and their number is at most a few thousand.

Table 4: Examples of Detecting more than one Candidate Expression

	Expression	Example sentence (English translation)	Usage
(9)	という (to-iu)	それが試合 [という] ものの難しさだ。 (That's why a match is not so easy.)	functional (NP_1 という (to-iu) NP_2 = NP_2 *called as* NP_1)
(10)	というものの (to-iu-mono-no)	勝った [というものの]、スコアは悪い。 (*Although* he won, the score is bad.)	functional (∼ というものの (to-iu-mono-no) = *although* ∼)

Therefore, it is easy to enumerate all the compound functional expressions and their morpheme sequences. Then, in the process of detecting candidates of compound functional expressions in a text, the text are matched against the morpheme sequences of the compound functional expressions considered.

Here, most of the 125 major functional expressions we consider in this paper are compound expressions which consist of one or more content words as well as functional words. As we introduced with the examples of Table 1, it is often the case that they have both a compositional *content word* usage as well as a non-compositional *functional* usage. For example, in Table 3, the expression "となると (to-naru-to)" in the sentence (2) has the meaning " *that (something) becomes* ∼", which corresponds to a literal concatenation of the usages of the constituents: the post-positional particle "と", the verb "なる", and the post-positional particle "と", and can be regarded as a *content word* usage. On the other hand, in the case of the sentence (1), the expression "となると (to-naru-to)" has a non-compositional functional meaning "*if*". Based on this discussion, we classify the usages of those expressions into two classes: *functional* and *content*. Here, *functional* usages include both non-compositional and compositional functional usages, although most of the functional usages of those 125 major expressions can be regarded as non-compositional. On the other hand, *content* usages include compositional content word usages only.

More practically, in the process of detecting candidates of compound functional expressions in a text, it can happen that more than one candidate expression is detected. For example, in Table 4, both of the candidate compound functional expressions "という (to-iu)" and "というものの (to-iu-mono-no)" are detected in the sentence (9). This is because the sequence of the two morphemes "と (to)" and "いう (iu)" constituting the candidate expression "という (to-iu)" is a subsequence of the four morphemes constituting the candidate expression "というものの (to-iu-mono-no)" as below:

Morpheme sequence
と (to)　いう (iu)　もの (mono)　の (no)

Candidate expression という (to-iu)
[と (to)　いう (iu)]　もの (mono)　の (no)

Candidate expression というものの (to-iu-mono-no)
[と (to)　いう (iu)　もの (mono)　の (no)]

This is also the case with the sentence (10).

Here, however, as indicated in Table 4, the sentence (9) is an example of the *functional* usage of the compound functional expression "という (to-iu)", where the sequence of the *two* morphemes "と (to)" and "いう (iu)" should be identified and chunked into a compound functional expression. On the other hand, the sentence (10) is an example of the *functional* usage of the compound functional expression "というものの (to-iu-mono-no)", where the sequence of the *four* morphemes "と (to)", "いう (iu)", "もの (mono)", and "の (no)" should be identified and chunked into a compound functional expression. Actually, in the result of our preliminary corpus study, at least in about 20% of the occurrences of Japanese compound functional expressions, more than one candidate expression can be detected. This result indicates that it is necessary to consider more than one candidate expression in the task of identifying a Japanese compound functional expression, and also in the task of classifying the functional/content usage of a candidate expression. Thus, in this paper, based on this observation, we formalize the task of identifying Japanese compound functional expressions as a *chunking* problem, rather than a *classification* problem.

Table 5: Number of Sentences collected from 1995 Mainichi Newspaper Texts (for 337 Expressions)

	# of expressions
$50 \leq$ # of sentences	187 (55%)
$0 <$ # of sentences < 50	117 (35%)
# of sentences $= 0$	33 (10%)

2.3 Developing an Example Database

We developed an example database of Japanese compound functional expressions, which is used for training/testing a chunker of Japanese compound functional expressions (Tsuchiya et al., 2005). The corpus from which we collect example sentences is 1995 Mainichi newspaper text corpus (1,294,794 sentences, 47,355,330 bytes). For each of the 337 expressions, 50 sentences are collected and chunk labels are annotated according to the following procedure.

1. The expression is morphologically analyzed by ChaSen, and its morpheme sequence[6] is obtained.
2. The corpus is morphologically analyzed by ChaSen, and 50 sentences which include the morpheme sequence of the expression are collected.
3. For each sentence, every occurrence of the 337 expressions is annotated with one of the usages *functional/content* by an annotator[7].

Table 5 classifies the 337 expressions according to the number of sentences collected from the 1995 Mainichi newspaper text corpus. For more than half of the 337 expressions, more than 50 sentences are collected, although about 10% of the 377 expressions do not appear in the whole corpus. Out of those 187 expressions with more than 50 sentences, 52 are those with balanced distribution of the *functional/content* usages in the newspaper text corpus. Those 52 expressions can be regarded as among the most difficult ones in the task of identifying and classifying *functional/content* usages. Thus, this paper focuses on those 52 expressions in the training/testing of chunking compound functional expressions. We extract 2,600 sentences (= 52 expressions $\times$ 50 sentences) from the whole example database and use them for training/testing the chunker. The number of the morphemes for the 2,600 sentences is 92,899. We ignore the chunk labels for the expressions other than the 52 expressions, resulting in 2,482/701 chunk labels for the functional/content usages, respectively.

[6]For those expressions whose constituent has conjugation and the conjugated form also has the same usage as the expression with the original form, the morpheme sequence is expanded so that the expanded morpheme sequences include those with conjugated forms.

[7]For the most frequent 184 expressions, on the average, the agreement rate between two human annotators is 0.93 and the Kappa value is 0.73, which means allowing tentative conclusions to be drawn (Carletta, 1996; Ng et al., 1999). For 65% of the 184 expressions, the Kappa value is above 0.8, which means good reliability.

3 Chunking Japanese Compound Functional Expressions with SVMs

3.1 Support Vector Machines

The principle idea of SVMs is to find a separate hyperplane that maximizes the margin between two classes (Vapnik, 1998). If the classes are not separated by a hyperplane in the original input space, the samples are transformed in a higher dimensional features space.

Giving $\mathbf{x}$ is the context (a set of features) of an input example; $\mathbf{x}_i$ and $y_i (i = 1, ..., l,\ \mathbf{x}_i \in \mathbf{R}^n,\ y_i \in \{1, -1\})$ indicate the context of the training data and its category, respectively; The decision function f in SVM framework is defined as:

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (1)$$

where K is a kernel function, $b \in \mathbf{R}$ is a threshold, and α_i are weights. Besides, the weights α_i satisfy the following constraints:

$$0 \leq \alpha_i \leq C\ (i = 1, ..., l) \quad (2)$$

$$\textstyle\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (3)$$

where C is a misclassification cost. The $\mathbf{x}_i$ with non-zero α_i are called support vectors. To train an SVM is to find the α_i and the b by solving the optimization problem; maximizing the following under the constraints of (2) and (3):

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \quad (4)$$

The kernel function K is used to transform the samples in a higher dimensional features space. Among many kinds of kernel functions available, we focus on the d-th polynomial kernel:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d \quad (5)$$

Through experimental evaluation on chunking Japanese compound functional expressions, we compared polynomial kernels with $d = 1$, 2, and 3. Kernels with $d = 2$ and 3 perform best, while the kernel with $d = 3$ requires much more computational cost than that with $d = 2$. Thus, throughout the paper, we show results with the quadratic kernel ($d = 2$).

3.2 Chunking with SVMs

This section describes details of formalizing the chunking task using SVMs. In this paper, we use an SVMs-based chunking tool YamCha[8] (Kudo and Matsumoto, 2001). In the SVMs-based chunking framework, SVMs are used as classifiers for assigning labels for representing chunks to each token. In our task of chunking Japanese compound functional expressions, each sentence is represented as a sequence of morphemes, where a morpheme is regarded as a token.

3.2.1 Chunk Representation

For representing proper chunks, we employ IOB2 representation, one of those which have been studied well in various chunking tasks of natural language processing (Tjong Kim Sang, 1999; Kudo and Matsumoto, 2001). This method uses the following set of three labels for representing proper chunks.

- **I** Current token is a middle or the end of a chunk consisting of more than one token.
- **O** Current token is outside of any chunk.
- **B** Current token is the beginning of a chunk.

As we described in section 2.2, given a candidate expression, we classify the usages of the expression into two classes: *functional* and *content*. Accordingly, we distinguish the chunks of the two types: the *functional* type chunk and the *content* type chunk. In total, we have the following five labels for representing those chunks: **B-functional, I-functional, B-content, I-content**, and **O**. Table 6 gives examples of those chunk labels representing chunks.

Finally, as for exending SVMs to multi-class classifiers, we experimentally compare the *pairwise* method and the *one vs. rest* method, where the *pairwise* method slightly outperformed the *one vs. rest* method. Throughout the paper, we show results with the *pairwise* method.

[8] http://chasen.org/~taku/software/yamcha/

3.2.2 Features

For the feature sets for training/testing of SVMs, we use the information available in the surrounding context, such as the morphemes, their parts-of-speech tags, as well as the chunk labels. More precisely, suppose that we identify the chunk label c_i for the i-th morpheme:

	$\longrightarrow$ Parsing Direction $\longrightarrow$				
Morpheme	m_{i-2}	m_{i-1}	m_i	m_{i+1}	m_{i+2}
Feature set at a position	F_{i-2}	F_{i-1}	F_i	F_{i+1}	F_{i+2}
Chunk label	c_{i-2}	c_{i-1}	$\boxed{c_i}$		

Here, m_i is the morpheme appearing at i-th position, F_i is the feature set at i-th position, and c_i is the chunk label for i-th morpheme. Roughly speaking, when identifying the chunk label c_i for the i-th morpheme, we use the feature sets F_{i-2}, F_{i-1}, F_i, F_{i+1}, F_{i+2} at the positions $i-2$, $i-1$, i, $i+1$, $i+2$, as well as the preceding two chunk labels c_{i-2} and c_{i-1}.

The detailed definition of the feature set F_i at i-th position is given below. The feature set F_i is defined as a tuple of the *morpheme feature* $MF(m_i)$ of the i-th morpheme m_i, the *chunk candidate feature* $CF(i)$ at i-th position, and the *chunk context feature* $OF(i)$ at i-th position.

$$F_i \;=\; \langle\, MF(m_i),\, CF(i),\, OF(i)\,\rangle$$

The morpheme feature $MF(m_i)$ consists of the lexical form, part-of-speech, conjugation type and form, base form, and pronunciation of m_i.

The chunk candidate feature $CF(i)$ and the chunk context feature $OF(i)$ are defined considering the candidate compound functional expression, which is a sequence of morphemes including the morpheme m_i at the current position i. As we described in section 2, the class of Japanese compound functional expressions can be regarded as closed and their number is at most a few thousand. Therefore, it is easy to enumerate all the compound functional expressions and their morpheme sequences. Chunk labels other than **O** should be assigned to a morpheme only when it constitutes at least one of those enumerated compound functional expressions. Suppose that a sequence of morphemes $m_j \ldots m_i \ldots m_k$ including m_i at the current position i constitutes a candidate functional expression E as below:

$$m_{j-2}\ m_{j-1}\ \underbrace{\boxed{m_j\ \ldots\ m_i\ \ldots\ m_k}}_{\substack{\text{candidate } E \text{ of}\\ \text{a compound}\\ \text{functional expression}}}\ m_{k+1}\ m_{k+2}$$

where the morphemes m_{j-2}, m_{j-1}, m_{k+1}, and m_{k+2} are at immediate left/right contexts of E. Then, the chunk candidate feature $CF(i)$ at i-th position is defined as a tuple of the number of morphemes constituting E and the position of m_i in E. The chunk context feature $OF(i)$ at i-th position is defined as a tuple of the morpheme features

Table 6: Examples of Chunk Representation and Chunk Candidate/Context Features

(a) Sentence (7) of Table 3

Morpheme	(*English translation*)	Chunk label	Chunk candidate feature	Chunk context feature
この (kono)	(*this*)	**O**	$\emptyset$	$\emptyset$
議論 (giron)	(*discussion*)	**O**	$\emptyset$	$\emptyset$
が (ga)	(*NOM*)	**O**	$\emptyset$	$\emptyset$
終わっ(owatt)	(*finish*)	**O**	$\emptyset$	$\emptyset$
たら (tara)	(*after*)	**O**	$\emptyset$	$\emptyset$
休憩 (kyuukei)	(*break*)	**O**	$\emptyset$	$\emptyset$
し (shi)	(*have*)	**O**	$\emptyset$	$\emptyset$
て (te)	(*may*)	**B-functional**	$\langle 2, 1 \rangle$	$\langle\, MF($休憩 (kyuukei)$), \emptyset,\ MF($し (shi)$), \emptyset,$
いい (ii)		**I-functional**	$\langle 2, 2 \rangle$	$MF($。 (period)$), \emptyset,\ \emptyset,\ \emptyset\,\rangle$
。(period)	(period)	**O**	$\emptyset$	$\emptyset$

(b) Sentence (8) of Table 3

Morpheme	(*English translation*)	Chunk label	Chunk candidate feature	Chunk context feature
この (kono)	(*this*)	**O**	$\emptyset$	$\emptyset$
かばん (bag)	(*discussion*)	**O**	$\emptyset$	$\emptyset$
は (ha)	(*TOP*)	**O**	$\emptyset$	$\emptyset$
大きく (ookiku)	(*big*)	**O**	$\emptyset$	$\emptyset$
て (te)	(*because*)	**B-content**	$\langle 2, 1 \rangle$	$\langle\, MF($は (ha)$), \emptyset,\ MF($大きく (ookiku)$), \emptyset,$
いい (ii)	(*nice*)	**I-content**	$\langle 2, 2 \rangle$	$MF($。 (period)$), \emptyset,\ \emptyset,\ \emptyset\,\rangle$
。(period)	(period)	**O**	$\emptyset$	$\emptyset$

as well as the chunk candidate features at immediate left/right contexts of E.

$$\begin{aligned} CF(i) &= \langle \text{ length of } E,\ \text{position of } m_i \text{ in } E \,\rangle \\ OF(i) &= \langle\, MF(m_{j-2}),\ CF(j-2), \\ &\quad MF(m_{j-1}),\ CF(j-1), \\ &\quad MF(m_{k+1}),\ CF(k+1), \\ &\quad MF(m_{k+2}),\ CF(k+2)\,\rangle \end{aligned}$$

Table 6 gives examples of chunk candidate features and chunk context features

It can happen that the morpheme at the current position i constitutes more than one candidate compound functional expression. For example, in the example below, the morpheme sequences $m_{i-1}m_im_{i+1}$, $m_{i-1}m_i$, and $m_im_{i+1}m_{i+2}$ constitute candidate expressions E_1, E_2, and E_3, respectively.

Morpheme sequence	m_{i-1}	m_i	m_{i+1}	m_{i+2}
Candidate E_1	m_{i-1}	m_i	m_{i+1}	
Candidate E_2	m_{i-1}	m_i		
Candidate E_3		m_i	m_{i+1}	m_{i+2}

In such cases, we prefer the one starting with the leftmost morpheme. If more than one candidate expression starts with the leftmost morpheme, we prefer the longest one. In the example above, we prefer the candidate E_1 and construct the chunk candidate features and chunk context features considering E_1 only.

4 Experimental Evaluation

The detail of the data set we use in the experimental evaluation was presented in section 2.3. As we show in Table 7, performance of our SVMs-based chunkers as well as several baselines including existing Japanese text processing tools is evaluated in terms of precision/recall/$F_{\beta=1}$ of identifying *functional* chunks. Performance is evaluated also in terms of accuracy of classifying detected candidate expressions into *functional/content* chunks. Among those baselines, "majority (= *functional*)" always assigns *functional* usage to the detected candidate expressions. "Hand-crafted rules" are manually created 145 rules each of which has conditions on morphemes constituting a compound functional expression as well as those at immediate left/right contexts. Performance of our SVMs-based chunkers is measured through 10-fold cross validation.

As shown in Table 7, our SVMs-based chunkers significantly outperform those baselines both in $F_{\beta=1}$ and classification accuracy[9]. We also evaluate the effectiveness of each feature set, i.e., the morpheme feature, the chunk candidate feature, and the chunk context feature. The results in the table show that the chunker with the chunk candidate feature performs almost best even without the chunk context feature[10].

[9]Recall of existing Japanese text processing tools is low, because those tools can process only 50∼60% of the whole 52 compound functional expressions, and for the remaining 40∼50% expressions, they fail in identifying all of the occurrences of functional usages.

[10]It is also worthwhile to note that training the SVMs-based chunker with the full set of features requires computational cost three times as much as training without the chunk

Table 7: Evaluation Results (%)

		Identifying *functional* chunks			Acc. of classifying *functional/content* chunks
		Prec.	Rec.	$F_{\beta=1}$	
Baselines	majority (= *functional*)	78.0	100	87.6	78.0
	Juman/KNP	89.2	49.3	63.5	55.8
	ChaSen/CaboCha	89.0	45.6	60.3	53.2
	hand-crafted rules	90.7	81.6	85.9	79.1
SVM (feature set)	morpheme	88.0	91.0	89.4	86.5
	morpheme + chunk-candidate	91.0	93.2	**92.1**	**89.0**
	morpheme + chunk-candidate/context	91.1	93.6	**92.3**	**89.2**

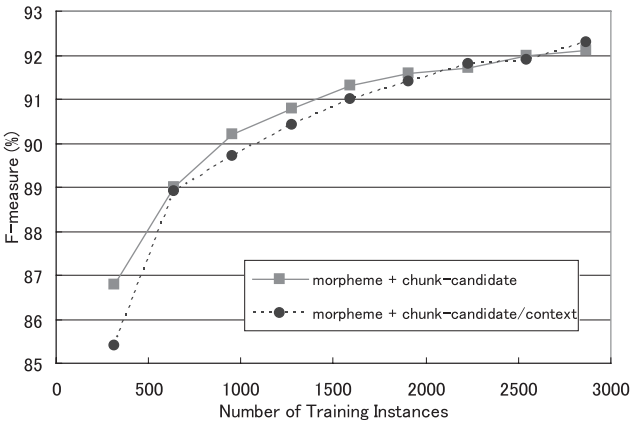

Figure 1: Change of $F_{\beta=1}$ with Different Number of Training Instances

For the SVMs-based chunker with the chunk candidate feature with/without the chunk context feature, Figure 1 plots the change of $F_{\beta=1}$ when training with different number of labeled chunks as training instances. With this result, the increase in $F_{\beta=1}$ seems to stop with the maximum number of training instances, which supports the claim that it is sufficient to collect and manually annotate about 50 training examples per expression.

5 Concluding Remarks

The Japanese language has various types of compound functional expressions, which are very important for recognizing the syntactic structures of Japanese sentences and for understanding their semantic contents. In this paper, we formalized the task of identifying Japanese compound functional expressions in a text as a chunking problem. We applied a machine learning technique to this task, where we employed that of Support Vector Machines (SVMs). We showed that the proposed method significantly outperforms existing Japanese text processing tools. The proposed framework has advantages over an approach based on manually created rules such as the one in (Shudo et al., 2004), in that it requires human cost to manually create and maintain those rules. On the other hand, in our framework based on the machine learning technique, it is sufficient to collect and manually annotate about 50 training examples per expression.

context feature.

References

J. Carletta. 1996. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22(2):249–254.

Group Jamashii, editor. 1998. *Nihongo Bunkei Jiten*. Kuroshio Publisher. (in Japanese).

T. Kudo and Y. Matsumoto. 2001. Chunking with support vector machines. In *Proc. 2nd NAACL*, pages 192–199.

J. Mayfield, P. McNamee, and C. Piatko. 2003. Named entity recognition using hundreds of thousands of features. In *Proc. 7th CoNLL*, pages 184–187.

Y. Morita and M. Matsuki. 1989. *Nihongo Hyougen Bunkei*, volume 5 of *NAFL Sensho*. ALC. (in Japanese).

National Language Research Institute. 2001. *Gendaigo Hukugouji Youreishu*. (in Japanese).

H. T. Ng, C. Y. Lim, and S. K. Foo. 1999. A case study on inter-annotator agreement for word sense disambiguation. In *Proc. ACL SIGLEX Workshop on Standardizing Lexical Resources*, pages 9–13.

I. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proc. 3rd CICLING*, pages 1–15.

K. Shudo, T. Tanabe, M. Takahashi, and K. Yoshimura. 2004. MWEs as non-propositional content indicators. In *Proc. 2nd ACL Workshop on Multiword Expressions: Integrating Processing*, pages 32–39.

E. Tjong Kim Sang. 1999. Representing text chunks. In *Proc. 9th EACL*, pages 173–179.

M. Tsuchiya, T. Utsuro, S. Matsuyoshi, S. Sato, and S. Nakagawa. 2005. A corpus for classifying usages of Japanese compound functional expressions. In *Proc. PACLING*, pages 345–350.

V. N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.