# 5-4 Fundamental Natural Language Processing Tools

KAZAMA Jun'ichi, WANG Yiou, and KAWADA Takuya

In this paper, we describe the fundamental natural language processing tools (evaluative expression analyzer, morphological analyzer, and syntactic parser) that we have developed and released through Advanced Language Information Forum (ALAGIN).

***Keywords***
Natural language processing, Evaluative expression, Morphological analysis, Syntactic analysis, Advanced Language Information Forum (ALAGIN)

## 1 Introduction

In order to acquire useful information and knowledge from documents written in natural languages and use them on various applications, the documents first need to be transformed into appropriate forms so that computers can (understand the contents and) handle them more easily. In this paper, transform processings whose usefulness has gained consensus to a certain degree will be called fundamental natural language processings. Typical among them are "morphological analysis" where sentences are segmented into words and each word is assigned a part of speech, and "dependency parsing" where dependency structures between constituent elements such as words and phrases are determined. Evaluative information analysis which we are going to introduce in this paper is also becoming popular as a fundamental natural language processing. It is an analytical processing to judge whether a given expression denotes a positive or negative opinion. We have been developing several systems for fundamental natural language processing. To return what we have gained to the society, we make those systems available to the public through ALAGIN. One of them is our evaluative information analysis system that will be presented in Section **2**. Evaluative information analysis technology was used for the information analysis system WISDOM (http://wisdom-nict.jp/), and we have organized related technologies and dictionaries to make them available to the public. In Sections **3** and **4**, our morphological analyzer and dependency parser are presented. Morphological analysis and dependency parsing are relatively old fields of study. Japanese analyzers have been widely used and proved to be precise enough, but those for such languages as Chinese do not have enough precision since those languages have not been studied long enough despite the fact that many researchers are now actively engaging in their study. To cope with increasing demands for processing those languages, we have conducted researches on multi languages focusing on Chinese and developed some systems with the world's highest level precision. We will describe our Chinese morphological analyzer and dependency parser in Sections **3** and **4**.

## 2 Evaluative information analysis system

Evaluative information analysis that can mine people's evaluations and opinions from texts has been drawing more attention. In evaluative information analysis, a given sentence is judged whether it represents an evaluation or opinion about a certain target and if judged so, it is automatically judged whether it is a positive or negative opinion. In the background of their prominence is advancement and expansion of information media including the Web. Many people are now able to publicly express their opinions about various things through the Web. On the other hand, their ever-increasing evaluations and opinions have kept accumulating and technologies to efficiently extract and organize them are being awaited. To cope with this problem, we have been developing evaluative information analysis systems that can automatically extract and organize positive and negative evaluations and opinions. We will describe these systems in the following sections.

### 2.1 Evaluative information

People express their evaluations and opinions in various ways. In this paper, evaluative information is defined as a unit of information which represents a positive (or negative) judgment or attitude toward a certain target. More specifically, it is a unit of information which basically consists of “a person or organization who asserts the opinion expression (evaluation holder)”, “a target of evaluation (evaluation target)”, “linguistically expressed judgment or attitude (evaluative expression)”, “an evaluation type” and “an evaluation polarity”. Example 1 is interpreted as a sentence describing “Taro”’s positive emotion toward “Aomori apples”. The word “loves” is extracted as the “evaluative expression” since it linguistically expresses evaluation. “Taro” is the one who evaluates and therefore is extracted as the evaluation holder and “Aomori apples” is what Taro evaluates, therefore it is extracted as the evaluation target. In the following part of this section, evaluation targets will be underlined and evaluation holders will be written in bold. In many cases, the evaluation holder and the author are identical and many of such evaluation holders are not explicitly written. If a phrase or word to denote an evaluation holder appears in a sentence, it will be written in italic.

Example 1: *Taro* **loves**
Evaluation holder Evaluation expression
Aomori apples.
Evaluation target (emotion +)

In actual texts, evaluations are expressed in various ways. Some are emotional and others are based on one’s experience. We have classified them into the following types according to certain criteria such as subjectivity and their evaluation polarity (+ and – represent positive and negative polarities respectively).

(1) Emotion+ / Emotion– : Subjective and emotional
Ex. 2: I **Love** Kyoto. (Emotion+)
Ex. 3: Taro **is not interested in** the product A. (Emotion–)

(2) Comment+ / Comment – : Subjective and expressing a certain attitude such as approval/disapproval and praise/criticism
Ex. 4: Kyoto **is beautiful**. (Comment +)
Ex. 5: The system A **has too many problems.** (Comment –)

(3) Merit+ / Merit– : Expressing merits and demerits
Ex. 6: These coupons **can be used anytime**. (Merit+)
Ex. 7: The product A **is hard to handle**. (Merit–)

(4) Adoption+ / Adoption– : Positively adopting or promoting something
Ex. 8: *Company A* **has decided to adopt** electric money. (Adoption+)
Ex. 9: The product A **is unpopular**. (Adoption–)

(5) Event+ / Event– : Expressing a good or bad event or experience
Ex. 10: The product A **was awarded the Good Design Award**. (Event+)
Ex. 11: The product B **broke down on the third day after purchase**. (Event–)

(6) Deontic: Expressing an obligation, proposal, advice or countermeasure
Ex. 12: Electric money **should be adopted**. (Deontic)
Ex. 13: The citizen judge system **should gain national consensus to be adopted**. (Deontic)

(7) Request: Expressing request or hope

Ex. 14: (I) **I hope that** electric money is available here. (Request)

For proposals or requests (6 and 7), no evaluation polarity will be indicated since they do not always explicitly show their positive (or negative) attitude toward a certain target (e.g. "The citizen judge system" in Example 13).

## 2.2 Evaluative information corpus

To extract a wide variety of evaluative information has been considered a difficult task. To cope with this problem, we have constructed an evaluative corpus [1]. We selected 100 topics such as "electric cars" and "pension system issues" and for each topic, collected 200 sentences from Web documents, making the total number of sentences in the corpus 20,000. Each sentence is annotated with evaluative information presented in Subsection **2.1** and its relevancy to the topic. For example, the sentence "there is an interesting study of the citizen jury system in an article of this website" selected for the topic "citizen jury system" does not evaluate "the citizen jury system" itself. Rather, the sentence evaluates the website. Such information, or information that evaluates not the topic but something else, is indicated that it is irrelevant to the topic. The corpus can be used as a training data set for machine learning or a test data set for benchmark tests.

## 2.3 Evaluative Expression Dictionary

Evaluative Expression Dictionary consists of sets of evaluative expressions and their evaluation polarity (e.g. "well-regulated +" and "sugary –"). The dictionary is used as basic knowledge for evaluative information analysis. The dictionary was constructed by following the procedure below. A small set of evaluative expressions annotated with evaluation polarity was first prepared for being used as seed expressions. Expressions that are contextually similar to the seed expressions were extracted as candidate evaluative expressions by using the Database of Similar Context Terms [2] and Support Tool for Customized Word Set Generation [3] (both for generating sets of words of similar meaning) based on the assumption that such expressions are highly possibly evaluative expressions. The candidate evaluative expressions were then manually judged whether they had an evaluation polarity or not. Candidates judged to have a polarity were listed in the dictionary as evaluative expressions along with their polarity. The newly added evaluative expressions were then used as a new set of seed expressions to create another set of evaluative expression entries, and the procedure was repeated in a bootstrapping manner to increase the number of evaluative expression entries in the dictionary. Moreover, entries in List of Burden and Trouble Expressions [4] were also listed in the dictionary as evaluative expressions with a negative polarity. The total number of evaluative expressions in the dictionary amounted to 36,981. The dictionary is available to the public as a model data for "opinion extraction tools" through ALAGIN.

## 2.4 Extraction of evaluative information

### *2.4.1 Procedure for evaluative information extraction*

Figure 1 is a flowchart of evaluative expression extraction performed by the evaluative expression analysis system. First, the user inputs raw sentences. Then the system extracts the evaluative expressions form the input sentences (1), identifies the evaluation holder (2), determines the evaluation type (3) and evaluation polarity (4), and finally, outputs the results. The following section describes each step of the procedure.

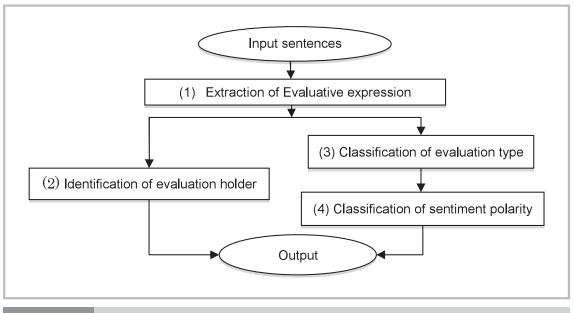

Input sentences
(1) Extraction of Evaluative expression
(3) Classification of evaluation type
(2) Identification of evaluation holder
(4) Classification of sentiment polarity
Output

**Fig.1** *Flow of the evaluative information analysis*

#### 2.4.2 Extraction of evaluative expressions

Evaluative expressions are extracted based on sequence labeling utilizing conditional random fields (CRFs). In this method, each morpheme is attached one of three types of tags according to its position in the constituting chunk: "B" for beginning morphemes, "I" for inside morphemes, and "O" for outside morphemes [5]. The method has been widely used for extracting such information as named entities. Sequence labeling is considered appropriate here since evaluative expressions can appear at any position in a sentence. Words that are frequently used for expressing evaluation are very useful for evaluative expression extraction. We used the above mentioned Evaluative Expression Dictionary. For CRF features, we used the following information of the current morpheme and two preceding and succeeding morphemes: the surface form, original form, coarse-grained POS tag, fine-grained POS tag and polarity in the evaluative polarity dictionary.

#### 2.4.3 Identification of evaluation holder

Evaluation holders are identified in two steps. First, a given evaluative expression is judged whether its evaluation holder is identical to the author of the expression by using SVMs (support vector machines). The surface form, original form, coarse-grained POS tag and fine-grained POS tag of the morpheme in the evaluative expression are used as features. If the holder is not the author, the word(s) to denote the evaluative holder is extracted from the evaluative expression by using CRFs. For CRF features, each morpheme's surface form, original form, coarse-grained POS tag, fine-grained POS tag and positional relationship to the evaluative expression are used.

#### 2.4.4 Classification of evaluation types

Each of the given evaluative expressions is classified into one of the seven evaluation types described in Subsection **2.1** by using an SVM modified for multi-value classification by the pairwise method. The surface form, original form, coarse-grained and fine-grained POS tags and their combination of each morpheme in the evaluative expression are used as SVM features.

#### 2.4.5 Classification of evaluation polarity

Automatic polarity classification has been studied by many researchers [6][7]. One of the most typical ways of approaching the classification is supervised machine learning using bag-of-words features. The method determines the polarity of an evaluative expression by treating the expression as a set of individual words contained in the expression. However, the method does not work well when an evaluation polarity is reversed, which is actually a frequent case. For example, an evaluative expression "kill cancer cells" has a negative-meaning component "cancer cells", but that negativity is denied by the word "kill" and therefore, the negative polarity based on "cancer cells" is reversed and the expression is judged to be positive as a whole. Thus, the positive (or negative) evaluative polarity of a word in an evaluative expression does not always mean the whole expression also has a positive (or negative) polarity. Therefore, we have not to treat them as independent elements but to consider the impact of interaction between words. Based on this idea, we use "CRFs with hidden variables" for our classification of evaluation polarity to take the impact of interaction between words into consideration [8]. In this method, the dependency structure of an evaluation expression is first analyzed and the evaluation polarity of each dependency subtree is represented by a hidden variable. The final classification of evaluation polarity is performed based on the interaction between the hidden variables.

As an example, the evaluative expression "have effects of reducing anxiety and tension" has the negative polarity words "anxiety" and "tension", but when those words depend on the word "reducing", their polarities are reversed, which leads to a possible conclusion that the subtree "reducing anxiety and tension" has a positive polarity. The subtrees "effects of reducing anxiety and tension" and "have effects of reducing anxiety and tension" also have a positive polarity. This means that every subtree in an evaluative expression has its own evaluation polarity.

We use a probabilistic model illustrated by the graph in Fig. 2. In this model, each word in an evaluative expression is considered to have a random variable as illustrated in Fig. 2 with oval nodes. The evaluation polarity of a subtree is indicated by a random variable given to the root of the subtree. A random variable is affected by not only the word itself but also by the random variables of syntactically related words. The model offers the information that a phrase (bunsetsu) that contains a positive (or negative) word tends to have a positive (or negative) polarity and two phrases (bunsetsu) with head-dependent relation tend to have opposite polarities, if the head contains a word that can reverse the polarity. A higher classification precision was achieved by using this method compared to the one that treated an evaluative expression as a simple set of independent words [8].

### 2.5 Performance evaluation

The performance of the evaluative information analysis system was measured by using the evaluative information corpus described in Subsection **2.2**. We randomly divided the corpus into 10 equal sized data sets and performed 10-fold cross validation. Each module was independently used and evaluated. The recall (the number of correctly extracted evaluative expressions divided by the number of evaluative expressions in the correct data set), precision (the number of correctly extracted evaluative expressions divided by the total number of extracted evaluative expressions) and F-measure (harmonic mean of recall and precision) were used for evaluating the system. An evaluative expression extracted by the system and an evaluative expression in the correct data set are considered a matched pair if their headwords (a word to represent the principal meaning of an element, or a morpheme at the end of an element in most Japanese phrases) match. Evaluation holder identification and evaluation type classification performances were measured by their accuracies (the number of correct outputs divided by the total number of evaluative expressions in the test set). Table 1 shows the results of performance evaluation of the evaluative information analysis system.

The inter-annotation agreement between two human annotators is presented in Table 2 to show the difficulty of evaluative expression extraction. For constructing a manually annotated evaluative information corpus, each sentence was annotated by two different annotators to ensure the quality of the corpus. The annotation results generated by one annotator were considered correct. The results generated by the other were then compared with the correct results. Table 2 shows the recall, precision and F-measure for the latter annotator's results. The results show that to achieve a high inter-annotation agreement in evaluative expression extraction is very hard, and considering this fact, the performance of the system

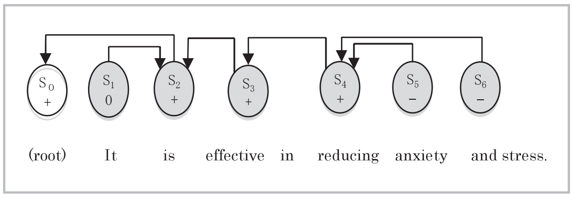


**Fig.2** *Example of head-dependent tree for sentiment polarity*

**Table 1** *The performance of evaluative information analysis system*

| | | |
|---|---|---|
| Evaluative expression extraction | Recall | 0.4077 |
| Evaluative expression extraction | Precision | 0.6020 |
| Evaluative expression extraction | F-measure | 0.4860 |
| Evaluation holder identification | Accuracy | 0.6919 |
| Evaluation type determination | Accuracy | 0.6515 |
| Sentiment polarity determination | Accuracy | 0.8703 |

**Table 2** *The annotation agreement on evaluative expression*

| | |
|---|---|
| Recall | 0.67 |
| Precision | 0.71 |
| F-measure | 0.69 |

shown in Table 1 is not very low. The system achieved a high accuracy of 0.87 in evaluation polarity classification by using CRFs with hidden variables described in Subsection ***2.4.5*** and the dictionary described in Subsection **2.3**.

### 2.6 Distribution through ALAGIN

The system is an open source software and available on the ALAGIN website (http://alaginrc.nict.go.jp/opinion/index.html). ALAGIN also provides a database containing the model parameters (a set of words and numbers to control the program's behavior) for the evaluative information analysis system. The database contains four model files "evaluative expression extraction", "evaluation holder identification", "evaluation type classification" and "evaluation polarity determination" for different processing flows.

## 3 High-precision Chinese morphological analyzer

This section presents a method to improve the precision of Chinese morphological analysis based on semi-supervised learning using large scale unlabeled data. More specifically, N-grams obtained by automatic analysis of large scale unlabeled data using a baseline model, cluster information obtained by word clustering, and lexicographical information obtained through cross validation are used as additional features. In an experiment using Penn Chinese Treebank, a standard evaluation data, our proposed method achieved a higher analysis precision than the baseline and other existing methods that do not adopt semi-supervised learning.

Like Japanese, Chinese does not have a boundary between words. Therefore, morphological analysis is the most basic and important task for processing Chinese. The technique requires high precision because it is used in the preprocessing phase of many tasks including dependency parsers and information retrieval systems. In recent years, various studies on Chinese morphological analysis have been conducted. Studies on joint learning of word segmentation and POS tagging are especially actively pursued these days [9]-[13]. For example, we have achieved the world's highest level analysis precision by using a word-character hybrid model [11].

A machine learning method called "semi-supervised learning" which uses a huge amount of data without any correct labeling is now becoming popular. Previous studies have reported that semi-supervised learning had improved the performance of certain natural language processing tasks, e.g. text chunking [14], POS tagging and named entity extraction [15], and dependency parsing [16]-[18]. However, few studies have been reported to have used semi-supervised learning for Chinese morphological analysis. Mochihashi et al. [19] succeeded in improving the precision of Chinese word segmentation by using the semi-supervised learning method, but it was a very small improvement since the unlabeled data they used was not large enough.

In this paper, we propose a method to improve the precisions of Chinese word segmentation and POS tagging by using large scale unlabeled data on a pipeline system which is more easily implementable than the joint learning technique.

### 3.1 System overview

We use a more easily implementable two-step pipeline system partly to cut down the development cost. For word segmentation, a character-based CRF is used and for POS tagging, a word-based CRF is used. For implementing CRFs, an open source toolkit, CRF++ (version 0.54)[*1] is used. The features for the baseline word segmentation model are the current character and one preceding and succeeding characters, indication of not being a character and the character type. Each character in each word is attached the following tags: "S" for single character words, "B" for the beginning characters, "B2" for the second characters, "B3" for the third characters, "M" for

[*1] http://crfpp.sourceforge.net/

other inside characters, and "E" for the ending characters. The features for the baseline POS tagging model are the current word and two preceding and succeeding words, beginning and ending characters of a word and the length of a word.

To realize high-precision morpheme analysis system, we propose a new approach: introduction of new features, i.e. information obtained from unlabeled data. This approach takes the following steps. First, large scale unlabeled data is auto-analyzed by using the baseline model to extract various types of lexicographical information which then will be used for the generation of new word-segmentation and POS-tagging features. The words in the segmented data are clustered to obtain cluster information which will be used as a POS tagging feature. Additionally, lexicographical information obtained from labeled data through cross validation will be added to the list of new features. Figure 3 illustrates the flow of our approach. In the following sections, our new features will be presented.

## 3.2 New features for word segmentation

### 3.2.1 Semi-supervised N-gram features

First, we preprocess unlabeled data using the baseline word segmentation model and obtain auto-segmented data. We then extract character N-gram lists from auto-segmented sentences. Finally, we generate N-gram features for word segmentation.

Each character $c_i$ is assigned a tag $t_i$ by using the baseline word segmentation model. When the number of characters in a word is $L$, an auto-segmentation result is expressed by the sequence $\{(c_i,t_i)\}_{i=1}^{L}$. An N-gram list $\{(g, seg, f(g, seg))\}$ is then extracted from the auto-segmentation results. "*g*" denotes a character-level N-gram (e.g. unigram $c_i$, bi-gram $c_ic_{i+1}$ and tri-gram $c_{i-1}c_ic_{i+1}$) and "*seg*" denotes the segmentation profile of "*g*". A segmentation profile consists of a tag $t_i$ or a combination of tags (e.g. $t_i$ or $t_it_{i+1}$ for bi-gram $c_ic_{i+1}$). $f(g, seg)$ denotes the frequency obtained when the segmentation profile of an N-gram *g* is *seg*.

The obtained lists are then divided into three sets according to their frequencies: high frequency (HF, top 5%), medium frequency (MF, next 15%) and low frequency (LF, bottom 80%). Then, the lists $L_{ng}=\{(g, seg, FL(g, seg))\}$ will be obtained. $FL(g, seg)$ denotes a frequency label obtained by the procedure above.

We attempted to encode the information of the above N-gram list into a new type of features. We tried several feature representations and generation methods and found that the feature derived from the bi-gram list with $seg = t_i$ was most effective. By using those lists, the feature for a given character $c_0$ is generated as below: from $L_{ng}$, obtain a subset ($L_m$) where N-grams *g* match the bi-gram $c_0c_1$, and generate features defined as below for each entry in $L_m$:

(a) ***seg-FL*(*g, seg*)**

Then, the features of each entry in $L_m$ are concatenated into one N-gram feature.

For example, the N-gram feature for $c_0$ in "幸/福" where $L_m$ is {(幸/福, *B*, *HF*), (幸/福,

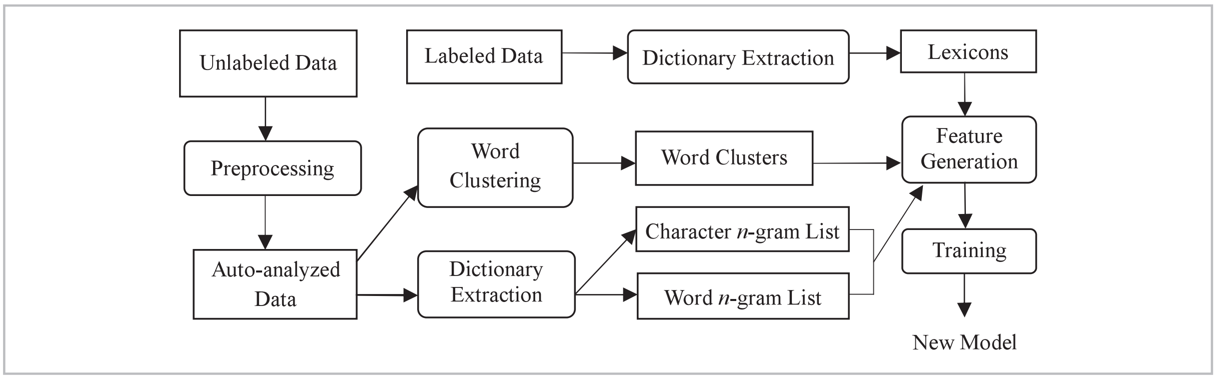


Fig.3 Overview of the proposed approach

$B_2$, *MF*), (幸/福, *E*, *LF*)} is "*B-HF*|$B_2$-*MF*|*E-LF*"

### 3.2.2 Lexicon features

Character-based word segmentation models show a higher precision in analyzing unknown words, while they are known for their inferiority in analyzing known words. It has been generally said that the precision for analyzing known words can be improved by introducing dictionaries. A dictionary of known words can be easily constructed by extracting words from a labeled training data set, and we used such resources for our research by introducing features obtained from dictionaries. We call the features "lexicon features".

A dictionary is compiled by collecting words and all corresponding POS tags from a training data set. For example, the word "交流 (exchange)" is listed as (交流, *NN-VV*) in the training data set, and "*NN-VV*" is the result of concatenating all POS tags assigned to "交流" in the data set.

However, when a system is trained with features generated from a training-data-extracted dictionary, there is a possibility of over-fitting to the training data, i.e. the system's overtrust in lexicon features. To cope with this problem, we adopt the cross validation technique for constructing our dictionary as below:

○ Divide the training data into 10 equal sized data sets.

○ Construct a dictionary per set by using the remaining nine sets and generate lexicon features from those dictionaries.

○ For the test data set, construct dictionaries by using the whole training data and generate lexicon features from those dictionaries.

Words for generating features are selected by conducting left-most longest prefix matching with the dictionary. A feature defined as below is then added to each character $c_k$ in each word *w*:

(b) $\boldsymbol{P(c_k)/LEN(w)\text{-}POSs(w)}$

$LEN(w)$ denotes the length of a word *w*, $P(c_k)$ denotes the position of a character $c_k$ in the word *w*, and $POSs(w)$ denotes the combination of POS tags assigned to the word *w* in a dictionary. For example, if a character string $c_0c_1$ "幸/福" matches a dictionary entry "幸福, *JJ-NN-VA*", the lexicon feature of the $c_0$ "幸" and that of the $c_1$ "福" are "*1/2-JJ-NN-VA*" and "*2/2-JJ-NN-VA*" respectively.

## 3.3 New features for POS tagging

### 3.3.1 Semi-supervised N-gram features

Word-level N-gram list $L_{wg} = \{(w, pos, FL(w, pos))\}$ can be obtained by analyzing automatically segmented unlabeled data by using a POS tagging model. *w* is a word-level N-gram and *pos* is the POS information of the word-level N-gram. N-gram features for POS tagging will be generated by using the N-gram lists. The results of a preliminary experiment showed that the maximum effect can be obtained when *w* is a unigram and *pos* is the POS of *w*. We extracted a subset of $L_{wg}$. where *w* matches the given current word $w_0$ and represent it by $L_s$. For example, when $w_0$ is "研究 (research)", the matching entries are (研究, *VV*, *HF*), (研究, *VA*, *LF*) and (研究, *CD*, *LF*). As the result of error analysis, POS tagging errors were found to occur frequently. Therefore, the following limitations have been applied to the acquisition of subsets $L_s$. $N(X)$ denotes the number of entries when $FL(w, pos) = X$ holds.

i. When $N(HF)$ is equal to or larger than 2, $L_s$ should consist of matching entries with $FL(w, pos) = HF$.

ii. When $N(HF)$ is smaller than 2 and $N(HF)+N(MF)$ is equal to or larger than 2, $L_s$ should consist of matching entries with $FL(w, pos) = HF$ or $FL(w, pos) = MF$.

iii. When $N(HF)+N(MF)$ is smaller than 2, all entries become matching entries.

For example, the $L_s$ of the example "研究" is {(研究, *NN*, *HF*), (研究, *VV*, *HF*)}. Like word segmentation, a feature generated for each entry in $L_s$ is defined as below:

(c) $\boldsymbol{pos\text{-}FL(w, pos)}$

Then, the features of each entry in $L_s$ are concatenated into one N-gram feature. For example, when $w_0$ is "研究", the N-gram feature of $w_0$ is "*NN-HF*|*VV-HF*".

### 3.3.2 Semi-supervised cluster features

For generating cluster features, word clustering is conducted by using the automatically analyzed data, and based on the method proposed by Koo et al. [18], cluster features of various granularities are acquired by using the prefix of cluster hierarchy generated by the Brown clustering algorithm [20]. As the result of a preliminary experiment, we have decided to use the following cluster features:

(d) All bits in the hierarchical bit representation of $w_{-1}$, $w_0$ and $w_1$

The first 6 bits in the hierarchical bit representation of $w_{-1}$, $w_0$ and $w_1$

In the preliminary experiment, we achieved the highest precision when we used the above cluster features in Bigram template.

### *3.3.3 Lexicon features*

Lexicon features are added by using the same dictionary as the one used for word segmentation. A feature defined as below is assigned to a given word $w_0$.

(e) ***POSs*(*w₀*)**

*POSs*($w_0$) is a set of concatenated POS tags of a word $w_0$ in the dictionary.

## 3.4 Experiment

### *3.4.1 Data sets*

(1) Labeled Data

Penn Chinese Treebank data sets were used for our experiment. More specifically, we used CTB5 (LDC2005T01), CTB6 (LDC2007T36) and CTB7 (LDC2010T07). As shown in Table 3, each corpus was divided into three sets: a training data set, a development data set and a test data set. Many of the existing studies have used CTB5. The credibility of the performance evaluation will be enlarged by adding CTB6 or CTB7 since their development and test sets are larger than those of CTB5.

**Table 3** *The statistics of the corpora*

| | Sentence number of training set | Sentence number of development set | Sentence number of test set |
|---|---|---|---|
| CTB5 | 18,089 | 350 | 348 |
| CTB6 | 23,420 | 2,079 | 2,796 |
| CTB7 | 31,131 | 10,136 | 10,180 |

(2) Unlabeled Data

204 million words from the XIN_CMN portion of Chinese Gigaword Version 2.0 (LDC2009T14) were used for the unlabeled data set. We excluded the portions that were possibly contained in CTBs. A million words in the data set were used for word clustering.

### *3.4.2 Results*

We conducted experiments on Chinese word segmentation (Seg) and POS tagging (Seg & Tag) to evaluate the effectiveness of the proposed method. F-measures were used for evaluation. Table 4 shows the results from previous studies and our experiments both using CTB5. All the results from the previous studies were quoted from their research papers. As seen in the results in the table, we have achieved the highest performance in both Seg and Seg & Tag.

Moreover, we conducted a comparative experiment among our proposed method and the methods proposed by Kruengkrai et al. [10] and Kruengkrai et al. [11] using CTB6 and CTB7. The results are shown in Table 5. You can see that our proposed method has achieved

**Table 4** *Comparison with previous studies (CTB5)*

| Method | Seg | Seg & Tag |
|---|---|---|
| Proposed | **0.9812** | **0.9420** |
| Baseline | 0.9753 | 0.9318 |
| Zhang et al. [1] | 0.9778 | 0.9367 |
| Kruengkrai et al. [2] | 0.9787 | 0.9367 |
| Kruengkrai et al. [3] | 0.9798 | 0.9400 |
| Jiang et al. [4] | 0.9785 | 0.9341 |
| Nakagawa et al. [5] | 0.9796 | 0.9338 |

**Table 5** *Comparison with previous studies (CTB6とCTB7)*

| Methods | CTB6 | | CTB7 | |
|---|---|---|---|---|
| | Seg | Seg & Tag | Seg | Seg & Tag |
| Proposed | **0.9579** | **0.9113** | **0.9566** | **0.9051** |
| Baseline | 0.9513 | 0.8999 | 0.9498 | 0.8937 |
| Kruengkrai et al. [2] | 0.9550 | 0.9050 | 0.9540 | 0.8986 |
| Kruengkrai et al. [3] | 0.9551 | 0.9053 | 0.9546 | 0.8990 |

the highest performance even with larger scale data sets.

### 3.5 Distribution of the system

The system incorporating the proposed technique will be released as an open source software under the name of CSP (Chinese Word Segmenter and POS Tagger) through the ALAGIN language resource website (http://alaginrc.nict.go.jp/csp/index.html). ALAGIN also plans to provide a database containing the model parameters (a set of words and numbers to control program’s behavior) for CSP. The database will contain models trained with CTB5, CTB6 and CTB7 and corresponding N-gram lists, information about clustering and other related resources.

## 4 High-precision Chinese dependency parsing

Morphological analysis is usually followed by a processing called syntactic analysis to determine sentence structures. A type of syntactic analysis that has been especially actively studied in recent years is dependency parsing where the relations (dependency) between words such as the relations between a verb and the subject or the object are determined. This section presents our high-precision dependency parser trained by semi-supervised learning [21][22]. The system has ranked among the highest level Chinese parsers.

Figure 4 shows the flow of morphological analysis of a Chinese sentence “布朗一行于今晚离沪赴广州 / Brown and his party will leave Shanghai for Guangzhou tonight” followed by dependency parsing of the same sentence. Dependency relations are represented by arrows and expressed by using the word “depend (on)” as in “the word positioned at the rear end of an arrow ’depends on’ the word at the head of the arrow.” Hereafter, we will call such arrows arcs. Arcs are sometimes assigned labels to show certain relations (e.g. “subj” to denote the subject and “obj” to denote the object). “ROOT” is a provisional word to indicate the position of the head (main) verb. The whole relationship is represented by a tree and the ROOT as its root. No arcs in a Chinese dependency tree should cross each other when each word is positioned on a row by order of their appearance as in the figure. Japanese trees have additional restriction that arcs should always proceed from left to right. In fact, Japanese and Chinese trees both have a few exceptional cases where arcs have to cross each other, but in many cases, those exceptions are assumed not to happen for the sake of efficiency[*2].

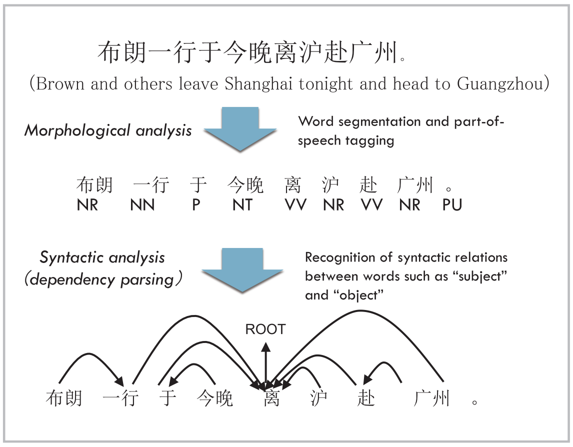

布朗一行于今晚离沪赴广州。
(Brown and others leave Shanghai tonight and head to Guangzhou)
Morphological analysis
Word segmentation and part-of-speech tagging
布朗 一行 于 今晚 离 沪 赴 广州 。
NR NN P NT VV NR VV NR PU
Syntactic analysis (dependency parsing)
Recognition of syntactic relations between words such as “subject” and “object”
ROOT
布朗 一行 于 今晚 离 沪 赴 广州 。

Fig.4 The flow of Chinese dependency parsing

Various parsing techniques have been proposed and in recent years, graph-based parsing has been widely used because of its high precision [23][24]. The graph-based parsing model sees each word in a sentence as a node and draws a graph where bidirectional arcs link nodes. Among the spanning trees (tree-structured subgraphs containing all nodes) in the graph, it tries to find the non-crossing (if specified so) tree with the maximum weight. The method is called MST parsing and the tree with the maximum weight is called the maximum spanning tree. There are several ways to assign weights to arcs including the first-order model where a single arc is assigned a weight [23] and the second-order [24] model where

[*2] Arcs in some languages like Czech often have to cross each other. Non-projective parsing models that allow crossing are used in such cases.

two arcs are assigned a weight. The weight of a spanning tree is represented by the sum of all weights in the spanning tree. The first-order and second-order models are most frequently used since a higher order of the model (the number of arcs involved in the score assignment) makes the cost of parsing larger. We used the first-order [23] and second-order [24] models, too. Each weight is broken down to various feature functions such as words and combinations of words. In the first-order model, it is defined as below:

$$w(x,y)=\sum_{(i,j)\in y} w(i,j)=\sum_{(i,j)\in y} \alpha\cdot f(x,i,j)$$

“ $x$ ” denotes an input word sequence and “ $y$ ” denotes a spanning tree. “$(i,j)$” denotes an arc from the $i$ ’th word to the $j$ ’th word. “ $f(x,i,j)$ $i$” is the feature vector to represent various characters and “ $\alpha$ ” is the weight vector to indicate the weight of each feature. A weight vector “ $\alpha$ ” is automatically obtained by machine learning from a manually annotated correct data set.

### 4.1 Application of subtree features

The proposed system uses the method that incorporates semi-supervised learning in order to improve analysis precision. Semi-supervised learning is a method to improve systems’ precision by using a large amount of raw sentences (raw corpora). The system uses a first-order MST parser (the baseline model) trained with a correct data set to parse a large amount of sentences, and extracts first-order and second-order subtrees. The extracted subtrees are then classified according to their frequencies and assigned one of the following labels: HF (high frequency, top 10%), MF (medium frequency, next 10%), LF (low frequency, bottom 80%) and ZERO (zero, no appearance). The labels assigned here are used as features for parsing (for details, see the reference [21]). The baseline model results cannot be always correct, but intuitively, we believe that we can get certain tendencies such as combinations of words that tend to have a dependency relation and those that hardly have a dependency relation if we statistically analyze the baseline model results since it contains relatively easily parsable sentences as well. Information obtained this way may be helpful in training the system with the correct data set.

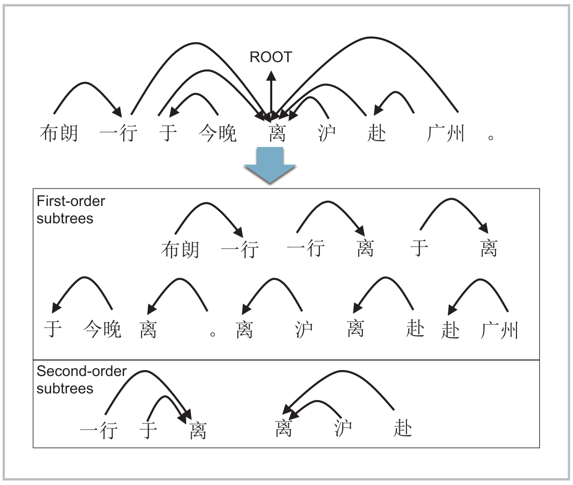


**Fig.5** *Extractions of subtrees*

Figure 5 illustrates extraction of subtrees from the analysis results. Since the second-order model [23] proposed in the reference [21] limits arcs to two adjacent ones, second-order subtrees extracted there are also limited that way. On the other hand, the method proposed in the reference [22] uses a higher-level second-order model [25] to use second-order subtrees in the form of “parent-child-grandchild”.

### 4.2 Experiment

We evaluated the proposed system by using English and Chinese data. The results shown here are based on those presented in the reference [22]. The Penn Treebank data set, a standard training and validation data set, and Chinese Penn Treebank (Version 4.0) which is also a standard training and validation data set were used as the English and the Chinese data sets respectively. As the raw corpora, 43 million word BLLIP Corpus and 311 million word Chinese Gigaword Version 2.0 were used for English and Chinese respectively. We measured the system quality by the percentage of correctly identified dependee(s) of each word excluding full stops (UAS: Unlabeled Attachment Score) and the percentage of sen-

**Table 6** *Experimental results (English)*

| | UAS | Complete |
|---|---|---|
| 1st-order | 90.95 | 37.45 |
| 1st-order+subtree | 91.76 | 40.68 |
| 2nd-order | 91.92 | 44.28 |
| 2nd-order+subtree | 92.89 | 47.97 |
| 2nd-order+subree +clustering+integration | 93.55 | 49.95 |
| KOO08-dep2c [6] | 93.16 | N/A |
| Carreras2008 [8] | 93.5 | N/A |
| Suzuki2009 [9] | 93.79 | N/A |

**Table 7** *Experimental results (Chinese)*

| | UAS | Complete |
|---|---|---|
| 1st-order | 86.38 | 40.80 |
| 1st-order+subtree | 88.11 | 43.10 |
| 2nd-order | 88.59 | 48.85 |
| 2nd-order+subtree | 91.77 | 54.31 |
| 2nd-order+subtree +integration | 91.93 | 55.45 |
| Yu2008 [10] | 87.26 | N/A |
| Zhao2009 [11] | 87.0 | N/A |

tences where all dependency relations identified by the system match the results given by the correct data set (Complete). Tables 6 and 7 show the results of the English and Chinese experiments respectively. You can see that subtree features had greatly improved the precision in both English and Chinese cases. Moreover, both systems can be further improved by combining the proposed features with cluster features [26] or integrated features obtained from other parsers' results [27]. In a comparative analysis with previous studies available in English, our system has ranked among the highest level systems. Suzuki 2009 applies the basic idea of semi-supervised learning, but it requires more complex implementation than ours. As for Chinese, our system has largely surpassed the performance of the best reported systems and as far as we know, it is now the world's best Chinese parser[*3].

### 4.3 Distribution through ALAGIN

The Chinese parser incorporating the proposed technique is available as an open source software under the name of CNP (A ChiNese dependency Parser) through the ALAGIN language resource website (http://alaginrc.nict.go.jp/cnp/index.html). ALAGIN also provides a database containing the model parameters for processing Chinese documents.

## 5 Conclusion

We have presented the fundamental natural language processing tools (the evaluative information analysis system, the morphological analyzer and the dependency parser) that have been developed by Information Analysis Laboratory and are available to the public through ALAGIN. In Section **2**, the evaluative expression analysis system incorporating such techniques as evaluative expression extraction, classification of evaluative expression types, identification of evaluation holders and evaluation polarity classification has been described. The performance of the system was evaluated based on the experimental results using the evaluative expression corpus. The future tasks for the system are to improve its performance by enriching the features or expanding the dictionary and corpus, and to expand the range of languages to cover. In Section **3**, the easily implementable but effective semi-supervised learning method for Chinese word segmentation on a pipeline system and Chinese POS tagging has been presented. The proposed method improves analysis precision by obtaining morphological information from large scale unlabeled data partly utilizing labeled data as well. Experimental results showed that the proposed method could achieve higher precisions than the baseline or known methods. In Section **4**, the semi-supervised learning technique for dependency parsing that utilizes subtrees extracted from the results of large scale raw corpus analysis using a baseline model has been proposed. With the proposed

[*3] As of the time of the publication and review of the referred papers.

method, we have achieved the world's highest precision for Chinese parsing. All these fundamental natural language processing tools and CSP that incorporates the technique presented in Subsection **3.5** are now being widely used in various researches and projects not only by our laboratory but by external institutions. Now, we are aiming at further improvement of the tools and development of additional fundamental processing technologies.

## *References*

1 Takuya Kawada, Tetsuji Nakagawa, Ritsuko Morii, Hisashi Miyamori, Susumu Akamine, Kentaro Inui, Sadao Kurohashi, and Yutaka Kidawara, "Construction of Evaluative Information Corpus on the Web," In Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing 2008. (in Japanese)

2 http://alaginrc.nict.go.jp/resources/nictmastar/resource-info/abstract.html#A-1

3 http://alaginrc.nict.go.jp/resources/nictmastar/resource-info/abstract.html#D-1

4 http://alaginrc.nict.go.jp/resources/nictmastar/resource-info/abstract.html#A-3

5 Eric Breck, Yejin Choi, and Claire Cardie, "Identifying expressions of opinion in context," Proceedings-IJCAI-2007, 2007.

6 Bo Pang and Lillian Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, Vol. 2, No. 1-2, pp. 1–135, 2008.

7 Takashi Inui and Manabu Okumura, "A Survey of Sentiment Analysis," Journal of natural language processing 13(3), 201-241, 2006. (in Japanese)

8 Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi, "Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables," In Proceedings of HLT-NAACL 2010, 2010.

9 Yue Zhang and Stephen Clark, "A Fast Decoder for Joint Word Segmentation and POS Tagging Using a Single Discriminative Model," In Proceedings of EMNLP-2010, 2010.

10 Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara, "An Error-Driven Word-Character Hybird Model for Joint Chinese Word Segmentation and POS Tagging," In Proceedings of ACL-IJCNLP-2009, 2009.

11 Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara, "Joint Chinese Word Segmentation and POS Tagging Using an Error-Driven Word-Character Hybrid Model," IEICE transactions on information and systems 92 (12), 2009.

12 Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lu, "A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging," In Proceedings of ACL-2008, 2008.

13 Tetsuji Nakagawa and Kiyotaka Uchimoto, "Hybrid Approach to Word Segmentation and POS Tagging," In Proceedings of ACL Demo and Poster Sessions, 2007.

14 Rie Kubota Ando and Tong Zhang, "A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data," Journal of Machine Learning Research, 2005.

15 Jun Suzuki and Hideki Isozaki, "Semi-Supervised Sequential Labeling and Segmentation using Gigaword Scale Unlabeled Data," In Proceedings of ACL-08: HLT, 2008.

16 Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins, "An Empirical Study of Semi-supervised Structured Conditional Models for Dependency Parsing," In Proceedings of EMNLP-2009, 2009.

17 Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa, "Improving Dependency Parsing with Subtrees from auto-Parsed Data," In Proceedings of EMNLP-2009, 2009.

18 Terry Koo, Xavier Carreras, and Michael Collins, "Simple Semi-supervised Dependency Parsing," In Proceedings of ACL-2008. 2008.

19 Daichi Mochihashi, Jun Suzuki, and Akinori Fujino. “Semi-supervised morphological analysis by the integration of conditional random fields and hierarchical Bayesian language models,” In Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing 2011. (in Japanese)

20 Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jenifer C. Lai, and Robert L.Mercer, “Class-based N-gram models of natural language,” Computational Linguistics, 18 (1992), pp. 467–479, 1992.

21 Wenliang Chen, Jun’ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa, “Improving Dependency Parsing with Subtrees from auto-Parsed Data,” In Proceedings of EMNLP 2009, 2009.

22 Wenliang Chen, Jun’ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa, “Exploiting Subtrees in Auto-Parsed Data to Improve Dependency Parsing,” Computational Intelligence, Vol. 28, Issue 3, pp. 426-451, 2012.

23 Ryan McDonald, Koby Crammer, and Fernando Pereira, “Online large-margin training of dependency parsers,” In Proceedings of ACL 2005, 2005

24 Ryan McDonald and Fernando Pereira, “Online learning of approximate dependency parsing algorithms,” In Proceedings of EACL2006, 2006.

25 Xavier Carreras, “Experiments with a higher-order projective dependency parser,” In Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, 2007

26 Terry Koo, Xavier Carreras, and Michael Collins, “Simple semi-supervised dependency parsing,” In Proceedings of ACL-08: HLT, 2008.

27 Joakim Nivre and Ryan McDonald, “Integrating graph-based and transition-based dependency parsers,” In Proceedings of ACL-08: HLT, 2008.

28 Xavier Carreras, Michael Collins, and Terry Koo, “Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing,” In Proceedings of CoNLL 2008, 2008.

29 Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins, “An empirical study of semi-supervised structured conditional models for dependency parsing,” In Proceedings of EMNLP 2009, 2009.

30 Kun Yu, Daisuke Kawahara, and Sasao Kurohashi, “Chinese dependency parsing with large scale automatically constructed case structures,” In Proceedings of COLING 2008, 2008.

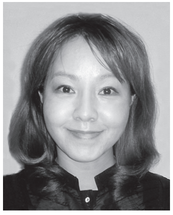
31 Hai Zhao, Yan. Song, Chunyun Kit, and Guodong Zhou, “Cross language dependency parsing using a bilingual lexicon,” In Proceedings of ACL-IJCNLP 2009, 2009.

(Accepted June 14, 2012)

***KAZAMA Jun’ichi**, Ph.D.*

*Senior Researcher, Information Analysis Laboratory, Universal Communication Research Institute*

*Natural Language Processing, Machine Learning*



***WANG Yiou**, Ph.D.*

*Researcher, Information Analysis Laboratory, Universal Communication Research Institute*

*Morphylogical Analysis, Opinion Analysis, Machine Translation, Constructing Language Resources*

***KAWADA Takuya**, Ph.D.*

*Researcher, Information Analysis Laboratory, Universal Communication Research Institute*

*Linguistics*