Sentence Boundary Detection Using Sequential Dependency Analysis Combined with CRF-based Chunking

Takanobu Oba, Takaaki Hori, Atsushi Nakamura

NTT Communication Science Laboratories, NTT Corporation,
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan

{oba,hori,ats}@cslab.kecl.ntt.co.jp

Abstract

In spoken language, sentence boundaries are much less explicit than in written language. Since conventional natural language processing (NLP) techniques are generally designed assuming the sentence boundaries are already given, it is crucial to detect the boundaries accurately for applying such NLP techniques to spoken language. Classification frameworks, such as Support Vector Machines (SVMs) and Conditional Random Fields (CRFs), can be used to detect the boundaries. With these methods, the sentence boundaries are determined based on local sentence-end-like word sequences around the boundaries. However, the methods do not evaluate whether or not each block determined by the boundaries is appropriate as a sentence. We have proposed sequential dependency analysis (SDA), which extracts the dependency structure of unsegmented word sequences with a subsidiary mechanism of sentence boundary detection. In this paper, we extend SDA by combining it with CRFs to reflect both the properties of local word sequences and the appropriateness as a sentence. In this way we achieve more accurate sentence boundary detection. The experimental result shows that our proposed method provides better detection accuracy than that obtained with SVMs or CRFs alone. Our method can also work sequentially because it is based on the SDA framework and can be used for on-line spoken applications.
Index Terms: sentence boundary detection, CRF, sequential dependency analysis

1. Introduction

Natural language processing (NLP) is a key technology for retrieving useful information automatically from raw text data by several types of linguistic analysis, such as tagging and lexical-dependency analysis. Automatic speech recognition expands the applicability of NLP, since it can provide text transcriptions of speech contained in raw audio data. Thus any NLP technique has the potential to deal with audio data through its transcribed text.

However, problems arise when analyzing such transcribed text data. For example, in most types of linguistic analysis, sentence boundaries are assumed to be given and many NLP applications, such as language translation and summarization, regard sentences as basic processing blocks. It is known, however, that sentence boundaries are much less explicit in spoken language than in written language. When applying NLP to spoken language, this makes the accurate detection of boundaries between sentences or sentence-like blocks in speech-transcribed texts a crucial issue.

To detect the boundaries in an unpunctuated text, we can utilize such classification frameworks as Hidden Markov Models (HMMs), Support Vector Machines (SVMs), and Conditional Random Fields (CRFs) [1]. These approaches learn whether a local word sequence has the properties of a sentence end, and detect sentence boundaries based on the patterns of the sequence of a few words. However, a sentence end can be expressed in various ways in spoken language and the sentence boundaries should be detected considering the appropriateness as a sentence.

We have recently proposed sequential dependency analysis (SDA) [2], which extracts dependency relationships between words with a subsidiary mechanism of sentence boundary detection. In this paper, SDA is expanded by combining it with a chunking method based on CRFs. CRFs divide an input word sequence into meaningful clusters, such as base-phrases, and provide the posterior probabilities that are for sentence boundary and for the other boundary of meaningful clusters. Then SDA is employed to detect sentence boundaries taking account of the posterior probabilities indicated by the CRFs while extracting dependency structures between the meaningful clusters. With our proposed method, the appropriateness of a sentence boundary is evaluated from two standpoints, namely the properties of a local word sequence estimated by the CRFs, and the appropriateness of a sentence estimated by analyzing the dependency structure. This significantly improves the accuracies of both sentence boundary detection and dependency analysis, which can synergistically enhance the overall performance of spoken language analysis.

In addition, our proposed method has the advantage of being applicable to on-line systems such as simultaneous interpretation, speech summarization for on-line captioning, and spoken dialogue systems. As with the original SDA, the algorithm can accept a word sequence sent continuously from a speech recognizer, and detect sentence boundaries. This paper is organized as follows. In section 2, we present a boundary detection framework for word clusters and sentences based on the CRF chunking approach. Section 3 describes parsing and training algorithms for SDA. In section 4, we present our proposed combination of CRF chunking and SDA for the accurate detection of sentence boundaries. The experimental results reported in section 5 clearly show the effectiveness of our approach for detecting sentence boundaries.

2. Chunking based on CRFs

2.1. Chunking

Chunking in NLP is the procedure used to divide a token (word) sequence into groups. One group consists of continuous tokens

and is called a chunk.

Most conventional chunking methods distinguish the groups by labeling each token. Some label representations are proposed in [3]. With IOB2[1], which is one type of representation, the label "B" means the token is at the beginning of a chunk, "I" means the token is involved in the current chunk, and "O" means the token is not involved in the current chunk.

It is possible to distinguish different types of chunks by employing different labels. In this paper, since we use a chunking method to detect both sentence boundaries and meaningful cluster boundaries, we introduce two labels Bs and Bb instead of B, where Bs represents the first token of a sentence and Bb represents the first token of a meaningful cluster that is not at the beginning of a sentence.

2.2. Conditional random fields

CRFs are discriminative models designed for sequence labeling problems such as tagging, named-entity extraction, and chunking.

Suppose that the random variable sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent input sequences and label sequences, respectively, and the generic input and label sequences are denoted as $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

A CRF on $(\boldsymbol{X}, \boldsymbol{Y})$ is specified by a local feature vector $\boldsymbol{f}$ and the corresponding weight vector $\boldsymbol{\lambda}$. The CRF's global feature vector is given by $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}, i)$. i denotes the position number. Then, the conditional probability distribution based on the CRF is defined as

$$P_{\boldsymbol{\lambda}}(\boldsymbol{Y}|\boldsymbol{X}) = \frac{\exp(\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{Y}, \boldsymbol{X}))}{Z_{\boldsymbol{\lambda}}(\boldsymbol{X})} \quad (1)$$

where $Z_{\boldsymbol{\lambda}}(\boldsymbol{X}) = \sum_{\boldsymbol{y}} \exp \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{X})$. The most probable label sequence $\hat{\boldsymbol{y}}$ for input sequence $\boldsymbol{x}$ is

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$$

and can be searched for efficiently using the Viterbi algorithm.

Now we represent the transition matrix $M_i(\boldsymbol{x}) = [M_i(y, y'|\boldsymbol{x})]$ from y to y' for position i as $M_i(y, y'|\boldsymbol{x}) = \exp \boldsymbol{\lambda} \cdot \boldsymbol{f}(y, y', \boldsymbol{x}, i)$. We can efficiently calculate the observation probability of y at position i given $\boldsymbol{x}$ with the forward-backward algorithm.

$$P_{\boldsymbol{\lambda}}(\boldsymbol{Y}_i = y|\boldsymbol{x}) = \frac{\boldsymbol{\alpha}_i(y|\boldsymbol{x})\boldsymbol{\beta}_i(y|\boldsymbol{x})}{Z_{\boldsymbol{\lambda}}(\boldsymbol{x})} \quad (2)$$

where the forward and backward vectors, $\boldsymbol{\alpha}_i(\boldsymbol{x})$ and $\boldsymbol{\beta}_i(\boldsymbol{x})$, which are initialized on $\boldsymbol{\alpha}_0 = 1$ and $\boldsymbol{\beta}_{|\boldsymbol{x}|} = 1$, are defined by

$$\begin{aligned} \boldsymbol{\alpha}_i(\boldsymbol{x}) &= \boldsymbol{\alpha}_{i-1}(\boldsymbol{x})M_i(\boldsymbol{x}) && \text{where } 0 < i \leq |\boldsymbol{x}| \\ \boldsymbol{\beta}_i(\boldsymbol{x}) &= M_{i+1}(\boldsymbol{x})\boldsymbol{\beta}_{i+1}(\boldsymbol{x}) && \text{where } 1 \leq i < |\boldsymbol{x}|. \end{aligned}$$

Equation (2) represents the appropriateness of the label being y at position i.

CRFs can decide the label of a token considering the labels of the anteroposterior positions of the token, and this is an advantage over HMMs and SVMs. However, in practice, the decision as regards each label is seldom affected by long distance tokens because the influence of the local features $\boldsymbol{f}$, which express the property extracted from the local tokens, is too strong.

[1]IOB2 was newly proposed, adding a slight difference to the original IOB, which was renamed IOB1 in [3].

3. Sequential dependency analysis

The object of dependency analysis is to extract the dependency structure of a sentence, which constitutes the modification relationships between units (meaningful clusters or words). Generally, when a unit u modifies a unit v, it is said that u links to v and this is represented as $u \rightarrow v$. u is called the modifier and v is called the head. A unit that does not modify any units in a sentence is called a sentence head. A dependency structure $\boldsymbol{D}$ is represented with a set of head units "$v_1, v_2, \cdots, v_N$" corresponding to modifier units "$u_1, u_2, \cdots, u_N$".

Dependency analysis finds the most appropriate dependency structure $\boldsymbol{D}^*$ from the hypothetical structures of a sentence. The most general method is based on probabilistic parsing. Where a unit sequence $\boldsymbol{U}$ is given,

$$\boldsymbol{D}^* = \arg\max_{\boldsymbol{D}} P(\boldsymbol{D}|\boldsymbol{U}) \quad (3)$$

$$P(\boldsymbol{D}|\boldsymbol{U}) = \prod_{i=1}^{N} P(u_i \rightarrow v_i|\boldsymbol{\Phi}(u_i, v_i, \boldsymbol{U})). \quad (4)$$

$\boldsymbol{\Phi}(u_i, v, \boldsymbol{U})$ is a linguistic feature vector. $P(u_i \rightarrow v|\boldsymbol{\Phi}(u_i, v, \boldsymbol{U}))$ is modeled based on the maximum entropy method as

$$P(u_i \rightarrow v|\boldsymbol{\Phi}(u_i, v, \boldsymbol{U})) = \frac{\exp(\boldsymbol{w} \cdot \boldsymbol{\Phi}(u_i, v))}{\sum_{c \in \boldsymbol{C_i}} \exp(\boldsymbol{w} \cdot \boldsymbol{\Phi}(u_i, c))},$$

where $\boldsymbol{C_i}$ represents the set of candidates for the head of u_i. Using parsed training data, the weight vector $\boldsymbol{w}$ can be estimated based on the maximum entropy criterion so as to distinguish the correct head of u_i from the other candidates.

We have already proposed sequential dependency analysis (SDA) [2]. In spoken language, sentence boundaries are unknown and also the input unit sequence from a speech recognizer does not necessarily end with the sentence boundary at an arbitrary time. SDA enables us to detect sentence boundaries while analyzing the dependency structure of the input sequence. Thus SDA solves the problem of conventional dependency analysis in which it is assumed that the sentence boundaries are already given. In SDA, meta symbols <*c*> and <*b*> are introduced to represent an arbitrary unseen unit and a sentence boundary, respectively. The link $u_i \rightarrow$<*c*> denotes a dependent relationship where u_i modifies a unit in the unseen part of the input sequence. Figure 1 shows an example dependency structure of an incomplete sentence u_1, u_2, u_3. SDA can obtain the dependency structure of incomplete sentences by attaching <*c*> to the end of the input sequence observed until the current time.

Figure 1: Dependency structure for incomplete sentences.

SDA can also detect sentence boundaries by adding the meta symbol <*b*> to the set of head candidates $\boldsymbol{C_i}$. If the most likely dependency structure $\boldsymbol{D}^*$ has links to <*b*>, a sentence boundary can be detected at each position where <*b*> appears as a head. This means finding the most likely structure from all possible structures obtained from sequences including and not including <*b*>. But, it is assumed that <*b*> itself never links to another unit. Figure 2 shows two dependency structures for the same input sequence. A

sentence boundary is detected on the left-hand side while it is not detected on the right-hand side. Finally the better structure is selected based on $P(\boldsymbol{D}|\boldsymbol{U})$, and consequently it is decided whether a sentence boundary exists or not. For sequential analysis, when a new input sequence is received, SDA updates the links to <c> in the already analyzed sequence and processes the new sequence. This procedure is repeated for every input of new units.

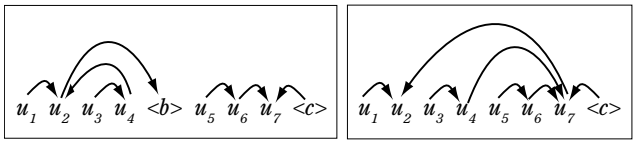

Figure 2: Search for sentence boundaries based on dependency analysis.

4. Sentence boundary detection based on the combination with CRFs and SDA

4.1. Incremental chunking

We employ CRF-based chunking together with SDA, and expand the method of chunking to increase the accuracy of boundary detection by an incremental approach. This also enables us to employ on-line processing for successive inputs of spoken language. Processing each word sequence between two pauses is a natural way of chunking. Since most pauses exist at boundaries of meaningful clusters, the basic units are not broken by pauses. In addition, the sequence usually has an appropriate length for processing. Thus it is suitable for detecting meaningful-cluster boundaries. It is, however, unsuitable for detecting sentence boundaries. The preceding context of the boundary is particularly important for the detection of sentence ends. The simple pause-based approach cannot provide sufficient context information.

We avoid this problem by using the following procedure. First, an input sequence before a pause is divided into meaningful clusters by chunking. The result corresponds to the first chunks depicted in Figure 3. Before the chunking for the second interval, we add the inputs that constituted the last meaningful cluster in the first chunks (w_3 to w_6) to the beginning of the next input sequence. This procedure enables us to associate words prior to w_7 with the local features $\boldsymbol{f}$ at the position $i = 7$ and potentially provides more accurate labeling.

In addition, we apply the rule that once labels are decided they are never changed by subsequent analysis because we might obtain different labels at the overlapping part. In Figure 3, the overlapping part is from w_3 to w_6 and the labels in the final chunks are reflected in the result for the first chunks.

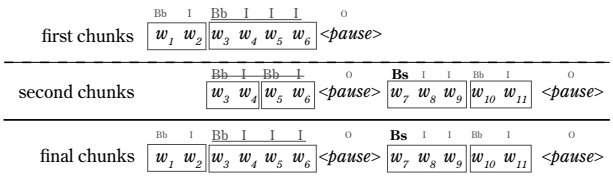

Figure 3: Incremental chunking.

4.2. SDA considering chunking probability

With our proposed method, the dependency structure is extracted by SDA while the sentence boundaries are being detected using both CRFs and SDA. The appropriateness of a sentence boundary is estimated based on both equation 2 of the CRFs and the probability of linking <b> given by SDA. That is, SDA analyzes the most appropriate unit sequence given by the CRFs where the dependency probability $P(u_i \rightarrow v|\boldsymbol{\Phi}(u_i, v, \boldsymbol{U}))$ is replaced with the following probability.

$$P_{\boldsymbol{W}}(u_i \rightarrow v|\boldsymbol{\Phi}(u_i, v, \boldsymbol{U})) = \begin{cases} P_{\boldsymbol{\lambda}}(\boldsymbol{Y}_{(<b>)} = \mathrm{Bs}|\boldsymbol{W})^{\alpha} \cdot P(u_i \rightarrow v|\boldsymbol{\Phi}) & \text{if } v = <b> \\ P_{\boldsymbol{\lambda}}(\boldsymbol{Y}_{(<b>)} = \mathrm{Bb}|\boldsymbol{W})^{\alpha} \cdot P(u_i \rightarrow v|\boldsymbol{\Phi}) & \text{otherwise} \end{cases} \quad (5)$$

$\mathrm{Y}_{(<b>)}$ denotes the label given to a word that is next to a hypothetic sentence boundary. This corresponds to the label of w_6 in Figure 4, which considers the possibility of a sentence boundary existing between w_5 and w_6. $\boldsymbol{W}$ is an input word sequence. $\boldsymbol{\Phi}(u_i, v, \boldsymbol{U})$ is abbreviated to $\boldsymbol{\Phi}$. α is a scaling parameter.

In Figure 4, for example, assume that the word sequence is divided into four units that are represented as rectangles. If $P_{\boldsymbol{\lambda}}(\boldsymbol{Y}_{(<b>)} = \mathrm{Bs}|\boldsymbol{W}) \approx P_{\boldsymbol{\lambda}}(\boldsymbol{Y}_{(<b>)} = \mathrm{Bb}|\boldsymbol{W})$, it is difficult to decide the label of w_6 solely by using the information provided by local words around w_6. However, we can say that a sentence boundary exists between w_5 and w_6 if the probability that the head of unit u_2 is <b> is much higher than for the other head candidates of u_2. Both the local and global views directly evaluate the sentence boundary appropriateness in our proposed method.

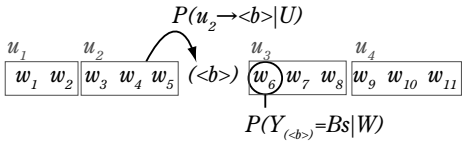

Figure 4: Combination of chunking and dependency analysis for sentence boundary detection.

5. Experiment

The Corpus of Spontaneous Japanese (CSJ) [4] contains 604 hours of speech talk data and includes many annotations such as transcriptions, sentence boundaries, dependency structures, pause intervals and *bunsetsu* boundaries.

*Bunsetsu*s are phrasal units in Japanese and consist of one or more content words, such as nouns and verbs, and zero or more function words, such as postpositions. All *bunsetsu*s are continuous, that is, every word belongs to one of the *bunsetsu*s. A sentence boundary can exist where a *bunsetsu* boundary is located. Pauses in speech are recognized as *bunsetsu* boundaries in CSJ. We adopted *bunsetsu*s as meaningful clusters and tried to detect *bunsetsu* boundaries and sentence boundaries simultaneously.

We divided 189 talk data into a training set, a development data, and a test set. All experiments were done for transcriptions.

	talks	sentences	*bunsetsu*s	words
training set	169	16,885	184,409	419,742
development data	10	715	11,193	26,539
test set	10	1,012	11,162	26,122

To indicate the advantages of our proposed method, we prepared the most general situation as a baseline. Here, normal de-

pendency analysis was used to extract the dependencies between *bunsetsu*s after the input sequence had been provided with both *bunsetsu* and sentence boundaries by the chunking method. We define normal dependency analysis as dependency analysis given sentence boundaries.

SVMs and CRFs were applied to the chunking method. First, we indicate their chunking accuracy. The input sequences consisted not only of words but also of marks indicating pauses of over 0.2 $msec$. The features were surfaces and Parts-of-Speech (POSs) and POS-subcategories of inputs from position $i-3$ to $i+3$, and their combinations. In addition, labels of positions $i-3$ to $i-1$ were used for SVMs, and Bi-gram features of each label were used for CRFs.

Table 1 shows the F-values for the sentence boundary detection and the *bunsetsu* detection. The excellent discrimination ability of SVMs resulted in highly accurate *bunsetsu* detection. However CRFs that considered a wider label sequence were better than SVMs for sentence boundary detection. We believe that sentence boundary detection is more difficult than *bunsetsu* detection, and that analysis using only local information is insufficient for accurate sentence boundary detection.

Table 1: Sentence boundary and *bunsetsu* detection accuracy of SVMs and CRF chunking.

	sentence boundary detection accuracy	*bunsetsu* detection accuracy
SVMs	84.5	96.8
CRFs	85.5	96.5

Next, we employed normal dependency analysis for the above *bunsetsu* sequences given by the SVMs and CRFs. The result was compared with that of our proposed method where the input sequence was directly analyzed using CRFs and SDA, incrementally and sequentially.

The feature vectors of dependency analysis $\mathbf{\Phi}$ were surface forms, POSs, POS-subcategories, inflection types, inflection forms, beginning, distance, and their combinations. With our proposed method, surface forms of meta symbols were added to the feature vectors as well as those of other regular words.

We determined the value of α in our proposed method, used in equation (5), so as to maximize the accuracy of the sentence boundary detection for the development data. The value used in this experiment was 1.2. We confirmed that analysis accuracy is not too sensitive around this value.

To calculate the dependency accuracy, we defined a correct link as a link where the *bunsetsu* pair consisting of a head and a modifier was correctly extracted and the *bunsetsu* boundaries were correctly detected.

Table 2 shows the F-values of the dependency analysis and sentence boundary detection. Baseline 1 corresponds to a normal dependency analysis after SVM chunking, and baseline 2 corresponds to that after CRF chunking had been employed. Therefore, the sentence boundary detection of baselines 1 and 2 is achieved by chunking and the accuracies are the same as those described in Table 1.

Our proposed method performed more accurately than the baselines, and the accuracy of the sentence boundary detection was significantly improved. The analysis that took account of the dependency relationships worked effectively for the detection of sentence boundaries.

Table 2: The accuracy of dependency analysis and sentence boundary detection.

	Dep Acc	SBD Acc
baseline 1 NormDep (SVM chunking)	76.3	84.5
baseline 2 NormDep (CRF chunking)	75.7	85.5
proposed method	**76.6**	**88.4**

Dep Acc:Dependency Accuracy, SBD Acc:Sentence Boundary Detection Accuracy, NormDep:Normal Dependency analysis.

The dependency accuracy of baseline 1 was better than that of baseline 2. This is based on the high *bunsetsu* detection rate of SVMs. However, this difference is very small from a practical standpoint because it is largely based on the *bunsetsu* detection errors and, in fact, the extracted dependency relationships maintain semasiological appropriateness.

In contrast, the practical implication of the improvement in dependency accuracy achieved when using our proposed method is considerable, because this improvement is based on the improvement in sentence boundary detection accuracy. The dependency links between two sentences are extremely inappropriate and there is a risk that they will generate an inexplicable semantic connection.

6. Conclusion

We proposed a sentence boundary detection method based on the combination of CRFs and SDA, which can simultaneously extract the dependency structures of meaningful clusters, such as basephrases. With our proposed method, sentence boundaries are detected from two standpoints, the local properties of the word sequence evaluated by CRFs, and the appropriateness of the sentence, evaluated through analysis of the dependencies between meaningful clusters.

We showed that our method could detect sentence boundaries more accurately than the single use of chunking based on SVMs and CRFs. In addition, we found that accurate sentence boundary detection improves the accuracy of language processing.

In this paper, we focused on linguistic information for more accurate sentence boundary detection, but we also think acoustic information, such as prosody, should be considered in future work.

7. References

[1] John Lafferty, Andrew McCallum, Fernando Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. ICML*, pp. 282-289, 2001.

[2] Takanobu Oba, Takaaki Hori, Atsushi Nakamura, "Sequential Dependency Analysis for Spontaneous Speech Understanding," *Proc. ASRU*, pp. 284-289, 2005.

[3] Erik F. Tjong Kim Sang, Jorn Veenstra, "Representing Text Chunks," *Proc. EACL*, pp. 173-179, 1999.

[4] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, Hitoshi Isahara, "Spontaneous Speech Corpus of Japanese," *Proc. LREC*, pp. 947-952, 2000.