Can Symbol Grounding Improve Low-Level NLP? Word Segmentation as a Case Study

Hirotaka Kameko†, Shinsuke Mori‡, and Yoshimasa Tsuruoka†
†Graduate School of Engineering, The University of Tokyo
Hongo, Bunkyo-ku, Tokyo, Japan
{kameko, tsuruoka}@logos.t.u-tokyo.ac.jp
‡Academic Center for Computing and Media Studies, Kyoto University
Yoshida Honmachi, Sakyo-ku, Kyoto, Japan
forest@i.kyoto-u.ac.jp

Abstract

We propose a novel framework for improving a word segmenter using information acquired from symbol grounding. We generate a term dictionary in three steps: generating a pseudo-stochastically segmented corpus, building a symbol grounding model to enumerate word candidates, and filtering them according to the grounding scores. We applied our method to game records of Japanese chess with commentaries. The experimental results show that the accuracy of a word segmenter can be improved by incorporating the generated dictionary.

1 Introduction

Today we can easily obtain a large amount of text associated with multi-modal information, and there is a growing interest in the use of non-textual information in the natural language processing (NLP) community. Many of these studies aim to output natural language sentences from a nonlinguistic modality, such as image (Farhadi et al., 2010; Yang et al., 2011; Rohrbach et al., 2013). Kiros et al. (2014) showed that multi-modal information improves the performance of a language model.

Inspired by these studies, we explore a method for improving the performance of a low-level NLP task using multi-modal information. In this work, we focus on the task of word segmentation (WS) in Japanese. WS is often performed as the first processing step for languages without clear word boundaries, and it is as important as part-of-speech (POS) tagging in English. We assume that a large set of pairs of non-textual data and sentences describing them is available as the information source. In our experiments, the pairs consist of game states in *Shogi* (Japanese chess) and textual comments on them, which were made by Shogi experts. We enumerate substrings (character sequences) in the sentences and match them with *Shogi* states by a neural network model. The rationale here is that substrings which match with non-language data well tend to be real words.

Our method consists of three steps (see Figure 1). First, we segment commentary sentences for a game state in various ways to produce word candidates. Then, we match them with game states of a Shogi playing program. Finally, we compile the symbol grounding results at all states and incorporate them to an automatic WS. To the best of our knowledge, this is the first result reporting a performance improvement in an NLP task by symbol grounding.

2 Stochastically Segmented Corpus

Before symbol grounding, we need to segment the text into words that include probable candidate words. For this purpose, we use a stochastically segmented corpus (SSC) (Mori and Takuma, 2004). Then we propose to simulate it by a normal (deterministically) segmented corpus to avoid the problem of computational cost.

2.1 Stochastically Segmented Corpora

An SSC is defined as a combination of a raw corpus C_r (hereafter referred to as the character sequence $\boldsymbol{x}_1^{n_r}$) and word boundary probabilities of the form P_i, which is the probability that a word boundary exists between two characters x_i and x_{i+1}. These probabilities are estimated by a model based on logistic regression (LR) (Fan et al., 2008) trained on a manually segmented corpus by referring to the surrounding characters[1]. Since there are word boundaries before the first character and after the last character of the corpus, $P_0 = P_{n_r} = 1$. The expected frequency of a word

[1]In the experiment we used the same features as those used in Neubig et al., (2011). .

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2298–2303,
Lisbon, Portugal, 17-21 September 2015.

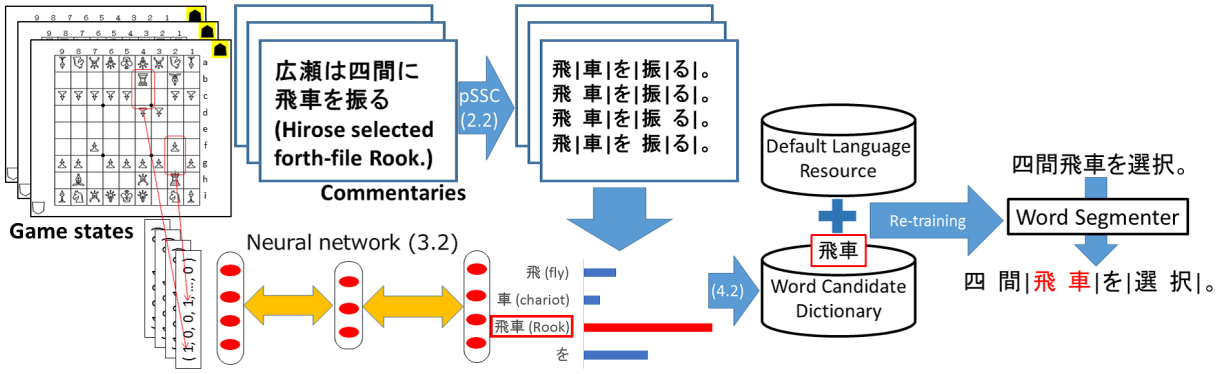

Figure 1: Overview of our method.

$\boldsymbol{w}$ in an SSC is calculated as follows: $f_r(\boldsymbol{w}) = \sum_{i \in O} P_i \left\{ \prod_{j=1}^{k-1} (1 - P_{i+j}) \right\} P_{i+k}$, where $O = \{i \mid x_{i+1}^{i+k} = \boldsymbol{w}\}$ is the set of all the occurrences of the string matching with $\boldsymbol{w}$[2].

2.2 Pseudo-Stochastically Segmented Corpora

The computational cost (in terms of both time and space) for calculating the expected frequencies in an SSC is very high[3], so it is not a practical approach for symbol grounding. In this work, we approximate an SSC using a deterministically segmented corpus, which we call a pseudo-stochastically segmented corpus (pSSC). The following is the process we use to produce a pSSC from an SSC.

- For i = 1 to $n_r - 1$
  1. output a character x_i,
  2. generate a random number $0 \leq p < 1$,
  3. output a word boundary if $p < P_i$ or output nothing otherwise.

Now we have a corpus in the same format as a standard segmented corpus with variable (non-constant) segmentation, where x_i and x_{i+1} are segmented with the probability of P_i. We execute the above procedure m times and divide the counts by m. The law of large numbers guarantees that the approximation errors decrease to 0 when $m \to \infty$.

3 Symbol Grounding

As the target of symbol grounding, we use states (piece positions) of a Shogi game and commentaries associated with them. We should note, however, that our framework is general and applicable to different types of combinations such as image/description pairs (Regneri et al., 2013).

3.1 Game Commentary

The Japanese language is one of the languages without clear word boundaries and we need an automatic WS as the first step of NLP. In Shogi, there are many professional players and many commentaries about game states are available.

3.2 Grounding Words

We build a symbol grounding model using a Shogi commentary dataset. We use a set of pairs of a Shogi state S_i and a commentary sentence C_i as the training set. A Shogi state S_i is converted into a feature vector f(S_i). We generate m (in our experiment, $m = 4$) pSSC C'_i from C_i. C'_i contains m corpora of the same text body but with different word segmentation, C'_{ij} ($j = 1, \ldots, m$). We treat these as m pairs of a feature vector of Shogi state f(S_i) and a sequence of words C'_{ij}. We train a model which predicts words in C'_{ij} using f(S_i) as input.

We use a multi-layer perceptron as the prediction model. The input is a vector of the features of a state. The hidden layer is a 100-dimensional vector and is activated by a bipolar sigmoid function. Its output is a d-dimensional real-valued vector, each of whose elements indicates whether a word in the vocabulary of d words appears in the commentary or not. The output layer is activated by a binary sigmoid function.

We use features of Shogi states which a computer Shogi program called Gekisashi (Tsuruoka et al., 2002) uses to evaluate the states in game tree search as input. The features of Shogi states used in this experiment are below:

a) Positions of pieces (e.g. my rook is at 2h).

[2]For a detailed explanation and a mathematical proof of this method, please refer to Mori and Takuma (2004) .

[3]This is because an SSC has many words and word fragments. Additionally, word 1-gram frequencies must be calculated using floating point numbers instead of integers.

b) Pieces captured (e.g. the opponent has a bishop).
c) Combinations of a) and b) (e.g. my king is at 7h and the opponent's rook is at 7b).
d) Other heuristic features.

Among them, a), b) and c) occupy the majority.

Unlike normal symbol grounding, the vocabulary contains many word candidates appearing in the pSSC generated from the commentaries. Some are real words and some are wrong fragments. These wrong fragments will appear more or less randomly in the commentaries than real words. The perceptron therefore cannot acquire strong relation between states and fragments and the output values of the perceptron will be smaller than those of real words.

4 Word Segmentation Using Symbol Grounding Result

This section describes a baseline automatic word segmenter and a method for incorporating the symbol grounding result to it.

4.1 Baseline Word Segmenter

Among many Japanese WS and morphological analyzers (word segmentation and POS tagging), we adopt pointwise WS (Neubig et al., 2011), because it is the only word segmenter which is capable of adding new words without POS information.

The input of the pointwise WS is an unsegmented character sequence $\boldsymbol{x} = x_1 x_2 \cdots x_k$. The word segmenter decides if there is a word boundary $t_i = 1$ or not $t_i = 0$ by using support vector machines (SVMs) (Fan et al., 2008). The features are character n-grams and character type n-grams ($n = 1, 2, 3$) around the decision points in a window with a width of 6 characters. Additional features are triggered if character n-grams in the window match with character sequences in the dictionary.

4.2 Training a Word Segmenter with Grounded Words

As a first trial for incorporating symbol grounding results to an NLP task, we propose to generate a dictionary based on the symbol grounding result. We can expect that the word candidates that are given high scores by the perceptron in the symbol grounding result have strong relationship to the positions. In other words, we can make a good dictionary by selecting word candidates in descending order of the scores. As a method for taking all the occurrences into account, we test the following three functions:

sum: the summation of the scores of all the output vectors,
ave: the average of them,
max: the maximum in them.

First, we acquire a V-dimensional real-valued vector for each Shogi state S_i as the result of symbol grounding. Then, for each candidate in C'_{ij}, we get the element of the vector which corresponds to the candidate as the score of the candidate. After that, we get the summation of, the average of, or the maximum in the scores of the same candidate over the whole dataset.

Finally we select the top R percent of word candidates in descending order of the value of **sum**, **ave**, or **max** and add them to the WS dictionary and retrain the model.

Table 1: Corpus specifications.

	#sent.	#words	#char.
Training			
BCCWJ	56,753	1,324,951	1,911,660
Newspaper	8,164	240,097	361,843
Conversation	11,700	147,809	197,941
Develepment			
Shogi-dev.	170	2,501	3,340
Test			
BCCWJ-test	6,025	148,929	212,261
Shogi-test	3,299	24,966	32,481

5 Evaluation

We conducted word segmentation experiments in the following settings.

5.1 Corpora

The annotated corpus we used to build the baseline word segmenter is the manually annotated part (core data) of the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa, 2008), plus newspaper articles and daily conversation sentences. We also used a 234,652-word dictionary (UniDic) provided with the BCCWJ. A small portion of the BCCWJ core data is reserved for testing. In addition, we manually segmented sentences randomly obtained from Shogi commentaries. We divided these sentences into two parts: a development set and a test set. Table 1 shows the details of these corpora.

To make a pSSC, we prepared 33,151 pairs of a Shogi position and a commentary sentence. The

Table 2: WS accuracy on BCCWJ.

	Recall	Prec.	F-meas.
Baseline	98.99	99.06	99.03
+ Sym.Gro.	99.03	99.01	99.02

Table 3: WS accuracy on Shogi commentaries.

	Recall	Prec.	F-meas.
Baseline	90.12	91.43	90.77
+ Sym.Gro.	90.60	91.66	91.13

sentences are converted into pSSC $m = 4$ times by an LR word segmentation model trained from the training data in Table 1 and sent to the symbol grounding module.

5.2 Word Segmentation Systems

We built the following two word segmentation models (Neubig et al., 2011) to evaluate our framework.

Baseline: The model is trained from training data shown in Table 1 and UniDic.

+Sym.Gro.: The model is trained from the language resources for the **Baseline** and the symbol grounding result.

To decide the function and the value of R for **+Sym.Gro.** (see Section 4.2), we measured the accuracies on the development set of all the combinations. The best combination was **sum** and $R = 0.011$[4]. In this case, 127 words were added to the dictionary.

5.3 Results and Discussion

Following the standard in word segmentation experiments, the evaluation criteria are recall, precision, and F-measure (their harmonic mean).

Table 2 and 3 show WS accuracies on BCCWJ-test and Shogi-test, respectively. The difference in accuracy of the baseline method on BCCWJ-test and Shogi-test shows that WS of Shogi commentaries is very difficult. Like many other domains, Shogi commentaries contain many special words and expressions, which decrease the accuracy.

When we compare the F-measures on Shogi-test (Table 3), **+Sym.Gro.** outperforms **Baseline**. The improvement is statistically significant (at 5% level). The error reduction ratio is comparable to a natural annotation case (Liu et al., 2014), despite the fact that our method is unsupervised except for a hyperparameter. Thus we can say that WS improvement by symbol grounding is as valuable as the annotation additions.

From a close look at the comparison of the recall and the precision, we see that the improvement in the recall is higher than that of the precision. This result shows that the symbol grounding successfully acquired new words with a few erroneous words. As the final remark, the result on the general domain (Table 2) shows that our framework does not cause a severe performance degradation in the general domain.

6 Related Work

The NLP task we focus on in this paper is word segmentation. One of the first empirical methods was based on a hidden Markov model (Nagata, 1994). In parallel, there were attempts at solving Chinese word segmentation in a similar way (Sproat and Chang, 1996). These methods take words as the modeling unit.

Recently, Neubig et al. (2011) have presented a method for directly deciding whether there is a word boundary or not at each point between characters. For Chinese word segmentation, there are some attempts at tagging characters with BIES tags (Xue, 2003) by a sequence labeller such as CRFs (Lafferty et al., 2001), where B, I, E, and S means the beginning of a word, intermediate of a word, the end of a word, and a single character word, respectively. The pointwise WS can be seen as character tagging with the BI tag system, in which there is no constraint between neighboring tags. For Japanese WS, our preliminary experiments showed that the combination of the BI tag system with SVMs is slightly better than the BIES tag system with CRFs. This is another reason why we used the former in this paper. Our extension of word segmentation is, however, applicable to the BIES/CRFs combination as well.

The method we describe in this paper is unsupervised and requires a small amount of annotated data to tune the hyperparameter. From this viewpoint, the approach based on natural annotation (Yang and Vozila, 2014; Jiang et al., 2013; Liu et al., 2014) may come to readers' mind. In these studies, tags in hyper-texts were regarded as partial annotations and used to improve WS performance using CRFs trainable from such data (Tsuboi et al., 2008). Mori and Nagao (1996) proposed a method for extracting new words from a large amount of raw text. Murawaki and Kuro-

[4] In addition we measured the accuracies on the test set of all the combinations and found that the same function and the value of the parameter are the best. This indicates the stability of the function and the parameter.

hashi (2008) proposed an online method in a similar setting. In contrast to these studies, this paper proposes to use other modalities, game states as the first trial, than languages.

7 Conclusion

We have described an unsupervised method for improving word segmentation based on symbol grounding results. To extract word candidates from raw sentences, we first segment sentences stochastically, and then match the word candidate sequences with game states that are described by the sentences. Finally, we selected word candidates referring to the grounding scores. The experimental results showed that we can improve word segmentation by using symbol grounding results. Our framework is general and it is worth testing on other NLP tasks. As future work, we will apply other deep neural network models to our approach. It is interesting to apply the symbol grounding results to an embedding model-based word segmentation approach (Ma and Hinrichs, 2015). It is also interesting to extend our method to deal with other types of non-textual information such as images and economic indices.

Acknowledgment

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by JSPS Grants-in-Aid for Scientific Research Grant Number 26540190.

References

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the 11th European Conference on Computer Vision*, pages 15–29.

Wenbin Jiang, Meng Sun, Yajuan Lu, Yating Yang, and Qun Liu. 2013. Discriminative learning with natural annotations: Word segmentation as a case study. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 761–769.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. 2014. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 595–603.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Yijia Liu, Yue Zhang, Wangxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 864–874.

Jianqiang Ma and Erhard Hinrichs. 2015. Accurate linear-time Chinese word segmentation via embedding matching. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1733–1743.

Kikuo Maekawa. 2008. Balanced corpus of contemporary written Japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*, pages 101–102.

Shinsuke Mori and Makoto Nagao. 1996. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 1119–1122.

Shinsuke Mori and Daisuke Takuma. 2004. Word *n*-gram probability estimation from a Japanese raw corpus. In *Proceedings of the Eighth International Conference on Speech and Language Processing*, pages 1037–1040.

Yugo Murawaki and Sadao Kurohashi. 2008. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 429–437.

Masaaki Nagata. 1994. A stochastic Japanese morphological analyzer using a forward-DP backward-A* N-best search algorithm. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 201–207.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 529–533.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36.

Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *Proceedings of the 14th International Conference on Computer Vision*, pages 433–440.

Richard Sproat and Chilin Shih William Gale Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 897–904.

Yoshimasa Tsuruoka, Daisaku Yokoyama, and Takashi Chikayama. 2002. Game-tree search algorithm based on realization probability. *ICGA Journal*, 25(3):145–152.

N. Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese*, 8(1):29–48.

Fan Yang and Paul Vozila. 2014. Semi-supervised Chinese word segmentation using partial-label learning with conditional random fields. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 90–98.

Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 444–454.