Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition

Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura and **Tomoko Ohkuma**
Fuji Xerox Co., Ltd.
{misawa.shotaro, motoki.taniguchi, yasuhide.miura, ohkuma.tomoko}@fujixerox.co.jp

Abstract

Recently, neural models have shown superior performance over conventional models in NER tasks. These models use CNN to extract sub-word information along with RNN to predict a tag for each word. However, these models have been tested almost entirely on English texts. It remains unclear whether they perform similarly in other languages. We worked on Japanese NER using neural models and discovered two obstacles of the state-of-the-art model. First, CNN is unsuitable for extracting Japanese sub-word information. Secondly, a model predicting a tag for each word cannot extract an entity when a part of a word composes an entity. The contributions of this work are (i) verifying the effectiveness of the state-of-the-art NER model for Japanese, (ii) proposing a neural model for predicting a tag for each character using word and character information. Experimentally obtained results demonstrate that our model outperforms the state-of-the-art neural English NER model in Japanese.

1 Introduction

Named Entity Recognition (NER) is designed to extract entities such as location and product from texts. The results are used in sophisticated tasks including summarizations and recommendations. In the past several years, sequential neural models such as long-short term memory (LSTM) have been applied to NER. They have outperformed the conventional models (Huang et al., 2015). Recently, Convolutional Neural Network (CNN) was introduced into many models for extracting sub-word information from a word (Santos and Guimaraes, 2015; Ma and Hovy, 2016). The models achieved higher performance because CNN can capture capitalization, suffixes, and prefixes (Chiu and Nichols, 2015). These models predict a tag for each word assuming that words can be separated clearly by explicit word separators (e.g. blank spaces). We refer to such model as a "word-based model", even if inputs include characters.

When Japanese NER employs a recent neural model, two obstacles arise. First, extracting sub-word information by CNN is unsuitable for Japanese language. The reasons are that Japanese words tend to be shorter than English and Japanese characters have no capitalization. Secondly, the word-based model cannot extract entities when a part of a word composes an entity. Japanese language has no explicit word separators. Word boundaries occasionally become ambiguous. Therefore, the possibility exists that entity boundary does not match word boundaries. We define such phenomena as "boundary conflict". To avoid this obstacle, NER using finer-grained compose units than words are preferred in Japanese NERs (Asahara and Matsumoto, 2003; Sassano and Utsuro, 2000). We follow these approaches and expand the state-of-the-art neural NER model to predict a tag for each character: a "character-based model".

The contributions of our study are: (i) application of a state-of-the-art NER model to Japanese NER and verification of its effectiveness, and (ii) proposition of a "character-based" neural model with concatenating words and characters. Experimental results show that our model outperforms the state-of-the-art neural NER model in Japanese.

2 Related Work

Conventional Models: Conventional NER systems employ machine learning algorithms that use

Proceedings of the First Workshop on Subword and Character Level Models in NLP, pages 97–102,
Copenhagen, Denmark, September 7, 2017. ©2017 Association for Computational Linguistics.

inputs which are hand-crafted features such as POS tags. Support Vector Machine (Isozaki and Kazawa, 2002), maximum entropy models (Bender et al., 2003), Hidden Markov Models (Zhou and Su, 2002) and CRF (Klinger, 2011; Chen et al., 2006; Marcinczuk, 2015) were applied.

Word-based Neural Models: A neural model was applied to sequence labeling tasks also in NER (Collobert et al., 2011). Modified models using Bi-directional LSTM (BLSTM) or Stacked LSTM were proposed (Huang et al., 2015; Lample et al., 2016). Recently, new approaches introducing CNN or LSTM for extracting subword information from character inputs have been found to outperform other models (Lample et al., 2016). Rei et al. (2016) proposed the model using an attention mechanism whose inputs are words and characters. Above all, BLSTM-CNNs-CRF (Ma and Hovy, 2016) achieved state-of-the-art performance on the standard English corpus: CoNLL2003 (Tjong Kim Sang and De Meulder, 2003).

Character-based Neural Models: Kuru et al. (2016) proposed a character-based neural model. This model, which inputs only characters, exhibits good performance on the condition that no external knowledge is used. This model predicts a tag for each character and forces that predicted tags in a word are the same. Therefore, it is unsuitable for languages in which boundary conflicts occur.

Japanese NER: For Japanese NER, many models using conventional algorithms have been proposed (Iwakura, 2011; Sasano and Kurohashi, 2008). Most such models are character-based models to deal with boundary conflicts.

Tomori et al. (2016) applied a neural model to Japanese NER. This study uses non-sequential neural networks with inputs that are hand-crafted features. This model uses no recent advanced approaches for NER, such as word embedding or CNN to extract sub-word information. Therefore, the effectiveness of recent neural models for Japanese NER has not been evaluated.

3 Japanese NER and Characteristics

One common definition of entity categories for Japanese NER is Sekine's extended named entity hierarchy (Sekine et al., 2002). This definition includes 30 entity categories. This study used the corpus annotated in Mainichi newspaper articles (Hashimoto et al., 2008).

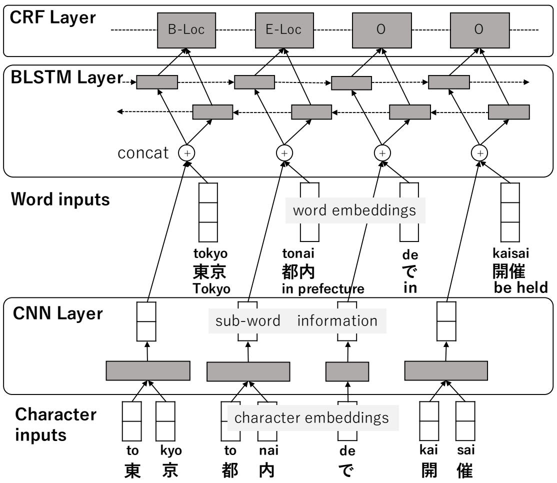

Figure 1: Structure of BLSTM-CNNs-CRF. The superscripts on Japanese show pronunciations. The subscripts on Japanese words are translations.

Japanese language is written without blank spaces. Therefore, word segmentations that are made using morphological analysis are needed to use word information. However, some word segmentations cause boundary conflicts. As an example, one can consider the extraction of the correct entity "*Tokyoto*" (Tokyo prefecture) from "*Tokyotonai*" (in Tokyo prefecture).

Tokyo|*tonai* (Tokyo / in pref.): word boundary

Tokyoto|*nai* (Tokyo pref. / in): entity boundary

These slashes show word and entity boundaries. The entity boundary does not match the word boundary. Therefore, the entity candidates by word-based models are "*Tokyo*" and "*Tokyotonai*." It is impossible to extract the entity "*Tokyoto*".

Word lengths of Japanese language tend to be shorter than those of English. The average word length in entities in CoNLL 2003 (Reuters news service) is 6.43 characters. That in the Mainichi newspaper corpus is 1.95. Therefore, it is difficult to extract sub-word information in Japanese in a manner that is suitable for English.

4 NER Models

4.1 Word-based neural model

In this study, we specifically examine BLSTM-CNNs-CRF (Ma and Hovy, 2016) because it achieves state-of-the-art performance in the CoNLL 2003 corpus. Figure 1 presents the architecture of this model. This word-based model combines CNN, BLSTM, and CRF layers. We describe each layer of this model as the following.

CNN Layer: This layer is aimed at extracting subword information. The inputs are character embeddings of a word. This layer consists of convolution and pooling layers. The convolution layer produces a matrix for a word with consideration of the sub-word. The pooling layer compresses the matrix for each dimension of character embedding.

BLSTM Layer: BLSTM (Graves and Schmidhuber, 2005) is an approach to treat sequential data. The output of CNN and word embedding are concatenated as an input of BLSTM.

CRF Layer: This layer was designed to select the best tag sequence from all possible tag sequences with consideration of outputs from BLSTM and correlations between adjacent tags. This layer introduces a transition score for each transition pattern between adjacent tags. The objective function is calculated using the sum of the outputs from BLSTM and the transition scores for a sequence.

4.2 Character-based neural model

To resolve the obstacles when applying a recent neural model, we propose character-based BLSTM-CRF model (Char-BLSTM-CRF). This model, which consists of BLSTM and CRF layers, predicts a tag for every character independently. Figure 2 presents the model structure.

This model gives an input for each character to predict a tag for a character independently. Additionally, we introduce word information with character information as inputs of this model. Character information is a character embedding and word information is the embedding of the word containing the character. That is, the same word embedding will be used as inputs of characters constructing a word. This enables us to utilize pre-training of word embeddings with the effectiveness shown in English (Ma and Hovy, 2016).

We assume that it is difficult for the CNN layer to extract the Japanese sub-word information. Moreover, we assume that sufficient information can be extracted from a simple character input. Consequently, the model uses no CNN layer.

5 Experiments

5.1 Experiment Conditions

We evaluate our models using the Mainichi newspaper corpus. We specifically examine the four categories of the highest frequency: *Product*, *Location*, *Organization*, *Time*. Table 1 presents statistics related to this corpus. We prepared pre-trained word embeddings using skip-gram model (Mikolov et al., 2013). Seven years (1995–1996 and 1998–2002) of Mainichi newspaper articles which include almost 500 million words are used for pre-training. We conduct parameter tuning using the development dataset. We choose the unit number of LSTM as 300, the size of word embedding as 500, that of character embedding as 50, the maximum epoch as 20, and the batch size as 60. We use Adam (Kingma and Ba, 2014), with the learning rate of 0.001 for optimization. We use MeCab (Kudo, 2005) for word segmentation. Other conditions are the same as those reported for an earlier study (Ma and Hovy, 2016).

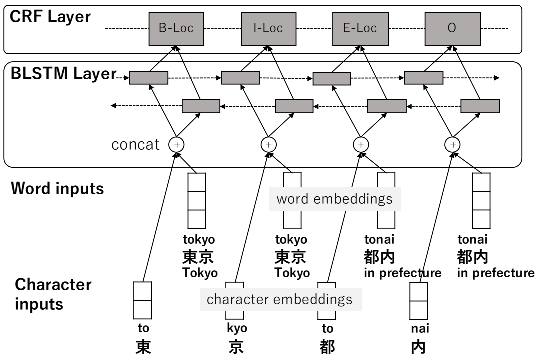

Figure 2: Structure of Char-BLSTM-CRF.

	Train	Dev.	Test
Articles	5,424	678	682
Sentences	62,373	8,032	7,689
Words	1,591,781	200,843	197,649
Product (NE)	39,734	5,087	5,120
Location (NE)	24,981	3,238	3,251
Organization (NE)	19,119	2,535	2,690
Time (NE)	17,252	2,216	2,148

Table 1: Statistics of the corpus.

5.2 Results

Table 2 presents F1 scores of models. We compare BLSTM-CNNs-CRF, Char-BLSTM-CRF, and character-based conventional CRF. To verify the effectiveness of the CNN layer and the CRF layer in BLSTM-CNNs-CRF, we use additional word-based models of two types with a component changed from BLSTM-CNNs-CRF. BLSTM-CRF is a model with eliminated the CNN layer and character inputs. BLSTM-CNNs is a model with the CRF layer replaced by a softmax layer. To evaluate the performance improvement of character inputs, word inputs and pre-training, we prepared additional three configurations of Char-BLSTM-CRF: without word, without character, without pre-training.

	BLSTM -CRF	BLSTM -CNNs	BLSTM -CNNs -CRF	CRF	Char-BLSTM -CRF w/o word	Char-BLSTM -CRF w/o char	Char-BLSTM -CRF w/o pretraining	Char- BLSTM -CRF
input	word	word+character			character	word	word+character	
output	word			character				
Product	†83.89	80.83	83.82	80.72	78.12	84.28	80.13	**84.46**
Location	†88.57	87.52	88.46	87.54	86.77	91.30	87.63	**91.47**
Organization	85.87	82.07	†**85.99**	79.62	77.72	85.26	80.79	85.56
Time	†94.39	92.50	93.51	93.00	93.35	94.03	93.55	**94.44**
Average	†87.16	84.59	86.98	84.25	82.81	87.80	84.33	**88.06**

Table 2: F1 score of each models. Average is a weighted average. † expresses the best result in the word-based models for each entity category. Bold means the best result in all models for each entity category. "output" means the unit of prediction; "input" shows information used as inputs.

Entity Category	*Pro.*	*Loc.*	*Org.*	*Time*
Word Length in Entity	1.99	2.07	2.51	1.20

Table 3: Averaged word length in entity.

	Pro.	*Loc.*	*Org.*	*Time*
Total Conflicts	66	75	23	7
Extracted Entities	25	68	8	3

Table 4: Number of entities with boundary conflicts and that of entities extracted by Char-BLSTM-CRF.

Word-based Neural Models: Among word-based models, BLSTM-CNNs-CRF is the best model for *Organization*. Also, BLSTM-CRF is the best model for *Product*, *Location*, and *Time*. We confirm that cutting-edge neural models are suitable for Japanese language because each model outperforms CRF.

When comparing BLSTM-CNNs and BLSTM-CNNs-CRF, the CRF layer contributes to improvement by 2.39 pt. When comparing BLSTM-CRF and BLSTM-CNNs-CRF, the CNN layer is worse by 0.18 pt. Here, the CNN layer and the CRF layer improve by 2.36 pt and 1.75 pt in English (Ma and Hovy, 2016). Therefore, the CRF layer performs similarly but the CNN layer performs differently. The CNN layer enhances the model flexibility. Nevertheless, this layer can scarcely extract information from characters because Japanese words are shorter than English, according to Section 3.

Table 3 shows the averaged word length after splitting an entity into words for each entity category. Words composing *Time* entities is the shortest. Therefore, information is scarce, especially in *Time*. In contrast, the words composing *Organization* is long. Therefore, CNN can extract information from characters in a word in *Organization*. This is the reason why BLSTM-CNNs-CRF performs better than BLSTM-CRF in *Organization*.

Character-based Neural Models: The results of averaged F1 scores show that Char-BLSTM-CRF is more suitable for Japanese than word-based models. When comparing four configurations of Char-BLSTM-CRF, pre-training is critically important for performance. Character input also contributes to the performance improvement in Char-BLSTM-CRF, although the input degrades the performance in a word-based model.

Total Conflicts in the table 4 is the total number of entities with boundary conflicts in the test data. Extracted Entities in the table is the number of entities that Char-BLSTM-CRF extracts among the entities with boundary conflicts. Results show that the model extracts entities with boundary conflicts which cannot be extracted by word-based models. The number of entities with boundary conflicts extracted by Char-BLSTM-CRF is the largest in *Location*. When comparing the performance of Char-BLSTM-CRF and BLSTM-CRF for each entity category, the largest performance improvement of 2.90 pt is achieved in *Location*. By extracting 68 entities with boundary conflicts in *Location*, Char-BLSTM-CRF achieves about 2 pt improvement out of total 2.90 pt. It can be said that almost all improvements of Char-BLSTM-CRF are from extracting these entities.

In contrast, Char-BLSTM-CRF is inappropriate for *Organization*. The averaged word length of entities that are not extracted accurately by BLSTM-CNNs-CRF is 4.07; that by Char-BLSTM-CRF is 4.87. It can be said that Char-BLSTM-CRF is unsuitable for extracting long words. We infer that the inputs of LSTM become redundant and that LSTM does not work efficiently. Especially, the averaged word length of *Organization* is long according to Table 3. For that reason, the character-based model is inappropriate in *Organization*.

6 Conclusions

As described in this paper, we verified the effectiveness of the state-of-the-art neural NER model for Japanese. The experimentally obtained results show that the model outperforms conventional CRF in Japanese. Results show that the CNN layer works improperly for Japanese because words of the Japanese language are short.

We proposed a character-based neural model incorporating words and characters: Char-BLSTM-CRF. This model outperforms a state-of-the-art neural NER model in Japanese, especially for the entity category consisting of short words. Our future work will examine reduction of redundancy in character-based model by preparing and combining different LSTMs for word and character inputs. Also to examine the effects of pre-training of characters in Char-BLSTM-CRF is our future work.

References

Masayuki Asahara and Yuji Matsumoto. 2003. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 7th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 8–15.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the 7th Conference on North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume4*, pages 148–151.

Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. 2006. Chinese named entity recognition with conditional random fields. In *Proceedings of the 5th Special Interest Group of Chinese Language Processing Workshop*, pages 118–121.

Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Taiichi Hashimoto, Takashi Inui, and Koji Murakami. 2008. Constructing extended named entity annotated corpora (in japanese). *The Special Interest Group Technical Reports of the Information Processing Society of Japan*, 113:113–120.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7.

Tomoya Iwakura. 2011. A named entity recognition method using rules acquired from unlabeled data. In *Proceedings of the 8th International Conference on Recent Advances in Natural Language Processing*, pages 170–177.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Roman Klinger. 2011. Automatically selected skip edges in conditional random fields for named entity recognition. In *Proceedings of the 8th International Conference on Recent Advances in Natural Language Processing*, pages 580–585.

Taku Kudo. 2005. Mecab: Yet another part-of-speech and morphological analyzer. *http://taku910.github.io/mecab/*.

Onur Kuru, Arkan Ozan Can, and Deniz Yuret. 2016. Charner: Character-level named entity recognition. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 911–921.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Michal Marcinczuk. 2015. Automatic construction of complex features in conditional random fields for named entities recognition. In *Proceedings of the 10th International Conference on Recent Advances in Natural Language Processing*, pages 413–419.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Marek Rei, Gamal Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 309–318.

Cicero Nogueira dos Santos and Victor Guimaraes. 2015. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008*.

Ryohei Sasano and Sadao Kurohashi. 2008. Japanese named entity recognition using structural natural language processing. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 607–612.

Manabu Sassano and Takehito Utsuro. 2000. Named entity chunking techniques in supervised learning for japanese named entity recognition. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 705–711.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of 3rd International Conference on Language Resources and Evaluation*, pages 1818–1824.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th conference on North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 4*, pages 142–147.

Suzushi Tomori, Takashi Ninomiya, and Shinsuke Mori. 2016. Domain specific named entity recognition referring to the real world by deep neural networks. In *proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 236–242.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480.