Fine-tuning for Named Entity Recognition Using Part-of-Speech Tagging

Masaya Suzuki Kanako Komiya Minoru Sasaki Hiroyuki Shinnou
Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
{17nm708n, kanako.komiya.nlp}@vc.ibaraki.ac.jp
{minoru.sasaki.01, hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp

Abstract

In recent years, machine learning methods beyond the confines of conventional supervised learning have been used along with deep learning methods and intensively investigated. Fine-tuning, which improves the performance of one task by re-learning using the weights of a model learned for another task as initial values, is one such example. This paper proposes fine-tuning named entity recognition (NER) using part-of-speech tagging. The experiments revealed that fine-tuning improves the performance of NER. They also revealed that there was no performance improvement even if POS tag set included tags that corresponded to an NE tag, when there was a difference in definitions between these tags.

1 Introduction

In recent years, many researchers use machine learning methods beyond the confines of conventional supervised learning along with deep learning methods and intensively investigate them. In general, it is difficult to learn a model of a task with high performance using small training data when we use conventional supervised learning. However, if we had larger corpora of the other tasks that are similar to and related to the task, it could help to achieve high performance. Fine-tuning, which improves the performance of one task by re-learning using the weights of a model learned for another task as initial values, is one way to realize it.

Named entity recognition (NER) involves seeking to locate and classify elements in texts into predefined categories such as the names of people, organizations, and locations and is one of the sequential labeling tasks. NER is related to part-of-speech (POS) tagging because POS tagging is also a sequential labeling task and POS are popular features for NER. Therefore, fine-tuning for NER using POS tagging would be a good way to learn an NER model with high performance when we have a small NE corpus and a large POS corpus at the same time. This paper proposes fine-tuning NER using POS tagging. We targeted at Japanese NER and POS tagging in the current study. There are many researches on Japanese NER like (Komiya et al., 2018) and (Iwakura et al., 2016). In addition, many researchers investigated NER using multitasking learning or joint learning like (Qu et al., 2016) and (Peng and Dredze, 2015). We investigated Japanese NER using fine-tuning (See Section 2).

The proposed model is developed on the basis of a model described in (Ma and Hovy, 2016). We simplified the model by excluding a CNN for the character-level representations (See Section 3). We evaluated precision, recall, and F-measure based on the gold standard with experiments using the models (described in Section 4). We discuss the results by comparing them to those of the existing method without fine-tuning (See Section 5) and conclude our work (See Section 6).

2 Related Work

NER has been studied for a long time. When we focus on Japanese NER, the Information Retrieval and Extraction Exercise (IREX) (Sekine and Isa-

32nd Pacific Asia Conference on Language, Information and Computation
Hong Kong, 1-3 December 2018

hara, 2000)[1] is a famous shared task that defined nine tags including eight NE tag types. Iwakura et al. (2016) annotated BCCWJ NE corpus[2] with the tags of the definition in IREX to the Balanced Corpus of Contemporary Japanese (BCCWJ) (Maekawa et al., 2014)[3]. Ichihara et al. (2015) investigated the performance of the existing NE recognizer and showed that the errors increased for document types that were very different from the training data of the NE recognizer. Komiya et al. (2018) compared two methods for annotating NE corpus using non-expert annotators.

In the research on the sequence labeling, NER and POS tagging are often selected as the target tasks at the same time. For example, Ma and Hovy (2016) proposed an end-to-end sequence labeling system that automatically learns a model from word-level and character-level representations and obtained state-of-the-art performance in both tasks.

There have been researches on NER using transfer learning. The followings are some examples. Qu et al. (2016) proposed a transfer learning method for NER in the case where not only domains but also labels of NER do not match. Peng and Dredze (2016) improved the performance of NER on Chinese social media (Peng and Dredze, 2015) by multitask learning of Chinese word segmentation. Peng and Dredze (2017b) improved the performance of the task using a modified dataset created with (He and Sun, 2017)[4]. Peng and Dredze (2017a) proposed a multitask domain adaptation considering Chinese word segmentation and NER.

Transfer learning methods were also used for a shared task on emerging and rare entity recognition (Derczynski et al., 2017)[5] in the 3rd workshop on noisy user-generated text (WNUT-2017). The shared task defined emerging and rare entities and provided datasets of social media for detecting these entities. (Aguilar et al., 2017), which is a paper that got the first position in this shared task, used multi-task learning of NE segmentation (i.e. a binary classification of whether a given token is an NE or not) and fine-grained NE categorization. Von Däniken and Cieliebak (2017), who wrote a paper that got the second position, also used a multitask learning method. They shared the lower layers of the network by a corpus of WNUT 2017 and a corpus of WNUT 2016 (Strauss et al., 2016)[6], which includes different NE tags from corpus of WNUT 2017. They also used sentence level features.

These researches investigated NER using multitasking learning or joint learning. However, we investigated NER using fine-tuning.

Fine-tuning attracts attentions from researchers and (Kanako Komiya and Hiroyuki Shinnou, 2018) investigated effective parameters for fine-tuning.

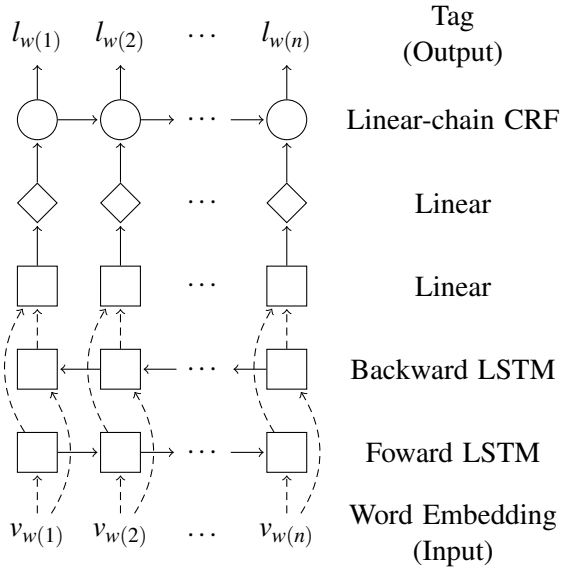

Figure 1: The neural network architecture of the proposed method. The dashed arrows indicate the dropout procedures applied to both the input and output vectors of the bi-directional LSTM. Square nodes indicate the layers to be fine-tuned. Diamond nodes indicate the layers only used for the target task.

3 Neural Network Architecture

Figure 1 shows the detailed neural network architecture of the proposed method. The model is developed on the basis of a model described in (Ma and

[1] https://nlp.cs.nyu.edu/irex/index-e.html

[2] https://sites.google.com/site/projectnextnlpne/en

[3] http://pj.ninjal.ac.jp/corpus_center/bccwj/en/

[4] https://github.com/hltcoe/golden-horse

[5] http://noisy-text.github.io/2017/emerging-rare-entities.html

[6] http://noisy-text.github.io/2016/ner-shared-task.html

Hovy, 2016). We simplified the model by excluding a CNN for the character-level representations. Therefore, this model consists of three layers: bidirectional LSTM, linear, and linear-chain CRF layers. We fine-tuned the bidirectional LSTM and linear layers. Dropout procedures were applied to both the input and output vectors of the bi-directional LSTM.

3.1 Learning the Tags of the Source Task

For fine-tuning, we have to train the tags of the source task. Here, tag $l_{w(t)}$, which is a tag of word $w(t)$ at time t, is obtained as follows for the source task, where $w(1)$, $w(2)$, $\cdots w(q)$ denote input sentence and L_1, L_2, $\cdots$, L_n denote the tag set of the source task.

1. A word embedding $v_{w(t)}$ of the word $w(t)$ is input to the bidirectional LSTM and an intermediate representation $h_{w(t)}$ is generated.

2. The intermediate representation $h_{w(t)}$ is input to the linear layer and $P(L_1|w(t))$, $P(L_2|w(t))$, $\cdots$, $P(L_n|w(t))$, occurrence probabilities of each tag, are obtained.

3. $P(L_1|w(t))$, $P(L_2|w(t))$, $\cdots$, $P(L_n|w(t))$ are input to the linear-chain CRF and the tag $l_{w(t)}$ corresponding to the time t is selected from $l_{w(1)}$, $l_{w(2)}$, $\cdots$, $l_{w(q)}$, the tag sequence of the source task in the input sentence $w(1)$, $w(2)$, $\cdots$, $w(q)$, using Viterbi algorithm.

3.2 Learning the Tags of the Target Task

We train the model of the target task using the weights of the model of the source task. Here, tag $l'_{w(t)}$, which is a tag of word $w(t)$ at time t, is obtained as follows for the target task, when $w(1)$, $w(2)$, $\cdots$, $w(q)$ denote input sentence, L_1, L_2, $\cdots$, L_n denote the tag set of the source task, and L'_1, L'_2, $\cdots$, L'_m denote the tag set of the target task:

1. A word embedding $v_{w(t)}$ of the word $w(t)$ is input to the bidirectional LSTM, and an intermediate representation $h_{w(t)}$ is generated.

2. The intermediate representation $h_{w(t)}$ is input to the linear layer, and $P(L_1|w(t))$, $P(L_2|w(t))$, $\cdots$, $P(L_n|w(t))$, occurrence probabilities of each tag of the source task, are obtained.

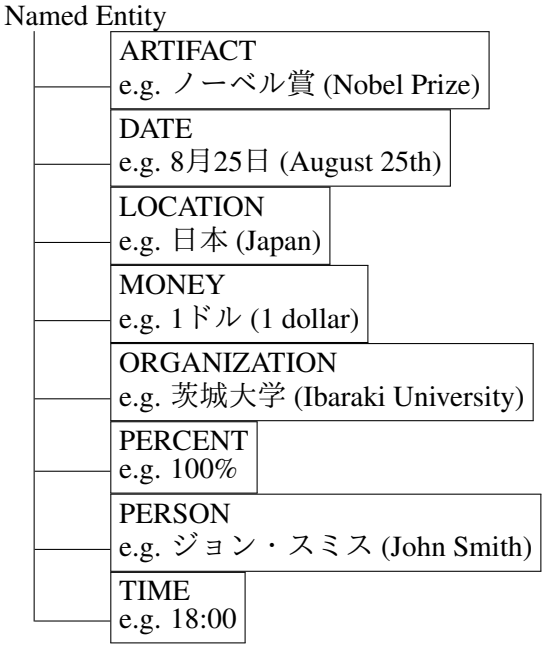

Figure 2: NE tag types

3. $P(L_1|w(t))$, $P(L_2|w(t))$, $\cdots$, $P(L_n|w(t))$ are input to the linear layer, and $P(L'_1|w(t))$, $P(L'_2|w(t))$, $\cdots$, $P(L'_m|w(t))$, occurrence probabilities of each tag of the target task, are obtained.

4. $P(L'_1|w(t))$, $P(L'_2|w(t))$, $\cdots$, $P(L'_m|w(t))$ are input to the linear-chain CRF and the tag $l'_{w(t)}$ corresponding to the time t is selected from $l'_{w(1)}$, $l'_{w(2)}$, $\cdots$, $l'_{w(q)}$, the tag sequence of the target task in the input sentence $w(1)$, $w(2)$, $\cdots$, $w(q)$, using Viterbi algorithm.

4 Experiments

We compared the proposed method (`POS2NER`), fine-tuning of the NER using the POS tagging task, to the conventional method (`NER`), the NER without fine-tuning.

4.1 Data

We used POS tagging as a source task and NER as a target task.

In NER task, we used nine kinds of tags defined by IREX, i.e., eight NE tag types (See Figure 2) and OPTIONAL tag[7] for ambiguous NEs. Table 1

[7] OPTIONAL tags do not have to be predicted.

Table 1: Summary of number of documents and NE tags

Documents		1,174
Tags	ARTIFACT	747
	DATE	3,567
	LOCATION	5,463
	MONEY	390
	ORGANIZATION	3,676
	PERCENT	492
	PERSON	3,840
	TIME	502
	OPTIONAL	585
	All	19,262

shows a summary of the number of documents and NE tags. We used IREX CRL data (CRL) (Sekine and Isahara, 2000)[8], which is an annotated corpus that consists of 1,174 articles of the Japanese newspaper "The Mainichi Shimbun" collected from January 1st to 10th.

In POS tagging, we used 21 POS tag types (See Figures 3 and 4) extracted from POS tag types used in UniDic[9]. POS tags are annotated for the same articles as the NER task. In other words, we used the Mainichi Shimbun annotated with POS and NER tags.

We used the IOBES format as a tagging scheme. We used NWJC2vec (Masayuki Asahara, 2018)[10], which is a 200 dimensional word2vec (Mikolov et al., 2013) model. This model trained from National Institute for Japanese Language and Linguistics (NINJAL) Web Japanese Corpus (NWJC) (Asahara et al., 2014)[11] containing ten billion words.

4.2 Tools and Settings

We used MeCab 0.996[12] as a morphological analyzer and UniDic[9] as a Japanese dictionary. We used Python 3.5.2 and Chainer v1.24.0 (Tokui et al., 2015)[13] for implementation of the neural network. We used GeForce GTX 1050 to train models. We used Adam (Kingma and Ba, 2014) as parameter op-

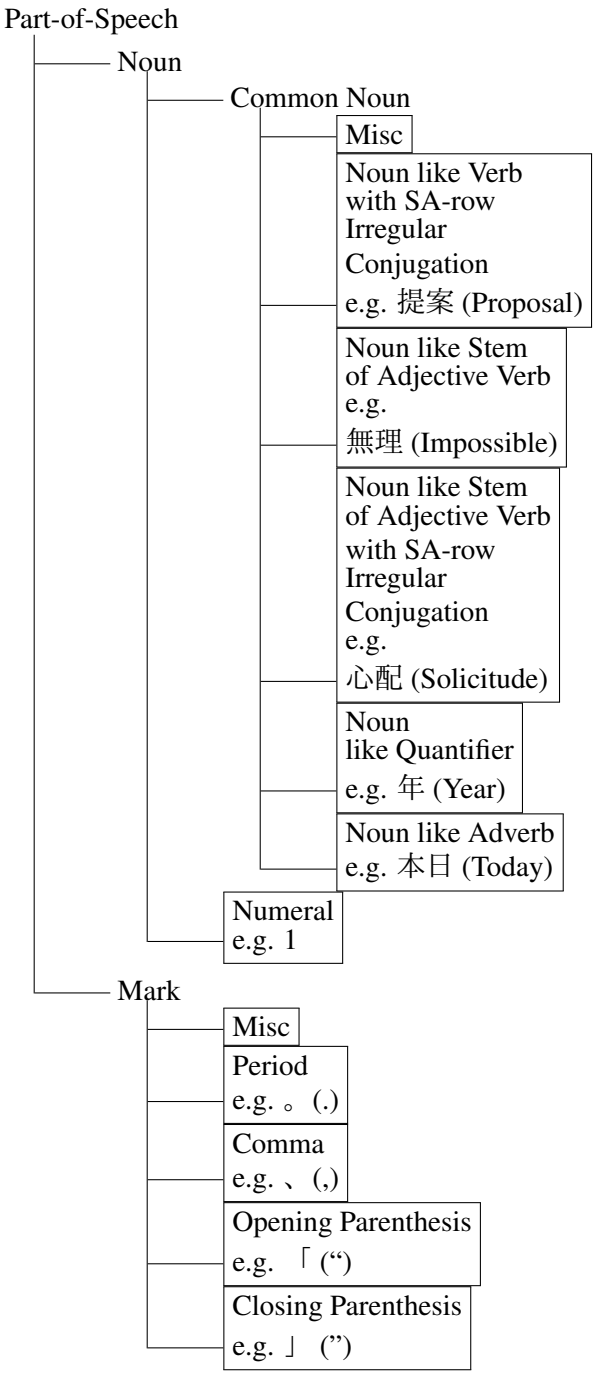

Figure 3: POS tag types (1/2)

[8] https://nlp.cs.nyu.edu/irex/index-e.html

[9] http://unidic.ninjal.ac.jp/ (in Japanese)

[10] http://nwjc-data.ninjal.ac.jp/ (in Japanese)

[11] http://pj.ninjal.ac.jp/corpus_center/nwjc/ (in Japanese)

[12] http://taku910.github.io/mecab/ (in Japanese)

[13] https://chainer.org/

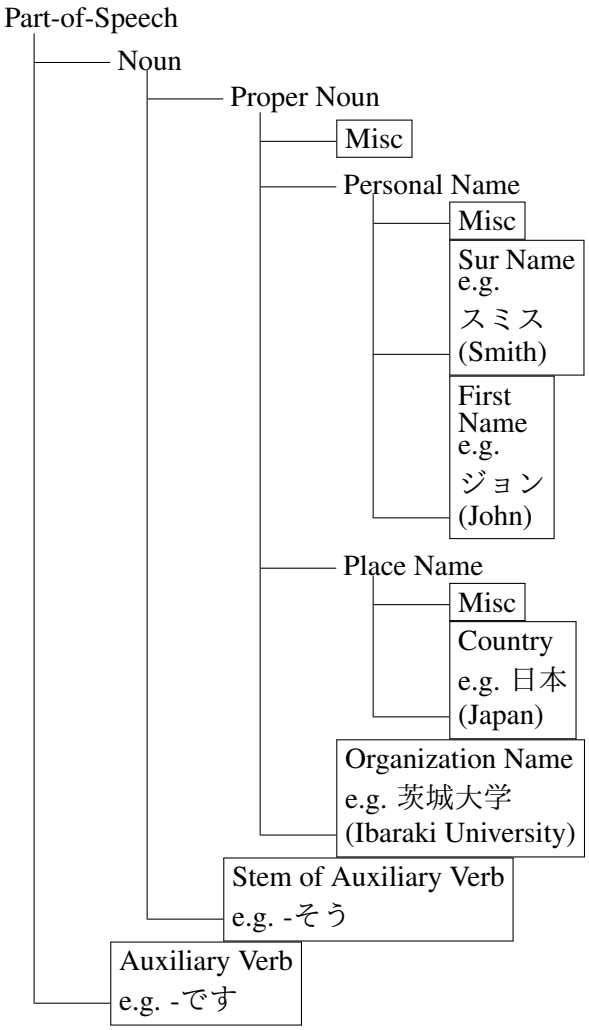

Figure 4: POS tag types (2/2)

Table 2: Corpus division ratio

Data	Ratio
Training Data	3
Development Set	1
Test Data	1

Table 3: Micro-averaged precision (P), recall (R), and F-measure (F) of each method

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	81.79	77.50	79.59
10	NER	82.58	76.80	79.59
20	POS2NER	82.49	77.91	80.14
20	NER	82.26	78.02	80.08
30	POS2NER	82.62	78.52	80.52
30	NER	82.59	77.77	80.11
40	POS2NER	82.88	78.57	80.67
40	NER	82.89	78.37	80.57
50	POS2NER	82.95	78.62	80.73
50	NER	82.80	78.05	80.36
60	POS2NER	82.83	78.82	80.78
60	NER	82.60	78.26	80.38

timization algorithm. In order to prevent overfitting, we used early stopping (Caruana et al., 2001) based on the performance on the development set. We set the number of dimensions of the intermediate representation to 200, identical to those of NWJC2vec. We set the dropout rate to 0.33 for all the dropout procedures. We set the number of epochs when learning of the source task in POS2NER to 10. We carried out five-fold cross validations and evaluated the precision (P), the recall (R), and the F-measure (F) based on the gold standard. when we carried out five-fold cross validations, we divided the corpus as shown in Table 2.

5 Results and Discussion

Table 3 shows the micro-averaged precision (P), recall (R), and F-measure (F) of each method for the whole dataset. Tables 4, 5, 6, 7, 8, 9, 10, and 11 show the micro-averaged precision (P), recall (R), and F-measure (F) for each tag. The higher values for the precision, recall, and F-measure among the two methods are written with underline.

Table 3 shows that POS2NER is always better than NER in the precision, recall, and F-measure except

Table 4: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of ARTIFACT tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	56.65	28.55	37.97
10	NER	61.41	29.22	39.60
20	POS2NER	57.65	28.25	37.92
20	NER	54.15	29.76	38.41
30	POS2NER	56.58	30.56	39.69
30	NER	55.53	30.97	39.76
40	POS2NER	54.36	31.77	40.10
40	NER	56.76	30.97	40.07
50	POS2NER	56.71	32.31	41.16
50	NER	60.23	28.42	38.62
60	POS2NER	57.82	31.23	40.56
60	NER	57.34	27.21	36.91

Table 5: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of DATE tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	90.60	94.67	92.59
10	NER	90.41	94.02	92.18
20	POS2NER	91.31	94.77	93.01
20	NER	90.36	95.11	92.67
30	POS2NER	90.88	95.20	92.99
30	NER	90.79	94.82	92.76
40	POS2NER	91.07	94.94	92.97
40	NER	90.47	95.03	92.69
50	POS2NER	90.95	94.88	92.87
50	NER	91.28	94.38	92.80
60	POS2NER	90.61	95.45	92.96
60	NER	90.98	94.91	92.90

Table 6: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of LOCATION tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	88.20	90.27	89.22
10	NER	88.16	90.81	89.47
20	POS2NER	88.01	91.37	89.66
20	NER	88.52	90.36	89.43
30	POS2NER	88.28	90.98	89.61
30	NER	88.44	90.45	89.44
40	POS2NER	88.43	90.63	89.52
40	NER	89.00	90.98	89.98
50	POS2NER	88.80	90.91	89.84
50	NER	88.65	90.65	89.64
60	POS2NER	88.58	90.86	89.71
60	NER	88.61	91.35	89.96

Table 7: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of MONEY tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	98.97	98.21	98.58
10	NER	99.21	96.67	97.92
20	POS2NER	98.72	99.23	98.98
20	NER	97.47	98.97	98.22
30	POS2NER	98.98	99.23	99.10
30	NER	98.97	98.72	98.84
40	POS2NER	98.47	99.23	98.85
40	NER	98.71	98.45	98.58
50	POS2NER	98.72	98.72	98.72
50	NER	98.71	98.46	98.59
60	POS2NER	98.71	98.46	98.59
60	NER	98.47	98.72	98.59

Table 8: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of ORGANIZATION tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	82.72	74.54	78.42
10	NER	84.32	72.12	77.75
20	POS2NER	83.32	74.13	78.46
20	NER	82.44	75.21	78.66
30	POS2NER	83.49	74.95	78.99
30	NER	83.55	74.99	79.04
40	POS2NER	83.56	74.75	78.91
40	NER	84.28	74.66	79.18
50	POS2NER	83.41	74.65	78.79
50	NER	83.03	74.88	78.75
60	POS2NER	83.67	75.67	79.47
60	NER	83.21	74.19	78.44

Table 9: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of PERCENT tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	95.13	95.33	95.23
10	NER	95.87	94.31	95.08
20	POS2NER	94.04	96.14	95.08
20	NER	95.71	95.33	95.52
30	POS2NER	95.94	96.14	96.04
30	NER	95.89	94.92	95.40
40	POS2NER	94.46	96.95	95.69
40	NER	93.85	96.14	94.98
50	POS2NER	95.75	96.14	95.94
50	NER	95.56	96.14	95.85
60	POS2NER	94.80	96.34	95.56
60	NER	95.57	96.54	96.06

Table 10: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of PERSON tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	88.16	83.97	86.02
10	NER	89.21	82.21	85.57
20	POS2NER	88.34	82.65	85.40
20	NER	89.38	83.29	86.23
30	POS2NER	88.95	83.75	86.27
30	NER	89.51	83.27	86.28
40	POS2NER	89.08	83.62	86.27
40	NER	88.70	83.46	86.00
50	POS2NER	88.79	84.14	86.40
50	NER	88.77	83.12	85.85
60	POS2NER	89.34	83.93	86.55
60	NER	88.51	83.85	86.12

Table 11: Micro-averaged precision (P), recall (R), and F-measure (F) of each method of TIME tag

Epoch	Task	P (%)	R (%)	F (%)
10	POS2NER	87.48	89.04	88.25
10	NER	89.61	87.65	88.62
20	POS2NER	88.85	90.44	89.63
20	NER	87.64	91.83	89.69
30	POS2NER	86.35	93.23	89.66
30	NER	88.26	89.84	89.04
40	POS2NER	88.72	94.02	91.30
40	NER	87.41	94.02	90.60
50	POS2NER	88.28	93.03	90.59
50	NER	86.95	94.22	90.44
60	POS2NER	86.13	95.22	90.44
60	NER	86.47	94.22	90.18

the precision and F-measure of 10 epochs, the recall of 20 epochs, and the precision of 40 epochs. Here, let us compare the precisions, recalls, and F-measures POS2NER at n epochs and NER at $n+10$ epochs because the model of POS2NER leaned 10 epochs more than NER; POS2NER learned the model of POS tagging 10 epochs. When we compare them, POS2NER is better than NER in the precisions, recalls, and F-measures when $n = 40$ and 50, the recall and F-measure when $n = 20$, and the recall when $n = 30$. These results indicate that the main reason for effectiveness of POS2NER is not increase of the number of epochs but fine-tuning itself.

Tables 3 and 6 show that the results of LOCATION tag of NER are often better than those of POS2NER compared with other tags. We thought this was strange because there are two POS tags corresponding to LOCATION NER-tag, namely, "Noun-Proper Noun-Place Name-Misc" and "Noun-Proper Noun-Place Name-Country." We expected that the NER performance would increase for the tags that had the corresponding POS tag. Therefore, we calculated the micro- and macro-averaged F-measure of POS tagging, which is the source task of POS2NER, for "Noun-Proper Noun-Place Name-Misc" tag and "Noun-Proper Noun-Place Name-Country" tags. However, the reason for low performances on LOCATION tags seems not the low performance of the POS tagging because the micro- and macro-averaged F-measure for both corresponding tags were approximately 90%. However, when comparing the place names tagged with these tags in the corpus, there was a difference between two tasks. Tags are only given to place names in the narrow scope such as "足立" ("Adachi")[14] for POS, whereas tags are given to place names in the broad scope such as "東京都足立区" ("Adachi, Tokyo")[14] and "グァンタナモ米軍基地" ("Guantanamo Bay Naval Base") for NER. We believe that this difference in definition made the NER performance on LOCATION tags lower.

6 Conclusions

We proposed fine-tuning for NER using POS tagging to learn an NER model with high performance when we have a small NE corpus and a large POS corpus. Our experiments were with Japanese NER and POS tagging. We evaluated the precision, recall, and F-measure based on the gold standard. The experiments revealed that fine-tuning improved the NER performance. They also revealed that there was no performance improvement even if POS tag set included tags that corresponded to an NE tag, when there is a difference in definitions between these tags.

[14] This is one of the special districts located in Tokyo, Japan.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 18K11421 and research grant of Woman Empowerment Support System of Ibaraki University.

References

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. 2017. A Multi-task Approach for Named Entity Recognition in Social Media Data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153.

Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi. 2014. Archiving and analysing techniques of the ultra-large-scale web-based Corpus Project of NINJAL, Japan. *Alexandria*, 25(1-2):129–148.

Rich Caruana, Steve Lawrence, and C Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Hangfeng He and Xu Sun. 2017. F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 713–718.

Masaaki Ichihara, Kanako Komiya, Tomoya Iwakura, and Maiko Yamazaki. 2015. Error Analysis of Named Entity Recognition in BCCWJ. *In Proceedings of Error Analysis Workshop, Natural Language Processing 2015*.

Tomoya Iwakura, Ryuichi Tachibana, and Kanako Komiya. 2016. Constructing a Japanese Basic Named

Entity Corpus of Various Genres. *ACL 2016*, pages 41–46.

Kanako Komiya and Hiroyuki Shinnou. 2018. Investigating Effective Parameters for Fine-tuning of Word Embeddings Using Only a Small Corpus. In *DeepLo 2018, Workshop of ACL 2018*, pages 60–67.

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

Kanako Komiya, Masaya Suzuki, Tomoya Iwakura, Minoru Sasaki, and Hiroyuki Shinnou. 2018. Comparison of Methods to Annotate Named Entity Corpora. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(4):No.34.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written Japanese. *Language resources and evaluation*, 48(2):345–371.

Masayuki Asahara. 2018. NWJC2Vec: Word embedding dataset from 'NINJAL Web Japanese Corpus'. *Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication*, 24(2):7–25, Feb.

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

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

Nanyun Peng and Mark Dredze. 2016. Improving Named Entity Recognition for Chinese Social Media with Word Segmentation Representation Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 149–155.

Nanyun Peng and Mark Dredze. 2017a. Multi-task Domain Adaptation for Sequence Tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100.

Nanyun Peng and Mark Dredze. 2017b. Supplementary Results for Named Entity Recognition on Chinese Social Media with an Updated Dataset.

Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, and Timothy Baldwin. 2016. Named Entity Recognition for Novel Types by Transfer Learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 899–905.

Satoshi Sekine and Hitoshi Isahara. 2000. IREX: IR and IE Evaluation project in Japanese. In *Proceedings of the 2nd International Conference on Language Resources & Evaluation*.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, volume 5.

Pius Von Däniken and Mark Cieliebak. 2017. Transfer learning and sentence level features for named entity recognition on tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 166–171.