Multi-Task Learning for Chemical Named Entity Recognition with Chemical Compound Paraphrasing

Taiki Watanabe[1,3]* watanabe-taiki@fujitsu.com
Akihiro Tamura[2,3] **Takashi Ninomiya**[2,3] {tamura,ninomiya}@cs.ehime-u.ac.jp
Takuya Makino[1,3] **Tomoya Iwakura**[1,3] {makino.takuya,iwakura.tomoya}@fujitsu.com
[1] Fujitsu Laboratories, Ltd. [2] Ehime University [3] RIKEN AIP-FUJITSU Collaboration Center

Abstract

We propose a method to improve named entity recognition (NER) for chemical compounds using multi-task learning by jointly training a chemical NER model and a chemical compound paraphrase model. Our method enables the long short-term memory (LSTM) of the NER model to capture chemical compound paraphrases by sharing the parameters of the LSTM and character embeddings between the two models. The experimental results on the BioCreative IV's CHEMDNER task show that our method improves chemical NER and achieves state-of-the-art performance (+1.43 F-score).

1 Introduction

Named Entity Recognition (NER) is one of the important basic technologies for Natural Language Processing (NLP) such as Information Extraction and Entity Linking. LSTM-CRF NER models, which combine a conditional random field (CRF) and a long short-term memory (LSTM), have achieved high performance (Lample et al., 2016; Ma and Hovy, 2016). The LSTM-CRF models that use a neural language model (NLM) pre-trained from a large-scale unlabeled corpus (Akbik et al., 2018; Peters et al., 2018) have shown state-of-the-art performances on the CoNLL 2003 shared task (Sang et al., 2003).

Chemical compound databases are widely used for investigating properties of chemical compounds or for developing new chemical products. However, updating the databases by hand is hard because new findings on chemical compounds are mainly reported in scientific papers and patents every day. Hence, chemical NER has been studied to recognize chemical compounds from chemical documents (Leaman et al., 2015; Lu et al., 2015; Lin et al., 2018).

One of the problems in chemical NER is notation variants of chemical compounds. For example, *Phenylalanine* has different notations such as *L-β-phenylalanine* and *(S)-2-Benzylglycine*. The average number of notations of each compound in PubChem[1] is approximately 3.88.[2]

If these expressions are dealt with differently, the statistics of the same compound can be dispersed, especially for low frequency compounds. In other words, the more distinct these representations are, the more difficult identifying chemical compound entities becomes. However, existing chemical NER methods do not deal with notation variants of chemical compounds, derived from the partial structures or the notation fluctuation peculiar to these chemical compounds.

We propose *HanPaNE*, which *Han*dles *Pa*raphrase in *NE*R by utilizing chemical compound paraphrase pairs as multi-task learning.

To train expression identity of different notations, *HanPaNE* learns shared parameters between paraphrases using multi-task learning on NER and paraphrase generation. This contrasts with widely used approaches that automatically augment training data by replacing NEs with other NEs (Yi et al., 2004). To train paraphrase generation, we use an attention-based neural machine translation (ANMT) model (Luong et al., 2015; Bahdanau et al., 2015) that shares parameters with the NER model in the translation encoder. The experimental results on the BioCreative IV 's CHEMDNER task (Krallinger et al., 2015) show that our method achieves the best accuracy (+1.43 F-score).

*Taiki Watanabe belonged to Ehime University when this work was done.

[1] https://pubchem.ncbi.nlm.nih.gov/
[2] We surveyed the average number on May, 2019.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 6244–6249, Hong Kong, China, November 3–7, 2019.

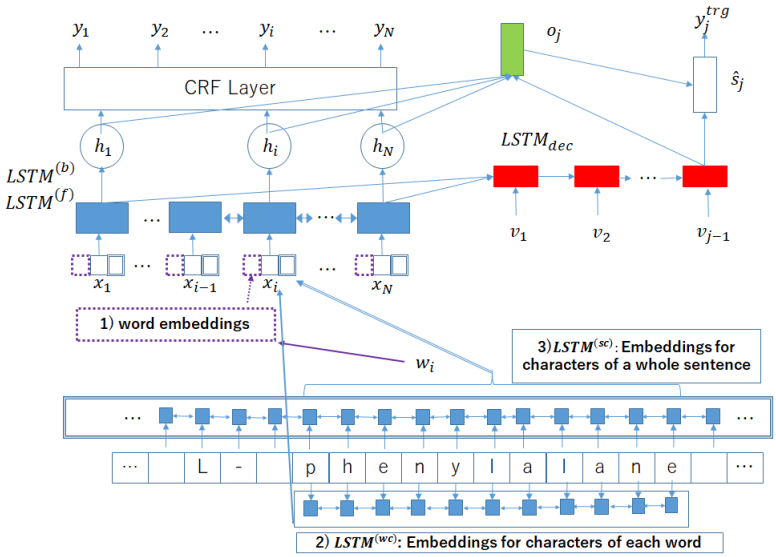

Figure 1: An overview of the method *HanPaNE*. The parameters of $LSTM^{(f)}$, $LSTM^{(b)}$, $LSTM^{(wc)}$ and $LSTM^{(sc)}$ are shared in NER and paraphrase generation.

2 Proposed Method

Figure 1 shows an overview of the method *HanPaNE*. We first describe our NER and chemical compound paraphrase model, and then describe our multi-task learning of the NER and the paraphrase model.

2.1 NER using Character-level NLM

This section describes the NER model using a character-level NLM (Akbik et al., 2018), which is our baseline comparison with our approach. This model showed state-of-the-art performance on the CoNLL 2003 shared task.

For this NER model, given an input sentence $\mathbf{w} = w_1, w_2, \cdots, w_N$, each word w_i is converted into a vector $\mathbf{x}_i$ represented as the concatenation of the following three types of embeddings.

- 1) word embeddings obtained by looking up a parameter matrix.
- 2) CWSE is the hidden state of a BiLSTM ($LSTM^{(wc)}$) on the character sequence consisting of a word (Lample et al., 2016).
- 3) CSE is the hidden state of a BiLSTM ($LSTM^{(sc)}$) on the character sequence consisting of a whole sentence (Akbik et al., 2018).

CWSE for a word is the concatenation of the last hidden states of the forward LSTM and the backward LSTM of $LSTM^{(wc)}$. CSE for a word is the concatenation of the hidden state of the last character of the word for the forward LSTM of $LSTM^{(sc)}$, and the hidden state of the first character of the word for the backward LSTM of $LSTM^{(sc)}$. We note that $LSTM^{(sc)}$ is pretrained as character-level NLMs with log-likelihood. For example, a forward LSTM is trained with the following:

$$P(\mathbf{c}_{1:T}) = \prod_{t=1}^{T} p(c_t|\mathbf{c}_{1:t-1}), \quad (1)$$

where c_t is the t-th character in a sentence.

After obtaining the vector representations of the input words $\mathbf{X} = \mathbf{x}_1, \cdots, \mathbf{x}_N$, the hidden state $\mathbf{h}_i$ of BiLSTM and the scores of NE tags $\mathbf{e}_i$ for each word w_i are computed as:

$$\overrightarrow{\mathbf{h}_i} = LSTM^{(f)}(\mathbf{x}_i, \overrightarrow{\mathbf{h}_{i-1}}), \quad (2)$$
$$\overleftarrow{\mathbf{h}_i} = LSTM^{(b)}(\mathbf{x}_i, \overleftarrow{\mathbf{h}_{i+1}}), \quad (3)$$
$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}], \quad (4)$$
$$\mathbf{e}_i = W_e \mathbf{h}_i, \quad (5)$$

where the dimension of $\mathbf{h}_i$ is d, $W_e \in \mathcal{R}^{k \times d}$ is a weight matrix, and k is the number of NE types to be identified.

The BiLSTM-CRF+CSE model predicts a tag sequence with the score matrix $\mathbf{P} = (\mathbf{e}_1, \cdots, \mathbf{e}_N)^T \in \mathcal{R}^{N \times k}$ using CRF. Note that $P_{i,j}$ is the score of the j-th tag for the i-th word. In particular, the probability of the output tag sequence

$\mathbf{y} = y_1, \cdots, y_N$ is calculated as follows:

$$p(\mathbf{y}|\mathbf{w}) = \frac{e^{s(\mathbf{w},\mathbf{y})}}{\sum_{\tilde{\mathbf{y}}\in\mathbf{Y_w}} e^{s(\mathbf{w},\tilde{\mathbf{y}})}}, \quad (6)$$

$$s(\mathbf{w},\mathbf{y}) = \sum_{i=-1}^{N} A_{y_i,y_{i+1}} + \sum_{i=1}^{N} P_{i,y_i}, \quad (7)$$

where $\mathbf{Y_w}$ is the set of all possible tag sequences, $\mathbf{A}$ is a matrix of transition scores, $A_{i,j}$ represents a score that transits from the i-th tag to the j-th tag, and y_{-1} and y_{N+1} are the special tag for the start of the sentence and the end of sentence, respectively.

During training, the model maximizes the following equation using the correct tag sequences:

$$\log(p(\mathbf{y}|\mathbf{w})) = s(\mathbf{w},\mathbf{y}) - \log(\sum_{\tilde{\mathbf{y}}\in\mathbf{Y_w}} e^{s(\mathbf{w},\tilde{\mathbf{y}})}), \quad (8)$$

where $\mathbf{y}$ is the correct tag sequence of $\mathbf{w}$. When recognizing NEs, the model outputs a tag sequence that maximizes the score calculated by the following equation: $\mathbf{y}^* = \arg\max_{\tilde{\mathbf{y}}\in\mathbf{Y_w}} s(\mathbf{w},\tilde{\mathbf{y}})$.

2.2 Paraphrase Model

We used the ANMT model (Luong et al., 2015; Bahdanau et al., 2015), which is a standard in machine translation, as a chemical compound paraphrase model. The ANMT model converts an input sequence $\mathbf{w}$ into another sequence $\mathbf{y}^{trg} = y_1^{trg}, y_2^{trg}, \cdots, y_T^{trg}$, which is a paraphrase of the input in our model, using an RNN encoder and an RNN decoder. The RNN encoder converts an input sequence to a multiset of fixed-length vectors, and then the RNN decoder generates an output sequence from the converted fixed length vector. We use a bidirectional LSTM defined by Eq. (2) and Eq. (3) as the encoder, and the vector representations $\mathbf{x}_i$ as the word embedding vectors of input sentences. The concatenation of the final hidden states of the bidirectional LSTM encoder, $\overrightarrow{\mathbf{h}}_N$ of Eq. (2) and $\overleftarrow{\mathbf{h}}_1$ of Eq. (3), was used as the initial state of the LSTM decoder: $\mathbf{s}_1 = [\overrightarrow{\mathbf{h}}_N; \overleftarrow{\mathbf{h}}_1]$.

The decoder calculates the j-th hidden state $\mathbf{s}_j$ as follows: $\mathbf{s}_j = LSTM_{dec}([\mathbf{v}_{j-1}; \hat{\mathbf{s}}_{j-1}], \mathbf{s}_{j-1})$, where $\mathbf{v}_{j-1}$ is the word embedding vector of y_{j-1}^{trg}, and $\hat{\mathbf{s}}_{j-1}$ is the $(j-1)$-th attention vector. Then, the model calculates the j-th attention vector $\hat{\mathbf{s}}_j$ using the context vector $\mathbf{o}_j$: $\hat{\mathbf{s}}_j = \tanh(W_e[\mathbf{s}_j : \mathbf{o}_j])$. where W_e is a weight matrix and tanh is the hyperbolic tangent function. The context vector $\mathbf{o}_j$ is a weighted average of the encoder's hidden states:

$$\mathbf{o}_j = \sum_{i=1}^{N} \alpha_j(i)\mathbf{h}_i, \quad (9)$$

$$\alpha_j(i) = \frac{\exp(\mathbf{h}_i \cdot \mathbf{s}_j)}{\sum_{k=1}^{N} \exp(\mathbf{h}_k \cdot \mathbf{s}_j)}, \quad (10)$$

where exp is the natural exponential function. The decoder generates an output based on the probability distribution of the j-th token: $p(y_j^{trg}|\mathbf{y}_{<j}^{trg}, \mathbf{w}) = \mathrm{softmax}(W_s\hat{\mathbf{s}}_j)$, where W_s is a weight matrix. The objective function is defined as follows:

$$J(\theta) = -\sum_{(\mathbf{w},\mathbf{y^{trg}})\in\mathbf{D}} \log p(\mathbf{y}^{trg}|\mathbf{w}), \quad (11)$$

where D is the training data and θ is the set of model parameters.

2.3 Handling Paraphrases in NER

Our method, *HanPaNE*, learns the NER model described in Section 2.1 and the chemical compound paraphrase model described in Section 2.2 at the same time through multi-task learning. In the multi-task learning, the character embedding weight matrix and the LSTM parameters in Eq. (2), Eq. (3) and the parameters of $LSTM^{(wc)}$ are shared between the two models. By the parameter sharing, the LSTM part of the NER model is expected to convert the same compound with different notations into a similar vector expression. The objective functions of the two models are Eq. (8) and Eq. (11), respectively.

3 Experiments

3.1 Experimental Settings

We used the BioCreative IV CHEMDNER data set, which was preprocessed by Luo et al (2018)[3]. The word embedding layer with Word2vec and the character-based NLM were independently pretrained from the MEDLINE abstracts from the PubMed website (PubMedAbs)[4]. We created PubChemDic, a set of paraphrases of chemical compounds compiled from PubChem. The number of dimensions of the word embedding layer, the character embedding layer, the character LSTM, the LSTM of the NER model (the LSTM encoder of the paraphrase model), the LSTM decoder of the paraphrase model, and the LSTM of the character

[3] https://github.com/lingluodlut/Att-ChemdNER

[4] https://www.nlm.nih.gov/databases/download/pubmed_medline.html

Model	Precision	Recall	F-score
Baseline	92.75	92.15	92.45
VE-P	93.11	91.40	92.25
HanPaNE-P	92.71	91.94	92.32
VE+P	**93.15**	91.79	92.47
HanPaNE+P (Proposed)	92.81	**92.33**	**92.57**

Table 1: Experimental results. +P indicates consideration of paraphrases and -P does not.

NLM are set to 100, 25, 50, 200, 400, and 2048, respectively.

We compared the following methods. "+P" indicates with consideration of paraphrases in PubChemDic and "-P" indicates without consideration of paraphrases.

- A BiLSTM-CRF+CSE of (Akbik et al., 2018) described in Section 2.1 was used as our Baseline.
- VE-P is a baseline trained with virtual examples (VEs) created by randomly replacing NEs of training data with chemical compounds in the PubChemDic similar to (Yi et al., 2004).
- VE+P is a baseline trained with VEs created by replacing NEs of training data with their corresponding paraphrases in the PubChemDic.
- *HanPaNE*-P is a multi-task for NER and paraphrasing trained with randomly generated sentence pairs with PubChemDic.
- *HanPaNE*+P is the proposed method trained with generated sentence pairs by replacing NEs in sentences with their corresponding paraphrases in the PubChemDic.

The baseline was trained with 55,458 sentences of CHEMDNER training data. VE-P was trained with 110,916 sentences in total consisting of the original CHEMDNER training data and 55,458 sentences automatically generated from the CHEMDNER training data. VE+P was trained with 59,033 sentences consisting of 3,575 sentences automatically generated from the original CHEMDNER training data with paraphrases of chemical compounds and the original CHEMDNER training data.[5] For *HanPaNE*-P and *HanPaNE*+P, we used randomly selected 100,000 sentences from PubMedAbs for training paraphrasing and the original CHEMDNER training data for NER.

For example, for +P, "... *Phenylalanine* is ..." are converted into "... *L-β-phenylalanine* is ..." and ".. *(S)-2-Benzylglycine* is ...", where *L-β-phenylalanine* and *(S)-2-Benzylglycine* are paraphrases of *Phenylalanine*. As for -P, "... *Phenylalanine* is ..." is converted into like "... *ethylene* is ...." where *ethylene* is randomly sampled from chemical compounds.

3.2 Experimental Results

Table 1 shows the experimental results. We can see that *HanPaNE*+P showed the highest accuracy and *HanPaNE*+P and VE+P, with consideration of paraphrases, showed a higher accuracy than Baseline. In contrast, *HanPaNE*-P and VE-P, without consideration of paraphrases, did not. The results indicate that the use of paraphrases contributed to improved accuracy.

We also conducted the following two types of hypothesis testing. The first one is a McNemar paired test on the labeling disagreements of words assigned by *HanPaNE* and the others as in (Sha and Pereira, 2003). All the results except for Baseline were significantly different ($p < 0.01$). The second one is a bi-nominal test used in (Sasano and Kurohashi, 2008). For this test, the number of the entities correctly recognized by only *HanPaNE* and the number of entities correctly recognized by only the other method are counted. Then, based on the assumption that outputs have the binomial distribution, we apply a binomial test. All the results were significantly different for this test ($p < 0.05$). These results showed that *HanPaNE* works better than augmented training data.

We also evaluated the accuracy on NEs not covered by training data and PubChemDic. From Table 2, we can see that our method also showed the best performance for both of the NEs not covered by the training data, PubChemDic and the covered NEs. The results indicate that our method contributed to recognizing NEs not covered by existing large data sets. The training data includes 8,508 types of chemical terms and covers 60.76% of chemical terms in the test data. PubChemDic includes 337,289,536 types of chemical compound names and covers 10.02% of the chem-

[5] VE+P had a smaller number of training data than VE-P because we only replaced NEs with their paraphrases included in PubChemDic.

Model	Acc.NC	Acc.C
Baseline	0.8475	0.9672
VE+P	0.8463	0.9623
HanPaNE+P (Proposed)	**0.8510**	**0.9680**

Table 2: Accuracy of NEs covered (Acc.C) or not covered (Acc. NC) by the training data and PubChemDic.

Table 3: Comparison with best results on BioCreative IV's CHEMDNER task.

Method	Precision	Recall	F-score
Leaman et al. (2015)	89.09	85.75	87.39
Lu et al. (2015)	88.73	87.41	88.06
Lin et al. (2018)	92.29	90.01	91.14
This paper	**92.81**	**92.33**	**92.57**

ical terms in the test data. The joint coverage of both is 61.79%. This means almost 1/3 of the NEs in the test data are not covered by the training data and PubChemDic even if these data include over 337 million names of chemical compounds and terms.

4 Related Works

BioCreative IV's CHEMDNER task

Table 3 shows a comparison with the previous best results. (Leaman et al., 2015) and (Lu et al., 2015) proposed a feature-based approach to improve the chemical NER performance. (Lin et al., 2018) proposed a neural network approach that treats document level information to maintain tagging consistency across sentences. By learning paraphrasing, our method showed the best accuracy on the CHEMDNER task.

Multi-task learning

Multi-task learning is often utilized to leverage the performance of NLP systems (Liu et al., 2015; Luong et al., 2016; Dong et al., 2015; Hashimoto et al., 2017), including NER. Liu et al. (2018) and Rei (2017) studied multi-task learning of sequence labeling with language models. Aguilar et al. (2018) and Cao et al. (2018) proposed multi-task learning of NER with word segmentation. Peng and Dredze (2017)'s method of multi-task learning leverages the performance of domain adaptation. Clark et al. (2018)'s method utilizes multi-task learning of NER with several NLP tasks such as POS tagging and parsing. Crichton et al. (2017) and Wang et al. (2018) leverage the performance of NER by multi-task learning of several tasks of biomedical NLP.

5 Conclusions

We proposed a multi-task learning method on an NER model and a paraphrase model to utilize paraphrase pairs efficiently. The evaluations on the BioCreative IV's CHEMDNER task showed that our method achieved the best performance.

References

Gustavo Aguilar, Adrian Pastor López Monroy, Fabio González, and Thamar Solorio. 2018. Modeling noisiness to recognize named entities using multi-task neural networks on social media. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1401–1412.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th COLING*, pages 1638–1649.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd ICLR*.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925. Association for Computational Linguistics.

Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. 2017. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd ACL and the 7th IJCNLP (Volume 1: Long Papers)*, pages 1723–1732.

Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 EMNLP*, pages 1923–1933.

Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. 2015. Chemdner: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, 7(1):S1.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 NAACL-HLT*, pages 260–270.

Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. 2015. tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of Cheminformatics*, 7(1):S3.

Hongfei Lin, Jian Wang, Ling Luo, Pei Yang, Zhihao Yang, Yin Zhang, and Lei Wang. 2018. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, pages 1381–1388.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 NAACL-HLT*, pages 912–921.

Yanan Lu, Donghong Ji, Xiaoyuan Yao, Xiaomei Wei, and Xiaohui Liang. 2015. Chemdner system with mixed conditional random fields and multi-scale word clustering. *Journal of Cheminformatics*, 7(1):S4.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 EMNLP*, pages 1412–1421.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th ACL*, pages 1064–1074.

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 NAACL-HLT*, pages 2227–2237.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130. Association for Computational Linguistics.

Tjong Kim Sang, Erik F., and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

Ryohei Sasano and Sadao Kurohashi. 2008. Japanese named entity recognition using structural natural language processing. In *Proc of . IJCNLP'08*, pages 607–612.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. of NAACL HLT'03*, pages 134–141.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, page bty869.

Eunji Yi, Gary Geunbae Lee, Yu Song, and Soo-Jun Park. 2004. Svm-based biological named entity recognition using minimum edit-distance feature boosted by virtual examples. In *IJCNLP'04*, pages 807–814.