Named Entity Recognition for Social Media Texts with Semantic Augmentation

Yuyang Nie[♢*], **Yuanhe Tian**[♥*], **Xiang Wan**[♡], **Yan Song**[♠♡†], **Bo Dai**[♢]
[♢]University of Electronic Science and Technology of China
[♥]University of Washington [♡]Shenzhen Research Institute of Big Data
[♠]The Chinese University of Hong Kong (Shenzhen)
[♢]nyy207@gmail.com [♥]yhtian@uw.edu [♡]wanxiang@sribd.cn
[♠]songyan@cuhk.edu.cn [♢]daibo@uestc.edu.cn

Abstract

Existing approaches for named entity recognition suffer from data sparsity problems when conducted on short and informal texts, especially user-generated social media content. Semantic augmentation is a potential way to alleviate this problem. Given that rich semantic information is implicitly preserved in pre-trained word embeddings, they are potential ideal resources for semantic augmentation. In this paper, we propose a neural-based approach to NER for social media texts where both local (from running text) and augmented semantics are taken into account. In particular, we obtain the augmented semantic information from a large-scale corpus, and propose an attentive semantic augmentation module and a gate module to encode and aggregate such information, respectively. Extensive experiments are performed on three benchmark datasets collected from English and Chinese social media platforms, where the results demonstrate the superiority of our approach to previous studies across all three datasets.[1]

1 Introduction

The increasing popularity of microblogs results in a large amount of user-generated data, in which texts are usually short and informal. How to effectively understand these texts remains a challenging task since the insights are hidden in unstructured forms of social media posts. Thus, named entity recognition (NER) is a critical step for detecting proper entities in texts and providing support for downstream natural language processing (NLP) tasks (Pang et al., 2019; Martins et al., 2019).

However, NER in social media remains a challenging task because (i) it suffers from the data sparsity problem since entities usually represent a small part of proper names, which makes the task hard to be generalized; (ii) social media texts do not follow strict syntactic rules (Ritter et al., 2011). To tackle these challenges, previous studies tired to leverage domain information (e.g., existing gazetteer and embeddings trained on large social media text) and external features (e.g., part-of-speech tags) to help with social media NER (Peng and Dredze, 2015; Aguilar et al., 2017). However, these approaches rely on extra efforts to obtain such extra information and suffer from noise in the resulted information. For example, training embeddings for social media domain could bring a lot unusual expressions to the vocabulary. Inspired by studies using semantic augmentation (especially from lexical semantics) to improve model performance on many NLP tasks (Song and Xia, 2013; Song et al., 2018a; Kumar et al., 2019; Amjad et al., 2020), it is also a potential promising solution to solving social media NER. Figure 1 shows a typical case. "*Chris*", supposedly tagged with "*Person*" in this example sentence, is tagged as other labels in most cases. Therefore, in the predicting process, it is difficult to label "*Chris*" correctly. A sound solution is to augment the semantic space of "*Chris*" through its similar words, such as "*Jason*" and "*Mike*", which can be obtained by existing pre-trained word embeddings from the general domain.

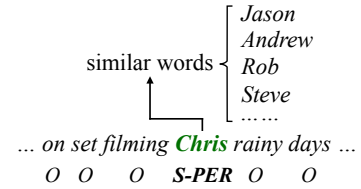

Figure 1: An example shows that an NE tagged with "*PER*" (Person) is suggested by its similar words.

In this paper, we propose an effective approach to NER for social media texts with semantic augmentation. In doing so, we augment the semantic

[*]Equal contribution.
[†]Corresponding author.
[1]The code and the best performing models are available at https://github.com/cuhksz-nlp/SANER.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1383–1391,
November 16–20, 2020.

space for each token from pre-trained word embedding models, such as GloVe (Pennington et al., 2014) and Tencent Embedding (Song et al., 2018b), and encode semantic information through an attentive semantic augmentation module. Then we apply a gate module to weigh the contribution of the augmentation module and context encoding module in the NER process. To further improve NER performance, we also attempt multiple types of pre-trained word embeddings for feature extraction, which has been demonstrated to be effective in previous studies (Akbik et al., 2018; Jie and Lu, 2019; Kasai et al., 2019; Kim et al., 2019; Yan et al., 2019). To evaluate our approach, we conduct experiments on three benchmark datasets, where the results show that our model outperforms the state-of-the-arts with clear advantage across all datasets.

2 The Proposed Model

The task of social media NER is conventionally regarded as sequence labeling task, where an input sequence $\mathcal{X} = x_1, x_2, \cdots, x_n$ with n tokens is annotated with its corresponding NE labels $\widehat{\mathcal{Y}} = \widehat{y}_1, \widehat{y}_2, \cdots, \widehat{y}_n$ in the same length. Following this paradigm, we propose a neural model with semantic augmentation for the social media NER. Figure 2 shows the architecture of our model, where the backbone model and the semantic augmentation module are illustrated in white and yellow backgrounds, respectively. For each token in the input sentence, we firstly extract the most similar words of the token according to their pre-trained embeddings. Then, the augmentation module use an attention mechanism to weight the semantic information carried by the extracted words. Afterwards, the weighted semantic information is leveraged to enhance the backbone model through a gate module.

In the following text, we firstly introduce the encoding procedure for augmenting semantic information. Then, we present the gate module to incorporate augmented information into the backbone model. Finally, we elaborate the tagging procedure for NER with the aforementioned enhancement.

2.1 Attentive Semantic Augmentation

The high quality of text representation is the key to obtain good model performance for many NLP tasks (Song et al., 2017; Sileo et al., 2019). However, obtaining such text representation is not easy in the social media domain because of data sparsity problem. Motivated by this fact, we propose semantic augmentation mechanism for social media NER by enhancing the representation of each token in the input sentence with the most similar words in their semantic space, which can be measured by pre-trained embeddings.

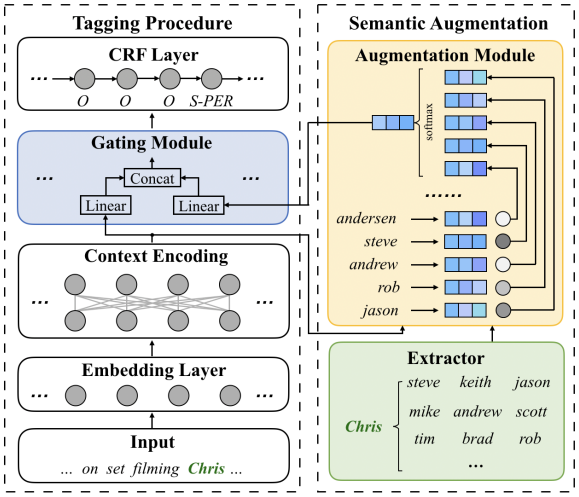

Figure 2: The overall architecture of our proposed model with semantic augmentation. An example sentence and its output NE labels are given, where the augmented semantic information for the word "*Chris*" are also illustrated with the processing through the augmentation module and the gate module.

In doing so, for each token $x_i \in \mathcal{X}$, we use pre-trained word embeddings (e.g., GloVe for English and Tencent Embedding for Chinese) to extract the top m words that are most similar to x_i based on cosine similarities and denote them as

$$C_i = \{c_{i,1}, c_{i,2}, \cdots, c_{i,j}, \cdots, c_{i,m}\} \quad (1)$$

Afterwards, we use another embedding matrix to map all extracted words $c_{i,j}$ to their corresponding embeddings $\mathbf{e}_{i,j}$. Since not all $c_{i,j} \in C_i$ are helpful for predicting the NE label of x_i in the given context, it is important to distinguish the contributions of different words to the NER task in that context. Consider that the attention and weight based approaches are demonstrated to be effective choices to selectively leverage extra information in many tasks (Kumar et al., 2018; Margatina et al., 2019; Tian et al., 2020a,d,b,c), we propose an attentive semantic augmentation module (denoted as AU) to weight the words according to their contributions to the task in different contexts. Specifically, for each token x_i, the augmentation module assigns a weight to each word $c_{i,j} \in C_i$ by

$$p_{i,j} = \frac{exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j})}{\sum_{j=i}^{m} exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j})}, \quad (2)$$

where $\mathbf{h}_i$ is the hidden vector for x_i obtained from

the context encoder with its dimension matching that of the embedding (i.e., $\mathbf{e}_{i,j}$) of $c_{i,j}$. Then, we apply the weight $p_{i,j}$ to the word $c_{i,j}$ to compute the final augmented semantic representation by

$$\mathbf{v}_i = \sum_{j=1}^{m} p_{i,j}\mathbf{e}_{i,j}, \quad (3)$$

where $\mathbf{v}_i$ is the derived output of AU, and contains the weighted semantic information. Therefore, the augmentation module ensures that the augmented semantic information are weighted based on their contributions and important semantic information is distinguished accordingly.

2.2 The Gate Module

We observe that the contribution of the obtained augmented semantic information to the NER task could vary in different contexts and a gate module (denoted by GA) is naturally desired to weight such information in the varying contexts. Therefore, to improve the capability of NER with the semantic information, we propose a gate module to aggregate such information to the backbone NER model. Particularly, we use a $reset$ gate to control the information flow by

$$\mathbf{g} = \sigma(\mathbf{W}_1 \cdot \mathbf{h}_i + \mathbf{W}_2 \cdot \mathbf{v}_i + \mathbf{b}_g), \quad (4)$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable matrices and $\mathbf{b}_g$ the corresponding bias term. Afterwards, we use

$$\mathbf{u}_i = [\mathbf{g} \circ \mathbf{h}_i] \oplus [(\mathbf{1} - \mathbf{g}) \circ \mathbf{v}_i] \quad (5)$$

to balance the information from context encoder and the augmentation module, where $\mathbf{u}_i$ is the derived output of the gate module; $\circ$ represents the element-wise multiplication operation and $\mathbf{1}$ is a 1-vector with its all elements equal to 1.

2.3 Tagging Procedure

To provide $\mathbf{h}_i$ to the augmentation module, we adopt a context encoding module (denoted as CE) proposed by Yan et al. (2019). Compared with vanilla Transformers, this encoder additionally models the direction and distance information of the input, which has been demonstrated to be useful for the NER task. Therefore, the encoding procedure of the input text can be denoted as

$$\mathbf{H} = CE(\mathbf{E}), \quad (6)$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n]$ and $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n]$ are lists of hidden vectors and embeddings of $\mathcal{X}$, respectively. In addition, since pre-trained word embeddings contain substantial context information from large-scale corpus, and different types of them may contain diverse information, a straightforward way of incorporating them is to concatenate their embedding vectors by

$$\mathbf{e}_i = \mathbf{e}_i^1 \oplus \mathbf{e}_i^2 \oplus \ldots \oplus \mathbf{e}_i^T, \quad (7)$$

where $\mathbf{e}_i$ is the final word embedding for x_i and T the set of all embedding types. Afterwards, a trainable matrix $\mathbf{W}_u$ is used to map $\mathbf{u}_i$ obtained from the gate module to the output space by $\mathbf{o}_i = \mathbf{W}_u \cdot \mathbf{u}_i$. Finally, a conditional random field (CRF) decoder is applied to predict the labels $\widehat{y}_i \in L$ (where L is the set with all NE labels) in the output sequence $\widehat{\mathcal{Y}}$ by

$$\widehat{y}_i = \arg\max_{y_i \in L} \frac{exp(\mathbf{W}_c \cdot \mathbf{o}_i + \mathbf{b}_c)}{\sum_{y_{i-1}y_i} exp(\mathbf{W}_c \cdot \mathbf{o}_i + \mathbf{b}_c)}, \quad (8)$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are the trainable parameters to model the transition for y_{i-1} to y_i.

Language	Dataset		Train	Dev	Test
English	W16	#Sent.	2,394	1,000	3,850
		#Ent.	1,496	661	3,473
		%Uns.	-	52.1	80.0
	W17	#Sent.	3,394	1,008	1,287
		#Ent.	1,975	835	1,079
		%Uns.	-	34.8	84.5
Chinese	WB	#Sent.	1,350	270	270
		#Ent.	1,885	389	414
		%Uns.	-	51.4	45.2

Table 1: The statistics of all benchmark datasets w.r.t. the number of sentences (# Sent.), named entities (# Ent.) and the percentage of unseen entities (% Uns.).

3 Experiments

3.1 Settings

In our experiments, we use three social media benchmark datasets, including WNUT16 (W16) (Strauss et al., 2016), WNUT17 (W17) (Derczynski et al., 2017), and Weibo (WB) (Peng and Dredze, 2015), where W16 and W17 are English datasets constructed from Twitter, and WB is built from Chinese social media platform (Sina Weibo). For all three datasets, we use their original splits and report the statistics of them in Table 1 (e.g., the number of sentences (#Sent.), entities (#Ent.), and the percentage of unseen entities (%Uns.) with respect to the entities appearing in the training set).

For model implementation, we follow Lample et al. (2016) to use the BIOES tag schema to represent the NE labels of tokens in the input sentence. For the text input, we try two types of embeddings

ID	*SE*	*GA*	W16	W17	WB
1	N	N	54.79	48.41	65.36
2	DS	N	55.03	48.36	65.01
3	DS	Y	56.28	48.98	66.24
4	AU	N	56.86	49.26	68.21
5	AU	Y	**57.94**	**50.02**	**69.32**

(a) Development Set

ID	*SE*	*GA*	W16	W17	WB
1	N	N	52.98	48.82	66.02
2	DS	N	53.11	48.71	65.78
3	DS	Y	54.02	49.56	67.52
4	AU	N	54.29	49.81	68.46
5	AU	Y	**55.01**	**50.36**	**69.80**

(b) Test Set

Table 2: $F1$ scores of the baseline model and ours enhanced with semantic augmentation ("SE") and the gate module ("GA") on the development (a) and test (b) sets. "DS" and "AU" represent the direct summation and attentive augmentation module, respectively. Y and N denote the use and non-use of corresponding modules.

for each language.[2] Specifically, for English, we use ELMo (Peters et al., 2018) and BERT-cased large (Devlin et al., 2019); for Chinese, we use Tencent Embedding (Song et al., 2018b), and ZEN (Diao et al., 2019).[3] In the context encoding module, we use a two-layer transformer-based encoder proposed by Yan et al. (2019) with 128 hidden units and 12 heads. To extract similar words carrying augmented semantic information, we use the pre-trained word embeddings from GloVe for English and those embedding from Tencent Embeddings for Chinese to extract the most similar 10 words (i.e., $m = 10$) [4]. In the augmentation module, we randomly initialize the embeddings of the extracted words (i.e., $\mathbf{e}_{i,j}$ for $c_{i,j}$) to represent the semantic information carried by those words.[5]

During the training process, we fix all pre-trained embeddings in the embedding layer and use Adam (Kingma and Ba, 2015) to optimize negative log-likelihood loss function with the learning rate set to $\eta = 0.0001$, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We train 50 epochs for each method with the batch size set to 32 and tune the hyper-parameters on the development set[6]. The model that achieves the best performance on the development set is evaluated on the test set with the $F1$ scores obtained from the official conlleval toolkits[7].

[2]We report the results of using each individual type of embeddings in Appendix A.

[3]We obtain the pre-trained BERT model from https://github.com/google-research/bert, Tencent Embeddings from https://ai.tencent.com/ailab/nlp/embedding.html, and ZEN from https://github.com/sinovation/ZEN. Note that we use ZEN because it achieves better performance than BERT on different Chinese NLP tasks. For reference, we report the results of using BERT in Appendix B.

[4]The results of using other embeddings as sources to extract similar words are reported in the Appendix C.

[5]We also try other ways (e.g., GloVe for English and Tencent Embedding for Chinese) to initialize the word embeddings, but do not find significant differences.

[6]We report the details of hyperparameter settings of different models in the Appendix D.

[7]The script to evaluate all models in the experiments is obtained from https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt.

3.2 Overall Results

To explore the effect of the proposed attentive semantic augmentation module (AU) and the gate module (GA), we run different settings of our model with and without the modules. In addition, we also try baselines that use direct summation (DS) to leverage the semantic information carried by the similar words, where the embeddings of the words are directly summed without weighting through attentions. The experimental results ($F1$) of the baselines and our approach on the development and test sets of all datasets are reported in Table 2(a) and (b), respectively.

There are some observations from the results on the development and test sets. First, compared to the baseline without semantic augmentation (ID=1), models using direct summation (DS, ID=2) to incorporate different semantic information undermines NER performance on two of three datasets, namely, W17 and WB; on the contrary, the models using the proposed attentive semantic augmentation module (AU, ID=4) consistently outperform the baselines (ID=1 and ID=2) on all datasets. It indicates that AU could distinguish the contributions of different semantic information carried by different words in the given context and leverage them accordingly to improve NER performance. Second, comparing the results of models with and without the gate module (GA) (i.e. ID=3 vs. ID=2 and ID=5 vs. ID=4), we find that the models with gate module achieves superior performance to the others without it. This observation suggests that the importance of the information from the context encoder and AU varies, and the proposed gate module is effective in adjusting the weights according to their contributions.

Moreover, we compare our model under the best setting with previous models on all three datasets in Table 3, where our model outperforms others on all datasets. We believe that the new state-of-the-

Model	W16	W17	WB
Zhang and Yang (2018)	-	-	58.79
Yan et al. (2019)	54.06	48.98	65.03
Zhu and Wang (2019)	-	-	59.31
Gui et al. (2019)	-	-	59.92
Sui et al. (2019)	-	-	63.09
Akbik et al. (2019)	-	49.59	-
Zhou et al. (2019)	53.43	42.83	-
Devlin et al. (2019)	54.36	49.52	67.33
Meng et al. (2019)	-	-	67.60
Xu et al. (2019)	-	-	68.93
Ours	**55.01**	**50.36**	**69.80**

Table 3: Comparison of $F1$ scores of our best performing model (the full model with augmentation module and gate module) with that reported in previous studies on all English and Chinese social media datasets.

art performance is established. The reason could be that compared to previous studies, our model is effective to alleviate the data sparsity problem in social media NER with the augmentation module to encode augmented semantic information. Besides, the gate module can distinguish the importance of information from the context encoder and AU according to their contribution to NER.

4 Analysis

4.1 Performance on Unseen Named Entities

Since this work focuses on addressing the data sparsity problem in social media NER, where the unseen NEs are one of the important factors that hurts model performance. To analyze whether our approach with attentive semantic augmentation (AU) and the gate module (GA) can address this problem, we report the recall of our approach (i.e., "+AU+GA") to recognize the unseen NEs on the test set of all datasets in Table 4. For reference, we also report the recall of the baseline without AU and GA, as well as our runs of previous studies (marked by "*"). It is clearly observed that our approach outperforms the baseline and previous studies on unseen NEs on all datasets, which shows that it can appropriately leverage semantic information carried by similar words and thus alleviate the data sparsity problem.

4.2 Case Study

To demonstrate how the augmented semantic information improves NER with the attentive augmentation module and the gate module, we show the extracted augmented information for the word "*Chris*" and visualize the weights for each augmented term in Figure 3, where deeper color refers to higher

Model	W16	W17	WB
# of Unseen NEs	2778	912	189
*Devlin et al. (2019)	49.02	46.73	45.98
*Yan et al. (2019)	48.97	46.89	45.71
Baseline	49.04	46.72	45.79
Ours (+AU +GA)	**51.27**	**49.45**	**48.81**

Table 4: The recall of our models with and without the attentive semantic augmentation (AU) and the gate module (GA) on unseen named entities (whose numbers are also reported at the first row) on all three datasets. The results of our runs of previous models (marked with "*") are also reported for references.

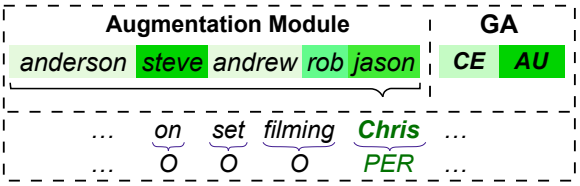

Figure 3: An example of helping recognize the NE "*Chris*" by augmented semantic information (darker color refers to greater value). "CE" and "AU" represent the context encoder and attentive augmentation module, respectively.

weight. In this case, the words "*steve*" and "*jason*" have higher weights in AU. The explanation could be that in all cases, these two words are a kind of "*Person*". Thus, higher attention to these terms helps our model to identify the correct NE label. On the contrary, the term "*anderson*" and "*andrew*" never occur in the dataset, and therefore provide no helpful effect in this case and eventually end with the lower weights in AU. In addition, a model can also mislabel "*Chris*" as "*Music-Artist*", because "*Chris*" belongs to that NE type in most cases and there is a word "*filming*" in its context. However, our model with the gate module can distinguish that the information from semantic augmentation is more important and thus make correct prediction.

5 Conclusion

In this paper, we proposed a neural-based approach to enhance social media NER with semantic augmentation to alleviate data sparsity problem. Particularly, an attentive semantic augmentation module is suggested to encode semantic information and a gate module is applied to aggregate such information to tagging process. Experiments conducted on three benchmark datasets in English and Chinese show that our model outperforms previous studies and achieves the new state-of-the-art result.

References

Gustavo Aguilar, Suraj Maharjan, Adrián 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, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 148–153.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled Contextualized Embeddings for Named Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649.

Maaz Amjad, Grigori Sidorov, and Alisa Zhila. 2020. Data Augmentation using Machine Translation for Fake News Detection in the Urdu Ulanguage. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2537–2542, Marseille, France.

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, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2019. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. *Arxiv*, abs/1911.00720.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019. CNN-Based Chinese NER with Lexicon Rethinking. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4982–4988.

Zhanming Jie and Wei Lu. 2019. Dependency-Guided LSTM-CRF for Named Entity Recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3860–3870.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir R. Radev, and Owen Rambow. 2019. Syntax-aware Neural Semantic Role Labeling with Supertags. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 701–709.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic Sentence Matching with Densely-Connected Recurrent and Co-Attentive Information. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6586–6593.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Abhishek Kumar, Daisuke Kawahara, and Sadao Kurohashi. 2018. Knowledge-Enriched Two-Layered Attention Network for Sentiment Analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 253–258, New Orleans, Louisiana.

Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and Wlliam Campbell. 2019. A Closer Look at Feature Space Data Augmentation for Few-Shot Intent Classification. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 1–10, Hong Kong, China.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford Corenlp Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60.

Katerina Margatina, Christos Baziotis, and Alexandros Potamianos. 2019. Attention-based Conditioning Methods for External Knowledge Integration. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3944–3951, Florence, Italy.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2019. Joint Learning of Named Entity Recognition and Entity Linking. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*, pages 190–196.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for Chinese Character Representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 2742–2753.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Lixin Su, and Xueqi Cheng. 2019. HAS-QA: Hierarchical Answer Spans Model for Open-Domain Question Answering. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6875–6882.

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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 548–554.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1524–1534.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining Discourse Markers for Unsupervised Sentence Representation Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song, Shuming Shi, and Jing Li. 2018a. Joint Learning Embeddings for Chinese Words and their Components via Ladder Structured Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4375–4381.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018b. Directional Skip-Gram: Explicitly Distinguishing Left and Right Context for Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 175–180.

Yan Song and Fei Xia. 2013. A Common Case of Jekyll and Hyde: The Synergistic Effect of Using Divided Source Training Data for Feature Augmentation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 623–631, Nagoya, Japan.

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, NUT@COLING 2016, Osaka, Japan, December 11, 2016*, pages 138–144.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3828–3838.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint Chinese Word Segmentation and Part-of-speech Tagging via Two-way Attentions of Auto-analyzed Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Supertagging Combinatory Categorial Grammar with Attentive Graph Convolutional Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuanhe Tian, Yan Song, Fei Xia, and Tong Zhang. 2020c. Improving Constituency Parsing with Span Attention. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020d. Improving Chinese Word Segmentation with Wordhood Memory Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285.

Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting Multiple Embeddings for Chinese Named Entity Recognition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2269–2272.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv*, abs/1911.04474.

Yue Zhang and Jie Yang. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1554–1564.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual Adversarial Neural Transfer for Low-Resource Named Entity Recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3461–3471.

Yuying Zhu and Guoxin Wang. 2019. CAN-NER: Convolutional Attention Network for Chinese Named Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3384–3393.

Appendix A: Effect of Using Different Embeddings

Model	Emb.	W16	W17	WB
Baseline	ELMo	52.16	47.31	-
+AU+GA		54.31	48.76	-
Baseline	BERT	52.09	48.33	-
+AU+GA		**54.16**	**49.57**	-
Baseline	Tencent	-	-	60.54
+AU+GA		-	-	63.12
Baseline	ZEN	-	-	66.09
+AU+GA		-	-	**68.96**

Table 5: Experimental results ($F1$ scores) of our approach with semantic augmentation (AU) and gate module (GA) on all datasets, where only one type of embeddings is used in the embedding layer to represent the input sentence. The results of their corresponding baseline without AU and GA are also reported.

In our main experiments, we use two types of embeddings for each language: ELMo (Peters et al., 2018) and BERT-cased large (Devlin et al., 2019) for English, and Tencent Embedding (Song et al., 2018b) and ZEN (Diao et al., 2019) for Chinese. In Table 5, we report the results ($F1$ scores) of our model with the best setting (i.e. the full model with semantic augmentation (AU) and gate module (GA)) as well as the baselines without AU and GA, where either one of the two types of embedding is used to represent the input sentence. From the results, it is found that our model with AU and GA can consistently outperforms the baseline models with different settings of embeddings.

Appendix B: Comparison Between BERT and ZEN on WB

Embeddings	WB
BERT + Tencent	69.56
ZEN + Tencent	**69.80**

Table 6: Experimental results ($F1$ scores) of our model with AU and GA on the WB dataset, where BERT or ZEN is used as one of the two types of embeddings (the other one is Tencent Embedding) to represent the input sentence for the embedding layer.

In our main experiments, we use ZEN (Diao et al., 2019) instead of BERT (Devlin et al., 2019) as the embedding to represent the input for Chinese. The reason is that ZEN achieves better performance compared with BERT, which is confirmed by Table 6 with its results ($F1$ scores) showing the performance of our approach with the best settings (i.e. two types of embeddings with AU and GA.) on

WB dataset. Either BERT or ZEN is used as one of the two types of embeddings (the other type of embedding is Tencent Embedding).

Appendix C: Effect of Using Different Embeddings to Extract Similar Words

Model	Source	W16	W17	WB
Baseline		49.56	49.11	66.02
Ours (+AU+GA)	Word2vec	54.94	50.22	-
	GloVe	**55.01**	**50.36**	-
	Giga	-	-	69.68
	Tencent	-	-	**69.80**

Table 7: Experimental results ($F1$ scores) of our best performing models (i.e., the ones with AU and GA) using different types of pre-trained embeddings as the source to extract similar words. The results of baseline (the one without AU and GA) are also reported.

In addition to use embeddings for input sentence representation, we also try different embedding sources (i.e. pre-trained word embeddings) to extract similar words for each token in the input sentence. For English, we use Word2vec (Mikolov et al., 2013) and Glove (Manning et al., 2014); for Chinese, we use Giga (Zhang and Yang, 2018) and Tencent Embedding (Song et al., 2018b).[8] The experimental results of our model with the best setting (i.e., the one with AU and GA) using different sources are reported in Table 7. The result of the baseline model without AU and GA is also reported for reference. The results show that our approach can consistently outperforms the baseline with different sources to find similar words, which demonstrates the robustness of our approach.

Appendix D: Hyper-parameter Settings

	Values	Best
Dropout rate	0, 0.1, 0.2, 0.3	0.2
Learning rate	e^{-5}, e^{-4}, e^{-3}	e^{-4}
Batch size	8, 16, 32	32
Number of layers	1, 2, 4	2
Number of head	4, 8, 12	12
Hidden units	64, 128, 256	128
# of similar of words (m)	5, 10, 20	10

Table 8: All values of different hyper-parameters as well as the best one used in our experiments.

We report all values of the hyper-parameters tried for our models in Table 8, where we try different combinations of them and find the best hyper-parameter configurations (which is also reported in Table 8) on the development set of each dataset.

[8] We obtain Word2vec from https://code.google.com/archive/p/word2vec/, GloVe from https://nlp.stanford.edu/projects/glove/, Giga from https://github.com/jiesutd/LatticeLSTM.