SSMI: Semantic Similarity and Mutual Information Maximization Based Enhancement for Chinese NER

Pengnian Qi, Biao Qin*

School of Information, Renmin University of China, Beijing, China
{pengnianqi,qinbiao}@ruc.edu.cn

Abstract

The Chinese NER task consists of two steps, first determining entity boundaries and then labeling them. Some previous work incorporating related words from pre-trained vocabulary into character-based models has demonstrated to be effective. However, the number of words that characters can match in the vocabulary is large, and their meanings vary widely. It is unreasonable to concatenate all the matched words into the character's representation without making semantic distinctions. This is because words with different semantics also have distinct vectors by the distributed representation. Moreover, mutual information maximization (MIM) provides a unified way to characterize the correlation between different granularity of embeddings. We find MIM can be used to enhance the features in our task. Consequently, this paper introduces a novel Chinese NER model named *SSMI* based on semantic similarity and MIM. We first match all the potential word boundaries of the input characters from the pre-trained vocabulary and employ BERT to segment the input sentence to get the segmentation containing these characters. After computing their semantic similarity, we obtain the word boundary with the highest similarity and the word group with similarity score larger than a specific threshold. Then, we concatenate the most relevant word boundaries with character vectors. We further calculate the mutual information maximization of group, character and sentence, respectively. Finally, we feed the result from the above steps to our novel network. The results on four Chinese public NER datasets show that our *SSMI* achieves state-of-the-art performance.

Introduction

Named Entity Recognition (NER) aims to automatically recognize named entities in plain text. In English NER, LSTM-CRF models (Lample et al. 2016; Ma and Hovy 2016; Liu et al. 2018; Chiu and Nichols 2016; Huang, Xu, and Yu 2015) have achieved state-of-the-art results by integrating character information into word representations. Compared with English NER, Chinese NER is more complicated because it has no obvious word boundaries. One intuitive way is to first perform word segmentation using Chinese word segmentation tools and then employ the word-based methods (Jie et al. 2016; He and Sun 2017b). However, such methods suffer from error propagation because named entities may encounter out-of-vocabulary problems in segmentation. Consequently, some works that applied the character-based methods have been empirically proven to be effective (He and Wang 2008; Liu, Zhu, and Zhao 2010; Li et al. 2014; Sui et al. 2019; Ding et al. 2019; Liu et al. 2019).

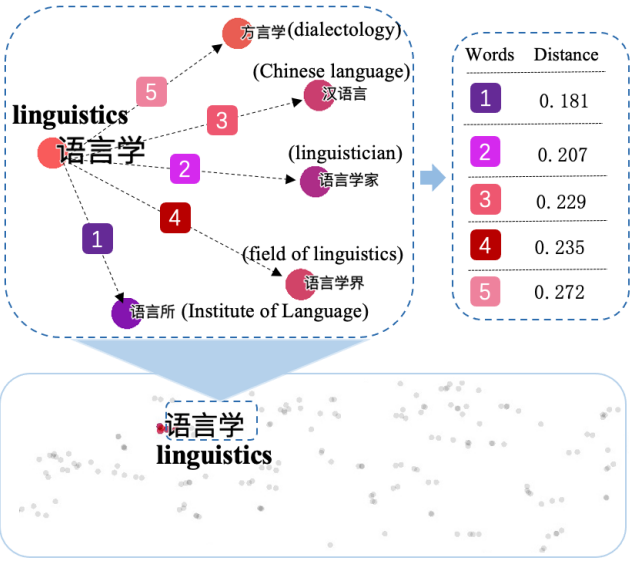

Figure 1: A visual example of some latent word boundaries matched by "言". Specifically, we first perform Chinese word segmentation of the input sentence to obtain "语言学", then calculate the spatial distance of all word embeddings with respect to it, and show the five shortest distances.

A drawback of the character-based model is that explicit word information is not fully exploited. Therefore, Zhang and Yang (2018) proposed Lattice-LSTM that incorporates the related words into the character representation, which not only injects the word's feature but also avoids the error propagation of word segmentation. However, Lattice-LSTM is limited by the structure of no-parallelizable sequential LSTM, and the architecture is quite complicated that it is a challenging to transfer to other neural networks. With this consideration, LR-CNN (Gui et al. 2019a) uses CNN instead of LSTM to improve the model parallel ability. FLAT

*Corresponding author.

(Li et al. 2020) also converts the Lattice into a flat structure that enables characters to directly interact with any potential words. LGN (Gui et al. 2019b) and CGN (Sui et al. 2019) utilize graph neural networks to fuse word information.

The above methods employ different networks to solve the issues of Lattice. While other methods (Liu et al. 2019; Ding et al. 2019; Ma et al. 2020) fuse character and word information into the embedding that is a representation model and does not concern what kind of networks is used subsequently. For example, SoftLexicon (Ma et al. 2020) integrates all matched words from the lexicon into the character representation according to weight obtained by frequency of occurrence. However, the lexicon contains a large number of potential word boundaries. It need be considered for embedding-level fusion to choose the most useful ones according to semantic similarity rather than concatenate all matched words. To the best of our knowledge, we are the first to explore this critical issue. This paper believes that words with close context semantics also have certain similarities in vector space after distributed representation. We visualize all the words matched by the character "言 (speech)" in the vocabulary as shown in Figure 1, where we can find that different words do have considerable differences in spatial representation.

In information theory, the mutual information of two random variables measures how much one variable tells us about the other. Moreover, mutual information maximization (MIM) provides a unified way to characterize the correlation between different granularity of embeddings. To the best of our knowledge, we are the first to employ MIM to enhance features in Chinese NER too. Consequently, using semantic similarity and MIM, we propose a novel Chinese NER model called *SSMI*. In our model, we first tokenize the input using BERT to get the word span containing characters, match all of its latent word boundaries and calculate their semantic similarity. Then, we obtain a potential word boundary with the most significant similarity and the set of word boundaries where similarity score is larger than a certain threshold, respectively. And we further concatenate the most relevant word boundaries with corresponding characters. Finally, we gain the set of words to compute mutual information with characters and sentences respectively, and we input the enhanced data into different stages of the proposed neural network architecture. The main contributions of our work are as follows:

- We propose an innovative method for Chinese NER, which first uses the semantic similarity to find highly relevant word boundaries and incorporate them into our model and then utilizes MIM as a unified way to characterize the correlation between different granularities of embeddings.
- We devise a neural framework to adapt our augmented representation approach. It inputs the enhanced data to different stages of the neural network to fully fuse features.
- The results show that our *SSMI* yields state-of-the-art (SOTA) results in all four datasets.

Background

Our work is inspired by previous work, including text similarity and mutual information maximization.

Text Similarity

From the perspective of information theory (Lin 1998), the similarity is defined as the commonality between the two text fragments. The existing text similarity methods can be divided into two categories: text distance (Deza and Deza 2009; Kusner et al. 2015; Norouzi, Fleet, and Salakhutdinov 2012) and text representation (Irving and Fraser 1992; Le and Mikolov 2014; Wan et al. 2016). There are many kinds of methods for calculating distance. We utilize cosine distance (Deza and Deza 2009) in Figure 1 to compute similarity and visualize it. The distance between texts can be computed below:

$$D_c(t_A, t_B) = 1 - S_c(t_A, t_B)$$

$$S_c(t_A, t_B) = \frac{\vec{A} \cdot \vec{B}}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Here D_c is the cosine distance and S_c is the cosine similarity. The approaches based on textual representations can be divided into lexically and semantically similar. The former has a similar character sequence that is a text comparison at the literal level. The latter is the semantic similarity of the context, which is the extraction of text features or co-occurrence probability in a given corpus. The semantic similarity used in this paper is to obtain the corresponding distributed representation through the pre-trained vocabulary and then calculate the cosine similarity between them.

Mutual Information

In information theory, the mutual information of two random variables can measure the degree of mutual dependence between them. A fundamental technique in our approach is mutual information maximization. In simple terms, given two random variables X and Y, it can be understood as knowing how much X reduces the uncertainty of Y and vice versa (Zhou et al. 2020). The Mutual information between X and Y can be formulated as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \sum_{x} p(x) log \frac{1}{p(x)} - \sum_{y,x} p(y,x) log \frac{1}{p(x|y)} \\ &= \sum_{y,x} p(y,x) log \frac{p(y,x)}{p(x)p(z)}. \end{aligned}$$

Mutual information can be maximized by finding a lower bound on I(X,Y). InfoNCE (Kong et al. 2019; Logeswaran and Lee 2018) has been proven to be an effective computing method in practice. It is formulated as:

$$E_{p(X,Y)}[f_\theta(x,y) - E_{q(\hat{Y})}[log \sum_{\hat{y} \in \hat{Y}} exp f_\theta(x,\hat{y})]] + log|\hat{Y}|$$

where x and y denote different views of an input, f_θ is a function parameterized by θ (e.g., the encoded representation of a word and its context (Kong et al. 2019). $\hat{Y}$ represents a set of samples drawn from the proposal distribution $q(\hat{Y})$, which contains positive samples y and $|\hat{Y}|-1$ negative samples. If $\hat{Y}$ always contains all possible values of the random variable Y (i.e., $\hat{Y}=Y$) and they are uniformly distributed, maximizing InfoNCE is similar to maximizing the standard cross-entropy loss:

$$E_{p(X,Y)}[f_\theta(x,y) - log \sum_{\hat{y}\in\hat{Y}} exp f_\theta(x,\hat{y})]$$

The above equation shows InfoNCE is related to maximizing $p_\theta(y|x)$, approximating the sum of the elements in Y (i.e. the partition function) by negative sampling (Zhou et al. 2020). Using this formula, we can maximize the mutual information between latent word boundaries and sentences.

Model

In this section, we mainly introduce *SSMI* in details. The overall architecture is shown in Figure 2, which consists of latent word boundary selection, multi-view information enhancement, and model architecture.

Latent Word Boundary Selection

We first use each character in the input sequence to match words that contain this character from pre-trained vocabulary to obtain a set of latent word boundaries. Then we employ the Chinese word segmentation tool to segment the input sentence and check whether it exists in the vocabulary. If it does not exist, we use BERT (Devlin et al. 2018) to generate the corresponding embedding. Finally, we calculate cosine similarity based on the obtained word segmentation and latent word boundary embeddings of characters to get the most relevant words and a set of words whose similarity score is larger than a threshold.

Match Latent Word Boundaries Models based on character outperform others for Chinese NER tasks (Liu, Zhu, and Zhao 2010; Li et al. 2014; Sui et al. 2019). Therefore, this work also employs characters as the primary input and conducts information enhancement. The input sequence can be viewed as a character set S=$\{c_1,c_2,...,c_n\}$$\in V$, where V denotes the pre-trained vocabulary. Each character c_i can be embedded as follows:

$$x_i^c = e^c(c_i),$$

where e^c represents the character embedding lookup table. Since the essence of Chinese NER is to first determine entity boundaries and then perform sequence labelling, it is crucial to choose the correct boundaries for subsequent tasks. We think that the latent word boundary information of a character can effectively enhance its representation. Consequently, we create a word set W of all matched latent word boundaries that contain a specific character.

$$W(c_i) \leftarrow \{\forall w_j \in V, w_j \Rightarrow [c_b, c_m, c_e]\},$$

where $w_j \Rightarrow [c_b,c_m,c_e]$ denotes that c_b is the character in the starting position of w_j, and c_m, c_e are the characters of middle and end positions, respectively.

Input Segmentation and Embedding We have obtained the set of latent word boundaries and their corresponding embeddings for a specific character. Then we tokenize the input sequence to get the segmentation that contains this character. To get the segmentation representation, we need first determine whether it exists in the pre-training vocabulary and, if not, utilize BERT to generate it. The purpose of tokenization is to calculate the spatial similarity between boundaries and segmentation in order to evaluate their semantic distance.

$$E(s_i) \rightarrow \{S_i \in V; E = BERT(s_i)\},$$

where $E(s_i)$ represents the final segmentation embedding. S_i, E denote acquisition from vocabulary and BERT generation, respectively.

Compute Semantic Similarity We suppose that the result after tokenization is an expression containing contextual semantics in the input text. However, the number of matching latent word boundaries is large, and the semantics of different terms is quite distinct. So it is not a good enhancement approach to directly concatenate all matching words without distinguishing their semantics. Consequently, we take advantage of BERT's superior contextual learning capabilities to fetch a complete representation of the input characters, and then compute the cosine similarity to the words found in the vocabulary. Our goal is to choose a word with the highest similarity and a set whose similarity score is greater than a threshold. We can compute them below:

$$S_k = \frac{\vec{V_s}\cdot\vec{V_b}}{\|V_s\|\|V_b\|},$$
$$S_m = Max(S_k), \quad S_s = [S_k > \Delta],$$

where S_k represents the cosine similarity between segmentation and latent word boundaries, V_s and V_b are their vectorized representations. S_m is the boundary with the highest semantic similarity while S_s is a set whose cosine similarity is greater than the threshold Δ.

Multi-View Enhancement

This section mainly applies mutual information maximization for feature enhancement. We first fuse the obtained semantically closest latent word boundaries with their corresponding character representations. Then we maximize the local mutual information between the input sequence and the set of word boundaries from the character perspective. Finally, we utilize global semantic information for augmentation and maximize the mutual information among the input sentence and the set boundaries.

Integrate Most Relevant Boundary We believe that the closer the representations of texts are in the vector space, the more similar the contextual semantics they imply. Therefore, we concatenate the latent word boundary with the largest similarity and the corresponding character representation as our primary input. It can be described as follows:

$$I = [E_c \oplus E_{S_m}]$$

where I and $\oplus$ mean the final input representation and concatenation operations, respectively. E_c and E_{S_m} are the embeddings of character and S_m, respectively.

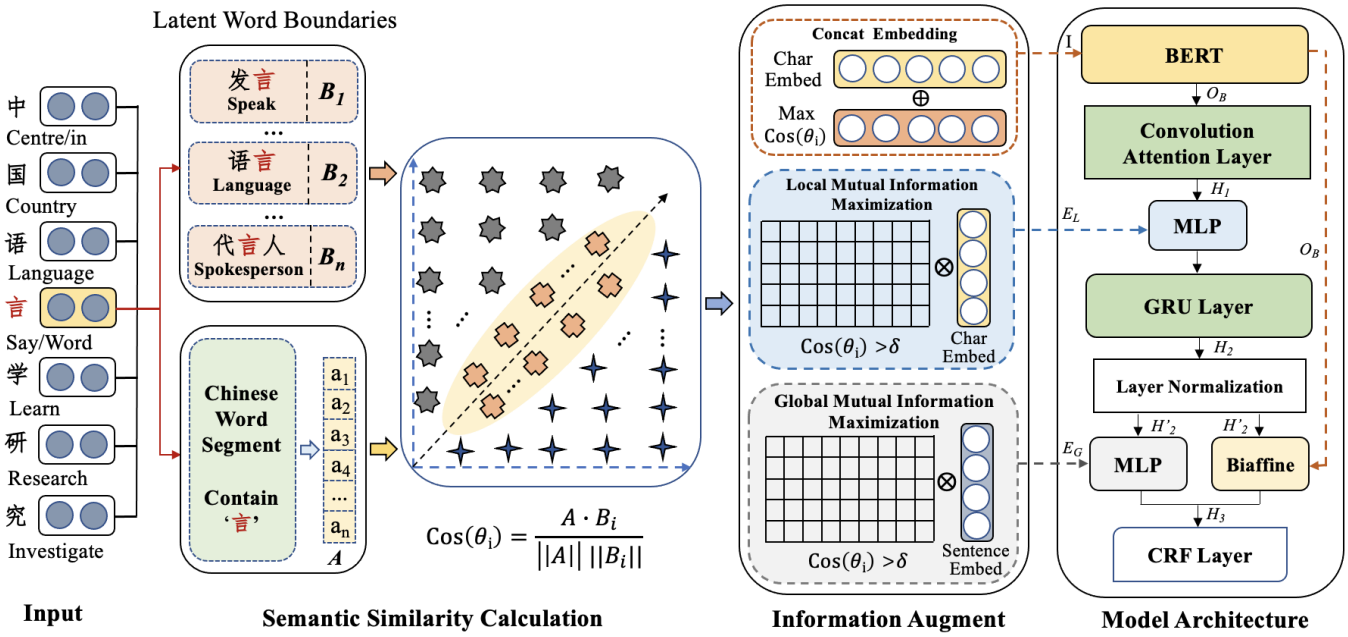

Figure 2: An overview of our *SSMI*. In the semantic similarity calculation step, we first utilize the Chinese word segmentation tool to obtain the fragments containing characters and then calculate their similarity with all potential word boundaries. We further enhance the word boundaries with high similarity in the information augmentation step and employ local and global mutual information maximizations. We finally design a novel neural architecture to implement our proposed method.

Local Mutual Maximization Mutual information is a criterion commonly used in statistical language modeling of word associations and related applications (Church and Hanks 1990; Wiener et al. 1995; Hartley 1928). Inspired by the work of Xu et al. (2007), we use the local semantic information contained in the characters to calculate the pointwise mutual information with the matched word boundaries. Given a boundary b and a character c, we have probabilities $p(b)$ and $p(c)$. Let A denote the number of times b and c co-occurrences, C denote the number of times c occurrences without b, and N represent the total number of matched latent boundaries in the pre-trained vocabulary. The pointwise mutual information criterion between c and b is defined as:

$$PI(c,b) = log\frac{p(b,c)}{p(b) \times p(c)} = log\frac{p(b \hat{} c)}{p(b) \times p(c)},$$

and is estimated as:

$$PI(c,b) \approx log\frac{A \times N}{A \times (A+C)},$$

Let $\{b_i\}_i^m$ be the set of all matched word boundaries. The pointwise mutual information between each b and character c can be computed as:

$$PI_{avg}(c) = \sum_{i=1}^{m} p(b_i) PI(c,b_i).$$

After calculating these criteria, we fetch the word boundaries with the largest PI_{avg}, which are then concatenated with the output of the convolutional attention layer and fed into the first multi-layer perceptron layer.

Global Mutual Maximization We maximize the mutual information between input sentences and word boundaries. For each sentence, word boundaries contain potential semantic information. Therefore, we jointly learn the global semantic information of sentences and the boundaries, whose similarities are larger than a threshold. In this way, it is expected to inject enhanced features into sentence representations. Given a sentence s and a filtered word boundaries set $B_i = \{b_1, ..., b_k\}$. Formally, let e_s represent the sentence embedding obtained by BERT, and e_{b_i} denote the embedding for the j-th word boundary $b_j \in B_i$. We introduce a loss function by the contrastive learning framework (Logeswaran and Lee 2018) to maximize the mutual information between the two perspectives. The loss function is shown below:

$$L(s, B_i) = E_{b_j \in B_i}[f(s,b_j) - log \sum_{\hat{b} \in B - B_i} exp(f(s,\hat{b}))],$$

where we use boundaries with very low similarity as negative samples to enhance the association between sentences and high similarity boundaries. The function $f(\cdot,\cdot)$ can be computed as:

$$f(s,b_j) = \sigma(e_s^\top \cdot W_{sb} \cdot e_{b_j}),$$

Here $W_{sb} \in \mathbb{R}^{d \times d}$ represents a learnable parameter matrix, the $\sigma(\cdot)$ denotes sigmoid function. Note that we define a loss function for each character's the set of high-similarity word boundaries. It is convenient to generalize $f(s,b_j)$ to the entire dataset.

Model Architecture

We design a novel neural architecture to implement our data augmentation method based on semantic similarity and MIM. The model structure consists of three parts: the input encoding, two types of feature enhancements and the decoding.

Input Encoding The primary role of this layer is to encode the original input. We first use BERT for basic encoding, which outputs O_B, and then input the encoded data into the convolution attention layer for deep fusion. The above steps can be processed by the following equation:

$$H_1 = CAN(\, BERT\,(I_1, ..., I_n \in I\,)),$$

where CAN (Zhu and Wang 2019) and BERT denote the convolution attention network and BERT, respectively. I_i is the input, which is concatenated from character embeddings and latent word boundaries with the highest similarity.

Local Enhancement The goal of this part is to augment the output of the previous stage. We first concatenate the word boundaries, which have the largest pointwise mutual information in the output of the convolution attention layer, then input it into a MLP layer and GRU layer successively. The following equation shows the implementation details:

$$H_2 = GRU(\, MLP_1\,(H_1 \oplus E_l)$$

where GRU and MLP_1 are the GRU and MLP layers respectively, and H_1 is the output of previous stage. E_l denotes the embedding from local mutual information maximization.

Global Enhancement The former part is enhanced with the local semantics of characters while this part is augmented by the global semantic information of sentences. We first perform layer normalization, which gets $H_2^{'}$, and then integrate the maximized global mutual information representation with $H_2^{'}$. At the same time, we employ Biaffine to fuse $H_2^{'}$ and BERT's output O_B. Finally, the above two results are concatenated and input into the CRF layer. Formally, the above steps can be shown as the following equation:

$$H_3 = MLP_2(LN(H_2) \oplus E_g) \oplus Biaffine(LN(H_2) \oplus O_B),$$

where E_g is the embedding from global mutual information maximization. LN represents the layer normalization that can be computed below:

$$H_2^{'} = LN(H_2) = f_\alpha(h_2^i) \odot (\frac{h_2^i - \mu}{\sigma}) + f_\beta(h_2^i),$$

where $f_\alpha(h_2^i) = W_\alpha h_2^i + b_\alpha$ and $f_\beta(h_2^i) = W_\beta h_2^i + b_\beta$ are the parameter and bias for layer normalization of $h_2^i \in H_2$. μ and σ denote the mean and standard deviation across the items of h_2^i, which are calculated as:

$$\mu = \frac{1}{d_h}\sum_{j=1}^{d_h} h_2^{ij},\ \sigma = \sqrt{\frac{1}{d_h}\sum_{j=1}^{d_h}(h_2^{ij} - \mu)^2}.$$

where h_2^{ij} is the j-th dimension of h_2^i.

Decoding Layer A standard CRF layer is used to capture the dependencies between successive labels. $h_3^i \in H_3$ is the final output of the last layer, and the probability of a label sequence y is:

$$P\,(y|s) = \frac{exp\left(\sum_i \left(W^{y_i} h_3^i + b_{(y_{i-1}, y_i)}\right)\right)}{\sum_{y'} exp\left(\sum_i \left(W^{y_i'} h_3^i + b_{(y_{i-1}', y_i')}\right)\right)}$$

Here $y^{'}$ is an arbitrary label sequence, and W^{y_i} is a model parameter specific to y_i, and $b_{(y_{i-1}, y_i)}$ is a bias specific to y_{i-1} and y_i. Given a set of manually annotated training data $\{(s_i, y_i)\}|_{i=1}^N$, we train the model by log-likelihood loss with L_2 regularization.

Experiments

Settings

We carry out extensive experiments to assess our *SSMI*. In addition, we compare its performance with recent classical models in the same settings.

Datasets

We evaluate our model on four Chinese NER datasets, including Ontonotes 4.0 (Weischedel et al. 2011), MSRA (Levow 2006), Weibo (Peng and Dredze 2015; He and Sun 2017a), and Resume (Zhang and Yang 2018). Both Ontonotes 4.0 and MSRA datasets are from the newswire domain in simplified Chinese. The former is annotated with four entity categories: PER (Person), ORG (Organization), LOC (Location), and GPE (Geo-Political Entity). And the latter contains three annotated entities: ORG, PER, and LOC. The Weibo dataset is a social media domain dataset drawn from Sina Weibo and annotated with the entity types of Ontonotes 4.0. The Chinese Resume dataset consists of resumes of senior executives that are annotated with eight kinds of named entities: CONT (Country), EDU (Educational Institution), LOC, PER, ORG, PRO (Profession), RACE (Ethnicity/Background), and TITLE (Job Title).

Baseline

To assess the effectiveness of our *SSMI*, we compare our approach with the following baseline methods:

(1) *Lattice-LSTM*. It first integrates word information into the character-based model to avoid the segmentation errors.

(2) *LR-CNN*. It (Gui et al. 2019a) applies rethinking mechanism of CNN fusion word to exploit parallelism.

(3) *LGN*. LGN (Gui et al. 2019b) is a graph neural network based method that alleviates the RNN-based models' weakness which is vulnerable to word ambiguities.

(4) *BERT*. It (Devlin et al. 2018) is a common Chinese version of BERT. We directly download from hugging face[1].

(5) *PLTE*. It (Xue et al. 2020) extends the transformer encoder that augments self-attention with positional relation representations to incorporate lattice structure.

(6) *SoftLexicon*. It (Ma et al. 2020) is a simpler implementation of lattice-LSTM to avoid complicated architecture.

[1]https://huggingface.co/

	OntoNote 4.0			MSRA			Weibo			Resume		
Models	**P**	**R**	**F1**	**P**	**R**	**F1**	**NE**	**NM**	**Overall**	**P**	**R**	**F1**
Lattice-LSTM	76.35	71.56	73.88	93.57	92.79	93.18	53.04	62.25	58.79	94.18	94.11	94.46
LR-CNN	76.40	72.60	74.45	94.50	92.93	93.71	57.14	66.67	59.92	95.37	94.84	95.11
LGN	76.13	73.68	74.89	94.19	92.73	93.46	55.34	64.98	60.21	95.28	95.46	95.37
SoftLexicon	77.28	74.07	75.64	94.63	92.70	93.66	59.08	62.22	61.42	95.30	95.77	95.53
+ bichar	77.13	75.22	76.16	94.73	93.40	94.06	58.12	64.20	59.81	95.71	95.77	95.74
PLTE	76.78	72.54	74.60	94.25	92.30	93.26	53.55	64.90	59.76	95.34	95.46	95.40
FLAT	-	-	75.70	-	-	94.35	-	-	63.42	-	-	94.93
MECT	77.57	76.27	76.92	94.55	94.09	94.32	61.91	62.51	63.30	96.40	95.39	95.89
BERT	80.53	79.88	79.95	95.37	94.56	94.78	67.18	68.75	67.30	95.14	95.27	95.29
ERNIE	79.43	76.23	77.65	95.69	94.83	95.08	-	-	67.96	95.04	94.89	94.82
ZEN	80.52	78.97	79.03	95.90	95.06	95.20	-	-	66.71	95.48	95.43	95.40
ChineseBERT	80.03	83.33	81.65	95.64	95.17	95.39	-	-	69.02	95.96	95.93	95.89
LEBERT	-	-	82.08	-	-	95.70	-	-	70.75	-	-	96.08
SoftLexicon $_{bert}$	83.41	82.21	82.81	95.75	95.10	95.42	70.94	67.02	70.50	96.08	96.13	96.11
MECT $_{bert}$	-	-	82.57	-	-	96.24	-	-	70.43	-	-	95.98
W^{2}NER	82.31	83.36	83.08	96.12	96.08	96.10	-	-	72.32	96.96	96.35	96.65
SSMI(our)	82.46	84.61	83.52	96.15	96.49	96.32	71.53	73.18	72.83	97.48	97.18	97.33

Table 1: Results on OntoNote 4.0, MSRA, Weibo, and Resume datasets

(7) *FLAT*. It (Li et al. 2020) converts the lattice-LSTM into a flat structure that can fully leverage the lattice advantage and has excellent parallelization ability.

(8) *MECT*. MECT (Wu et al. 2021) is a multi-metadata embedding based cross-transformer by fusing the structural information of Chinese characters.

(9) *W^{2}NER*. It (Li et al. 2021) is an alternative by modeling the unified NER as word-word relation classification.

(10) *ERNIE* (Sun et al. 2019). It is an extension of BERT enhanced by knowledge masking strategies, including entity-level masking and phrase-level masking.

(11) *ZEN* (Diao et al. 2019). It is a BERT-based Chinese text encoder enhanced by N-gram, where different combinations of characters are considered.

(12) *LEBERT* (Liu et al. 2021). It is a Multi-metadata Embedding based Cross-Transformer by fusing the structural information of Chinese characters.

(13) *ChineseBERT* (Sun et al. 2021). It is a pre-trained model that incorporates both the glyph and pinyin information into language model pretraining.

Results & Analysis

This section evaluates and analyzes our model relative to the state-of-the-art models of different periods. As shown in Table 1, we divide the baselines into three categories. The first augments data by fusing external lexicon information, the second uses a single transformer encoder as the main model architecture, and the last is based on BERT or employs it as the fundamental encoder.

OntoNotes. As shown in the second column of Table 1, we start from the perspective of the sequence encoding and observe that the F1 score is significantly improved by replacing CNN, LSTM and the single transformer encoder layer with BERT, which indicates BERT's excellent context encoding capability can be fully exploited on this dataset. Furthermore, we find that our *SSMI* can effectively augment Chinese expressiveness and achieve a relatively high precision, recall, and F1. Notably, the F1 score of our *SSMI* is 83.52%, which outperforms that of *SoftLexicon* combined with BERT. It demonstrates that employing the similarity of vectors to seek semantically close word boundaries is more reasonable than directly matching words.

MSRA. The results obtained on MSRA are shown in the third column. We know that the F1 score by utilizing BERT as a sequence encoder on this dataset is better than any other architectures. However, the improvement is not significant. This is because the improvement is related to the size and content of the dataset. As we can see in Table 1, our *SSMI* sets the SOTA F1 score.

Weibo. The text of the Weibo dataset is highly irregular. Consequently, the entity recognition of Weibo is the most difficult. As can be seen from the fourth column of Table 1, where NE, NM and Overall denote F1-scores for named entities, nominal entities (excluding named entities) and both respectively, the context semantics learned by the pre-training model in large-scale Chinese corpora demonstrates a considerable advantage on the Weibo dataset. Compared with the architecture without applying the transformer encoder (first block SOTA), the performance of BERT based models (third block SOTA) is improved by more than 10%, which is the greatest improvement among all datasets.

Resume. Since the Resume dataset is both regular and small, the improvement achieved by the baselines is not apparent with only 1.12% improvement from *SoftLexicon* to *W^{2}NER*. In contrast, the methods whose F1 is larger than 96% use BERT as the primary encoder and also utilize word information for enhancement, which suggests that this combination can positively affect this dataset. According to the results in the fifth column, we know that the maximization of mutual information yielded by semantic similarity can effectively perform data enhancement. Therefore, our *SSMI* achieves the highest F1 score on this dataset.

	Span F		Type Acc	
	Ontonote	**MSRA**	**Ontonote**	**MSRA**
SSMI	83.22	96.59	98.73	99.85
-BERT	82.68	96.42	98.46	99.32
-Cosine	81.89	95.16	98.02	99.57
-E_{loc}	78.31	95.40	97.96	99.41
-E_{glb}	73.85	95.28	98.53	99.48

Table 2: Two metrics of our *SSMI*. *-Cosine* represents latent word boundaries are chosen randomly instead of by calculating semantic similarity. *-BERT* means not to encode with BERT. *-E_{loc}* and *-E_{glb}* denote the removal of local and global mutual information maximization, respectively.

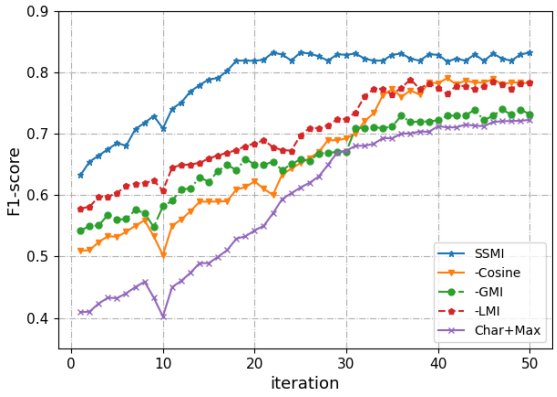

Figure 3: F1 against iterations. *-Cosine* means randomly choosing latent word boundaries. *-LMI* and *-GMI* denote not using local and global mutual information. *Char+Max* uses only cosine similarity to filter the most relevant words and fuse them into character embeddings.

Discussion

In this section, we first conduct a comprehensive analysis of the correctness of span and type, then discuss the changes of F1 under different iterations. Finally, we discuss the data augmentation methods and network architectures' effects on the final performance from a more fine-grained.

Correctness of span and type. The Chinese NER task is to determine the span of an entity and label its type. We use two metrics from FLAT (Li et al. 2020) to validate our method. *Span F* is the F-score of the span and *Type Acc* refers to the ratio of entirely correct predictions to span correct predictions. As shown in Table 2, the *Span F* and *Type Acc* decrease by less than 1% on both datasets without applying BERT as the underlying encoder. When semantic similarity is removed, the *Span F* on two datasets reduces more than 1%, but the *Type Acc* is not sensitive. However, when we do not use local and global MIM embeddings, the *Span F* of the Ontonotes dataset drops 4.91% and 9.37% respectively and that of MSRA also decreases by 1.19% and 1.31% respectively while *Type Acc* of both datasets does not apparently decline.

F1 against iterations. Figure 3 depicts the F1 score of our model against iterations on the OntoNotes. As the number of iterations increases, the F1 scores without semantic similarity, local or global MM are consistently smaller than those of SSMI. As we can see in Figure 3, both semantic similarity and MIM fusion are more significant enhancement than Char+Max. Furthermore, global MIM is more important than local MIM, which may be due to the stronger dependence on global semantics as sentence length increases.

	Ontonote	MSRA	Weibo	Resume
SSMI	**83.52**	**96.27**	**72.83**	**97.33**
-Cosine	79.19	94.90	68.55	95.07
-GMI	74.08	92.53	66.83	94.29
-LMI	78.57	94.84	69.66	95.12
Char+Max	72.23	93.37	62.77	93.32
-BERT	82.63	95.87	71.50	96.83
-CAN	83.03	95.92	72.37	97.11
-GRU	82.64	95.53	72.19	96.88
-MLP_1	81.80	94.62	70.94	95.63
-MLP_2	80.95	94.31	69.87	95.19
-LNorm	82.54	95.90	72.36	97.03
-Biaffine	83.20	96.10	72.43	96.93

Table 3: Removed different parts to verify our SSMI.

Ablation study. Using Table 3, we analyze their respective impact on performance from the level of data augmentation and network architecture with the smallest granularity. The first block in the table verifies our data augmentation method's effectiveness, and the second block evaluates the effect of each component on the network architecture.

In the first block, we can find that *Char+Max*'s F1 scores are the lowest on all datasets. However, *SSMI* can set the maximum F1 score on four datasets, which indicates the effectiveness of our model. Consequently, removing the latent word boundary filtering module based on semantic similarity, the global or local MIM will cause performance degradation. But the global mutual information has much greater influence, which is related to the length of the sentence.

In the second block, we remove each component of the neural network architecture and assess its influence on performance. From the table, we can find that removing any component will result in performance degradation, and the two MLP layers have the most significant influence. Therefore, it shows that each component in our network architecture has a positive enhancement effect.

Conclusion

In this paper, we design a novel Chinese NER framework called *SSMI* based on smantic similarity and mutual information maximization. In our approach, we filter all potential word boundaries according to the semantic similarity and employ mutual information maximization to compute the correlations between word boundaries, character, and sentence embeddings. We further introduce a novel network to implement our augmented representation approach. The experimental results show that our *SSMI* yields SOTA performance on four widely-used benchmark datasets. We hope that MIM will be beneficial for English NER.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. 61772534, and partially supported by Public Computing Cloud, Renmin University of China.

References

Chiu, J. P.; and Nichols, E. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4: 357–370.

Church, K.; and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1): 22–29.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deza, M. M.; and Deza, E. 2009. Encyclopedia of distances. In *Encyclopedia of distances*, 1–583. Springer.

Diao, S.; Bai, J.; Song, Y.; Zhang, T.; and Wang, Y. 2019. ZEN: Pre-training Chinese text encoder enhanced by n-gram representations. *arXiv preprint arXiv:1911.00720*.

Ding, R.; Xie, P.; Zhang, X.; Lu, W.; Li, L.; and Si, L. 2019. A neural multi-digraph model for Chinese NER with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1462–1467.

Gui, T.; Ma, R.; Zhang, Q.; Zhao, L.; Jiang, Y.-G.; and Huang, X. 2019a. CNN-Based Chinese NER with Lexicon Rethinking. In *IJCAI*, 4982–4988.

Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; and Huang, X. 2019b. A lexicon-based graph neural network for chinese ner. 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)*, 1040–1050.

Hartley, R. V. 1928. Transmission of information 1. *Bell System technical journal*, 7(3): 535–563.

He, H.; and Sun, X. 2017a. 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*, 713–718.

He, H.; and Sun, X. 2017b. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

He, J.; and Wang, H. 2008. Chinese named entity recognition and word segmentation based on character. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Irving, R. W.; and Fraser, C. B. 1992. Two algorithms for the longest common subsequence of three (or more) strings. In *Annual Symposium on Combinatorial Pattern Matching*, 214–229. Springer.

Jie, Y.; Teng, Z.; Zhang, M.; and Yue, Z. 2016. Combining Discrete and Neural Features for Sequence Labeling. In *International Conference on Intelligent Text Processing Computational Linguistics*.

Kong, L.; de Masson d'Autume, C.; Yu, L.; Ling, W.; Dai, Z.; and Yogatama, D. 2019. A Mutual Information Maximization Perspective of Language Representation Learning. In *International Conference on Learning Representations*.

Kusner, M.; Sun, Y.; Kolkin, N.; and Weinberger, K. 2015. From word embeddings to document distances. In *International conference on machine learning*, 957–966. PMLR.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Le, Q.; and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, 1188–1196. PMLR.

Levow, G. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 108–117.

Li, H.; Hagiwara, M.; Li, Q.; and Ji, H. 2014. Comparison of the Impact of Word Segmentation on Name Tagging for Chinese and Japanese. In *LREC*, 2532–2536.

Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; and Li, F. 2021. Unified named entity recognition as word-word relation classification. *arXiv preprint arXiv:2112.10070*.

Li, X.; Yan, H.; Qiu, X.; and Huang, X. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6836–6842.

Lin, D. 1998. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, 296–304. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558605568.

Liu, L.; Shang, J.; Xu, F.; Xiang, R.; and Han, J. 2018. Empower sequence labeling with task-aware neural language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Liu, W.; Fu, X.; Zhang, Y.; and Xiao, W. 2021. Lexicon enhanced chinese sequence labeling using bert adapter. *arXiv preprint arXiv:2105.07148*.

Liu, W.; Xu, T.; Xu, Q.; Song, J.; and Zu, Y. 2019. An encoding strategy based word-character LSTM for Chinese NER. 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)*, 2379–2389.

Liu, Z.; Zhu, C.; and Zhao, T. 2010. Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? In *International Conference on Intelligent Computing*, 634–640. Springer.

Logeswaran, L.; and Lee, H. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Ma, R.; Peng, M.; Zhang, Q.; Wei, Z.; and Huang, X.-J. 2020. Simplify the Usage of Lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5951–5960.

Ma, X.; and Hovy, E. 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)*.

Norouzi, M.; Fleet, D. J.; and Salakhutdinov, R. R. 2012. Hamming distance metric learning. *Advances in neural information processing systems*, 25.

Peng, N.; and Dredze, M. 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*, 548–554.

Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; and Liu, S. 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)*, 3830–3840.

Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; and Wu, H. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Sun, Z.; Li, X.; Sun, X.; Meng, Y.; Ao, X.; He, F., Q.and Wu; and Li, J. 2021. ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Wan, S.; Lan, Y.; Guo, J.; Xu, J.; Pang, L.; and Cheng, X. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Weischedel, R.; Pradhan, S.; Ramshaw, L.; Palmer, M.; Xue, N.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; and Belvin, R. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Wiener, E.; Pedersen, J. O.; Weigend, A. S.; et al. 1995. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval*, volume 317, 332. Citeseer.

Wu, S.; Song, X.; Zhang, Q.; Peng, M.; and Feng, Z. 2021. MECT: Multi-Metadata Embedding based Cross-Transformer for Chinese Named Entity Recognition. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*.

Xu, Y.; Jones, G.; Li, J.; Wang, B.; and Sun, C. 2007. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*, 3(3): 1007–1012.

Xue, M.; Yu, B.; Liu, T.; Zhang, Y.; and Wang, B. 2020. Porous Lattice Transformer Encoder for Chinese NER. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3831–3841.

Zhang, Y.; and Yang, J. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1554–1564.

Zhou, K.; Wang, H.; Zhao, W. X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; and Wen, J.-R. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, 1893–1902. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368599.

Zhu, Y.; and Wang, G. 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, Volume 1 (Long and Short Papers)*, 3384–3393. Minneapolis, Minnesota: Association for Computational Linguistics.