TENER: Adapting Transformer Encoder for Named Entity Recognition

Abstract

The Bidirectional long short-term memory networks (BiLSTM) have been widely used as an encoder in models solving the named entity recognition (NER) task. Recently, the Transformer is broadly adopted in various Natural Language Processing (NLP) tasks owing to its parallelism and advantageous performance. Nevertheless, the performance of the Transformer in NER is not as good as it is in other NLP tasks. In this paper, we propose TENER, a NER architecture adopting adapted Transformer Encoder to model the character-level features and word-level features. By incorporating the direction and relative distance aware attention and the un-scaled attention, we prove the Transformer-like encoder is just as effective for NER as other NLP tasks.

Introduction

The named entity recognition (NER) is the task of finding the start and end of an entity in a sentence and assigning a class for this entity. NER has been widely studied in the field of natural language processing (NLP) because of its potential assistance in question generation BIBREF0, relation extraction BIBREF1, and coreference resolution BIBREF2. Since BIBREF3, various neural models have been introduced to avoid hand-crafted features BIBREF4, BIBREF5, BIBREF6.

NER is usually viewed as a sequence labeling task, the neural models usually contain three components: word embedding layer, context encoder layer, and decoder layer BIBREF4, BIBREF5, BIBREF6, BIBREF7, BIBREF8, BIBREF9, BIBREF10. The difference between various NER models mainly lies in the variance in these components.

Recurrent Neural Networks (RNNs) are widely employed in NLP tasks due to its sequential characteristic, which is aligned well with language. Specifically, bidirectional long short-term memory networks (BiLSTM) BIBREF11 is one of the most widely used RNN structures. BIBREF4 was the first one to apply the BiLSTM and Conditional Random Fields (CRF) BIBREF12 to sequence labeling tasks. Owing to BiLSTM's high power to learn the contextual representation of words, it has been adopted by the majority of NER models as the encoder BIBREF5, BIBREF6, BIBREF9, BIBREF10.

Recently, Transformer BIBREF13 began to prevail in various NLP tasks, like machine translation BIBREF13, language modeling BIBREF14, and pretraining models BIBREF15. The Transformer encoder adopts a fully-connected self-attention structure to model the long-range context, which is the weakness of RNNs. Moreover, Transformer has better parallelism ability than RNNs. However, in the NER task, Transformer encoder has been reported to perform poorly BIBREF16, our experiments also confirm this result. Therefore, it is intriguing to explore the reason why Transformer does not work well in NER task.

In this paper, we analyze the properties of Transformer and propose two specific improvements for NER.

The first is that the sinusoidal position embedding used in the vanilla Transformer is aware of distance but unaware of the directionality. In addition, this property will lose when used in the vanilla Transformer. However, both the direction and distance information are important in the NER task. For example in Fig FIGREF3, words after “in" are more likely to be a location or time than words before it, and words before “Inc." are mostly likely to be of the entity type “ORG". Besides, an entity is a continuous span of words. Therefore, the awareness of distance might help the word better recognizes its neighbor. To endow the Transformer with the ability of direction- and distance-awareness, we adopt the relative positional encoding BIBREF17, BIBREF18, BIBREF19. instead of the absolute position encoding. We propose a revised relative positional encoding that uses fewer parameters and performs better.

The second is an empirical finding. The attention distribution of the vanilla Transformer is scaled and smooth. But for NER, a sparse attention is suitable since not all words are necessary to be attended. Given a current word, a few contextual words are enough to judge its label. The smooth attention could include some noisy information. Therefore, we abandon the scale factor of dot-production attention and use an un-scaled and sharp attention.

With the above improvements, we can greatly boost the performance of Transformer encoder for NER.

Other than only using Transformer to model the word-level context, we also tried to apply it as a character encoder to model word representation with character-level information. The previous work has proved that character encoder is necessary to capture the character-level features and alleviate the out-of-vocabulary (OOV) problem BIBREF6, BIBREF5, BIBREF7, BIBREF20. In NER, CNN is commonly used as the character encoder. However, we argue that CNN is also not perfect for representing character-level information, because the receptive field of CNN is limited, and the kernel size of the CNN character encoder is usually 3, which means it cannot correctly recognize 2-gram or 4-gram patterns. Although we can deliberately design different kernels, CNN still cannot solve patterns with discontinuous characters, such as “un..ily” in “unhappily" and “unnecessarily". Instead, the Transformer-based character encoder shall not only fully make use of the concurrence power of GPUs, but also have the potentiality to recognize different n-grams and even discontinuous patterns. Therefore, in this paper, we also try to use Transformer as the character encoder, and we compare four kinds of character encoders.

In summary, to improve the performance of the Transformer-based model in the NER task, we explicitly utilize the directional relative positional encoding, reduce the number of parameters and sharp the attention distribution. After the adaptation, the performance raises a lot, making our model even performs better than BiLSTM based models. Furthermore, in the six NER datasets, we achieve state-of-the-art performance among models without considering the pre-trained language models or designed features.

Related Work ::: Neural Architecture for NER

BIBREF3 utilized the Multi-Layer Perceptron (MLP) and CNN to avoid using task-specific features to tackle different sequence labeling tasks, such as Chunking, Part-of-Speech (POS) and NER. In BIBREF4, BiLSTM-CRF was introduced to solve sequence labeling questions. Since then, the BiLSTM has been extensively used in the field of NER BIBREF7, BIBREF21, BIBREF22, BIBREF5.

Despite BiLSTM's great success in the NER task, it has to compute token representations one by one, which massively hinders full exploitation of GPU's parallelism. Therefore, CNN has been proposed by BIBREF23, BIBREF24 to encode words concurrently. In order to enlarge the receptive field of CNNs, BIBREF23 used iterative dilated CNNs (ID-CNN).

Since the word shape information, such as the capitalization and n-gram, is important in recognizing named entities, CNN and BiLSTM have been used to extract character-level information BIBREF7, BIBREF6, BIBREF5, BIBREF23, BIBREF8.

Almost all neural-based NER models used pre-trained word embeddings, like Word2vec and Glove BIBREF25, BIBREF26. And when contextual word embeddings are combined, the performance of NER models will boost a lot BIBREF27, BIBREF28, BIBREF29. ELMo introduced by BIBREF28 used the CNN character encoder and BiLSTM language models to get contextualized word representations. Except for the BiLSTM based pre-trained models, BERT was based on Transformer BIBREF15.

Related Work ::: Transformer

Transformer was introduced by BIBREF13, which was mainly based on self-attention. It achieved great success in various NLP tasks. Since the self-attention mechanism used in the Transformer is unaware of positions, to avoid this shortage, position embeddings were used BIBREF13, BIBREF15. Instead of using the sinusoidal position embedding BIBREF13 and learned absolute position embedding, BIBREF17 argued that the distance between two tokens should be considered when calculating their attention score. BIBREF18 reduced the computation complexity of relative positional encoding from $O(l^2d)$ to $O(ld)$, where $l$ is the length of sequences and $d$ is the hidden size. BIBREF19 derived a new form of relative positional encodings, so that the relative relation could be better considered.

Related Work ::: Transformer ::: Transformer Encoder Architecture

We first introduce the Transformer encoder proposed in BIBREF13. The Transformer encoder takes in an matrix $H \in \mathbb {R}^{l \times d}$, where $l$ is the sequence length, $d$ is the input dimension. Then three learnable matrix $W_q$, $W_k$, $W_v$ are used to project $H$ into different spaces. Usually, the matrix size of the three matrix are all $\mathbb {R}^{d \times d_k}$, where $d_k$ is a hyper-parameter. After that, the scaled dot-product attention can be calculated by the following equations,

where $Q_t$ is the query vector of the $t$th token, $j$ is the token the $t$th token attends. $K_j$ is the key vector representation of the $j$th token. The softmax is along the last dimension. Instead of using one group of $W_q$, $W_k$, $W_v$, using several groups will enhance the ability of self-attention. When several groups are used, it is called multi-head self-attention, the calculation can be formulated as follows,

where $n$ is the number of heads, the superscript $h$ represents the head index. $[head^{(1)}; ...;

head^{(n)}]$ means concatenation in the last dimension. Usually $d_k \times n = d$, which means the output of $[head^{(1)}; ...; head^{(n)}]$ will be of size $\mathbb {R}^{l \times d}$. $W_o$ is a learnable parameter, which is of size $\mathbb {R}^{d \times d}$.

The output of the multi-head attention will be further processed by the position-wise feed-forward networks, which can be represented as follows,

where $W_1$, $W_2$, $b_1$, $b_2$ are learnable parameters, and $W_1 \in \mathbb {R}^{d \times d_{ff}}$, $W_2 \in \mathbb {R}^{d_{ff} \times d}$, $b_1 \in \mathbb {R}^{d_{ff}}$, $b_2 \in \mathbb {R}^{d}$. $d_{ff}$ is a hyper-parameter. Other components of the Transformer encoder includes layer normalization and Residual connection, we use them the same as BIBREF13.

Related Work ::: Transformer ::: Position Embedding

The self-attention is not aware of the positions of different tokens, making it unable to capture the sequential characteristic of languages. In order to solve this problem, BIBREF13 suggested to use position embeddings generated by sinusoids of varying frequency. The $t$th token's position embedding can be represented by the following equations

where $i$ is in the range of $[0, \frac{d}{2}]$, $d$ is the input dimension. This sinusoid based position embedding makes Transformer have an ability to model the position of a token and the distance of each two tokens. For any fixed offset $k$, $PE_{t+k}$ can be represented by a linear transformation of

PE_{t} BIBREF13.

Proposed Model

In this paper, we utilize the Transformer encoder to model the long-range and complicated interactions of sentence for NER. The structure of proposed model is shown in Fig FIGREF12. We detail each parts in the following sections.

Proposed Model ::: Embedding Layer

To alleviate the problems of data sparsity and out-of-vocabulary (OOV), most NER models adopted the CNN character encoder BIBREF5, BIBREF30, BIBREF8 to represent words. Compared to BiLSTM based character encoder BIBREF6, BIBREF31, CNN is more efficient. Since Transformer can also fully exploit the GPU's parallelism, it is interesting to use Transformer as the character encoder. A potential benefit of Transformer-based character encoder is to extract different n-grams and even uncontinuous character patterns, like “un..ily” in “unhappily” and “uneasily”. For the model's uniformity, we use the “adapted Transformer” to represent the Transformer introduced in next subsection.

The final word embedding is the concatenation of the character features extracted by the character encoder and the pre-trained word embeddings.

Proposed Model ::: Encoding Layer with Adapted Transformer

Although Transformer encoder has potential advantage in modeling long-range context, it is not working well for NER task. In this paper, we propose an adapted Transformer for NER task with two improvements.

Proposed Model ::: Encoding Layer with Adapted Transformer ::: Direction- and Distance-Aware Attention

Inspired by the success of BiLSTM in NER tasks, we consider what properties the Transformer lacks compared to BiLSTM-based models. One observation is that BiLSTM can discriminatively collect the context information of a token from its left and right sides. But it is not easy for the Transformer to distinguish which side the context information comes from.

Although the dot product between two sinusoidal position embeddings is able to reflect their distance, it lacks directionality and this property will be broken by the vanilla Transformer attention. To illustrate this, we first prove two properties of the sinusoidal position embeddings.

Property 1 For an offset k and a position t, $PE_{t+k}^TPE_{t}$ only depends on k, which means the dot product of two sinusoidal position embeddings can reflect the distance between two tokens.

Based on the definitions of Eq.(DISPLAY_FORM11) and Eq.(), the position embedding of t-th token is PEt = [c (c0t)

(c0t)

$\vdots $

(cd2-1t)

(cd2-1t)

], where d is the dimension of the position embedding, c_i is a constant decided by i, and its value is $1/10000^{2i/d}$.

Therefore,

where Eq.(DISPLAY_FORM17) to Eq.() is based on the equation $\cos (x-y) = \sin (x)\sin (y) + \cos (x)\cos (y)$.

Property 2 For an offset k and a position t, $PE_{t}^TPE_{t-k}=PE_{t}^TPE_{t+k}$, which means the sinusoidal position embeddings is unware of directionality.

Let $j=t-k$, according to property 1, we have

The relation between d, k and $PE_t^TPE_{t+k}$ is displayed in Fig FIGREF18. The sinusoidal position embeddings are distance-aware but lacks directionality.

However, the property of distance-awareness also disappears when PE_t is projected into the query and key space of self-attention. Since in vanilla Transformer the calculation between PE_t and PE_{t+k} is actually $PE_t^TW_q^TW_kPE_{t+k}$, where W_q, W_k are parameters in Eq.(DISPLAY_FORM7). Mathematically, it can be viewed as $PE_t^TWPE_{t+k}$ with only one parameter W. The relation between $PE_t^TPE_{t+k}$ and $PE_t^TWPE_{t+k}$ is depicted in Fig

FIGREF19.

Therefore, to improve the Transformer with direction- and distance-aware characteristic, we calculate the attention scores using the equations below:

where $t$ is index of the target token, $j$ is the index of the context token, $Q_t, K_j$ is the query vector and key vector of token $t, j$ respectively, $W_q, W_v \in \mathbb {R}^{d \times d_k}$. To get $H_{d_k}\in \mathbb {R}^{l \times d_k}$, we first split $H$ into $d/d_k$ partitions in the second dimension, then for each head we use one partition. $\mathbf {u} \in \mathbb {R}^{d_k}$, $\mathbf {v} \in \mathbb {R}^{d_k}$ are learnable parameters, $R_{t-j}$ is the relative positional encoding, and $R_{t-j} \in \mathbb {R}^{d_k}$, $i$ in Eq.() is in the range $[0, \frac{d_k}{2}]$. $Q_t^TK_j$ in Eq.() is the attention score between two tokens; $Q_t^TR_{t-j}$ is the $t$th token's bias on certain relative distance; $u^TK_j$ is the bias on the $j$th token; $v^TR_{t-j}$ is the bias term for certain distance and direction.

Based on Eq.(), we have

because $\sin (-x)=-\sin (x), \cos (x)=\cos (-x)$. This means for an offset $t$, the forward and backward relative positional encoding are the same with respect to the $\cos (c_it)$ terms, but is the opposite with respect to the $\sin (c_it)$ terms. Therefore, by using $R_{t-j}$, the attention score can distinguish different directions and distances.

The above improvement is based on the work BIBREF17, BIBREF19. Since the size of NER datasets is usually small, we avoid direct multiplication of two learnable parameters, because they can be represented by one learnable parameter. Therefore we do not use W_k in Eq.(DISPLAY_FORM22). The multi-head version is the same as Eq.(DISPLAY_FORM8), but we discard W_o since it is directly multiplied by W_1 in Eq.(DISPLAY_FORM9).

Proposed Model ::: Encoding Layer with Adapted Transformer ::: Un-scaled Dot-Product Attention

The vanilla Transformer use the scaled dot-product attention to smooth the output of softmax function. In Eq.(), the dot product of key and value matrices is divided by the scaling factor $\sqrt{d_k}$.

We empirically found that models perform better without the scaling factor $\sqrt{d_k}$. We presume this is because without the scaling factor the attention will be sharper. And the sharper attention might be beneficial in the NER task since only few words in the sentence are named entities.

Proposed Model ::: CRF Layer

In order to take advantage of dependency between different tags, the Conditional Random Field (CRF) was used in all of our models. Given a sequence $\mathbf {s}=[s_1, s_2, ..., s_T]$, the corresponding golden label sequence is $\mathbf {y}=[y_1, y_2, ..., y_T]$, and $\mathbf {Y}(\mathbf {s})$ represents all valid label sequences. The probability of $\mathbf {y}$ is calculated by the following equation

where $f(\mathbf {y}_{t-1},\mathbf {y}_t,\mathbf {s})$ computes the transition score from $\mathbf {y}_{t-1}$ to $\mathbf {y}_t$ and the score for $\mathbf {y}_t$. The optimization target is to maximize $P(\mathbf {y}|\mathbf {s})$. When decoding, the Viterbi Algorithm is used to find the path achieves the maximum probability.

Experiment ::: Data

We evaluate our model in two English NER datasets and four Chinese NER datasets.

(1) CoNLL2003 is one of the most evaluated English NER datasets, which contains four different named entities: PERSON, LOCATION, ORGANIZATION, and MISC BIBREF34.

(2) OntoNotes 5.0 is an English NER dataset whose corpus comes from different domains, such as telephone conversation, newswire. We exclude the New Testaments portion since there is no named entity in it BIBREF8, BIBREF7. This dataset has eleven entity names and seven value types, like CARDINAL, MONEY, LOC.

(3) BIBREF35 released OntoNotes 4.0. In this paper, we use the Chinese part. We adopted the same pre-process as BIBREF36.

(4) The corpus of the Chinese NER dataset MSRA came from news domain BIBREF37.

(5) Weibo NER was built based on text in Chinese social media Sina Weibo BIBREF38, and it contained 4 kinds of entities.

(6) Resume NER was annotated by BIBREF33.

Their statistics are listed in Table TABREF28. For all datasets, we replace all digits with “0”, and use the BIOES tag schema. For English, we use the Glove 100d pre-trained embedding BIBREF25. For the character encoder, we use 30d randomly initialized character embeddings. More details on models' hyper-parameters can be found in the supplementary material. For Chinese, we used the character embedding and bigram embedding released by BIBREF33. All pre-trained embeddings are finetuned during training. In order to reduce the impact of randomness, we ran all of our experiments at least three times, and its average F1 score and standard deviation are reported.

We used random-search to find the optimal hyper-parameters, hyper-parameters and their ranges are displayed in the supplemental material. We use SGD and 0.9 momentum to optimize the model. We run 100 epochs and each batch has 16 samples. During the optimization, we use the triangle learning rate BIBREF39 where the learning rate rises to the pre-set learning rate at the first 1% steps and decreases to 0 in the left 99% steps. The model achieves the highest development performance was used to evaluate the test set. The hyper-parameter search range and other settings can be found in the supplementary material. Codes are available at https://github.com/fastnlp/TENER.

Experiment ::: Results on Chinese NER Datasets

We first present our results in the four Chinese NER datasets. Since Chinese NER is directly based on the characters, it is more straightforward to show the abilities of different models without considering the influence of word representation.

As shown in Table TABREF29, the vanilla Transformer does not perform well and is worse than the BiLSTM and CNN based models. However, when relative positional encoding combined, the performance was enhanced greatly, resulting in better results than the BiLSTM and CNN in all datasets. The number of training examples of the Weibo dataset is tiny, therefore the performance of the Transformer is abysmal, which is as expected since the Transformer is data-hungry. Nevertheless, when enhanced with the relative positional encoding and unscaled attention, it can achieve even better performance than the

BiLSTM-based model. The superior performance of the adapted Transformer in four datasets ranging from small datasets to big datasets depicts that the adapted Transformer is more robust to the number of training examples than the vanilla Transformer. As the last line of Table TABREF29 depicts, the scaled attention will deteriorate the performance.

Experiment ::: Results on English NER datasets

The comparison between different NER models on English NER datasets is shown in Table TABREF32. The poor performance of the Transformer in the NER datasets was also reported by BIBREF16. Although performance of the Transformer is higher than BIBREF16, it still lags behind the BiLSTM-based models BIBREF5. Nonetheless, the performance is massively enhanced by incorporating the relative positional encoding and unscaled attention into the Transformer. The adaptation not only makes the Transformer achieve superior performance than BiLSTM based models, but also unveil the new state-of-the-art performance in two NER datasets when only the Glove 100d embedding and CNN character embedding are used. The same deterioration of performance was observed when using the scaled attention. Besides, if ELMo was used BIBREF28, the performance of TENER can be further boosted as depicted in Table TABREF33.

Experiment ::: Analysis of Different Character Encoders

The character-level encoder has been widely used in the English NER task to alleviate the data sparsity and OOV problem in word representation. In this section, we cross different character-level encoders (BiLSTM, CNN, Transformer encoder and our adapted Transformer encoder (AdaTrans for short)) and different word-level encoders (BiLSTM, ID-CNN and AdaTrans) to implement the NER task. Results on CoNLL2003 and OntoNotes 5.0 are presented in Table TABREF34 and Table TABREF34, respectively.

The ID-CNN encoder is from BIBREF23, and we re-implement their model in PyTorch. For different combinations, we use random search to find its best hyper-parameters. Hyper-parameters for character encoders were fixed. The details can be found in the supplementary material.

For the results on CoNLL2003 dataset which is depicted in Table TABREF34, the AdaTrans performs as good as the BiLSTM in different character encoder scenario averagely. In addition, from Table TABREF34, we can find the pattern that the AdaTrans character encoder outpaces the BiLSTM and CNN character encoders when different word-level encoders being used. Moreover, no matter what character encoder being used or none being used, the AdaTrans word-level encoder gets the best performance. This implies that when the number of training examples increases, the AdaTrans character-level and word-level encoder can better realize their ability.

Experiment ::: Convergent Speed Comparison

We compare the convergent speed of BiLSTM, ID-CNN, Transformer, and TENER in the development set of the OntoNotes 5.0. The curves are shown in Fig FIGREF37. TENER converges as fast as the BiLSTM model and outperforms the vanilla Transformer.

Conclusion

In this paper, we propose TENER, a model adopting Transformer Encoder with specific customizations for the NER task. Transformer Encoder has a powerful ability to capture the long-range context. In order to make the Transformer more suitable to the NER task, we introduce the direction-aware, distance-aware and un-scaled attention. Experiments in two English NER tasks and four Chinese NER tasks show that the performance can be massively increased. Under the same pre-trained embeddings and external knowledge, our proposed modification outperforms previous models in the six datasets.

Meanwhile, we also found the adapted Transformer is suitable for being used as the English character encoder, because it has the potentiality to extract intricate patterns from characters. Experiments in two English NER datasets show that the adapted Transformer character encoder performs better than BiLSTM and CNN character encoders.

Supplemental Material ::: Character Encoder

We exploit four kinds of character encoders. For all character encoders, the randomly initialized character embeddings are 30d. The hidden size of BiLSTM used in the character encoder is 50d in each direction. The kernel size of CNN used in the character encoder is 3, and we used 30 kernels with stride 1. For Transformer and adapted Transformer, the number of heads is 3, and every head is 10d, the dropout rate is 0.15, the feed-forward dimension is 60. The Transformer used the sinusoid position embedding. The number of parameters for the character encoder (excluding character embedding) when using BiLSTM, CNN, Transformer and adapted Transformer are 35830, 3660, 8460 and 6600 respectively. For all experiments, the hyper-parameters of character encoders stay unchanged.

Supplemental Material ::: Hyper-parameters

The hyper-parameters and search ranges for different encoders are presented in Table TABREF40, Table TABREF41 and Table TABREF42.