# DeepMet: A Reading Comprehension Paradigm for Token-level Metaphor Detection

**Chuandong Su**[1,2], **Fumiyo Fukumoto**[2], **Xiaoxi Huang**[1], **Jiyi Li**[2],
**Rongbo Wang**[1] **and Zhiqun Chen**[1]
[1]College of Computer Science and Technology, Hangzhou Dianzi University,
Hangzhou 310018, China
[2]Department of Computer Science and Engineering, University of Yamanashi,
Yamanashi 400-8510, Japan
{suchuandong,huangxx,wangrongbo,chenzq}@hdu.edu.cn
fukumoto@yamanashi.ac.jp
garfieldpigljy@gmail.com

## Abstract

Machine metaphor understanding is one of the major topics in NLP. Most of the recent attempts consider it as classification or sequence tagging task. However, few types of research introduce the rich linguistic information into the field of computational metaphor by leveraging powerful pre-training language models. We focus a novel reading comprehension paradigm for solving the token-level metaphor detection task which provides an innovative type of solution for this task. We propose an end-to-end deep metaphor detection model named DeepMet based on this paradigm. The proposed approach encodes the global text context (whole sentence), local text context (sentence fragments), and question (query word) information as well as incorporating two types of part-of-speech (POS) features by making use of the advanced pre-training language model. The experimental results by using several metaphor datasets show that our model achieves competitive results in the second shared task on metaphor detection.

## 1 Introduction

Metaphor is one of the figurative languages and often used to express our thoughts in daily conversations. It is deeply related to human cognitive processes (Lakoff and Johnson, 2003). Metaphor is used to implicitly refer one concept to another concept, usually triggered by a verb (Steen et al., 2010). For example, the verb "drink" in "*car drinks gasoline*" is a metaphorical usage. Other parts of speech can also be used metaphorically (Tsvetkov et al., 2014). For example, the noun "angel" in "*she is an angel*" and the adjective "bright" in "*your idea is very bright*" are also metaphorical uses. Metaphor computation technologies are helpful for most NLP tasks such as machine translation, dialogue systems, content analysis, and machine reading comprehension. Of these, token-level metaphor detection is the basic technology for metaphor understanding. Its task is to give a text sequence and determine whether a token in the given text sequence is a metaphor or literal. The second shared task on metaphor detection[1] aims to promote the development of metaphor detection technology. This task provides two data sets, VU Amsterdam Metaphor Corpus (VUA) (Steen, 2010) and TOEFI (a subset of ETS corpus of non-native written English) (Klebanov et al., 2018), each with two tasks. Each dataset has two tasks, i.e., verb metaphor detection and all POS metaphor detection. Previous research (Wu et al., 2018; Gao et al., 2018; Mao et al., 2019) has been limited to treat them as the text classification task or sequence tagging task without deeply investigating and leveraging the linguistic information that may be proper for the specific metaphor understanding task.

Motivated by the previous work mentioned in the above, we propose an end-to-end neural based method named DeepMet for detecting metaphor by transforming the token-level metaphor detection task into the reading comprehension task. Our approach encodes the global text, local text and question information as well as incorporating the POS features on two granularity. To improve the performance further, we also leverage the powerful pre-training language models. The F1 score of our best model reaches 80.4% and 76.9% in the verbal track and the all POS track of the VUA data set, and 74.9% and 71.5% in the verbal track and the all POS track of the TOEFL data set, respectively. Our source codes are available online[2].

The main contributions of our work can be summarized: (1) We propose a novel reading comprehension paradigm for token-level metaphor detection task. (2) We design a metaphor detec-

[1]https://competitions.codalab.org/competitions/22188
[2]https://github.com/YU-NLPLab/DeepMet



*Proceedings of the Second Workshop on Figurative Language Processing*, pages 30–39

https://doi.org/10.18653/v1/P17

tion model based on the reading comprehension paradigm which makes use of the advanced pre-training language model to encode global, local, and question information of the text as well as two types of POS auxiliary features. We also introduced a metaphor preference parameter in the cross-validation phase to improve the model performance. (3) The experimental results on several metaphor datasets show that our model is comparable to the state-of-the-art metaphor detection, especially we verified that fine-grained POS (FGPOS) features contribute to performance improvement in our model.

## 2 Related Work

### 2.1 Metaphor Detection

As a common language phenomenon, the metaphor was first studied by linguists and psycho-linguists (Wilks, 1975; Glucksberg, 2003; Group, 2007). Metaphor is related to the human cognitive process, and the essential mechanism of metaphor is the conceptual mapping from the source domain to the target domain (Lakoff and Johnson, 2003). Metaphor understanding involves high-level semantic analysis and thus requires special domain knowledge (Tsvetkov et al., 2014).

There are three types of metaphor detection methods. One is a lexicon and rule-based methods (Dodge et al., 2015; Mohler et al., 2013), while these methods need manual creation of rules which is extremely costly. The second is a corpus-based statistical algorithm. It has been studied to construct manual features such as unigrams (Klebanov et al., 2014), bag-of-words features (Köper and im Walde, 2016), concreteness, abstractness (Turney et al., 2011; Tsvetkov et al., 2014), and sensory features (Shutova et al., 2016). The disadvantage of this method is that it cannot detect rare usages of metaphors as we can hardly deal with all these unexpected linguistic phenomena. The third is a metaphor detection algorithm based on deep learning. With a recent surge of interest in neural networks, metaphor detection based on deep learning techniques has been intensively studied. Wu et al. (2018) proposed a metaphor detection model based on Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) (Graves and Schmidhuber, 2005). They utilized Word2Vec (Mikolov et al., 2013) as text representation, and POS and word clusters information for additional features. Their method performed the best in the NAACL-2018 metaphor shared task (Leong et al., 2018) with an ensemble learning strategy. Gao et al. (2018) proposed a metaphor detection model using global vectors for word representation (GloVe) (Pennington et al., 2014) and deep contextualized word representations (ELMo) (Peters et al., 2018) as text representations. They applied BiLSTM as an encoder. The accuracy of their method surpasses Wu et al.'s method. Mao et al. (2019) presented two metaphor detection models inspired by the theory of metaphor linguistics (Metaphor Identification Procedure (MLP) (Steen et al., 2010) and Selectional Preference Violation (SPV) (Wilks, 1975)), with BiLSTM as the encoder and Glove and ELMo as the word embeddings. The method is currently SOTA on metaphor detection tasks. Despite some successes, approaches explored so far use classification or sequence labeling and the encoder is based on shallow neural networks such as CNN or BiLSTM, ignoring to make use of different aspects of contexts simultaneously.

Several efforts have been made to cope with shallow neural network architectures. One attempt is Transformer based methods (Vaswani et al., 2017) such as GPT (Radford et al., 2018), BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019),and XL-Net (Yang et al., 2019). Our backbone network be based on RoBERTa, which uses robustly optimized BERT pretraining approach to improve the performance on many NLP tasks.

### 2.2 Reading Comprehension

The reading comprehension in NLP assesses a machine's understanding of NL by measuring its ability to answer questions based on a given text/document. The answer to this question may be either explicit or implicit in the text and needs to be inferred based on knowledge and logic (Seo et al., 2016; Wang and Jiang, 2016; Shen et al., 2017). It is a crucial task in NLP and a lot of approaches are presented. McCann et al. showed that many NLP tasks can be translated into reading comprehension tasks, e.g., the sentiment analysis task can be regarded as the reading comprehension task that answers the polarity of a sentence based on a given text (McCann et al., 2018). Levy et al. (2017) translated the information extraction task into the reading comprehension task with good results. Li et al. (2019) attempted to use reading comprehension to solve the NER task and also achieved good

performance on multiple NER datasets.

Inspired by the previous work mentioned in the above, we utilize a paradigm based on reading comprehension and propose a Transformer-based encoder for metaphor detection.

## 3 Methodology

### 3.1 A Reading Comprehension Paradigm for Token-level Metaphor Detection

Let $S$ ($|S| = n$) be a sentence and $w_i \in V$ be the $i$-th word within the sentence, where $V$ is the data set vocabulary and the total number of words of sentence is $n$. Similarly, let $Q$ ( $|Q| = m$) be a query word sequence within the sentence $S$ and $q_j \in V'$ be the $j$-th query word with in $Q$, where $V'$ is the query word vocabulary and the total number of query words is $m$. As shown in Figure 1, the task of the token-level metaphor detection is to predict a label sequence $Y$ ($|Y| = m$), where each $y_j \in Y$ refers to the predicted label of $q_j$ and $y_j \in \{1,0\}$ (1 denotes *metaphor* and 0 indicates *literal*). The goal of the task is to estimate the conditional probability $P(Y \mid S, Q)$.

We note that the length of the sequence $Q$ is smaller than that of $S$. This is because metaphors are generally triggered by some POS such as verbs, nouns, adjectives, and adverbs (Steen et al., 2010; Wilks, 1975). Other POS such as punctuation, prepositions, and conjunctions are unlikely to trigger metaphors. Therefore, we set the POS of a query sequence word to a verb, nouns, adjectives, and adverbs. We consider the token-level metaphor detection task to be a reading comprehension task based on a given context and query words, while previous research has regarded it as a classification or sequence tagging task.

The form of converted reading comprehension paradigm can be defined as triple $(S, q_j, y_j)$ $(S, q_j \in Q, y_j \in Y)$. The goal of the task is to estimate the conditional probability $P(y_j|S, q_j)$. For example, when the context is "car drinks gasoline" and the question is the query word "car", the correct label is 0 (literal). If the query word is changed to "*drink*", the correct label is 1 (metaphor).

Metaphor detection is a metaphor comprehension problem, and the reading comprehension task is more in line with the definition of natural language comprehension problems. In addition, reading comprehension paradigms can avoid unnecessary training. When constructing a training set triples, we can filter query words that can not be a metaphor.

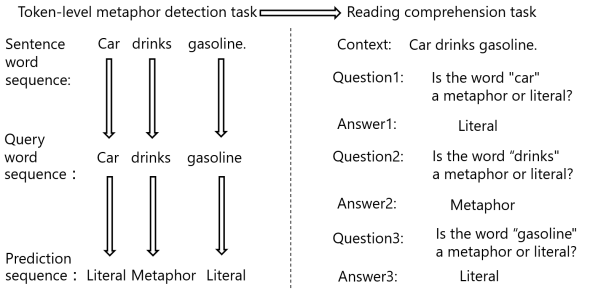


Figure 1: Schematic diagram of metaphor detection task translated into reading comprehension task.

### 3.2 DeepMet: An End-to-End Neural Metaphor Detector

We build an end-to-end neural metaphor detection model based on the reading comprehension paradigm, and the architecture is shown in Figure 2. We use the improved BERT embedding layer (Devlin et al., 2018) to represent the input information, use the byte pair encoding (BPE) algorithm (Shibata et al., 1999) to obtain the token, and use the position code represented by the yellow dots and the segment code represented by the blue dots to represent the position information of the token and distinguish the different token segments. A special classification token $[CLS]$ will be added before the first token, and special segment separation tokens $[SEP]$ will be added between different sentences. The final input is the addition of token, position encoding, and segment encoding. The improvement of our embedding layer is to use five features as input. The red dots represent the global text context, that is, the original text data. Green dots represent the local text context obtained by cutting the original text data with a comma. Orange dots indicate the features of the question, which is the query word. The purple dots indicate the general POS features, that is, the POS of the query word is represented by the POS of the verb, adjective, noun, etc. Light blue dots represent FGPOS features, using Penn Treebank POS Tags (Santorini, 1990) to represent POS, and FGPOS has a wider variety of POS features than general POS features. Different features are separated by a special segment separation token $[SEP]$.

The backbone network of our model (DeepMet) uses the Transformer encoder layer (Vaswani et al., 2017) of the siamese architecture, which uses two Transformer encoder layers to process different fea-

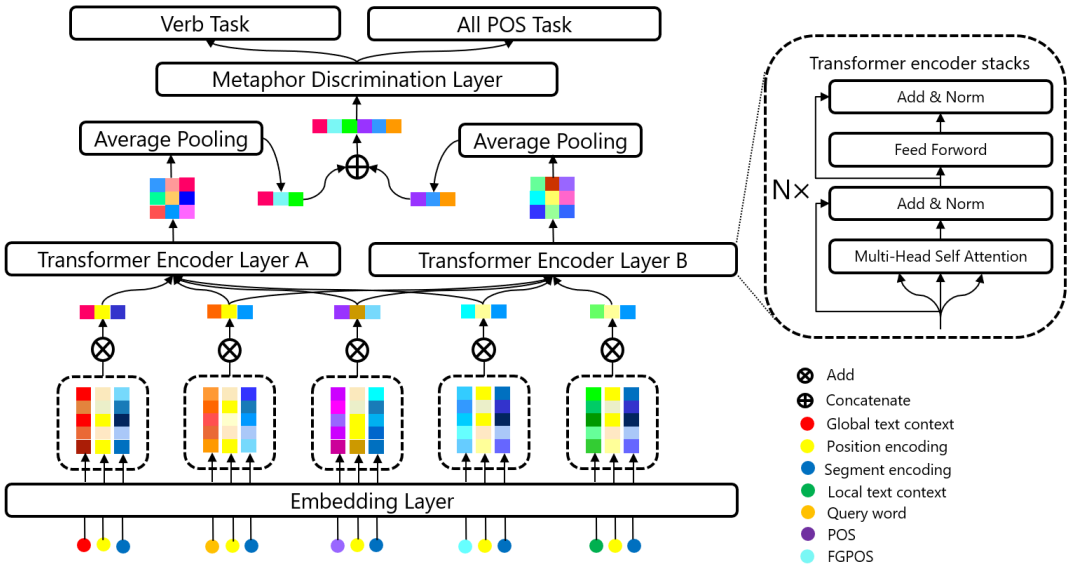


Figure 2: The overall architecture of our model (DeepMet).

ture combinations. The Transformer encoder layer A processes global text features, and the Transformer encoder layer B processes local text features. The query word and two POS features are shared by the two Transformer encoder layers. Specifically, the feature input order of Transformer coding layer A is global text context, query word, POS, FGPOS, and the feature input order of Transformer coding layer B is local text context, query word, POS, FGPOS, and the features are separated by special segment separation token $[SEP]$. The two Transformer encoder layers share weight parameters, which not only learns global and local information from different perspectives but also avoids double storage of weight parameters. The Transformer encoder layer is composed of stacked multi-headed self-attention encoders and its formula is shown in Formula (1)–(5).

$$Q^i, K^i, V^i = W_q h^{i-1}, W_k h^{i-1}, W_v h^{i-1} \quad (1)$$

$$S^i = softmax(\frac{Q^i K^i}{\sqrt{d_k}}) \quad (2)$$

$$Attention(Q, K, V) = h^i = S^i V^i \quad (3)$$

$$head_j = Attention(QW_q^j, KW_k^j, VW_v^j) \quad (4)$$

$$MultiHead(Q, K, V) = Concat_{j=1}^{n}(head_j)W_o \quad (5)$$

Among them, $i$ is the $i$-th self-attention block, $Q$, $K$, and $V$ are query matrix, key matrix, and value matrix, $h$ is the hidden state, $W_q$, $W_k$, $W_v$, $W_o$ are all self-attention mechanism weight matrices, $d_k$ is a scaling factor to counteract the effect of excessive dot product growth, $j$ is the $j$-th self-attention head and function $Concat$ is the tensor concatenation. The Transformer encoder also includes residual connections, feedforward networks (FFN), and batch normalization (BN) (Vaswani et al., 2017).

The output of these two Transformer encoder layers is a metaphor information matrix with dimensions of maximum sequence length and hidden state size, respectively. Then these two matrices are reduced by average pooling to obtain high-level metaphor feature vector with length of hidden state size, including global semantic features and local semantic features, respectively, and then stitching these two vectors into the metaphor discrimination layer. The metaphor discriminating layer first performs a dropout operation to alleviate overfitting then uses an FFN containing two neurons to obtain a metaphor discriminant vector with length equal to 2, and finally performs a $softmax$ function to obtain the metaphor and literal probability. As shown in formula (6).

$$y_\tau = softmax(V^T x + b) \quad (6)$$

$y_\tau$ is a real value vector representing metaphor and literal probability, $V$ and $b$ are the FFN parameter matrix. In the process of training the model, we use the parameter weight of pre-training language models published by Facebook (RoBERTa) (Liu et al., 2019) to fine-tune the Transformer encoder

layers. The metaphor discrimination layer will use the training method to train the model through the Adam optimizer with the adaptive learning rate. The final training goal is the cross-entropy loss function $\mathcal{L}$, which contains the loss functions $\mathcal{L}_0$ and $\mathcal{L}_1$ of the two subtasks (verb task and all POS task of metaphor detection), as shown in Formulas (7)–(9).

$$\mathcal{L}_0 = \mathcal{L}_1 = -\sum_{i=1}^{M}(\hat{y}\log y_{\tau 0} + (1-\hat{y})\log y_{\tau 1}) \quad (7)$$

$$\mathcal{L} = \mathcal{L}_0 T(t) + \mathcal{L}_1(1 - T(t)) \quad (8)$$

$$T(t) = \begin{cases} 1 & \text{if } t \text{ is } VERB \\ 0 & \text{if } t \text{ is } ALLPOS \end{cases} \quad (9)$$

where $T(t)$ $(t \in \{VERB, ALLPOS\})$ is the task selection function, $M$ is the number of training data samples, $\hat{y}$ is the real label of the data, $y_{\tau 0}$ and $y_{\tau 1}$ represent the prediction probability of whether the data belongs to metaphor and literal respectively, and $y_{\tau 0}, y_{\tau 1} \in [0, 1]$, $y_{\tau 0} + y_{\tau 1} = 1$. During the training process, we use the multi-task mode to train the metaphor detector to improve the training efficiency. Therefore, the final parameters in the task-specific metaphor feature extractor for the two subtasks is the same.

We use cross-validation to train the model to improve the training set utilization efficiency. We introduce a metaphor preference parameter $\alpha$ in this process to improve the metaphor recognition effect, as shown in formula (10).

$$P_i = \begin{cases} M & \frac{1}{N}\sum_{i=j}^{N} DeepMet_j(d_i) \geq \alpha \\ L & \frac{1}{N}\sum_{i=j}^{N} DeepMet_j(d_i) < \alpha \end{cases} \quad (10)$$

where $N$ is the number of cross-validation folds, the function $DeepMet_j$ $(0 \leq j \leq N)$ is the metaphor recognizer we designed, $d_i$ ($i$ is the index of the validation data) is the validation data and $P_i$ is the final prediction result and the results are $M$ (metaphor) and $L$ (literal meaning) respectively. Since the metaphor data sets are imbalanced, the model recall rate can be effectively improved by adjusting the metaphor preference parameter $\alpha$. For details, refer to the section 4.

## 4 Experiments and Analysis

### 4.1 Data Sets and Exploratory Data Analysis

We used four benchmark datasets: (1) VUA[3] (Steen, 2010) is currently the largest publicly available metaphor detection data set. Both of the NAACL-2018 metaphor shared task and second shared task on metaphor detection use VUA as the evaluation data set. There are two tracks, i.e., verbs and all POS metaphor detection. (2) TOEFI[4] (Klebanov et al., 2018) is a subset of ETS corpus of non-native written English. It is also used as the evaluation data set in the second shared task on metaphor detection with two tracks, verbs and all POS metaphor detection. (3) MOH-X[5] (Mohammad et al., 2016) is a verb metaphor detection database with the data from WordNet (Miller, 1998) example sentences. The average sentence length of MOH-X is the shortest among the four data sets. (4) The TroFi[6] (Birke and Sarkar, 2006) is a verb metaphor detection dataset consisting of sentences from the 1987-89 Wall Street Journal Corpus Release 1 (Charniak et al., 2000). The average sentence length of TroFi is the longest among the four data sets.

We first sampled the four data sets into four new $(S, q_i, y_j)$ triple data sets following the requirements of the reading comprehension paradigm. In this paper, we focus on the VUA and TOEFI as the evaluation data set. MOH-X and TOEFI are used as auxiliary data sets to verify the performance of our designed metaphor detector. We made exploratory data analysis on the data sets of VUA and TOEFI. The label distribution of data sets is shown in Figure 3.

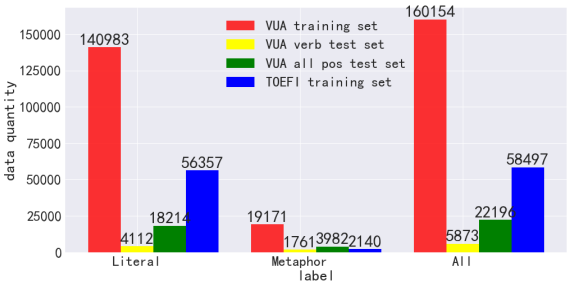


Figure 3: Distribution of label categories.

There are more literal data in VUA and TOEFI than metaphor data, indicating that both data sets are unbalanced. Unbalanced data sets may affect the performance of metaphor detectors. The distribution of the sentence length in the data set is shown in Figure 4.

As we can see from Figure 4 that the distribution of the sentence length distribution by both training

[3]http://ota.ahds.ac.uk/headers/2541.xml

[4]https://catalog.ldc.upenn.edu/LDC2014T06

[5]http://saifmohammad.com/WebPages/metaphor.html

[6]http://natlang.cs.sfu.ca/software/trofi.html

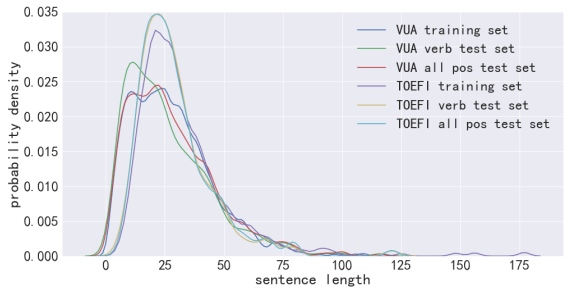


Figure 4: Sentence length distribution of different data sets.

and test set are similar. Similarly, the distribution of query word's POS and its label in the data sets are shown in Figure 5 and Figure 6. The most likely POS of query words triggering metaphor in the two data sets are verbs, nouns, and adjectives. We can delete the triplet data of query words whose POS are other than those POS as these query words are few possibilities as a trigger metaphor.

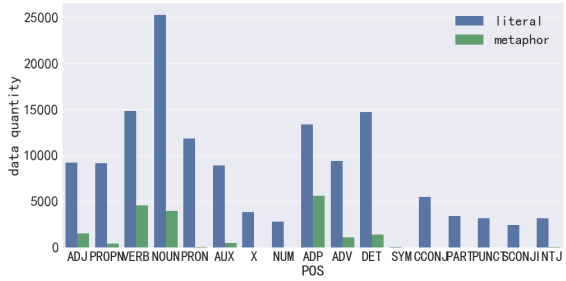


Figure 5: The relationship between POS and label of query words in VUA dataset.

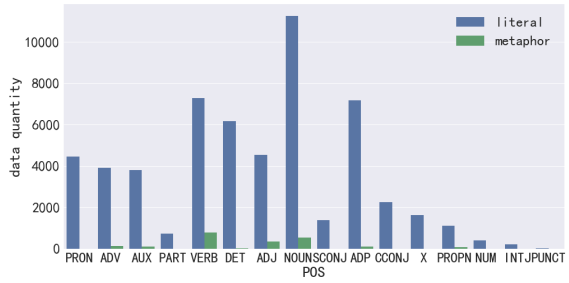


Figure 6: The relationship between POS and label of query words in TOEFI dataset.

### 4.2 Baselines

We use four baselines to compare the performance of different metaphor detectors: (1) Word2Vec+CNN+BiLSTM+Ensemble (Wu et al., 2018) is the best model in the NAACL-2018 Metaphor Shared Task. The model is based on a sequence tagging paradigm by using CNN and BiLSTM as encoders, Word2Vec, POS tags and word clusters as features, and it is further improved performance through ensemble learning. (2) ELMo+BiLSTM (Gao et al., 2018) is a metaphor detection model based on classification and sequence labeling paradigm by using ELMo as feature representations, and BiLSTM as an encoder. (3) Glove+ELMo+BiLSTM+Attention (Mao et al., 2019) is a metaphor detection model based on sequence tagging paradigm by using GloVe and ELMo as feature representations, BiLSTM and attention mechanism as encoders. To the best of our knowledge, this model is the best among others in the benchmark data sets. (4) BERT+BiLSTM (Mao et al., 2019) is a metaphor detection model based on the sequence labeling paradigm with BERT output vector as the feature and BiLSTM as the encoder.

Table 1: The value of the hyperparameters.

| Hyperparameters | Vaule |
|---|---|
| Sequencey length | 128 |
| Batches | 16 |
| Initial learning rate | 1e-5 |
| Dropout rate | 0.2 |
| Epochs | 3 |
| Cross-validation folds | 10 |

### 4.3 Data Preprocessing and Hyperparameters Setting

Our evaluation metrics for metaphor detection tasks are accuracy (A), precision (P), recall (R) and F1 measure (F1), which are the most commonly used evaluation metrics for metaphor detection tasks. We used the default hyperparameters of RoBERTa (Liu et al., 2019) and estimated them by using a grid search within a reasonable range. Each value of the hyperparameters is shown in Table 1.

First, we preprocess the data into the triple format $(S, q_i, y_j)$ required by the reading comprehension paradigm. We remove triples whose query words are punctuation marks, and it was included about 10% among the data. We use the Spacy[7] framework to obtain the query word POS and FG-POS features needed by the experiments. The pre-training language model directly encodes the data into dynamic word embeddings. The best model parameter weight in the validation set is the final model parameter weight. We divided the data into two folds, training and verification sets consisting of 90%and 10% of the data, respectively. We used ten folds cross-validation throughout the experi-

[7]https://spacy.io

ments.

### 4.4 Experimental results and Analysis

The results are shown in Table 2. Overall, we can see that our metaphor detector (DeepMet) attained at the best performance in each of the four metaphor detection data sets. To verify the factors that affect the performance of DeepMet, we conducted ablation experiments on the model. The results are shown in Table 3.

The experimental results show that FGPOS features have a greater impact on the model than POS features, which shows that the fine-grained POS information provided by FGPOS features is better than ordinary POS information. At the level of the model structure, we also designed corresponding ablation experiments. The experimental results show that the influence of Transformer encoder layer A on the model is greater than that of Transformer encoder layer B, which indicates that the global text information extracted by Transformer encoder layer A is better than local text information extracted by Transformer encoder layer B. Moreover, the ensemble learning of DeepMet with different hyperparameters can also improve about a 3% in the F1 score.

From the experimental results of metaphor detection on four datasets, we can see that the metaphor detection model based on the reading comprehension paradigm can achieve competitive results. Global and local information and two POS features are also helpful to improve the performance of the model. Global and local information contains two kinds of granularity context, which is helpful for the model to extract different granularity text features. FGPOS and POS contain two kinds of granularity POS information, which give the model more abundant query word features. POS features are related to the POS of query words, which can capture implicit knowledge of the model. One reason why DeepMet is better than the previous baseline is that the reading comprehension paradigm can model the nature of the metaphor comprehension problem better, and the Transformer encoder works well than that of general deep learning models such as CNN and BiLSTM.

Moreover, metaphors are used less frequently than ordinary words, and all of the experimental data are unbalanced data sets, i.e., the number of literal sentences are larger than those of metaphor sentences. We thus introduced the metaphor preference parameter $\alpha$ to help the recall value of the model. The results are shown in Figure 7.

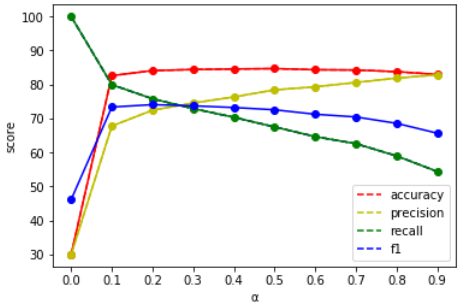


Figure 7: Influence of metaphor preference parameter $\alpha$ on model performance in VUA verb task test set.

As can be seen clearly from Figure 7, the recall score can be improved by using lower $\alpha$, while the accuracy will be reduced if $\alpha$ is too small. Our experiments show that the best F1 score can be obtained by controlling the metaphor preference parameter $\alpha$ to 0.2 or 0.3.

Through the experiments, we can conclude: (1) Metaphor detection based on the reading comprehension paradigm is feasible, and we obtained competitive results. (2) Ablation experiments indicate that global information, local information, and POS are helpful for metaphor detection. (3) In the cross-validation stage, the introduction of metaphor preference parameter and model ensemble learning can further improve the performance of the metaphor detector.

### 4.5 Error Analysis

We analyzed the data which could not predict correctly. The ambiguous annotation will make our model incorrectly predict. For example, "*The Health Secretary accused the unions of 'posturing and pretending' to run a 999 service yesterday*" (VUA ID: a7w-fragment01_29), in which the underlined words are labeled as metaphors. Although our model detects "accused" as the literal meaning, it is difficult for even human to judge whether "*accused*" is a metaphor or literal meaning. It is also challenging to detect metaphors triggered by multiple words. For example, "*I stared at Jackson Chatterton , and at last sensed the drama that lay behind his big calm presence.*" (VUA ID: ccw-fragment04_2095). In our model, the detection result of "*big*" is a false negative, and "*drama that lay behind his big calm presence*" triggers metaphor together. However, our model only questions one word at a time, so it causes misjudgment that "*big*"

Table 2: Performance of different models on different datasets. * indicates $p<0.01$ in two-tailed t-test, bold indicates best result.

| Model | Dataset | A | P | R | F1 |
|---|---|---|---|---|---|
| Word2Vec+CNN+BiLSTM+Ensemble | VUA-verb | - | 60.0 | 76.3 | 67.2 |
| ELMo+BiLSTM | VUA-verb | 81.4 | 68.2 | 71.3 | 69.7 |
| Glove+ELMo+BiLSTM+Attention | VUA-verb | 82.1 | 69.3 | 72.3 | 70.8 |
| BERT+BiLSTM | VUA-verb | 80.7 | 66.7 | 71.5 | 69.0 |
| DeepMet | VUA-verb | **88.0** | **78.9** | **81.9** | **80.4*** |
| Word2Vec+CNN+BiLSTM+Ensemble | VUA-allpos | - | 60.8 | 70.0 | 65.1 |
| ELMo+BiLSTM | VUA-allpos | 93.1 | 71.6 | 73.6 | 72.6 |
| Glove+ELMo+BiLSTM+Attention | VUA-allpos | **93.8** | 73.0 | 75.7 | 74.3 |
| BERT+BiLSTM | VUA-allpos | 92.9 | 71.5 | 71.9 | 71.7 |
| DeepMet | VUA-allpos | 91.6 | **75.6** | **78.3** | **76.9*** |
| DeepMet | TOEFI-verb | - | 73.3 | 76.6 | 74.9 |
| DeepMet | TOEFI-allpos | - | 69.5 | 73.5 | 71.5 |
| ELMo+BiLSTM | MOH-X | 77.2 | 79.1 | 73.5 | 75.6 |
| Glove+ELMo+BiLSTM+Attention | MOH-X | 79.8 | 77.5 | 83.1 | 80.0 |
| BERT+BiLSTM | MOH-X | 78.1 | 75.1 | 81.8 | 78.2 |
| DeepMet | MOH-X | **92.3** | **93.3** | **90.3** | **91.8*** |
| ELMo+BiLSTM | TroFi | 74.6 | 70.7 | 71.6 | 71.1 |
| Glove+ELMo+BiLSTM+Attention | TroFi | 75.2 | 68.6 | 76.8 | 72.4 |
| BERT+BiLSTM | TroFi | 73.4 | 70.3 | 67.1 | 68.7 |
| DeepMet | TroFi | **77.0** | **72.1** | **80.6** | **76.1*** |

Table 3: Experimental results of ablation experiments. w/o indicates ablation of features or network structures.

| Model | Dataset | A | P | R | F1 |
|---|---|---|---|---|---|
| DeepMet | VUA-verb | 88.0 | 78.9 | 81.9 | 80.4 |
| w/o POS | VUA-verb | 86.0 | 76.7 | 76.8 | 76.7 |
| w/o FGPOS | VUA-verb | 85.1 | 72.7 | 80.5 | 76.4 |
| w/o Transformer Encoder Layer A | VUA-verb | 85.7 | 75.4 | 77.3 | 76.4 |
| w/o Transformer Encoder Layer B | VUA-verb | 85.6 | 73.9 | 80.2 | 76.9 |
| w/o ensemble learning | VUA-verb | 86.2 | 76.2 | 78.3 | 77.2 |
| DeepMet | VUA-allpos | 91.6 | 75.6 | 78.3 | 76.9 |
| w/o POS | VUA-allpos | 90.5 | 74.7 | 71.2 | 72.9 |
| w/o FGPOS | VUA-allpos | 89.8 | 70.5 | 74.2 | 72.3 |
| w/o Transformer Encoder Layer A | VUA-allpos | 90.2 | 73.2 | 71.2 | 72.2 |
| w/o Transformer Encoder Layer B | VUA-allpos | 90.6 | 74.0 | 73.2 | 73.6 |
| w/o ensemble learning | VUA-allpos | 90.5 | 73.8 | 73.2 | 73.5 |

is not a metaphor.

## 5 Conclusion and Future Work

This paper proposed a reading comprehension paradigm for metaphor detection. According to this reading comprehension paradigm, we designed an end-to-end neural metaphor detector, which processes global and local information of the text through the transformer encoder, and introduces two POS with different granularity as additional features. Throughout the experiments on four metaphor detection data sets, we found that the model works well, and a competitive result is achieved good performance in the second metaphor detection sharing task. We also designed ablation experiments to verify the influence factors of the model and found that fine-grained POS and global text information is more helpful to the metaphor detection ability of the model.

There are a number of interesting directions for future work: (1) Metaphor is a special figurative language and we will extend our research methods to other figurative languages such as metonymy, simile, satire, and pun. (2) We will introduce linguistic theory into our framework to make a deep learning model more explanatory. (3) Through error analysis, we find that the multiple words trigger metaphor will affect the performance of the metaphor detection model. We will consider the multi-word question metaphor detection based on the reading comprehension paradigm.

## Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. This work was supported in part by the Grant-in-

aid for JSPS, the Support Center for Advanced Telecommunications Technology Research Foundation under Grant 17K00299, the Humanities and Social Sciences Research Program Funds from the Ministry of Education of China under Grant 18YJA740016, and the Major Projects of the National Social Science Foundation of China under Grant 18ZDA290.

## References

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

E Charniak, D Blaheta, N Ge, K Hall, J Hale, and M Johnson. 2000. Bllip 1987-89 wsj corpus release 1. linguistic data consortium ldc2000t43.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. MetaNet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*. Association for Computational Linguistics.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sam Glucksberg. 2003. The psycholinguistics of metaphor. *Trends in Cognitive Sciences*, 7(2):92–96.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.

Beata Beigman Klebanov, Ben Leong, Michael Heilman, and Michael Flor. 2014. Different texts, same metaphors: Unigrams and beyond. In *Proceedings of the Second Workshop on Metaphor in NLP*. Association for Computational Linguistics.

Beata Beigman Klebanov, Chee Wee (Ben) Leong, and Michael Flor. 2018. A corpus of non-native written english annotated for metaphor. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics.

Maximilian Köper and Sabine Schulte im Walde. 2016. Distinguishing literal and non-literal usage of german particle verbs. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

George Lakoff and Mark Johnson. 2003. *Metaphors We Live By*. University of Chicago Press.

Chee Wee (Ben) Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 VUA metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Rui Mao, Chenghua Lin, and Frank Guerin. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 27–35.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Beatrice Santorini. 1990. Part-of-speech tagging guidelines for the penn treebank project.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Gerard Steen. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*, volume 14. John Benjamins Publishing.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna A. Kaal, and Tina Krennmayr. 2010. Metaphor in usage. *Cognitive Linguistics*, 21(4).

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Peter D Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.

Chuhan Wu, Fangzhao Wu, Yubo Chen, Sixing Wu, Zhigang Yuan, and Yongfeng Huang. 2018. Neural metaphor detecting with CNN-LSTM model. In *Proceedings of the Workshop on Figurative Language Processing*. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.