

MelBERT: Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories

Minjin Choi¹, Sunkyung Lee¹, Eunseong Choi¹, Heesoo Park²,
Junhyuk Lee¹, Dongwon Lee³ and Jongwuk Lee¹

¹Sungkyunkwan University, Republic of Korea, ²Bering Lab, Republic of Korea,

³The Pennsylvania State University, United States

¹{zxcvxd, sk1027, eunseong, ljhjoon00, jongwuklee}@skku.edu

²heesoo.park@beringlab.com ³dongwon@psu.edu

Abstract

Automated metaphor detection is a challenging task to identify the metaphorical expression of words in a sentence. To tackle this problem, we adopt pre-trained contextualized models, *e.g.*, BERT and RoBERTa. To this end, we propose a novel metaphor detection model, namely *metaphor-aware late interaction over BERT (MelBERT)*. Our model not only leverages contextualized word representation but also benefits from linguistic metaphor identification theories to detect whether the target word is metaphorical. Our empirical results demonstrate that MelBERT outperforms several strong baselines on four benchmark datasets, *i.e.*, VUA-18, VUA-20, MOH-X, and TroFi.

1 Introduction

As the conceptual and cognitive mapping of words, a *metaphor* is a common language expression representing other concepts rather than taking literal meanings of words in context (Lakoff and Johnson, 1980; Lagerwerf and Meijers, 2008). For instance, in the sentence “hope is on the horizon,” the word “horizon” does not literally mean the line at the earth’s surface. It is a metaphorical expression to describe a positive situation. Therefore, the meaning of “horizon” is context-specific and different from its literal definition.

As the metaphor plays a key role in cognitive and communicative functions, it is essential to understand contextualized and unusual meanings of words (*e.g.*, metaphor, metonymy, and personification) in various natural language processing (NLP) tasks, *e.g.*, machine translation (Shi et al., 2014), sentiment analysis (Cambria et al., 2017), and dialogue systems (Dybala and Sayama, 2012). A lot of existing studies have developed various computational models to recognize metaphorical words in a sentence.

Automated metaphor detection aims at identifying metaphorical expressions using computational

models. Existing studies can be categorized into three pillars. First, feature-based models employ various hand-crafted features (Shutova et al., 2010; Turney et al., 2011; Shutova and Sun, 2013; Broadwell et al., 2013; Tsvetkov et al., 2014; Bulat et al., 2017). Although simple and intuitive, they are highly sensitive to the quality of a corpus. Second, some studies (Wu et al., 2018; Gao et al., 2018; Mao et al., 2019) utilize recurrent neural networks (RNNs), which are suitable for analyzing the sequential structure of words. However, they are limited to understanding the diverse meanings of words in context. Lastly, the pre-trained contextualized models, *e.g.*, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), have been used for detecting metaphors (Chen et al., 2020; Gong et al., 2020; Su et al., 2020). Owing to the powerful representation capacity, such models have been successful for addressing various NLP tasks (Wang et al., 2019) and document ranking in IR (Mittra and Craswell, 2018).

Based on such an advancement, we utilize a contextualized model using two metaphor identification theories, *i.e.*, *Metaphor Identification Procedure (MIP)* (Pragglejaz Group, 2007; Steen et al., 2010) and *Selectional Preference Violation (SPV)* (Wilks, 1975, 1978). For MIP, a metaphorical word is recognized if the literal meaning of a word is different from its contextual meaning (Haagsma and Bjerva, 2016). For instance, in the sentence “Don’t twist my words”, the contextual meaning of “twist” is “to distort the intended meaning”, different from its literal meaning, “to form into a bent, curling, or distorted shape.” For SPV, a metaphorical word is identified if the target word is unusual in the context of its surrounding words. That is, “twist” is metaphorical because it is unusual in the context of “words.” Although the key ideas of the two strategies are similar, they have different procedures for detecting metaphorical words and their contexts in the sentence.

To this end, we propose a novel metaphor detection model using metaphorical identification theories over the pre-trained contextualized model, namely *metaphor-aware late interaction over BERT* (MelBERT). MelBERT deals with a classification task to identify whether a target word in a sentence is metaphorical or not. As depicted in Figure 2, MelBERT is based on a siamese architecture that takes two sentences as input. The first sentence is a sentence S with a target word w_t and the second sentence is a target word w_t itself. MelBERT independently encodes S and w_t into each embedding vector, which avoids unnecessary interactions between S and w_t . Inspired by MIP, MelBERT then employs the contextualized and isolated representations of w_t to distinguish between the contextual and literal meaning of w_t . To utilize SPV, MelBERT employs the sentence embedding vector and the contextualized target word embedding vector. MelBERT identifies how much the surrounding words mismatch from the target word. Lastly, MelBERT combines two metaphor identification strategies to predict if a target word is metaphorical or not. Each metaphor identification theory is non-trivial for capturing complicated and vague metaphorical words. To overcome these limitations, we incorporate two linguistic theories into a pre-trained contextualized model and utilize several linguistic features such as POS features.

To summarize, MelBERT has two key advantages. First, MelBERT effectively employs the contextualized representation to understand various aspects of words in context. Because MelBERT is particularly based on a late interaction over contextualized models, it can prevent unnecessary interactions between two inputs and effectively distinguish the contextualized meaning and the isolated meaning of a word. Second, MelBERT utilizes two metaphor identification theories to detect whether the target word is metaphorical. Experimental results show that MelBERT consistently outperforms state-of-the-art metaphor detection models in terms of F1-score on several benchmark datasets, such as VUA-18, VUA-20, and VUA-Verb datasets.

2 Related Work

2.1 Metaphor Detection

Feature-based approach. Various linguistic features are used to understand metaphorical expressions. Representative hand-engineered features include word abstractness and concreteness (Tur-

ney et al., 2011), word imageability (Broadwell et al., 2013), semantic supersenses (Tsvetkov et al., 2014), and property norms (Bulat et al., 2017). However, they have difficulties handling rare usages of metaphors because the features rely on manually annotated resources. To address this problem, sparse distributional features (Shutova et al., 2010; Shutova and Sun, 2013) and dense word embeddings (Shutova et al., 2016; Rei et al., 2017), *i.e.*, Word2Vec (Mikolov et al., 2013), are used as better linguistic features. For details, refer to the survey (Veale et al., 2016).

RNN-based approach. Several studies proposed neural metaphor detection models using recurrent neural networks (RNNs). (Wu et al., 2018) adopts a bidirectional-LSTM (BiLSTM) (Graves and Schmidhuber, 2005) and a convolutional neural network (CNN) using Word2Vec (Mikolov et al., 2013) as text features in addition to part-of-speech (POS) and word clustering information as linguistic features. (Gao et al., 2018) employs BiLSTM as an encoder using GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018) as text input representation. (Mao et al., 2019) makes use of the metaphor identification theory on top of the architecture of (Gao et al., 2018). Despite their success, the shallow neural networks (*e.g.*, BiLSTM and CNN) have limitations on representing various aspects of words in context.

Contextualization-based approach. Recent studies utilize pre-trained contextualized language models, *e.g.*, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), for metaphor detection. Because the pre-trained model can encode rich semantic and contextual information, it is useful for detecting metaphors with fine-tuning training. DeepMet (Su et al., 2020) utilizes RoBERTa with various linguistic features, *i.e.*, global text context, local text context, and POS features. IlliniMet (Gong et al., 2020) combines RoBERTa with linguistic information obtained from external resources. (Chen et al., 2020) formulates the multi-task learning problem for both metaphor detection, and (Leong et al., 2020) reports the results of these models in the VUA 2020 shared task.

2.2 Semantic Matching over BERT

The key idea of neural semantic matching is that neural models encode a query-document pair into two embedding vectors and compute a relevance score between the query and the document (Mi-

tra and Craswell, 2018). The simple approach is to feed a query-document pair to BERT (Devlin et al., 2019) and compute a relevance score, where the query and the document are fully interacted (Nogueira et al., 2019; Dai and Callan, 2020). In contrast, SBERT (Reimers and Gurevych, 2019), TwinBERT (Lu et al., 2020), and ColBERT (Khattab and Zaharia, 2020) adopt late interaction architectures using siamese BERT, where the query and the document are encoded independently. Our work is based on the late interaction architecture. In other words, the sentence with the target word and the target word is encoded separately to represent contextualized and isolated meanings of the target word.

3 MelBERT

In this section, we propose a novel metaphor detection model over a pre-trained contextualized model. To design our model, we consider two metaphor detection tasks. Given a sentence $S = \{w_1, \dots, w_n\}$ with n words and a target word $w_t \in S$, the classification task predicts the metaphoricity (*i.e.*, metaphorical or literal) of w_t . Given a sentence S , the sequence labeling predicts the metaphoricity of each word w_t ($1 \leq t \leq n$) in S .

We aim at developing a metaphor detection model for the classification task. Our model returns a binary output, *i.e.*, 1 if the target word w_t in S is metaphorical or 0 otherwise. By sequentially changing the target word w_t , our model can be generalized to classify the metaphoricity of each word in a sentence, as in sequence labeling.

3.1 Motivation

The pre-trained language models, *e.g.*, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), usually take two sentences as input and return output to predict the relevance between two input sentences. We adopt RoBERTa as the contextualized backbone model because RoBERTa is known to outperform BERT (Liu et al., 2019). To design a metaphor detection model, we treat one input sentence as a single word (or a phrase).

As depicted in Figure 1, there are two paradigms for representing the interaction between two input sentences: *all-to-all interaction* and *late interaction*, as discussed in the document ranking problem (Khattab and Zaharia, 2020). While all-to-all interaction takes two input sentences together as an input, late interaction encodes two sentences

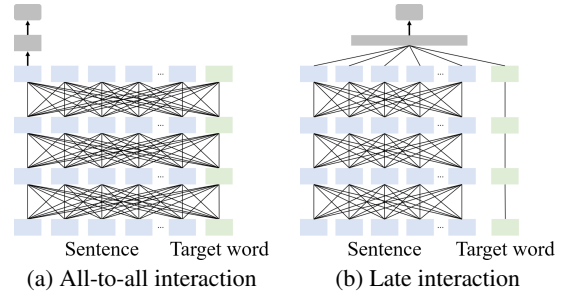


Figure 1: Two interaction paradigms over a contextualized model.

separately over a siamese architecture. Given a sentence S and a target word w_t , all-to-all interaction can capture all possible interactions within and across w_t and S , which incurs high computational cost. Moreover, when some interactions across w_t and S are useless, it may learn noisy information. In contrast, because late interaction encodes w_t and S independently, it naturally avoids unnecessary intervention across w_t and S . The sentence embedding vector also can be easily reused in computing the interaction with the target word. In other words, the cost of encoding the sentence vector can be amortized for that of encoding different target words.

Because our goal is to identify whether the contextualized meaning of the target word w_t is different from its isolated meaning, we adopt the late interaction paradigm for metaphor detection. Our model encodes a sentence S with a target word and a target word w_t into embedding vectors, respectively, and computes the metaphoricity score of the target word. (In Section 4, it is found that our model using late interaction outperforms a baseline model using all-to-all interaction.)

3.2 Model Architecture

We propose a novel metaphor detection model, namely, *metaphor-aware late interaction over BERT (MelBERT)* using metaphor identification theories, *i.e.*, *Metaphor Identification Procedure (MIP)* (Pragglejaz Group, 2007; Steen et al., 2010) and *Selectional Preference Violation (SPV)* (Wilks, 1975, 1978). Figure 2 illustrates the overall architecture of MelBERT, which consists of three components: a sentence encoder $Enc(S)$, a target word encoder $Enc(w_t)$, and a late interaction mechanism to compute a score.

We first explain the input layer for two encoders $Enc(S)$ and $Enc(w_t)$. Each word in the sentence is converted to tokens using an improved implementation of byte-pair encoding (BPE) (Radford

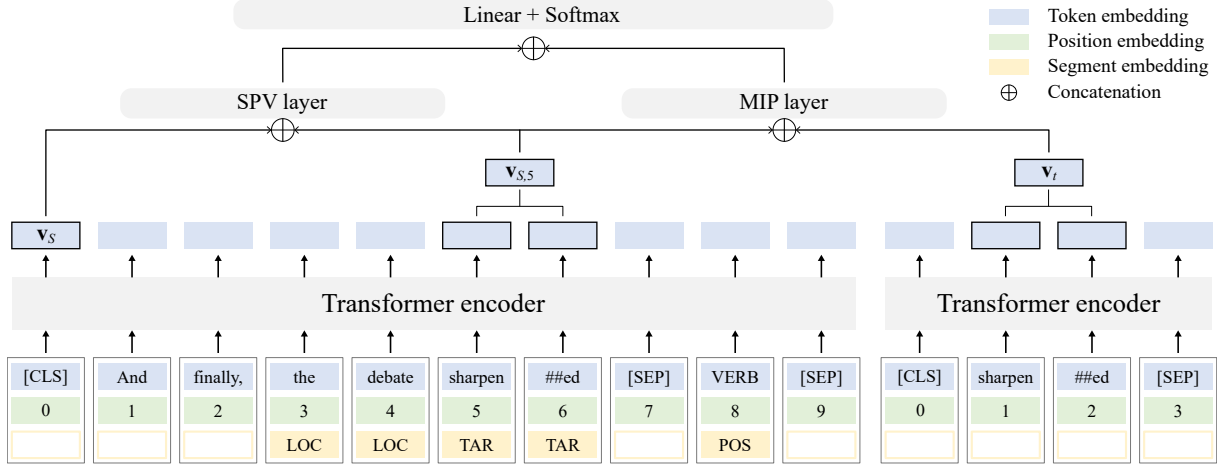


Figure 2: Model architecture of MelBERT. When a target word w_t is split into multiple tokens by BPE, the average pooling is used for the target word.

et al., 2019). As shown in the original BERT, the position embedding is used to represent the position of tokens. The segment embedding is used to distinguish target tokens (denoted as [TAR]) and their local context (denoted as [LOC]). When the sentence is represented as a composite sentence, the local context indicates a clause including target tokens. For simplicity, we represent the local context using comma separator (,) in the sentence. Besides, we add a special classification token [CLS] before the first token and a segment separation token [SEP] after the last token. To make use of the POS feature of the target word, we append the POS tag for the target word after [SEP], as used in (Su et al., 2020). The input representation is finally computed by the element-wise addition of token, position embedding, and segment embedding. For $Enc(w_t)$, the target word is converted to the tokens using BPE, but position and segment embedding are not used.

Given a sentence $S = \{w_1, \dots, w_n\}$, $Enc(S)$ encodes each word into a set of contextualized embedding vectors, $\{\mathbf{v}_S, \mathbf{v}_{S,1}, \dots, \mathbf{v}_{S,n}\}$ using the transformer encoder (Vaswani et al., 2017), where \mathbf{v}_S is the embedding vector corresponding to the [CLS] token and $\mathbf{v}_{S,i}$ is the i -th embedding vector for w_i in S . Similarly, $Enc(w_t)$ encodes a target word w_t into \mathbf{v}_t without context.

$$\mathbf{v}_S, \mathbf{v}_{S,1}, \dots, \mathbf{v}_{S,n} = Enc(["CLS], w_1, \dots, w_n, [SEP]") \quad (1)$$

$$\mathbf{v}_t = Enc(["CLS], w_t, [SEP]") \quad (2)$$

While \mathbf{v}_S reflects the interaction across all words in S , $\mathbf{v}_{S,t}$ considers the interaction between w_t and

other words in S . Therefore, $\mathbf{v}_{S,t}$ and \mathbf{v}_t can be interpreted as different meanings for w_t , i.e., $\mathbf{v}_{S,t}$ is contextualized representation of w_t and \mathbf{v}_t is isolated representation of w_t .

Then, we utilize two metaphor identification theories using contextualized embedding vectors.

MelBERT using MIP. The basic idea of MIP is that a metaphorical word is identified by the gap between the contextual and literal meaning of a word. To incorporate MIP into MelBERT, we employ two embedding vectors $\mathbf{v}_{S,t}$ and \mathbf{v}_t , representing a contextualized embedding vector and an isolated embedding vector for w_t , respectively. Using these vectors, we identify the semantic gap for the target word in context and isolation.

MelBERT using SPV. The idea of SPV is that a metaphorical word is identified by the semantic difference from its surrounding words. Unlike MIP, we only utilize the sentence encoder. Given a target word w_t in S , our key assumption is that \mathbf{v}_S and $\mathbf{v}_{S,t}$ show a semantic gap if w_t is metaphorical. Although \mathbf{v}_S and $\mathbf{v}_{S,t}$ are contextualized, the meanings of the two vectors are different; \mathbf{v}_S represents the interaction across all pair-wise words in S , but $\mathbf{v}_{S,t}$ represents the interaction between w_t and other words in S . In this sense, when w_t is metaphorical, $\mathbf{v}_{S,t}$ can be different from \mathbf{v}_S by the surrounding words of w_t .

Late interaction over MelBERT. Using the two strategies, MelBERT predicts whether a target word $w_t \in S$ is metaphorical or not. We can compute a hidden vector h_{MIP} by concatenating $\mathbf{v}_{S,t}$ and \mathbf{v}_t for MIP.

$$h_{MIP} = f([\mathbf{v}_{S,t}; \mathbf{v}_t]), \quad (3)$$

where $h_{MIP} \in \mathbb{R}^{h \times 1}$ and $f(\cdot)$ is a function for the MLP layer to learn the gap between two vectors $\mathbf{v}_{S,t}$ and \mathbf{v}_t .

We can also compute a hidden vector h_{SPV} using \mathbf{v}_S and $\mathbf{v}_{S,t}$ for SPV.

$$h_{SPV} = g([\mathbf{v}_S; \mathbf{v}_{S,t}],) \quad (4)$$

where $h_{SPV} \in \mathbb{R}^{h \times 1}$ and $g(\cdot)$ is a function for the MLP layer to learn the semantic difference between \mathbf{v}_S and $\mathbf{v}_{S,t}$.

We combine two hidden vectors h_{MIP} and h_{SPV} to compute a prediction score:

$$\hat{y} = \sigma(W^> [h_{MIP}; h_{SPV}] + b), \quad (5)$$

where $\sigma(\cdot)$ is the sigmoid function, $W \in \mathbb{R}^{2h \times 1}$ is the parameter, and b is a bias. To learn MelBERT, finally, we use the cross-entropy loss function for binary classification as follows:

$$\mathcal{L} = \sum_{i=1}^N y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i), \quad (6)$$

where N is the number of samples in the training set. y_i and \hat{y}_i are the true and predicted labels for the i -th sample in the training set.

4 Evaluation

In this section, we first present the experimental setup, then report empirical results by comparing our model against strong baselines.

4.1 Experimental Setup

Datasets. We use four well-known public English datasets. First, the VU Amsterdam Metaphor Corpus (VUA) has been released in metaphor detection shared tasks in 2018 and 2020. We use two versions of VUA datasets, called VUA-18 (Leong et al., 2018) and VUA-20 (Leong et al., 2020), where VUA-20 is the extension of VUA-18. Let $VUA-18_{tr}$, $VUA-18_{dev}$, $VUA-18_{te}$ denote the training, validation, and test datasets, split from VUA-18. $VUA-20_{tr}$ includes $VUA-18_{tr}$ and $VUA-18_{dev}$. $VUA-20_{te}$ also includes $VUA-18_{te}$, and $VUA-Verb_{te}$ is a subset of $VUA-18_{te}$ and $VUA-20_{te}$.

Because most of the tokens in a sentence are literal words in VUA-18, VUA-20 selectively chooses the tokens in the training and testing datasets. $VUA-18_{te}$ consists of four genres, including news, academic, fiction, and conversation. It can also be categorized into different POS tags, such as verb, noun, adjective, and adverb. Additionally, we employ MOH-X (Mohammad et al., 2016)

Dataset	#tokens	%M	#Sent	Sent len
VUA-18 _{tr}	116,622	11.2	6,323	18.4
VUA-18 _{dev}	38,628	11.6	1,550	24.9
VUA-18 _{te}	50,175	12.4	2,694	18.6
VUA-20 _{tr}	160,154	12.0	12,109	15
VUA-20 _{te}	22,196	17.9	3,698	15.5
VUA-Verb _{te}	5,873	30	2,694	18.6
MOH-X	647	48.7	647	8
TroFi	3,737	43.5	3,737	28.3

Table 1: Detailed statistics on benchmark datasets. #tokens is the number of tokens, %M is the percentage of metaphorical words, #Sent is the number of sentences, and Sent len is the average length of sentences.

and TroFi (Birke and Sarkar, 2006) for testing purposes only. MOH-X is a verb metaphor detection dataset with the sentences from WordNet and TroFi is also a verb metaphor detection dataset, including sentences from the 1987-89 Wall Street Journal Corpus Release 1. The sizes of these datasets are relatively smaller than those of VUA datasets, and they have metaphorical words of more than 40%, while VUA-18 and VUA-20 datasets have about 10% of metaphorical words. While MOH-X and TroFi only annotate verbs as metaphorical words, the VUA dataset annotates all POS tags as metaphorical words. In this sense, we believe that the VUA dataset is more appropriate for training and testing models. Table 1 summarizes detailed statistics on the benchmark datasets.

Baselines. We compare our models with several strong baselines, including RNN-based and contextualization-based models.

- **RNN_ELMo** and **RNN_BERT** (Gao et al., 2018): They employ the concatenation of the pre-trained ELMo/BERT and the GloVe (Pennington et al., 2014) embedding vectors as an input, and use BiLSTM as a backbone model. Note that they use contextualized models only for input vector representation.
- **RNN_HG** and **RNN_MHCA** (Mao et al., 2019): They incorporate MIP and SPV into **RNN_ELMo** (Gao et al., 2018). **RNN_HG** compares an input embedding vector (literal) with its hidden state (contextual) through BiLSTM. **RNN_MHCA** utilizes multi-head attention to capture the contextual feature within the window size.
- **RoBERTa_BASE**: It is a simple adoption of RoBERTa for metaphor detection. It takes a

target word and a sentence as two input sentences and computes a prediction score. It can be viewed as a metaphor detection model over an all-to-all interaction architecture.

- **RoBERTa_SEQ** (Leong et al., 2020): It takes one single sentence as an input, and a target word is marked as the input embedding token and predicts the metaphoricity of the target word using the embedding vector of the target word. This architecture is used as the BERT-based baseline in the VUA 2020 shared task.
- **DeepMet** (Su et al., 2020): It is the winning model in the VUA 2020 shared task. It also utilizes RoBERTa as a backbone model and incorporates it with various linguistic features, such as global context, local context, POS tags, and fine-grained POS tags.

Evaluation protocol. Because the ratio of metaphorical words is relatively small, we adopt three metrics, *e.g.*, precision, recall, and F1-score, denoted by Prec, Rec, and F1. MOH-X and TroFi datasets are too smaller than VUA datasets. Thus, we only used them as the test datasets; metaphor detection models are only trained in VUA datasets, and zero-shot transfer is conducted to evaluate the effectiveness of model generalization.

Implementation details. For four baselines, we used the same hyperparameter settings¹ in (Gao et al., 2018; Mao et al., 2019; Su et al., 2020). For DeepMet², we evaluated it with/without bagging technique. While DeepMet (Su et al., 2020) exploits two optimization techniques, bagging and ensemble, we only used a bagging technique for MelBERT and DeepMet. It is because we want to evaluate the effectiveness of model designs. The performance difference for DeepMet between the original paper and ours thus comes from the usage of the ensemble method. For contextualized models, we used a pre-trained RoBERTa³ with 12 layers, 12 attention heads in each layer, and 768 dimensions of the hidden state. For contextualized baselines, we set the same hyperparameters with MelBERT, which were tuned on VUA-18_{dev} based on F1-score. The batch size and max sequence length were set as 32 and 150. For training, the number of epochs was three with Adam optimizer.

¹<https://github.com/RuiMao1988/Sequential-Metaphor-Identification>

²<https://github.com/YU-NLPLab/DeepMet>

³<https://huggingface.co/roberta-base>

Dataset	Model	Prec	Rec	F1
VUA-18	RNN_ELMo	71.6	73.6	72.6
	RNN_BERT	71.5	71.9	71.7
	RNN_HG	71.8	<u>76.3</u>	74.0
	RNN_MHCA	73.0	<u>75.7</u>	74.3
	RoBERTa_BASE	79.4	75.0	77.1
	RoBERTa_SEQ	<i>80.4</i>	74.9	<i>77.5</i>
	DeepMet	82.0	71.3	<u>76.3</u>
	MelBERT	80.1	76.9	78.5
	DeepMet-CV	77.5	80.2	78.8
	MelBERT-CV	78.9	80.7	79.8
VUA-Verb	RNN_ELMo	68.2	71.3	69.7
	RNN_BERT	66.7	71.5	69.0
	RNN_HG	69.3	72.3	70.8
	RNN_MHCA	66.3	75.2	70.5
	RoBERTa_BASE	76.9	72.8	74.7
	RoBERTa_SEQ	<i>79.2</i>	69.8	74.2
	DeepMet	79.5	70.8	<i>74.9</i>
	MelBERT	78.7	<u>72.9</u>	75.7
	DeepMet-CV	76.2	78.3	77.2
	MelBERT-CV	75.5	78.7	77.1

Table 2: Performance comparison of MelBERT with baselines on VUA-18 and VUA-Verb (best is in **bold** and second best is in *italic underlined*). Let -CV denote the bagging technique for its base model (best is in **bold-italic**). * denotes $p < 0.05$ for a two-tailed t-test with the best competing model.

Dataset	Model	Prec	Rec	F1
VUA-20	RoBERTa_BASE	74.9	<u>68.0</u>	71.2
	RoBERTa_SEQ	76.9	66.7	<i>71.4</i>
	DeepMet	<u>76.7</u>	65.9	70.9
	MelBERT	76.4	68.6	72.3
	DeepMet-CV	73.8	73.2	73.5
	MelBERT-CV	74.1	73.7	73.9

Table 3: Performance comparison of MelBERT with baselines on VUA-20 (best is in **bold** and second best is in *italic underlined*). Let -CV denote the bagging technique for its base model (best is in **bold-italic**). * denotes $p < 0.05$ for a two-tailed t-test with the best competing model.

We increased the learning rate from 0 to $3e-5$ during the first two epochs and then linearly decreased it during the last epoch. We set the dropout ratio as 0.2. All experimental results were averaged over five runs with different random seeds. We conducted all experiments on a desktop with 2 NVidia TITAN RTX, 256 GB memory, and 2 Intel Xeon Processor E5-2695 v4 (2.10 GHz, 45M cache). We implemented our model using PyTorch. All the source code is available at our website⁴.

⁴<https://github.com/jin530/MelBERT>

Genre	Model	Prec	Rec	F1
Academic	RNN_ELMo	78.2	80.2	79.2
	RNN_BERT	76.7	76.0	76.4
	RNN_HG	76.5	83.0	79.6
	RNN_MHCA	79.6	80.0	79.8
	RoBERTa_BASE	<u>88.1</u>	79.5	<u>83.6</u>
	RoBERTa_SEQ	86.0	77.3	81.4
	DeepMet	88.4	74.7	81.0
	MelBERT	85.3	<u>82.5</u>	83.9
	Conversation	RNN_ELMo	64.9	63.1
RNN_BERT		64.7	64.2	64.4
RNN_HG		63.6	72.5	67.8
RNN_MHCA		64.0	71.1	67.4
RoBERTa_BASE		70.3	69.0	69.6
RoBERTa_SEQ		<u>70.5</u>	69.8	70.1
DeepMet		71.6	71.1	71.4
MelBERT		70.1	<u>71.7</u>	<u>70.9</u>
Fiction		RNN_ELMo	61.4	69.1
	RNN_BERT	66.5	68.6	67.5
	RNN_HG	61.8	<u>74.5</u>	67.5
	RNN_MHCA	64.8	<u>70.9</u>	67.7
	RoBERTa_BASE	<u>74.3</u>	72.1	73.2
	RoBERTa_SEQ	<u>73.9</u>	72.7	<u>73.3</u>
	DeepMet	76.1	70.1	73.0
	MelBERT	74.0	76.8	75.4
	News	RNN_ELMo	72.7	71.2
RNN_BERT		71.2	72.5	71.8
RNN_HG		71.6	76.8	74.1
RNN_MHCA		74.8	<u>75.3</u>	75.0
RoBERTa_BASE		<u>83.5</u>	71.8	<u>77.2</u>
RoBERTa_SEQ		82.2	74.1	77.9
DeepMet		84.1	67.6	75.0
MelBERT		81.0	73.7	77.2

Table 4: Model performance of different genres in VUA-18 (best is in **bold** and second best is in *italic underlined*).

4.2 Empirical Results

Overall results. Tables 2 and 3 report the comparison results of MelBERT against other baselines using RNNs and contextualized models on VUA-18, VUA-20, and VUA-Verb. It is found that MelBERT is consistently better than strong baselines in terms of F1-score. MelBERT outperforms (F1 = 78.5, 75.7, and 72.3) DeepMet (Su et al., 2020) with 2.8%, 1.0%, and 1.9% performance gains on the three datasets. MelBERT also outperforms contextualized baseline models (*i.e.*, RoBERTa_BASE and RoBERTa_SEQ), up to 1.2-1.5% gains on the three datasets, indicating that MelBERT effectively utilizes metaphorical identification theories.

When combining MelBERT and DeepMet with the bagging technique, both models (*i.e.*, MelBERT-CV and DeepMet-CV) show better performance than their original models by aggregating multiple models trained with 10-fold cross-validation process as used in (Su et al., 2020). MelBERT-

POS	Model	Prec	Rec	F1
Verb	RNN_ELMo	68.1	71.9	69.9
	RNN_BERT	67.1	72.1	69.5
	RNN_HG	66.4	75.5	70.7
	RNN_MHCA	66.0	76.0	70.7
	RoBERTa_BASE	<u>77.0</u>	72.1	74.5
	RoBERTa_SEQ	<u>74.4</u>	75.1	<u>74.8</u>
	DeepMet	78.8	68.5	73.3
	MelBERT	74.2	<u>75.9</u>	75.1
	Adjective	RNN_ELMo	56.1	60.6
RNN_BERT		58.1	51.6	54.7
RNN_HG		59.2	65.6	62.2
RNN_MHCA		61.4	<u>61.7</u>	61.6
RoBERTa_BASE		71.7	59.0	64.7
RoBERTa_SEQ		<u>72.0</u>	57.1	63.7
DeepMet		79.0	52.9	63.3
MelBERT		69.4	60.1	<u>64.4</u>
Adverb		RNN_ELMo	67.2	53.7
	RNN_BERT	64.8	61.1	62.9
	RNN_HG	61.0	66.8	63.8
	RNN_MHCA	66.1	60.7	63.2
	RoBERTa_BASE	78.2	<u>69.3</u>	<u>73.5</u>
	RoBERTa_SEQ	77.6	63.9	70.1
	DeepMet	<u>79.4</u>	66.4	72.3
	MelBERT	80.2	69.7	74.6
	Noun	RNN_ELMo	59.9	60.8
RNN_BERT		63.3	56.8	59.9
RNN_HG		60.3	66.8	63.4
RNN_MHCA		69.1	58.2	63.2
RoBERTa_BASE		77.5	60.4	<u>67.9</u>
RoBERTa_SEQ		76.5	59.0	66.6
DeepMet		<u>76.5</u>	57.1	65.4
MelBERT		75.4	<u>66.5</u>	70.7

Table 5: Model performance of different POS tags in VUA-18 (best is in **bold** and second best is in *italic underlined*).

CV still shows better performance for all metrics than DeepMet-CV in VUA-18 and VUA-20. Also, MelBERT-CV (Recall = 73.7) significantly improves the original MelBERT (Recall = 68.6) in terms of recall. It implies that MelBERT-CV can capture various metaphorical expressions by combining multiple models.

Besides, it is found that contextualization-based models show better performance than RNN-based models in VUA-18 and VUA-Verb. While RNN-based models show 71-74% F1-score, contextualization-based models show 76-78% F1-score on VUA-18. It is revealed that RNN-based models are limited in capturing various aspects of words in context. Compared to RNN_ELMo and RNN_BERT, it also indicates that utilizing contextualization-based models as backbone models can have a better effect than simply utilizing it as an extra input embedding vector in (Gao et al., 2018; Mao et al., 2019).

Dataset	Model	Prec	Rec	F1
MOH-X	RoBERTa_BASE	77.4	80.1	78.4
	RoBERTa_SEQ	80.6	77.7	<u>78.7</u>
	DeepMet	79.9	76.5	<u>77.9</u>
	MelBERT	79.3	<u>79.7</u>	79.2
TroFi	RoBERTa_BASE	54.6	74.3	62.9
	RoBERTa_SEQ	53.6	70.1	60.7
	DeepMet	<u>53.7</u>	72.9	61.7
	MelBERT	53.4	<u>74.1</u>	<u>62.0</u>

Table 6: Performance comparison of MelBERT with baselines over two datasets. Note that the models are trained on VUA-20, and these datasets are only used as the test datasets (best is in **bold** and second best is in *italic underlined*).

VUA-18 breakdown analysis. Table 4 reports the comparison results for four genres in the VUA-18 dataset. MelBERT still shows better than or comparable to all competitive models in both breakdown datasets. Compared to RNN-based models, MelBERT achieves substantial improvements, as high as 4.9% (Academic), 4.4% (Conversation), 10.2% (Fiction), and 2.8% (News) in terms of F1-score. Particularly, they show the lowest accuracy because Conversation and Fiction have more complicated or rare expressions than other genres. For example, Conversation contains colloquial expressions or fragmented sentences such as “ah”, “cos”, “yeah” and Fiction often contains the names of fictional characters such as “Tepilit”, “Laibon” which do not appear in other genres. Nonetheless, MelBERT shows comparable or the best performance in all genres. For Academic and Fiction, MelBERT particularly outperforms all the models in terms of F1-score.

Table 5 reports the comparison result for four POS tags in the VUA-18 dataset. For all POS tags, MelBERT consistently shows the best performance in terms of the F1-score. Compared to RNN-based models, MelBERT achieves as much as 5.9% (Verb), 3.4% (Adjective), 14.5% (Adverb), and 10.3% (Noun) gains in terms of F1-score. For all POS tags, MelBERT also outperforms DeepMet. It means that MelBERT using metaphorical identification theories can achieve consistent improvements regardless of POS tags of target words.

Zero-shot transfer on MOH-X and TroFi. We evaluate a zero-shot learning transfer across different datasets, where the models are trained with the VUA-20 training dataset, and MOH-X and TroFi are used as test datasets. Although it is a challenging task, it is useful for evaluating the gener-

Model	VUA-18			VUA-20		
	Prec	Rec	F1	Prec	Rec	F1
MelBERT	80.1	76.9	78.5	76.4	68.6	72.3
(-) MIP	77.8	75.8	76.7	74.7	67.8	71.1
(-) SPV	<u>79.5</u>	<u>76.3</u>	<u>77.9</u>	<u>74.9</u>	<u>68.6</u>	<u>71.7</u>

Table 7: Effect of different metaphorical identification theories on VUA-18 and VUA-20. (-) MIP and (-) SPV indicate MelBERT without MIP and SPV, respectively (best is in **bold** and second best is in *italic underlined*).

alization power of trained models. Table 6 reports the comparison results of MelBERT against other contextualization-based models. For the MOH-X dataset, MelBERT (F1 = 79.2) shows the best performance in terms of F1-score with 0.6–1.6% performance gains. It indicates that MelBERT is an effective generalization model. For the TroFi dataset, the overall performance of all the models is much lower than MOH-X. It is because the average length of the sentences in the TroFi dataset is much longer and sentences are more complicated than those in MOH-X. Also, note that we trained DeepMet with the VUA-20 training dataset for evaluating a zero-shot transfer, while (Su et al., 2020) reported the results for DeepMet trained and tested with the MOH-X and TroFi datasets. While the performance gap between models is much small in terms of precision, MelBERT is better than DeepMet in terms of recall. It means that MelBERT can capture complicated metaphorical expressions than DeepMet.

Ablation study of MelBERT. Table 7 compares the effectiveness of metaphor identification theories. It is found that MelBERT using both strategies consistently shows the best performance. Also, MelBERT without SPV shows better performance than MelBERT without MIP, indicating that MelBERT using late interaction is more effective for capturing the difference between contextualized and isolated meanings of target words. Nonetheless, MelBERT shows the best performance by synergizing both metaphor identification strategies.

Error analysis. Table 8 reports qualitative evaluation results of MelBERT. Based on the original annotation guideline⁵, we analyze several failure cases of MelBERT. For MelBERT without MIP, it is difficult to find common words with multiple meanings, e.g., *go* and *feel*. Also, when a sentence includes multiple metaphorical words, it mostly fails to detect metaphorical words. In this case,

⁵<http://www.vismet.org/metcor/documentation/home.html>

(-) MIP	(-) SPV	MeLBERT	Sentence
✓	✓		Manchester is not <i>alone</i> .
✓	✓		That’s an <i>old</i> trick.
✓	✓		Oh you rotten old <i>pig</i> , you’ve been sick.
✓	✓		"Are the twins <i>trash</i> ?"
	✓	✓	I know, what is <i>going</i> on!
	✓	✓	So who’s <i>covering</i> tomorrow?
	✓	✓	Do you <i>feel</i> better now?
	✓	✓	The day thrift turned into a <i>nightmare</i> .
			<i>Way</i> of the <i>World</i> : Farming notes
			So many places Barry are <i>going</i> down
			<i>Sensitivity</i> , though, is not enough.

Table 8: Examples of incorrect samples for MeLBERT on VUA-20. The metaphorical words in the sentence are in red italicized. ✓ marks correct model prediction.

the surrounding words of a target word are not a cue to detect metaphors using SPV. Meanwhile, MeLBERT without SPV has a failure case if target words are metaphorical for personification. That is, using MIP only, the target word can be closely interpreted by its literal meaning. As the most difficult case, MeLBERT often fails to identify metaphorical words for borderline or implicit metaphors, e.g., “*Way of the World*” is poetic.

5 Conclusion

In this work, we proposed a novel metaphor detection model, namely, *metaphor-aware late interaction over BERT (MeLBERT)*, marrying pre-trained contextualized models with metaphor identification theories. To our best knowledge, this is the first work that takes full advantage of both contextualized models and metaphor identification theories. Comprehensive experimental results demonstrated that MeLBERT achieves state-of-the-art performance on several datasets.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) (NRF-2018R1A5A1060031). Also, this work was supported by the Institute of Information & communications Technology Planning & evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00421, AI Graduate School Support Program and No.2019-0-01590, High-

Potential Individuals Global Training Program). The work of Dongwon Lee was in part supported by NSF awards #1742702, #1820609, #1909702, #1915801, and #1934782.

References

- Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- George Aaron Broadwell, Umit Boz, Ignacio Cases, Tomek Strzalkowski, Laurie Feldman, Sarah M. Taylor, Samira Shaikh, Ting Liu, Kit Cho, and Nick Webb. 2013. Using imageability and topic chaining to locate metaphors in linguistic corpora. In *Proceeding of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction (SBP)*, pages 102–110.
- Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. Modelling metaphor with attribute-based semantics. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 523–528.
- Erik Cambria, Soujanya Poria, Alexander F. Gelbukh, and Mike Thelwall. 2017. Sentiment analysis is a big suitcase. *IEEE Intell. Syst.*, 32(6):74–80.
- Xianyang Chen, Chee Wee Leong, Michael Flor, and Beata Beigman Klebanov. 2020. Go Figure! multi-task transformer-based architecture for metaphor detection using idioms: ETS team in 2020 metaphor shared task. In *The Second Workshop on Figurative Language Processing (Fig-Lang@ACL)*, pages 235–243.
- Zhuyun Dai and Jamie Callan. 2020. Context-aware term weighting for first stage passage retrieval. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1533–1536.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.
- Pawel Dybala and Kohichi Sayama. 2012. Humor, emotions and communication: Human-like issues of human-computer interactions. In *Annual Meeting of the Cognitive Science Society (CogSci)*.
- Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 607–613.

- Hongyu Gong, Kshitij Gupta, Akriti Jain, and Suma Bhat. 2020. IlliniMet: Illinois system for metaphor detection with contextual and linguistic information. In *The Second Workshop on Figurative Language Processing (Fig-Lang@ACL)*, pages 146–153.
- Alex Graves and Jürgen Schmidhuber. 2005. Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.
- Hessel Haagsma and Johannes Bjerva. 2016. Detecting novel metaphor using selectional preference information. In *Proceedings of the Fourth Workshop on Metaphor in NLP*.
- Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 39–48.
- Luuk Lagerwerf and Anoe Meijers. 2008. Openness in metaphorical and straightforward advertisements: Appreciation effects. *Journal of Advertising*, 37(2):19–30.
- George Lakoff and Mark Johnson. 1980. *Metaphors we live by*. University of Chicago Press.
- Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 VUA and TOEFL metaphor detection shared task. In *The Second Workshop on Figurative Language Processing (Fig-Lang@ACL 2020)*, pages 18–29.
- Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 VUA metaphor detection shared task. In *The Workshop on Figurative Language Processing (Fig-Lang@NAACL-HLT)*, pages 56–66.
- 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. *CoRR*, abs/1907.11692.
- Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed BERT models for large-scale retrieval. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2645–2652.
- 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 (ACL)*, pages 3888–3898.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119.
- Bhaskar Mitra and Nick Craswell. 2018. An introduction to neural information retrieval. *Found. Trends Inf. Retr.*, 13(1):1–126.
- Saif Mohammad, Ekaterina Shutova, and Peter D. Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM@ACL)*.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *CoRR*, abs/1904.08375.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237.
- Pragglejaz Group. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Technical report*, OpenAI.
- Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1537–1546.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. In *The Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3980–3990.
- Chunqi Shi, Toru Ishida, and Donghui Lin. 2014. Translation agent: A new metaphor for machine translation. *New Gener. Comput.*, 32(2):163–186.
- Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 160–170.
- Ekaterina Shutova and Lin Sun. 2013. Unsupervised metaphor identification using hierarchical graph factorization clustering. In *Proceeding of the Conference of the North American Chapter of the Asso-*

- ciation of Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 978–988.
- Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceeding of International Conference on Computational Linguistics (CoLING)*, pages 1002–1010.
- Gerard J Steen, Aletta G Dorst, J Berenike Herrman, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification. From MIP to MIPVU*. John Benjamin.
- Chuangdong Su, Fumiyo Fukumoto, Xiaoxi Huang, Jiyi Li, Rongbo Wang, and Zhiqun Chen. 2020. DeepMet: A reading comprehension paradigm for token-level metaphor detection. In *The Second Workshop on Figurative Language Processing (FigLang@ACL)*, pages 30–39.
- Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 248–258.
- 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 (EMNLP)*, pages 680–690.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.
- Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov. 2016. *Metaphor: A Computational Perspective*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*.
- Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artif. Intell.*, 6(1):53–74.
- Yorick Wilks. 1978. Making preferences more active. *Artif. Intell.*, 11(3):197–223.
- 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*, pages 110–114.