Multi-task Learning for Metaphor Detection with Graph Convolutional Neural Networks and Word Sense Disambiguation

Duong Minh Le[1], **My Thai**[2] **and Thien Huu Nguyen**[1,3]*
[1]VinAI Research, Hanoi, Vietnam
[2]Department of Computer and Information Science and Engineering, University of Florida
[3]Department of Computer and Information Science, University of Oregon, Eugene, Oregon 97403, USA
v.duonglm1@vinai.io, mythai@cise.ufl.edu, thien@cs.uoregon.edu

Abstract

The current deep learning works on metaphor detection have only considered this task independently, ignoring the useful knowledge from the related tasks and knowledge resources. In this work, we introduce two novel mechanisms to improve the performance of the deep learning models for metaphor detection. The first mechanism employs graph convolutional neural networks (GCN) with dependency parse trees to directly connect the words of interest with their important context words for metaphor detection. The GCN networks in this work also present a novel control mechanism to filter the learned representation vectors to retain the most important information for metaphor detection. The second mechanism, on the other hand, features a multi-task learning framework that exploits the similarity between word sense disambiguation and metaphor detection to transfer the knowledge between the two tasks. The extensive experiments demonstrate the effectiveness of the proposed techniques, yielding the state-of-the-art performance over several datasets.

Introduction

We study the problem of metaphor detection (MD) that aims to identify the metaphorical expressions/words in text. Metaphor is persuasive in daily communication, providing the vividness and clarity for our thoughts and information exchange. In the cognitive level, metaphor helps to conceptualize our concrete experience in the real world and transfer such knowledge across different domains (Shutova, Kiela, and Maillard 2016). More specifically, in an early and influential work, (Lakoff and Johnson 1980) describes metaphor as a phenomenon in which a systematic metaphorical association between two distinct concepts/domains is established in our cognition and presented in the language. For example, the words "*curing*" and "*transmitting*" in the phrases "*curing juvenile delinquency*" and "*corruption transmitting through the government ranks*" (respectively) are metaphorical as they are supposed to originate from the concept of *disease* (i.e., the source concept), but applied to the concept of *crime* (i.e., the target concept) (Rei et al. 2017).

The ubiquity of metaphor has made it an important problem in natural language processing (NLP) and recognizing metaphor correctly is crucial for the text understanding capacity of NLP systems for various purposes (e.g., Information Extraction, Opinion Mining, Machine Translation) (Shutova, Teufel, and Korhonen 2013). However, metaphor detection is a challenging problem, partly due to the fact that the systems need to understand the literal senses of the expressions and distinguish such senses from the non-literal meanings based on the analogical comparison in the specific context of the expressions. In order to capture the context for metaphor detection, the early work has adopted rule-based and machine learning systems (Mason 2004; Turney et al. 2011; Tsvetkov et al. 2014; Hovy et al. 2013) where extensive feature engineering and system development are required. Recently, deep learning and word representations have been applied to metaphor detection to minimize the feature engineering effort, offering the systems with the current state-of-the-art performance for this task (Gao et al. 2018; Wu et al. 2018; Mao, Lin, and Guerin 2019). In this work, we introduce two novel techniques for metaphor detection with deep learning that, to the best of our knowledge, have not been investigated in the previous work, i.e., graph convolutional neural networks (GCN) with dependency trees and multi-task learning with word sense disambiguation (WSD).

The first intuition for the proposed model is that given a word of interest in a sentence, some context words are more important than the others with respect to their contribution to the determination about metaphor for the given word, and effective modeling of such important words is critical for the metaphor detection systems. For instance, in the example sentence in Figure 1, the word "*house*" is crucial to correctly predict "*lay*" as a metaphorical word (i.e., "*lay*" is usually used for live animals instead of static objects like houses). The other words (e.g., "*seemed*", "*float*", and "*mist*"), on the other hand, are not necessary and might even introduce noise into the representations learned by the deep learning models to determine the literal or metaphorical sense for "*lay*" in this case. In order to properly capture the important context words, in this work, we propose to rely on the dependency trees of the sentences to guide the computation of the

[*] Corresponding author.

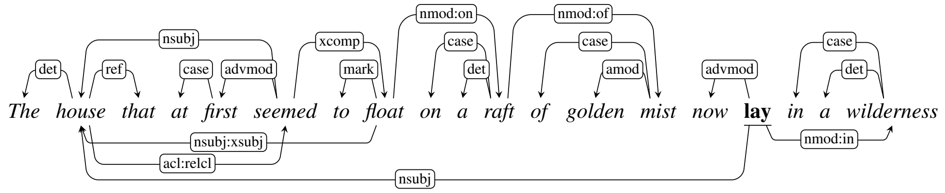

Figure 1: The dependency parse tree for an example sentence. The word "*lay*" is metaphorical in this case.

graph convolutional neural networks for metaphor detection. Our hypothesis is that the head-modifier connections between words in the dependency trees would help to directly link the important context words to the words of interest for metaphor detection. For instance, in the example sentence of Figure 1, the relevant context word "*house*" is syntactically next to the word of interest "*lay*" via the *nsubj* relation. In the graph convolutional neural network (Kipf and Welling 2016) performed over such dependency structure, the representation vector for a word at one step will be computed based on the representation vectors of the syntactic neighboring words in the previous step. Consequently, the representation vectors for the words in GCNs will mainly involve the information from the important/relevant context words in the sentences for metaphor detection, filtering the irrelevant words and potentially improving the performance for the models. Finally, as we often classify a single word or expression in the sentences for metaphor detection at a time, we propose a novel control technique to customize the GCN representation vectors so only the information relevant to the words of interest would be retained and propagated over the next computation step. Specifically, control vectors will be generated from the representation vectors of the word of interest that are then employed as the filters for the GCN vectors to perform metaphor prediction.

Second, for the multi-task learning with WSD, the main intuition is that WSD is highly related to the task of metaphor detection and we can transfer the knowledge from the reasoning process for WSD to improve the performance of metaphor detection. In particular, the goal of WSD is to identify the correct sense/meaning of a word/expression in its context among the possible senses that the word can have in general (e.g., the senses in WordNet). At the modeling level, both WSD and metaphor detection need to perform a classification problem for a word/expression based on its context in the sentence. At the semantic level, it has been shown in the previous studies that many metaphorical senses of the words are recorded in the sense inventory of WordNet (i.e., Section §3.4 (Shutova, Teufel, and Korhonen 2013)). For instance, the sense with the id "*drown%2:35:00::*" of "*drown*" (i.e., *cover completely or make imperceptible*) in the phrase "*drowned in work*" is metaphorical, in contrast to the literal sense of id "*drown%2:30:00::*" (i.e., *die from being submerged in water*) for "*drown*" in "*drowned in water*". Consequently, if a deep learning system is capable of learning effective representations to differentiate the senses of the words in the context, its induced representations might also be helpful for metaphor detection due to the close relatedness in semantic modeling for context. In order to employ such similarity between WSD and metaphor detection, in this work, we propose a novel multi-task learning framework that matches the representations induced for WSD and metaphor prediction to facilitate the knowledge transfer between the two tasks. Our framework explicitly handles the practical issue where the datasets are only annotated for a single task (i.e., either WSD or metaphor detection) by training two networks for the two tasks and encouraging the representations of the two networks to be similar if they are presented with the same sentences/contexts. Based on our literature survey, this is the first work on multi-task learning for metaphor detection with deep learning.

We extensively evaluate the proposed model over several benchmark datasets. The experimental results clearly demonstrate the effectiveness of the model and present the state-of-the-art performance for the such datasets.

Related Work

There have been several theories about metaphor in the past, including the Contextual Metaphor Theory (Lakoff and Johnson 1980) and the Selection Preference Violation (SPV) (Wilks 1978). In order to annotate metaphorical words in corpus, the popular principle is the metaphor identification procedure (MIP) that defines metaphor as the contrast between the literal meaning of a word and its meaning in the context (Group 2007). MIP is used to annotate several widely used datasets for metaphor detection, i.e., VUA (Gerard Steen and Pasma 2010) and MOH-X (Mohammad, Shutova, and Turney 2016).

In the early work for metaphor detection, various features and resources have been exploited to develop rule-based and machine learning systems, including semantic roles, domain types, word abstractness/concreteness, imageability, WordNet supersenses (Mason 2004; Turney et al. 2011; Strzalkowski et al. 2013; Tsvetkov et al. 2014; Klebanov et al. 2016)). The other words have also considered clustering and unsupervised learning techniques for metaphor detection (Shutova, Teufel, and Korhonen 2013; Shutova E. and Narayanan 2017; Mao, Lin, and Guerin 2018). Recently, many studies have applied deep learning to solve metaphor prediction (Rei et al. 2017; Wu et al. 2018; Mao, Lin, and Guerin 2019), resulting in the state-of-the-art performance on several benchmark datasets. We also propose a deep learning model in this work; however, we introduce GCNs and the multi-task learning framework with WSD that have never been explored before.

Finally, the previous works have applied multi-task learn-

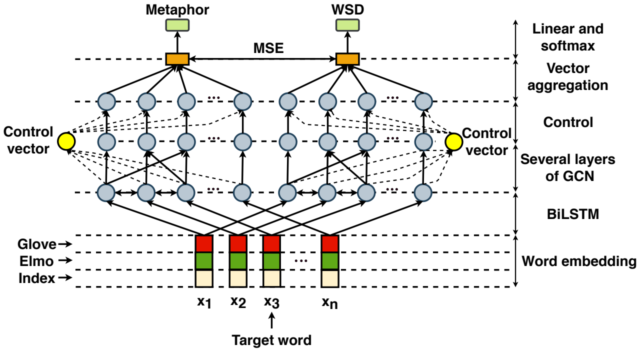

Figure 2: Model Overview.

ing to address various problems in NLP, including structured prediction tasks (Duong et al. 2015; Collobert et al. 2011; Guo et al. 2016) and sequence-to-sequence tasks (Dong et al. 2015). Our work is different from such prior works as we simultaneously model WSD and metaphor detection with deep learning for the first time.

Model

We cast metaphor detection as a binary classification problem where given a word and the sentence containing this word, we need to predict whether this word is metaphorical or not (Gao et al. 2018; Mao, Lin, and Guerin 2019). We call the word of interest for metaphor detection as the target word and the words in the sentence would constitute the context for the target word in our prediction task.

Formally, let $w = w_1, w_2, \ldots, w_n$ be a sentence of length n in which w_a ($1 \leq a \leq n$) is the target word for metaphor prediction (w_i is the i-th token in the sentence $\forall 1 \leq i \leq n$). The multi-task model for metaphor detection in this work consists of four modules: (i) the encoding module, (ii) the graph convolution module, (iii) the control module, and (iv) the multi-task learning module. Figure 2 shows an overview of the proposed model.

The Encoding Module

The first step in the proposed model is to transform the words in the sentence w into real-valued vectors for the deep learning architectures. In order to ensure a fair comparison with the previous works on metaphor detection (Gao et al. 2018; Mao, Lin, and Guerin 2019), in this work, we convert each word $w_i \in w$ into a vector x_i using the concatenation of the following three vectors:

• The uncontextualized pre-trained word embedding of w_i from Glove (Pennington, Socher, and Manning 2014). We obtain this vector by looking up the embedding table provided by Glove.

• The contextualized word embeddings of w_i from ELMo (Peters et al. 2018). In particular, we run the pre-trained ELMo model over the input sentence, resulting in three sequences of hidden vectors for the words in this sentence (each sequence corresponds to a layer in the ELMo architecture). Afterward, we take the weighted sum of the hidden vectors at the position i in each sequence to produce an accumulated representation vector for w_i. The weights for each sequence and the scalar parameter in ELMo are learned during our training process of the whole model.

• The index embedding: In order to specify that w_a is the word of interest for metaphor prediction, we assign a binary indicator b_i so $b_i = 1$ if $i = a$ and 0 otherwise. We then map such binary indicators into real-valued vectors using an index embedding table that is initialized randomly and fixed during training.

This word-to-vector conversion process transforms w into a sequence of real-valued vectors $x = x_1, x_2, \ldots, x_n$. Although the ELMo component already helps to encapsulate the contextual information over the whole sentence w into each vector x_i (due to its employment of the bidirectional long-short term memory (LSTM) networks (Hochreiter and Schmidhuber 1997)), the Glove and index components in x_i are still independent from each other and not yet well integrated with the ELMo component to form fully contextualized and rich representation vectors for the words in the input sentence. In order to better combine the three vector components in the vectors x_i, we perform a bidirectional LSTM network (BiLSTM) (Gao et al. 2018) over the vector sequence $x_1, x_2, \ldots, x_n$, generating the hidden vector sequence $h = h_1, h_2, \ldots, h_n$ as the output (i.e., the BiLSTM representation vectors). Due to the recurrent and bidirectional nature of BiLSTM, each vector h_i encodes the contextual information over the whole sentence x, smoothly wrapping the different components of x_i into a single contextualized representation space.

The Graph Convolution Module

Given the hidden vectors from BiLSTM $h = h_1, h_2, \ldots, h_n$, it is possible to directly aggregate such vectors to compute an overall representation vector for metaphor prediction as in (Gao et al. 2018). However, as we presented in the introduction, it is important for the metaphor detection models to recognize the relevant context words for the target word and explicitly model them to achieve good performance. As such important word modeling is not explicitly designed in the current bidirectional LSTM network, we further feed the hidden vectors h_i from this network into a graph convolutional neural network (Kipf and Welling 2016) that structures its computation over the dependency tree of the input sentence w.

The operation of GCNs requires an adjacency matrix A to encode the connections of the words in the dependency tree for x. In order to prepare the adjacency matrix in this work, besides the original directed connections in the dependency tree, we also add the reverse edges and the self loops into the tree. Such additional edges enable the governor word of w_i in the dependency tree (if any) and the word w_i itself to contribute to the computation of the GCN representation vector for w_i via the convolution operation. This helps to enrich the GCN representation vectors with syntactically important context for better metaphor prediction performance (Nguyen and Grishman 2018a).

Let $H^0 = [h_1, h_2, \ldots, h_n]$ be the matrix whose rows are the hidden vectors $h_1, h_2, \ldots, h_n$ from the bidirectional

LSTM network in the encoding module. The GCN module involves several layers of convolution; each layer takes as input the matrix $H_i (i \geq 0)$ from the previous layer i and computes the matrix H_{i+1} for the current layer based on (the bias is ignored for simplicity): $H_{i+1} = g(AH_iW_i^g)$. Here, W_i^g is the weight matrix for the i-th layer and g is a non-linear function. We optimize the number of layers for the GCN module based on the validation datasets in this work. For convenience, we call the row vectors of the matrix in the final convolution layer of the GCN module as $h^g = h_1^g, h_2^g, \ldots, h_n^g$ (i.e., the GCN representation vectors).

The Control Module

In order to perform metaphor prediction for w_a, the deep learning models need to define a method to compute an overall representation vector V, serving as the features for the prediction task. One way to generate such overall representation vector in our model is to directly aggregate the GCN representation vectors $h_1^g, h_2^g, \ldots, h_n^g$ in the previous module via the popular techniques such as the pooling (e.g., max, average) and attention mechanisms (Gao et al. 2018). A drawback of this approach is that it assumes the same level of importance for all the dimensions in the representation vectors $h_1^g, h_2^g, \ldots, h_n^g$. This is undesirable as the dimensions of the representation vectors are not constrained at all so far and some dimensions might have more impacts than the other dimensions for metaphor prediction. In addition, as the BiLSTM and GCN representation vectors have been abstracted away from the original word vectors x_i via the hidden layers (i.e., BiLSTM and GCN), the information about the position of the target word (i.e., the index embedding component in x_i) might have been blurred, causing the confusion of the representation vectors about the target word for metaphor prediction. In this work, we address these two problems by devising a novel control technique to regulate the GCN representation vectors to be more specific and aware of the target word w_a. Such regulation will be done at the dimension level (i.e., feature-wise) so the dimensions can be quantified appropriately according to their importance for the metaphor detection problem.

Our general strategy for representation vector regulation is to first compute the control vectors based on the representation vector of the target word w_a (e.g., h_a) and then apply such control vectors as the feature-wise filters for the other BiLSTM and GCN representation vectors. In particular, for the BiLSTM representation vectors $h = h_1, h_2, \ldots, h_n$, we first obtain the control vector c_h via: $c_h = Relu(W_h h_a)$.

This control vector then helps to filter the irrelevant information (with respect to the target word w_a) from the BiLSTM representation vectors h_i via the element-wise multiplication $\odot$: $\hat{h}_i = c_h \odot h_i \; \forall 1 \leq i \leq n$.

where $\hat{h}_i$ is the filtered BiLSTM vector and the element-wise multiplication manifests our mechanism to achieve the feature-wise/dimension-level manipulation of the representation vectors in this work. For the GCN representation vectors, besides the BiLSTM representation vector h_a of w_a, the GCN control vector c_g is also conditioned on the weighted sum of the vectors in h. This helps to inform the control vector c_g about the information already presented in the BiLSTM representation vectors h so when c_g is applied to the GCN vectors h_g, the important information in the BiLSTM vectors can be still preserved. For the weighted sum of the vectors in h, the filtered vectors $\hat{h}_i$ are employed to obtain the weights for the vectors in h to ensure that such weights are also customized for the target word w_a:

$$\alpha_i = \frac{\exp(W_\alpha \hat{h}_i)}{\sum_{j=1}^{n} \exp(W_\alpha \hat{h}_j)}$$
$$m = \sum_{i=1}^{n} \alpha_i h_i, \; c_g = Relu(W_g[h_a, m]) \quad (1)$$

In the next step, the control vector c_g is also applied to the GCN representation vectors h_i^g using the element-wise multiplication operation, producing $\hat{h}_i^g$ as the filtered GCN representation vectors: $\hat{h}_i^g = c_g \odot h_i^g$.

Finally, in order to aggregate the filtered GCN vectors $\hat{h}_i^g$ to form the overall representation vector V for metaphor detection, we use the following concatenation vector:

$$V = [\hat{h}_a^g, \max(\hat{h}_1^g, \hat{h}_2^g, \ldots, \hat{h}_n^g)] \quad (2)$$

In this formula, $\hat{h}_a^g$ captures the context information for the target word w_a while $\max(\hat{h}_1^g, \hat{h}_2^g, \ldots, \hat{h}_n^g)$ leverages the most important context information from the other words to enrich the representation V. For prediction, the overall vector V would be used as the input for a feed-forward neural network, followed by a softmax layer in the end to compute the probability distribution over two choices (i.e., metaphorical or not). The loss function to train the models in this work is the negative log-likelihood over the training datasets.

The Multi-task Learning Module

The goal of the multi-task learning module in this section is to transfer the knowledge from the datasets for word sense disambiguation to improve the performance for our metaphor prediction task as motivated in the introduction. In the multi-task learning setting for NLP, multiple different, but related tasks are solved simultaneously using their available training datasets. The common approach with deep learning for such multi-task learning techniques assumes a single deep learning model to learn the representation vectors for the text inputs (i.e., the encoder) that would then be used as the shared features by different classifiers specific to the tasks of interest (Dong et al. 2015; Duong et al. 2015; Guo et al. 2016). If the datasets for the related tasks share the text inputs (e.g., the same sentences are annotated for different tasks), we can utilize the joint training process that optimizes the joint loss function of the different tasks (Collobert and Weston 2008). However, if the datasets for the related tasks involve different input texts (e.g., a sentence for a dataset for a task only has the label for that corresponding task), we need to resort to the alternative training procedure that alternates the training process for the tasks of interest. In particular, at one iteration, one task would be selected with some probability and a minibatch of the dataset for that task would be sampled to compute the loss and update the model

(Guo et al. 2016). Note that the parameters of the encoders are updated at every iteration as they are shared across the tasks while only the parameters for the classifier of the currently selected task are affected at one training iteration. Our multi-task learning framework for WSD and metaphor detection falls under the later scenario as the available datasets for these two tasks do not share the input sentences, partly due to the separation of the evaluation campaign of the tasks. We thus consider the alternative training paradigm as the baseline for our multi-task learning framework in this work.

One problem with the baseline approach is that a single deep learning model is used as the encoder for the multiple related tasks of consideration. For WSD and metaphor detection, although they have some level of similarity in terms of the semantic classification for words/expressions in context, the representation vectors of the encoder for WSD might need to involve more fine-grained or detailed information than those for metaphor detection. This is caused by the fact that the labels (i.e., the senses) in WSD are in general more specific and exhaustive than the labels for metaphor detection. In particular, one word in WSD can have more than ten senses (e.g., "*play*") while a word in metaphor detection can only be assigned to two labels (i.e., metaphorical or not). On the other hand, some metaphorical senses might not be present in WordNet (Shutova, Teufel, and Korhonen 2013) for WSD, potentially requiring the representation vectors for metaphor detection to capture some different semantic information from those for WSD. Consequently, if a single encoder is used to induce the representation vectors for WSD and metaphor detection, the encoder might struggle to decide which semantic information/aspects it should focus on, leading to the reduced quality of the representations in the end. In order to overcome this problem, in this work, instead of using a single deep learning encoder, we propose to introduce two separate encoders to compute the representation vectors for WSD and metaphor detection. The knowledge is transferred between the two encoders by ensuring that they use the same network architecture as the one we propose in this work and that they produce similar representation vectors once presented with the same input sentence. The two separate encoder networks allow the flexibility to learn specific features for the individual tasks while the transferring knowledge mechanism facilitate the incorporation of the knowledge in WSD for metaphor detection.

Formally, let E^{wsd} and E^{md} be the encoders for the WSD and metaphor detection tasks, following the network architecture we present before. Also, let (w^t, p^t, y^t) be one example (at one iteration) from a dataset for either WSD or metaphor detection (i.e., t is the task indicator: $t \in \{wsd, md\}$) where w^t is the input sentence, p^t is the position for the target word, and y^t is the label of w^t for the task t). In order to perform knowledge transferring, we feed the input text (w^t, p^t) to both encoders E^{wsd} and E^{md}, resulting in two representation vectors V^{wsd} and V^{md} respectively:

$$V^{wsd} = E^{wsd}(w^t, p^t), V^{md} = E^{md}(w^t, p^t) \qquad (3)$$

For the task t, the representation vector V^t is sent to the task-specific classifier F^t (e.g., a feed-forward neural network followed by a softmax layer) to compute the probability distribution $P^t(.|w^t, p^t)$ for the possible labels for t. In the usual single-task training procedure, we would optimize the negative log-likelihood function $C(w^t, p^t, y^t) = -\log P^t(y^t|w^t, p^t)$ to search for the parameters for the encoder E^t and the classifier F^t for t. However, in our multi-task learning framework, we instead minimize the following loss function to update the parameters for F^t and both the encoders E^{wsd} and E^{md}:

$$\begin{aligned} C(w^t, p^t, y^t) = \\ -\log P^t(y^t|w^t, p^t) + \lambda \|V^{wsd} - V^{md}\|_2^2 \end{aligned} \qquad (4)$$

where λ is a trade-off parameter. The rationale for the second term is that as the V^{wsd} and V^{md} are the representation vectors for the same input sentence w^t from the related encoders E^{wsd} and E^{md}, they should be similar to each other. This allows the two encoders to communicate to each other so the knowledge from one task (e.g., WSD) can be back-propagated to the other task (e.g., metaphor detection) to improve the quality of the representation vectors. Note that we also follow the alternative training procedure to train the multi-task learning framework in this work. This completes the description of the proposed model for metaphor detection in the current work.

Experiments

Datasets, Parameters and Resources

In order to be compatible with the previous work (Gao et al. 2018; Mao, Lin, and Guerin 2019), we evaluate the proposed models using three widely used datasets for metaphor detection, i.e., **VUA** (Gerard Steen and Pasma 2010), **MOH-X** (Mohammad, Shutova, and Turney 2016) and **TroFi** (Birke and Sarkar 2006). **VUA** represents the largest public evaluation dataset for metaphor detection that is used by the NAACL-2018 Metaphor Shared Task (Leong, Beigman Klebanov, and Shutova 2018). The annotation for this dataset is based on MIP for which every word in the sentences is labeled for metaphor identification. Following the prior work (Gao et al. 2018; Mao, Lin, and Guerin 2019), we also consider two versions of this dataset, i.e., **VUA ALL POS** where words of all types (e.g., nouns, verbs, adjectives) are labeled, and **VUA VERB** that only focuses on the verbs for metaphor detection. For **MOH-X**, the sentences are shorter and simpler than those in the other datasets as they are sampled from WordNet. Only one single verb is labeled in each sentence in **MOH-X**. Finally, **TroFi** involves sentences from the 1987-89 Wall Street Journal Corpus Release 1. Similar to **MOH-X**, **TroFi** is also only annotated for a single target verb. Following the settings in the prior work (Gao et al. 2018; Mao, Lin, and Guerin 2019), we perform 10-fold cross validation on **MOH-X** and **TroFi** and split the **VUA** datasets into training, validation and test sets. We use the same data splits for all the three datasets as the previous work (Gao et al. 2018; Mao, Lin, and Guerin 2019) for the fair comparison.

We use the Semcor dataset (Miller et al. 1994) for the WSD dataset in this work. This dataset includes sentences whose words have been manually annotated for the WordNet sense ids. As the number of words in Semcor is much larger

Model	VUA All POS				VUA VERB				MOH-X				TroFi			
	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
Lexical Baseline	-	-	-	-	67.9	40.7	50.9	76.4	39.1	26.7	31.3	43.6	72.4	55.7	62.9	71.4
SimNet	-	-	-	-	-	-	-	-	73.6	76.1	74.2	74.8	-	-	-	-
CNN+BiLSTM†	60.8	70.0	65.1	-	60.0	76.3	67.2	-	-	-	-	-	-	-	-	-
RNN_CLS	-	-	-	-	53.4	65.6	58.9	69.1	75.3	84.3	79.1	78.5	68.7	74.6	72.0	73.7
RNN_SEQ_ELMo†	71.6	73.6	72.6	93.1	68.2	71.3	69.7	81.4	79.1	73.5	75.6	77.2	70.7	71.6	71.1	74.6
RNN_SEQ_BERT†	71.5	71.9	71.7	92.9	66.7	71.5	69.0	80.7	75.1	81.8	78.2	78.1	70.3	67.1	68.7	73.4
RNN_HG†	71.8	76.3	74.0	93.6	69.3	72.3	70.8	82.1	79.7	79.8	79.8	79.7	67.4	77.8	72.2	74.9
RNN_MHCA†	73.0	75.7	74.3	93.8	66.3	75.2	70.5	81.8	77.5	83.1	**80.0**	79.8	68.6	76.8	72.4	75.2
MUL_GCN	74.8	75.5	**75.1**	**93.8**	72.5	70.9	**71.7**	**83.2**	79.7	80.5	79.6	**79.9**	73.1	73.6	**73.2**	**76.4**

Table 1: The systems' performance. †indicates the models that apply the sequential labeling setting. P, R, and F1 are calculated for the metaphorical words while Acc indicate the overall accuracy of the models.

than those in the datasets for metaphor detection, we only sample a portion of Semcor to train the models in this work. In particular, we sample so that the numbers of examples in the WSD and metaphor detection datasets would be similar. For the metaphor detection datasets that only involve verbs as the targets (i.e., **VUA VERB**, **MOH-X**, **TroFi**), we also sample only the verbs in Semcor for WSD accordingly.

Regarding the pre-trained word embeddings, we also use the 300d Glove vectors (Pennington, Socher, and Manning 2014) and 1024d ELMo vectors (Peters et al. 2018) as in (Gao et al. 2018; Mao, Lin, and Guerin 2019). The dimension of the index embeddings is set to 50 as in the classification setting in (Gao et al. 2018) for a fair comparison. We fine-tune the other hyper-parameters of the proposed model for each dataset that results in the parameter values as follow. The numbers of hidden units for the BiLSTM networks and the GCN networks are both 200 while the number of the GCN layers is set to 2. The models are trained with shuffled minibatches of size 32, using the Adam optimizer to update the parameters. The trade-off parameter λ for multi-task learning in Equation 4 for **VUA ALL POS**, **VUA VERB**, **MOH-X** and **TroFi** are all set to 1.

Comparing to the State of the Art

This section compares the proposed model (called **MUL_GCN**) with the state-of-the-art models in metaphor detection. In particular, similar to the prior work (Gao et al. 2018; Mao, Lin, and Guerin 2019), the following baseline models are selected for comparison: **Lexical Baseline**: a simple system based on the metaphorical frequency of the words (Gao et al. 2018), **SimNet**: the neural similarity networks using skip-gram word embeddings in (Rei et al. 2017), **CNN+BiLSTM**: the ensemble model with Convolutional Neural Networks (CNN) and BiLSTM in (Wu et al. 2018). This is the best model among the participants of the NAACL-2018 Metaphor Shared Task (i.e., the workshop on Figurative Language Processing), **RNN_CLS**: the BiLSTM model with attention in (Gao et al. 2018) for the classification setting, **RNN_SEQ_ELMo**: the BiLSTM model in (Gao et al. 2018) for the sequential prediction setting, **RNN_SEQ_BERT**: this model (reported in (Mao, Lin, and Guerin 2019)) is similar to **RNN_SEQ_ELMo** except that the ELMo embeddings are replaced by the BERT embeddings (Devlin et al. 2019), **RNN_HG**: the BiLSTM model based on the MIP principle in (Mao, Lin, and Guerin 2019), and **RNN_MHCA**: the BiLSTM model with contextual attention and the SPV principle in (Mao, Lin, and Guerin 2019). **RNN_MHCA** is recently proposed and has the best reported performance for metaphor detection in the literature[1].

Table 1 presents the performance where F1 is the most important measure for this task (Mao, Lin, and Guerin 2019). There are two different settings/approaches to do metaphor detection in the literature, i.e., the sequential labeling setting and the classification setting. In the sequential labeling setting, the models are trained to predict a sequence of binary labels to indicate the metaphoricity of the words in the sentences (i.e., the models with †in Table 1) while the classification setting determines the metaphorcity of the words in the sentences independently as a word classification problem (i.e., the way we model metaphor detection in this work). On the one hand, Table 1 shows that among the model with the classification setting, the proposed model significantly outperform the previous state-of-the-art model (i.e., **RNN_CLS**). The performance gap is significant and substantial with respect to **VUA VERB** and **TroFi**. On the other hand, comparing the previous sequential labeling models with the proposed model **MUL_GCN**, we see that **MUL_GCN** also has significantly better F1 score than the previous models (e.g., the current state-of-the-art system **RNN_MHCA**) on three over four considered datasets ($p < 0.01$). The only exception is on the **MOH-X** dataset where **MUL_GCN** achieves comparable performance with **RNN_MHCA**. Such evidences clearly help to demonstrate the advantages of the proposed model over the ones in the previous work. One interesting point is predicting the metaphor labels of the context words (as in the sequential labeling setting) is suggested by the previous work (Gao et al. 2018; Mao, Lin, and Guerin 2019) as the better way to do metaphor detection than the classification setting. However, in this work, we show the contrary that the classification setting can still produce metaphor detection models with the state-of-the-art performance. We attribute such achievement to the proposal of multi-task learning, the control mechanism and the GCNs that helps to boost the performance of the model in this work significantly.

[1] We do not compare to (Shutova, Kiela, and Maillard 2016) as their experiment setting is different from the current works.

Model Variations

The main components of the neural network model in this work include the BiLSTM model in the encoding module, the GCN module and the control module. This section evaluates the effectiveness of such components when they are removed from the whole model. For the GCN module, we additionally evaluate the model when the GCN module is replaced by the popular multihead self-attention layer from Transformer (Vaswani et al. 2017) to demonstrate the necessity of GCNs in this work. Similar to the prior work (Gao et al. 2018), we use the **VUA VERB** dataset for such ablation studies[2]. Table shows the performance of the models in the test datasets of **VUA VERB**.

Model	P	R	F1	Acc
MUL_GCN	72.5	70.9	71.7	83.2
MUL_GCN - BiLSTM Layer	65.9	69.3	67.6	80.0
MUL_GCN - Control Module	69.0	67.6	68.3	81.2
MUL_GCN - GCN Module	74.6	60.9	67.0	82.0
Replace GCN with Self-Attention	72.6	67.4	69.9	82.6

Table 2: Ablation Study. The models' performance on the **VUA VERB** dataset.

As we can see from the table, each component (i.e., BiLSTM, Control and GCN) is important for the proposed model **MUL_GCN** as excluding any of them would hurt the performance significantly. The replacement of GCN with self-attention also worsens the model substantially that helps to further testify to the benefit of GCNs for selecting the appropriate context words for representation learning in metaphor detection.

The Necessity of Multi-task Learning

Another important module in this work is the multi-task learning framework between WSD and metaphor learning. In order to demonstrate the effectiveness of this module for metaphor detection, this section evaluates the following baseline techniques to train the models: (1) **Single Network**: only a single network for metaphor detection is trained (i.e., completely ignoring the network for WSD), (2) **Pre-training**: a single network is trained on the WSD dataset first and then retrained on the metaphor dataset later, and (3) **Alternative**: a single model is trained for both WSD and metaphor detection, following the alternative training procedure (Guo et al. 2016). Table 3 shows the performance of the methods on the **VUA VERB** test set.

Method	P	R	F1	Acc
Mul_GCN (proposed)	72.5	70.9	71.7	83.2
Single Network	69.7	68.1	68.9	81.5
Pre-training	70.8	68.5	69.6	82.1
Alternative	72.9	65.4	69.0	82.3

Table 3: The multi-task learning performance.

From the table, we can see that the multi-task learning framework can significantly improve the **Single Network** method with substantially better F1 score. This demonstrates the benefit of WSD for metaphor detection. It is also evident that the proposed method **Mul_GCN** significantly outperforms the multi-task learning baselines by large margin on the F1 score (i.e., up to 2.7% improvement over the absolute F1), thereby corroborating the advantages of the multi-task learning mechanism in this work for metaphor detection.

Representation Similarity Variations

In order to achieve the similarity of the two vectors V^{wsd} and V^{md} in the multi-task module (i.e., Equation 4), the proposed model employs the mean squared error $M = \|V^{wsd} - V^{md}\|_2^2$ (called *MSE*) as the measure of dissimilarity to be minimized via the overall loss function. In practice, there are several alternative dissimilarity measures M that can be added into the loss function for this purpose. In this section, we additionally investigate the following dissimilarity measures M for knowledge transferring in the multi-task learning module to better understand the effect of such measure choices for the model in this work:

•Kullback-Leibler divergence (*KL*): $M = KL(S^{wsd}, S^{md}) = -\sum_i S_i^{wsd} \log \frac{S_i^{wsd}}{S_i^{md}}$ where $S^{wsd} = softmax(V^{wsd})$ and $S^{md} = softmax(V^{md})$.

•Cosine (*Cosine*): $M = 1 - \cos(V^{wsd}, V^{md})$

•The Margin Loss (*Margin*): $M = 1 - s_{wsd} + s_{md}$ where $s_{wsd} = sigmoid(FF(V^{wsd})$ and $s_{md} = sigmoid(FF(V^{md})$ with FF as a feed-forward function to transform the vectors V^{wsd} and V^{md} into scalars.

Measure	P	R	F1	Acc
MSE (proposed)	72.5	70.9	71.7	83.2
KL	73.8	66.0	69.7	82.8
Cosine	71.5	65.5	68.4	81.8
Margin	72.6	64.4	68.3	82.0

Table 4: The performance of the dissimilarity measures.

Table 4 reports the performance of such dissimilarity measures on the **VUA VERB** test set when they are used in the proposed model (i.e., replacing the *MSE* measure). It is clear from the table that the *MSE* is significantly better than the other dissimilarity measures for the model, justifying for our choice of *MSE* in this work.

Conclusion

We present a multi-task learning model for metaphor detection that features graph convolutional neural networks to appropriately capture the important context words, the control mechanism to emphasize the target words, and the knowledge transferring from word sense disambiguation to improve the performance. We achieve the state-of-the-art performance on several benchmark datasets for metaphor detection. The experimental results demonstrate the effectiveness of the components proposed in this work. In the future, we plan to (1) explore more methods and related tasks to further improve the multi-task learning for metaphor detection, and (2) extend the proposed models to other related applications.

[2] We use the **VUA VERB** dataset for these experiments, but similar trends can be seen for the other datasets.

Acknowledgments

This work has been supported in part by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA18, IARPA BETTER, the NSF grant CNS-1747798 to the IUCRC Center for Big Learning, the NSF grant CNS-1814614, and Adobe Research Gift.

References

Birke, J., and Sarkar, A. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *EACL*.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural language processing (almost) from scratch. In *CoRR*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Dong, D.; Wu, H.; He, W.; Yu, D.; and Wang, H. 2015. Multi-task learning for multiple language translation. In *ACL-IJCNLP*.

Duong, L.; Cohn, T.; Bird, S.; and Cook, P. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL*.

Gao, G.; Choi, E.; Choi, Y.; and Zettlemoyer, L. 2018. Neural metaphor detection in context. In *EMNLP*.

Gerard Steen, Aletta Dorst, B. H. A. K. T. K., and Pasma, T. 2010. A method for linguistic metaphor identification: From mip to mipvu. In *John Ben-jamins Publishing, volume 14*.

Group, P. 2007. Mip: A method for identifying metaphorically used words in discourse. In *Metaphor and symbol, 22(1):1–39*.

Guo, J.; Che, W.; Wang, H.; Liu, T.; and Xu, J. 2016. A unified architecture for semantic role labeling and relation classification. In *COLING*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. In *Neural Computation*.

Hovy, D.; Srivastava, S.; Jauhar, S. K.; Sachan, M.; Goyal, K.; Li, H.; Sanders, W.; and Hovy, E. 2013. Identifying metaphorical word use with tree kernels. In *Proceedings of the First Workshop on Metaphor in NLP*.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Klebanov, B. B.; Leong, C. W.; Gutierrez, E. D.; Shutova, E.; and Flor, M. 2016. Semantic classifications for detection of verb metaphors. In *ACL*.

Lakoff, G., and Johnson, M. 1980. Metaphors we live by. In *University of Chicago Press, Chicago*.

Leong, C. W. B.; Beigman Klebanov, B.; and Shutova, E. 2018. A report on the 2018 VUA metaphor detection shared task. In *Workshop on Figurative Language Processing*.

Mao, R.; Lin, C.; and Guerin, F. 2018. Word embedding and WordNet based metaphor identification and interpretation. In *ACL*.

Mao, R.; Lin, C.; and Guerin, F. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *ACL*.

Mason, Z. J. 2004. CorMet: A computational, corpus-based conventional metaphor extraction system. In *CL*.

Miller, G. A.; Chodorow, M.; Landes, S.; Leacock, C.; and Thomas, R. G. 1994. Using a semantic concordance for sense identification. In *Proceedings of the workshop on Human Language Technology*.

Mohammad, S.; Shutova, E.; and Turney, P. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*.

Nguyen, T. H., and Grishman, R. 2018a. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Rei, M.; Bulat, L.; Kiela, D.; and Shutova, E. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. In *EMNLP*.

Shutova E., Sun L., G. D. E. L. P., and Narayanan, S. 2017. Multilingual metaphor processing: Experiments with semi-supervised and unsupervised learning. In *CL*.

Shutova, E.; Kiela, D.; and Maillard, J. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *NAACL-HLT*.

Shutova, E.; Teufel, S.; and Korhonen, A. 2013. Statistical metaphor processing. In *Computational Linguistics, 39(2) 301-353*.

Strzalkowski, T.; Broadwell, G. A.; Taylor, S.; Feldman, L.; Shaikh, S.; Liu, T.; Yamrom, B.; Cho, K.; Boz, U.; Cases, I.; and Elliot, K. 2013. Robust extraction of metaphor from novel data. In *the First Workshop on Metaphor in NLP*.

Tsvetkov, Y.; Boytsov, L.; Gershman, A.; Nyberg, E.; and Dyer, C. 2014. Metaphor detection with cross-lingual model transfer. In *ACL*.

Turney, P.; Neuman, Y.; Assaf, D.; and Cohen, Y. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *EMNLP*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wilks, Y. 1978. Making preferences more active. In *Artificial Intelligence, 11(3):197–223*.

Wu, C.; Wu, F.; Chen, Y.; Wu, S.; Yuan, Z.; and Huang, Y. 2018. Neural metaphor detecting with CNN-LSTM model. In *Proceedings of the Workshop on Figurative Language Processing*.