Adversarial Multi-task Learning for End-to-end Metaphor Detection

Shenglong Zhang **Ying Liu** [*]
Tsinghua University, Beijing, China, 100084
zsl18@mails.tsinghua.edu.cn
yingliu@mail.tsinghua.edu.cn

Abstract

Metaphor detection (MD) suffers from limited training data. In this paper, we started with a linguistic rule called Metaphor Identification Procedure and then proposed a novel multi-task learning framework to transfer knowledge in basic sense discrimination (BSD) to MD. BSD is constructed from word sense disambiguation (WSD), which has copious amounts of data. We leverage adversarial training to align the data distributions of MD and BSD in the same feature space, so task-invariant representations can be learned. To capture fine-grained alignment patterns, we utilize the multi-mode structures of MD and BSD. Our method is totally end-to-end and can mitigate the data scarcity problem in MD. Competitive results are reported on four public datasets. Our code and datasets are available [1].

1 Introduction

Metaphor involves a mapping mechanism from the source domain to the target domain, as proposed in Conceptual Metaphor Theory (Lakoff and Johnson, 2008).

e.g. *The police **smashed** the drug ring after they were tipped off*.

Smash in the above sentence means "hit hard" literally (source domain). However, it is employed in a creative way, indicating "overthrow or destroy" (target domain). The mapping from the source to the target makes the word a metaphor.

Understanding metaphors in human languages is essential for a machine to dig out the underlying intents of speakers. Thus, metaphor detection and understanding are crucial for sentiment analysis (Cambria et al., 2017), and machine translation(Mao et al., 2018), etc.

Metaphor detection (MD) requires a model to predict whether a specific word is literal or metaphorical in its current context. Linguistically, if there is a semantic conflict between the contextual meaning and a more basic meaning, the word is a metaphor (Crisp et al., 2007; Steen, 2010; Do Dinh et al., 2018). The advent of large Pretrained Language Models has pushed the boundaries of MD far ahead (Devlin et al., 2019; Liu et al., 2019b). However, MD suffers from limited training data, due to complex and difficult expert knowledge for data annotation (Group, 2007).

Recently, Lin et al. (2021) used self-training to expand MD corpus, but error accumulation could be a problem. Many researchers used various external knowledge like part of speech tags (Su et al., 2020; Choi et al., 2021), dictionary resources (Su et al., 2021; Zhang and Liu, 2022), dependency parsing (Le et al., 2020; Song et al., 2021), etc., to promote MD performance. These methods are not end-to-end, thus they impeded continuous training on new data.

To address the data scarcity problem in MD, we propose a novel task called basic sense discrimination (BSD) from word sense disambiguation (WSD). BSD regards the most commonly used lexical sense as a basic usage, and aims to identify whether a word is basic in a certain context. Both BSD and MD need to compare the semantic difference between the basic meaning and the current contextual meaning. Despite the lack of MD data, we can distill knowledge from BSD to alleviate data scarcity and overfitting, which leads to the usage of multi-task learning.

We design the **Ad**versarial **Mul**ti-task Learning Framework (AdMul) to facilitate the knowledge transfer from BSD to MD. AdMul aligns the data distributions for MD and BSD through adversarial training to force shared encoding layers (for example, BERT) to learn task-invariant representations. Furthermore, we leverage the internal multi-mode structures for fine-grained alignment. The literal senses in MD are forcibly aligned with basic senses

[*] Corresponding Author
[1] https://github.com/SilasTHU/AdMul

Findings of the Association for Computational Linguistics: ACL 2023, pages 1483–1497
July 9-14, 2023 ©2023 Association for Computational Linguistics

in BSD, which can push the literal senses away from the metaphorical. Similarly, the non-basic senses in BSD are aligned with metaphors in MD, which enlarges the discrepancy between basic and non-basic senses to enhance model performance.

The contributions of this paper can be summarized as follows:

- We proposed a new task, basic sense discrimination, to promote the performance of metaphor detection via a multi-task learning method. The data scarcity problem in MD can be mitigated via knowledge transfer from a related task.
- Our proposed model, AdMul, uses adversarial training to learn task-invariant representations for metaphor detection and basic sense discrimination. We also make use of multi-mode structures for fine-grained alignment. Our model is free of any external resources, totally end-to-end, and can be easily trained.
- Experimental results indicate that our model achieves competitive performance on four datasets due to knowledge transfer and the regularization effect of multi-task learning. Our zero-shot transfer result even surpasses fine-tuned baseline models.

2 Related Work

Metaphor Detection: Metaphor detection is a popular task in figurative language computing (Leong et al., 2018, 2020). With the progress of natural language processing, various methods have been proposed. Traditional approaches used different linguistic features like word abstractness, word concreteness, part of speech tags and linguistic norms, etc., to detect metaphors (Shutova and Sun, 2013; Tsvetkov et al., 2014; Beigman Klebanov et al., 2018; Wan et al., 2020). These methods are not end-to-end and rely heavily on feature engineering.

The rise of deep learning boosted the advancement of metaphor detection significantly. Gao et al. (2018), Wu et al. (2018) and Mao et al. (2019) used RNN and word embeddings to train MD models. Recently, lots of works combined the advantages of pre-trained language models and external resources to enhance the performance of metaphor detection (Su et al., 2020, 2021; Choi et al., 2021; Song et al., 2021; Zhang and Liu, 2022). Though great improvements have been made, these models still suffer from the lack of training data, which is well exemplified by their poorer performance on small datasets.

Multi-task Learning: Multi-task learning (MTL) can benefit a target task via related tasks. It has brought great success in computer vision and natural language processing. MTL learns universal representations for different task inputs, so all tasks share a common feature space, where knowledge transfer becomes possible. Previous studies trained MTL models by deep neural networks like CNN or RNN, achieving promising results in text classification (Liu et al., 2017; Chen and Cardie, 2018). Liu et al. (2019a) and Clark et al. (2019) combined MTL framework with BERT (Devlin et al., 2019), obtaining encouraging results on multiple GLUE tasks. There are some other successful MTL applications in machine translation (Dong et al., 2015), information extraction (Nishida et al., 2019), and sentiment analysis (Liang et al., 2020), etc. Dankers et al. (2019) applied MTL to study the interplay of metaphor and emotion. Le et al. (2020) combined WSD and MD for better metaphor detection results. However, to the best of our knowledge, we are the first to use adversarial MTL for metaphor detection based on the linguistic nature of metaphors.

3 Proposed Method

3.1 Metaphor Identification Procedure

Metaphor Identification Procedure (MIP) (Crisp et al., 2007) is the most commonly used linguistic rule in guiding metaphor detection. It is originally the construction guideline of VU Amsterdam Metaphor Corpus. MIP indicates that if a word contrasts with one of its more basic meanings but can be understood in comparison with it, then the word is a metaphor. A more basic meaning is more concrete, body-related, more precise, or historically older (Steen, 2010; Do Dinh et al., 2018).

Some researchers have pointed out that when a word is used alone, it is very likely to depict a more basic meaning (Choi et al., 2021; Song et al., 2021). We concatenate the target word and the sentence as input. In the input, the first segment is the target used alone, presenting a more basic meaning. The second segment is the whole sentence, which can encode the contextual meaning of the target. Then the model adopts MIP to detect metaphors.

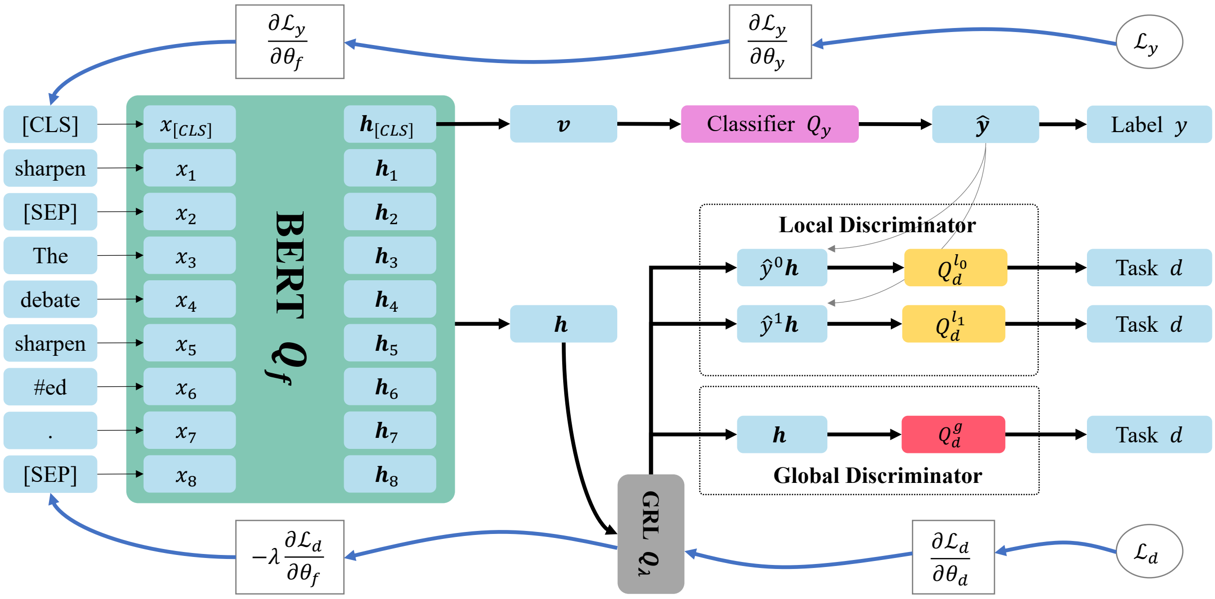

Figure 1: AdMul architecture. The **black** arrows mean forward propagation, while the **blue** ones denote back propagation. BERT is the shared feature extractor Q_f. GRL stands for gradient reversal layer. Classifier Q_y is task-specific to perform MD or BSD, and y is the label for MD or BSD. Global discriminator Q_d^g aligns overall data distribution to make BERT learn universal representations. Two local discriminators $Q_d^{l_c}$ are in line with two labels. Each is responsible for aligning the data in MD and BSD of label c. Both Q_d^g and $Q_d^{l_c}$ predict which task the input sentence comes from. Task $d \in \{0, 1\}$, 0 or MD and 1 for BSD. $\mathcal{L}_y$ is the loss for Q_y. $\mathcal{L}_d$ is the loss for Q_d^g or $Q_d^{l_c}$.

3.2 From WSD to BSD

Metaphor detection (MD) aims to identify whether a contextualized word is metaphorical. Word sense disambiguation (WSD) aims to determine the lexical sense of a certain word from a given sense inventory. The two tasks share the same nature: we should decide the sense of a given word according to its context.

A word may have multiple senses, so WSD is a multinomial classification task, whereas MD is a binary classification task. Integrating WSD with MD can be quite expensive. For example, the state-of-the-art model (Barba et al., 2021) regarded WSD as an information extraction task. It concatenated all the candidate senses and tried to extract the correct one. Such a method requires not only external dictionary resources, but also enormous computing resources since the input may be a very long sequence.

WordNet (Miller, 1995; Fellbaum, 1998) ranks the senses of a word according to its occurrence frequency[2]. The most commonly used lexical sense is at the top of the inventory list, which is usually a more basic meaning(Choi et al., 2021; Song et al., 2021; Zhang and Liu, 2022). Thus, we regard the most commonly used sense as a basic sense of a word, and try to figure out whether a word in a certain context is basic or not. We call this task basic sense discrimination (BSD). Obviously, BSD is a binary classification task and fits MD.

3.3 Task Description

Formally, given the MD dataset $\mathcal{D}_{\text{MD}} = \{(\boldsymbol{x}_i^{\text{MD}} y_i^{\text{MD}})_{i=1}^{n_{\text{MD}}}\}$ and the BSD dataset $\mathcal{D}_{\text{BSD}} = \{(\boldsymbol{x}_i^{\text{BSD}}, y_i^{\text{BSD}})_{i=1}^{n_{\text{BSD}}}\}$, they have n_{MD} and n_{BSD} labeled training samples respectively. $\boldsymbol{x} = ([\text{CLS}], \text{target}, [\text{SEP}], \text{sentence}, [\text{SEP}])$. Usually, MD and BSD have different data distributions p, so $p_{\text{MD}}(\boldsymbol{x}_{\text{MD}}) \neq p_{\text{BSD}}(\boldsymbol{x}_{\text{BSD}})$. Both $\mathcal{D}_{\text{MD}}$ and$\mathcal{D}_{\text{BSD}}$ will be used to train a multi-task learning model, which will align p_{MD} and p_{BSD} in a same feature space via adversarial training. Our goal is to minimize the risk $\epsilon = \mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{MD}}}[f(\boldsymbol{x}) \neq y]$. We actually use BSD as an auxiliary task and only care about the performance of MD.

3.4 Model Details

We present AdMul to tackle MD and BSD simultaneously. As Fig. 1 shows, AdMul has five key parts: shared feature extractor Q_f (BERT in our case, the green part), task-specific classifier Q_y (the purple part), gradient reversal layer Q_λ (the grey

[2] https://wordnet.princeton.edu/frequently-asked-questions

part), global task discriminator Q_d^g (the red part) and local task discriminators $Q_d^{l_c}$ (the yellow part).

3.4.1 Feature Extractor

AdMul adopts BERT as the feature extractor Q_f, which is shared by both MD and BSD. We take the BERT hidden state of [CLS] as a semantic summary of the input segment pair (Devlin et al., 2019). [CLS] can automatically learn the positions of two target words in the two segments, and then perceive the semantic difference via self-attention mechanism (Vaswani et al., 2017). The hidden state then goes through a non-linear activation function and produces semantic discrepancy feature $\boldsymbol{v}$:

$$\boldsymbol{v} = \tanh\left(Q_f\left(x_{[CLS]}\right)\right). \tag{1}$$

On the other hand, we use the whole input sequence $\boldsymbol{x}$ to generate sentence embedding $\boldsymbol{h}$ via average pooling:

$$\boldsymbol{h} = Q_f\left(\boldsymbol{x}\right). \tag{2}$$

3.4.2 Task-specific Classifier

Task-specific classifier Q_y takes semantic discrepancy feature $\boldsymbol{v}$ as input. For the sake of brevity, we only draw a single classifier in the diagram. Actually, we are using use different classifiers for MD and BSD.

$$\hat{\boldsymbol{y}} = Q_y(\boldsymbol{v}) = \mathrm{softmax}(W_{Q_y}\boldsymbol{v} + b_{Q_y}), \tag{3}$$

where $\hat{\boldsymbol{y}} \in \mathbb{R}^2$ is the predicted label distribution of $\boldsymbol{x}$. W_{Q_y} and b_{Q_y} are weights and bias of Q_y. Finally, we can compute classification losses:

$$\mathcal{L}_y^{\mathrm{MD}} = \frac{1}{|\mathcal{D}_{\mathrm{MD}}|}\sum\nolimits_{i=1}^{|\mathcal{D}_{\mathrm{MD}}|} L_{CE}\left(\hat{y}_i, y_i\right), \tag{4}$$

$$\mathcal{L}_y^{\mathrm{BSD}} = \frac{1}{|\mathcal{D}_{\mathrm{BSD}}|}\sum\nolimits_{i=1}^{|\mathcal{D}_{\mathrm{BSD}}|} L_{CE}\left(\hat{y}_i, y_i\right), \tag{5}$$

where L_{CE} is a cross-entropy loss function. $\hat{y}_i$ and y_i are the predicted probability and the ground truth label of the i-th training sample respectively.

3.4.3 Gradient Reversal Layer

Gradient Reversal Layer (GRL) Q_λ is the key point of adversarial learning (Ganin and Lempitsky, 2015). During the forward propagation, GRL works as an identity function. While during the back propagation, it will multiply the gradient by a negative scalar $-\lambda$ to reverse the gradient. The operations can be formulated as the following pseudo function:

$$Q_\lambda(\boldsymbol{h}) = \boldsymbol{h}, \tag{6}$$

$$\frac{\partial Q_\lambda(\boldsymbol{h})}{\partial \boldsymbol{h}} = -\lambda I, \tag{7}$$

where I is an identity matrix and λ can be computed automatically (see Section 4.3).

3.4.4 Global Discriminator

Sentence embedding $\boldsymbol{h} = Q_f(\boldsymbol{x})$ first goes through GRL, then global discriminator Q_d^g tries to predict which task $\boldsymbol{h}$ belongs to. The training objective of Q_d^g is:

$$\mathcal{L}_d^g = \frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}_i \in \mathcal{D}} L_{CE}\left(Q_d^g(Q_f(\boldsymbol{x}_i)), d_i\right), \tag{8}$$

where $\mathcal{D} = \mathcal{D}_{\mathrm{MD}} \cup \mathcal{D}_{\mathrm{BSD}}$. d_i is the task label for input $\boldsymbol{x}_i$ ($d = 0$ for MD and $d = 1$ for BSD).

The feature extractor Q_f tries to generate similar features to fool global task discriminator Q_d^g, so that Q_d^g cannot accurately discern the source task of the input feature. The features that cannot be used to distinguish the source are task-invariant (Liu et al., 2017; Chen and Cardie, 2018). As the model converges, Q_f will learn universal representations to align the distributions for MD and BSD.

3.4.5 Local Discriminator

We noticed some corresponding patterns between MD and BSD via simple linguistic analysis. As Fig. 2 illustrates.

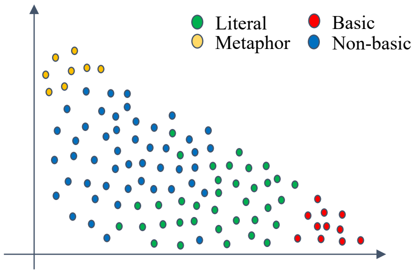

Figure 2: Multi-mode structures of MD and BSD data distributions.

The samples in MD can be classified as literal or metaphorical, while the samples in BSD can be categorized as basic or non-basic. A basic sense (red samples) must be literal (green samples), so they are clustered closer in the feature space. A metaphor (yellow samples) must be non-basic (blue samples), hence they are closer. Moreover, the metaphorical and the basic are significantly dissimilar, so they lie at different corners in the feature space, far from each other. If we bring the literal and the basic closer, then the dividing line between the metaphorical and the literal will be clearer. If the metaphorical and the non-basic get closer, then BSD will be promoted as well. Better performance of BSD will strengthen knowledge transfer from BSD to MD.

Such multi-mode patterns inspire us to apply fine-grained alignment (Pei et al., 2018; Yu et al., 2019). We forcibly push the class 0 samples (literal in MD and basic in BSD) closer, and cluster the class 1 samples (metaphor in MD and non-basic in BSD) closer. Therefore, we use two local discriminators. Each aligns samples from class $c \in \{0, 1\}$:

$$\mathcal{L}_d^l = \frac{1}{|\mathcal{D}|} \sum_{c=0}^{C} \sum_{\boldsymbol{x}_i \in \mathcal{D}} w_d L_{CE}(Q_d^{l_c}(\hat{y}_i^c Q_f(\boldsymbol{x}_i)), d_i), \tag{9}$$

where d_i is the task label and C is the number of classes. $d = 0$ for MD and $d = 1$ for BSD. w_d is a task weight. To maintain the dominance of MD in local alignment, we set $w_0 = 1$ and $w_1 = 0.3$ in all experiments. $\hat{y}_i^c$ comes from Eq. 3. The classifier Q_y will deliver a normalized label distribution for each sample $\boldsymbol{x}_i$, no matter which task it belongs to. We can view it as an attention mechanism. Q_y thinks $\boldsymbol{x}_i$ has a probability of $\hat{y}_i^c$ to be class c. Then we use the label distribution as attention weights to apply to the sample. In practice, it performs better than hard attention, because more information can be considered.

The training of local discriminators is also adversarial. The feature extractor Q_f generates task-invariant features to fool local discriminators $Q_d^{l_c}$, so that $Q_d^{l_c}$ cannot discern which task the features in class c come from.

3.4.6 Training Objective

The training of AdMul involves multiple objectives. It can be formulated as the loss function below:

$$\begin{aligned} &\mathcal{L}(\theta_f, \theta_d, \theta_y) = \\ &\quad \mathcal{L}_y^{\text{MD}} + \alpha \mathcal{L}_y^{\text{BSD}} - \lambda(\beta \mathcal{L}_d^g + \gamma \mathcal{L}_d^l), \end{aligned} \tag{10}$$

where α, β and γ are hyper-parameters to balance the loss magnitudes. θ_f, θ_d and θ_y are parameters of Q_f, Q_d (all discriminators) and Q_y respectively.

The optimization of $\mathcal{L}$ involves a mini-max game like Generative Adversarial Network (Goodfellow et al., 2014). The feature extractor Q_f tries to make the deep features as similar as possible, so that both global and local task discriminators cannot differentiate which task they come from. After the training converges, the parameters $\hat{\theta}_f$, $\hat{\theta}_y$ and $\hat{\theta}_d$ will deliver a saddle point of Eq. 10:

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} \mathcal{L}(\theta_f, \theta_y, \theta_d), \tag{11}$$

$$(\hat{\theta}_d) = \arg\max_{\theta_d} \mathcal{L}(\theta_f, \theta_y, \theta_d). \tag{12}$$

At the saddle point, θ_y will minimize classification loss $\mathcal{L}_y$ (combined by $\mathcal{L}_y^{\text{MD}}$ and $\mathcal{L}_y^{\text{BSD}}$). θ_d will minimize task discrimination loss $\mathcal{L}_d$ (combined by $\mathcal{L}_d^g$ and $\mathcal{L}_d^l$). θ_f will maximize the loss of task discriminators (features are task-invariant, so the task discrimination loss increases). AdMul can be easily trained via standard gradient descent algorithms. We take stochastic gradient descent (SGD) as an example:

$$\theta_f \longleftarrow \theta_f - \eta \left(\frac{\partial \mathcal{L}_y^i}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d^i}{\partial \theta_f} \right), \tag{13}$$

$$\theta_y \longleftarrow \theta_y - \eta \left(\frac{\partial \mathcal{L}_y^i}{\partial \theta_y} \right), \tag{14}$$

$$\theta_d \longleftarrow \theta_d - \eta \left(\frac{\partial \mathcal{L}_d^i}{\partial \theta_d} \right), \tag{15}$$

where i denotes the i-th training sample and η is learning rate. The update for θ_y and θ_d is the same as SGD. As for θ_f, if there is no minus sign for $\frac{\partial \mathcal{L}_d^i}{\partial \theta_f}$, then SGD will minimize the task discrimination loss $\mathcal{L}_d$, which means the features generated by Q_f are dissimilar across tasks (Ganin and Lempitsky, 2015).

4 Experiments

4.1 Datasets

Four metaphor detection datasets are used in our experiments. The information is shown in Table 1. **VUA All** (Steen, 2010) is the largest metaphor detection dataset to date. VUA All labels each word in a sentence. The sentences are from four genres, namely academic, conversation, fiction, and news. **VUA Verb** (Steen, 2010) is drawn from VUA All dataset. The target words are all verbs. **MOH-X** (Mohammad et al., 2016) is sampled from WordNet, with only verb targets included. WordNet

Dataset	#Sent.	#Tar.	%Met.	Avg. Len
VUA All$_{tr}$	6,323	116,622	11.19	18.4
VUA All$_{val}$	1,550	38,628	11.62	24.9
VUA All$_{te}$	2,694	50,175	12.44	18.6
VUA Verb$_{tr}$	7,479	15,516	27.90	20.2
VUA Verb$_{val}$	1,541	1,724	26.91	25.0
VUA Verb$_{te}$	2,694	5,873	29.98	18.6
MOH-X	647	647	48.69	8.0
TroFi	3,737	3,737	43.54	28.3

Table 1: MD Datasets information. **#Sent.**: Number of sentences. **#Tar.**: Number of target words. **%Met.**: Proportion of metaphors. **Avg. Len**: Average sentence length.

creates a sense inventory for each verb, of which some may have metaphorical senses. **TroFi** (Birke and Sarkar, 2006, 2007) is a dataset collected from 1987-1989 Wall Street Journal Corpus via an unsupervised method. TroFi only has verb targets as well.

We use a word sense disambiguation (WSD) toolkit (Raganato et al., 2017) to create the basic sense discrimination (BSD) dataset. The toolkit provides SemCor (Miller et al., 1994), the largest manually annotated dataset for WSD. We filter out the targets that have less than 3 senses to balance the magnitudes of WSD and MD datasets. The information of BSD dataset is shown in Table 2.

Dataset	#Sent.	#Tar.	%Basic	Avg. Len
SemCor$_{\text{BSD}}$	34,479	130,808	60.83	22.34

Table 2: BSD Dataset information. **%Basic**: Proportion of basic senses.

4.2 Compared Methods

RNN_ELMo and **RNN_BERT** (Gao et al., 2018) are two end-to-end models use both GloVe and ELMo embeddings.

RNN_HG and **RNN_MHCA** (Mao et al., 2019) are based on RNN. Both models regard the static GloVe embedding as literal, and dynamic ELMo embedding can present metaphorical senses. RNN_HG and RNN_MHCA also utilize linguistic rules.

MUL_GCN (Le et al., 2020) uses multi-task learning to transfer knowledge from WSD to MD. However, it does not use shared layers. The knowledge transfer is accomplished via a loss term. MUL_GCN also leverages dependency relations.

DeepMet (Su et al., 2020) is the winning method in the 2020 VUA and TOEFL Metaphor Detection Shared Task (Leong et al., 2020). DeepMet is built upon BERT, with various external resources like fine-grained part of speech tags utilized.

MelBERT (Choi et al., 2021) is designed upon RoBERTa. It uses a late-interaction mechanism to encode the literal meaning and the contextual meaning of a target respectively. MelBERT also leverages part of speech information.

MrBERT (Song et al., 2021) uses relation classification paradigm to detect metaphors. It embeds dependency relations into input to fine-tune pre-trained BERT, with various relation models applied.

MisNet (Zhang and Liu, 2022) is a linguistics-driven model. Two linguistic rules, namely Metaphor Identification Procedure and Selectional Preference Violation (Wilks, 1975, 1978) guide the model design. MisNet regards MD as semantic matching, with dictionary resources leveraged.

4.3 Implementation Details

We use DeBERTa$_{base}$ as the backbone (feature extractor Q_f in Fig. 1) for all experiments (He et al., 2021), through the APIs provided by HuggingFace (Wolf et al., 2020). The embedding dimension is 768. We set the maximum input sequence length as 150. The optimizer is AdamW (Peters et al., 2019). We let $\alpha = 0.2$, $\beta = 0.1$, and $\gamma = 0.1$ according to the model performance on VUA Verb, and apply them to the rest datasets. The total training epoch, batch size, and learning rate are specific for each dataset, as Table 3 shows.

Dataset	Epochs	Batch Size	LR
VUA All	8	64	3e-5
VUA Verb	5	64	3e-5
MOH-X	5	32	2e-5
TroFI	10	64	1e-5

Table 3: Hyper-parameters. **LR** stands for learning rate.

Instead of using a fixed constant, the parameter λ in GRL (Eq. 7) is set by $\lambda = \frac{m}{1+\exp(-10p)} - n$, where $m = 1.4$ and $n = 0.6$. $p = \frac{t}{T}$, where t and T are the current training step and the maximum training step respectively. λ is increasing from 0.1 to 0.8 in our case. Such a method stabilizes the training (Ganin and Lempitsky, 2015). At the beginning of the training, λ should be small so that the generated feature is not too hard for task discrimination. With training going on, adversarial

Model	VUA All				VUA Verb				MOH-X (10 fold)				TroFi (10 fold)			
	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.
RNN_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	70.1	71.6	71.1	74.6
RNN_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*
DeepMet	*82.0*	71.3	76.3	-	*79.5*	70.8	74.9	-	-	-	-	-	-	-	-	-
MelBERT	80.1	76.9	78.5	-	78.7	72.9	75.7	-	-	-	-	-	-	-	-	-
MrBERT	**82.7**	72.5	77.2	*94.7*	**80.8**	71.5	*75.9*	*86.4*	80.0	*85.1*	82.1	81.9	70.4	74.3	72.2	75.1
MisNet	80.4	*78.4*	**79.4**	**94.9**	78.3	73.6	*75.9*	86.0	*84.2*	84.0	*83.4*	*83.6*	67.5	77.6	71.9	73.6
AdMul	78.4	**79.5**	*79.0*	*94.7*	78.5	**78.1**	**78.3**	**87.0**	**87.4**	**88.8**	**87.9**	**88.0**	*70.5*	**79.8**	**74.7**	**76.5**

Table 4: MD Results on VUA All, VUA Verb, MOH-X, and TroFi. The first four baseline models are end-to-end. The best performance for each metric in **bold**, and the second best in *italic underlined*.

Model	Verb				Adjective				Adverb				Noun			
	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.
RNN_ELMo	68.1	71.9	69.9	-	56.1	60.6	58.3	-	67.2	53.7	59.7	94.8	59.9	60.8	60.4	-
RNN_BERT	67.1	72.1	69.5	87.9	58.1	51.6	54.7	88.3	64.8	61.1	62.9	94.8	63.3	56.8	59.9	88.6
RNN_HG	66.4	75.5	70.7	*88.0*	59.2	*65.6*	62.2	89.1	61.0	66.8	63.8	94.5	60.3	66.8	63.4	88.4
RNN_MHCA	66.0	76.0	70.7	87.9	61.4	61.7	61.6	89.5	66.1	60.7	63.2	*94.9*	69.1	58.2	63.2	89.8
DeepMet	**78.8**	68.5	73.3	-	**79.0**	52.9	63.3	-	*79.4*	66.4	72.3	-	*76.5*	57.1	65.4	-
MelBERT	74.2	75.9	*75.1*	-	69.4	60.1	64.4	-	**80.2**	69.7	**74.6**	-	75.4	66.5	*70.7*	-
MisNet	*77.5*	*77.7*	**77.6**	**91.4**	68.8	65.2	*67.0*	*91.2*	76.4	*70.5*	73.3	**96.3**	74.4	*67.2*	70.6	*91.6*
AdMul	77.2	**78.1**	**77.6**	**91.4**	*72.4*	**66.9**	**69.5**	**92.0**	76.3	**71.3**	*73.7*	**96.3**	**77.0**	**70.3**	**73.5**	**92.4**

Table 5: MD Breakdown results on VUA All for open word classes, the most important parts of metaphor detection. The first four baseline models are end-to-end.

training can be strengthened for better knowledge transfer. We choose the best model on the validation set for testing. Since MOH-X and TroFi do not have the training, validation, and testing split, we leverage 10-fold cross-validation. In each iteration, we pack MD and BSD samples into a mini-batch input. They have the same amount (half of the batch size). All experiments are done on an RTX 3090 GPU and CUDA 11.6.

5 Metaphor Detection Results

5.1 Overall Results

To be consistent with previous studies (Mao et al., 2018; Choi et al., 2021; Zhang and Liu, 2022), we mainly focus on the F1 score. As Table 4 shows, our proposed AdMul obtains great improvements compared with the baseline models. Best scores are reported on 3 out of 4 datasets, including VUA Verb, MOH-X, and TroFi. We attain a comparable result to the state-of-the-art model on VUA All as well. The average F1 score across 4 datasets is 79.98, which is 2.33 points higher than MisNet (77.65 on average). We notice that AdMul performs better on small datasets (VUA Verb, MOH-X, and TroFi) than the large dataset (VUA All). We attribute it to different dataset sizes. Deep learning models need numerous data to achieve good performance, so MTL can help. The knowledge distilled from BSD can greatly promote MD, especially when faced with severe data scarcity problems. MTL also works as a regularization method to avoid overfitting via learning task-invariant features (Liu et al., 2019a). However, VUA All is a large dataset, so there may be a marginal utility for more data from a related task. VUA All requires predictions for each word class as well, while BSD only has open class (i.e., verb, noun, adjective, and adverb) words. Consequently, the rest word class targets cannot get enough transferred knowledge.

The most significant enhancement is from MOH-X. BSD dataset and MOH-X are both built upon WordNet, so the data distributions can be very similar. In such a case, AdMul can easily align globally, and pay more attention to local alignment. The improvement from TroFi is barely satisfactory. TroFi is built via an unsupervised method, therefore it may contain many noises. Many baseline models perform mediocrely on TroFi as observed.

MUL_GCN is the only chosen baseline method in our experiments. MUL_GCN used an L2 loss term to force the encoder of MD and the encoder of WSD to generate similar deep features for both MD and WSD data. However, MUL_GCN only leveraged the features at the output layer, without using parameter-sharing strategy. Thus MUL_GCN did not allow latent interaction between different data distributions, and that is why our method performs better.

5.2 VUA All Breakdown Results

Table 5 shows a breakdown analysis of VUA All dataset. The most important part of MD is the model performance on open class words. As we can see, AdMul achieves the best F1 scores on 3 out of 4 word classes, and acquires a result similar to MelBERT on adverbs. The biggest gains are reported on nouns, with 2.8 absolute F1 score improvements against the strongest baseline MelBERT. The enhancement in adjectives is also encouraging (2.5 absolute improvements against MisNet). Though AdMul performs slightly less well than MisNet on VUA All, AdMul obtains better results on open class words. As we mentioned before, WordNet only has annotated knowledge for open class words, which demonstrates that AdMul can get benefits from MTL.

5.3 VUA All Genres

The sentences of VUA All dataset originate from four genres, namely academic, conversation, fiction, and news. The performance of our proposed AdMuL on the four genres is shown in Table 6.

Genre	**Pre.**	**Rec.**	**F1**	**Acc.**
Academic	83.9	83.5	83.7	94.4
Conversation	66.6	73.9	70.1	95.2
Fiction	74.5	81.7	77.9	95.7
News	81.1	76.4	78.7	93.7

Table 6: Performance of four genres in VUA All.

The performance of conversation is inferior to the others. Conversations have more closed word classes (e.g., conjunctions, interjections, prepositions, etc.). The performance on academic is the best, since it has more open class words, which are adequate in WordNet. VUA All dataset annotates metaphoricity for closed word classes as well. However, these cases may be confusing.

e.g. *She checks her appearance* ***in*** *a mirror.*

The preposition ***in*** in the above sentence is tagged as metaphorical. However, it is quite tricky even for humans to notice the metaphorical sense. As Table 7 shows, there are lots of words in closed classes, but our proposed AdMuL cannot get transferred knowledge from auxiliary task BSD.

	POS	**Train**	**Val**	**Test**
Open	VERB	20,917	7,152	9,872
	NOUN	20,514	6,859	8,588
	ADJ	9,673	3,213	3,965
	ADV	6,973	2,229	3,393
Closed	PART	2,966	1,137	1,463
	PRON	6,942	2,230	3,955
	ADP	13,310	4,556	5,300
	DET	10,807	3,541	4,118
	CCONJ	3,645	1,369	1,581
	INTJ	734	159	398

Table 7: Number for different word classes in VUA All dataset.

5.4 Zero-shot Transfer

We use AdMul trained on VUA All to conduct zero-shot transfer on two small datasets, i.e., MOH-X and TroFi. The results are shown in Table 8. Though the performance on VUA All is inferior to MisNet, AdMul has a stronger generalization ability, defeating the baseline models in all metrics across two datasets. It is worth mentioning that DeepMet and MelBERT are trained on an expanded version of VUA All (Choi et al., 2021), so they have more data than us. Our zero-shot performance on MOH-X is even better than fine-tuned MisNet, the previous state-of-the-art method (see Table 4).

Model	**MOH-X (Zero-shot)**				**TroFi (Zero-shot)**			
	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.
DeepMet	*79.9*	76.5	77.9	-	53.7	72.9	61.7	-
MelBERT	79.3	79.7	79.2	-	53.4	74.1	62.0	-
MrBERT	75.9	84.1	79.8	79.3	*53.8*	75.0	62.7	61.1
MisNet	77.8	*84.4*	*81.0*	*80.7*	*53.8*	*76.2*	*63.1*	*61.2*
AdMul	**82.3**	**85.4**	**83.8**	**83.9**	**55.7**	**77.1**	**64.7**	**63.3**

Table 8: Zero-shot transfer results.

5.5 Ablation Study

We carried out ablation experiments to prove the effectiveness of each module, as Table 9 shows. We removed global discriminator Q_d^g, local discriminators $Q_d^{l_c}$, and adversarial training (no discriminators used) respectively. Each setting hurts the performance of the MTL framework. It demonstrates that we cannot naively apply MTL to combine MD and BSD. Instead, we should carefully deal with the alignment patterns globally and locally for better knowledge transfer. In addition, we tested DeBERTa$_{base}$, a model trained only on MD dataset. DeBERTa$_{base}$ takes the target word and its context as input, thus it can be viewed as a realization of MIP. The performance of DeBERTa$_{base}$ is mediocre, which indicates that the progress of AdMul is not only due to the large pre-trained language model, but closely related to our adversarial multi-task learning framework.

Model	Pre.	Rec.	F1	Acc.
AdMul	*78.5*	*78.1*	**78.3**	**87.0**
w/o global disc.	75.0	77.3	*76.2*	85.5
w/o local disc.	71.9	**80.5**	76.0	84.7
w/o adv.	**79.3**	73.0	76.0	*86.2*
DeBERTa$_{base}$	78.2	71.3	74.6	85.4

Table 9: Ablation on VUA Verb. w/o denotes *without*.

5.6 Hyper-parameter Discussion

In Eq. 10, there are three hyper-parameters, i.e., α, β, and γ that balance the loss of BSD, global alignment loss, and local alignment loss respectively. Here we conduct experiments on VUA Verb dataset to see the impacts of different loss weight values. We tune each weight with the rest fixed. The results are shown in Fig. 3. If α is too small, then the model cannot get enough transferred knowledge from BSD. On the contrary, if α is too large, then BSD will dominate the training, leading to poorer performance of MD.

Two adversarial weights β and γ share the same pattern. If they are too small, then the data distributions cannot be aligned well globally or locally, resulting in inadequate knowledge transfer. On the contrary, if they are too big, distribution alignment will dominate the training. It is worth mentioning that the training is quite sensitive to γ, because our local alignment is based on a linguistic hypothesis. We should not pay much attention to local alignment, or it will disrupt the correct semantic space, leading to bad results.

5.7 Hyper-parameter Search

In this paper, the hyper-parameters are BSD loss weight α, global alignment loss weight β, local alignment loss weight γ, learning rate η, batch size, and total training epoch. We tune each hyper-parameter with the rest fixed. α, β, and γ are searched from 0.05 to 0.5, with an interval of 0.05. η is searched

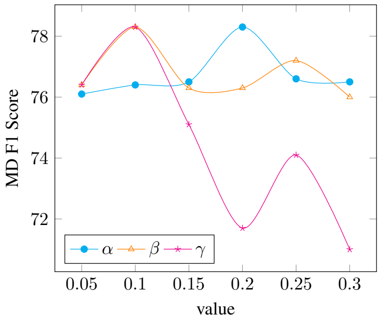

Figure 3: Impacts of hyper-parameters.

in $[1e-5, 2e-5, 3e-5, 4e-5, 5e-5]$. The batch size is selected from $[16, 32, 64]$. The total training epoch is selected from $[5, 8, 10]$. The best hyper-parameters are described in Section 4.3. As mentioned before, we tune all hyper-parameters on VUA Verb dataset, and apply them to the rest datasets, except η, batch size, and the total training epoch.

6 Conclusion

In this paper, we proposed AdMul, an adversarial multi-task learning framework for end-to-end metaphor detection. AdMul uses a new task, basic sense discrimination to promote MD, achieving promising results on several datasets. The zero-shot results even surpass the previous fine-tuned state-of-the-art method. The ablation study demonstrates that the strong ability of AdMul comes not only from the pre-trained language model, but also from our adversarial multi-task learning framework.

Acknowledgement

This work is supported by 2018 National Major Program of Philosophy and Social Science Fund (18ZDA238), and Tsinghua University Initiative Scientific Research Program (2019THZWJC38).

Limitations

Though we simply assume that the most commonly used lexical sense is a more basic sense and such an assumption fits most cases, it may not be accurate all the time. Take the verb ***dream*** as an example. The most commonly used sense of ***dream*** according to WordNet is "have a daydream; indulge in a fantasy", which is metaphorical and non-basic. While it has another literal and basic sense, meaning "experience while sleeping". We are expecting a more fine-grained annotation system to clarify the evolution of different senses: which sense is basic and how other senses are derived. Such a system will benefit both metaphor detection and linguistic ontology studies.

Due to computing convenience, our model cannot handle long texts. An indirect metaphor needs to be determined across several sentences. Such a case is beyond our capabilities (Zhang and Liu, 2022). We will also leave it as a future work.

Ethics Statement

Our proposed AdMul aims to detect metaphors in English, and the method can also be applied to other languages or multi-lingual cases. Though our manual observations did not show that there were biased metaphor detection cases for AdMul, there may still exist biases from the pre-trained language model.

We use DeBERTa$_{base}$ in all experiments, which is pre-trained on a variety of datasets, including Wikipedia, BookCorpus[3], and CommonCrawl, etc(He et al., 2021). The total pre-training data size is about 78GB. Since AdMul needs to fine-tune DeBERTa$_{base}$, AdMul may inherit poisonous languages from the pre-trained language model, like hate speech, gender bias, stereotypes, etc.

References

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021. ConSeC: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Beata Beigman Klebanov, Chee Wee (Ben) Leong, and Michael Flor. 2018. A corpus of non-native written English annotated for metaphor. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 86–91, New Orleans, Louisiana. Association for Computational Linguistics.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336, Trento, Italy. Association for Computational Linguistics.

Julia Birke and Anoop Sarkar. 2007. Active learning for the identification of nonliteral language. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 21–28, Rochester, New York. Association for Computational Linguistics.

Erik Cambria, Soujanya Poria, Alexander Gelbukh, and Mike Thelwall. 2017. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6):74–80.

Xilun Chen and Claire Cardie. 2018. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1226–1240, New Orleans, Louisiana. Association for Computational Linguistics.

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

Peter Crisp, Raymond Gibbs, Alice Deignan, Graham Low, Gerard Steen, Lynne Cameron, Elena Semino, Joe Grady, Alan Cienki, Zoltan Kövecses, et al. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.

Verna Dankers, Marek Rei, Martha Lewis, and Ekaterina Shutova. 2019. Modelling the interplay of metaphor and emotion through multitask learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2218–2229, Hong Kong, China. Association for Computational Linguistics.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Erik-Lân Do Dinh, Hannah Wieland, and Iryna Gurevych. 2018. Weeding out conventionalized metaphors: A corpus of novel metaphor annotations.

[3] https://github.com/butsugiri/homemade_bookcorpus

In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1424, Brussels, Belgium. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 607–613, Brussels, Belgium. Association for Computational Linguistics.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks.

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

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Duong Le, My Thai, and Thien Nguyen. 2020. Multitask learning for metaphor detection with graph convolutional neural networks and word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8139–8146.

Chee Wee (Ben) 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 *Proceedings of the Second Workshop on Figurative Language Processing*, pages 18–29, Online. Association for Computational Linguistics.

Chee Wee (Ben) Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 VUA metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66, New Orleans, Louisiana. Association for Computational Linguistics.

Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, Yulan He, and Ruifeng Xu. 2020. Aspect-invariant sentiment features learning: Adversarial multi-task learning for aspect-based sentiment analysis. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 825–834, New York, NY, USA. Association for Computing Machinery.

Zhenxi Lin, Qianli Ma, Jiangyue Yan, and Jieyu Chen. 2021. CATE: A contrastive pre-trained model for metaphor detection with semi-supervised learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3888–3898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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

Rui Mao, Chenghua Lin, and Frank Guerin. 2018. Word embedding and WordNet based metaphor identification and interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1222–1231, Melbourne, Australia. Association for Computational Linguistics.

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*, pages 3888–3898, Florence, Italy. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33, Berlin, Germany. Association for Computational Linguistics.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy. Association for Computational Linguistics.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2018. Multi-adversarial domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Ekaterina Shutova and Lin Sun. 2013. Unsupervised metaphor identification using hierarchical graph factorization clustering. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 978–988, Atlanta, Georgia. Association for Computational Linguistics.

Wei Song, Shuhui Zhou, Ruiji Fu, Ting Liu, and Lizhen Liu. 2021. Verb metaphor detection via contextual relation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4240–4251, Online. Association for Computational Linguistics.

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

Chang Su, Kechun Wu, and Yijiang Chen. 2021. Enhanced metaphor detection via incorporation of external knowledge based on linguistic theories. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1280–1287, Online. Association for Computational Linguistics.

Chuandong Su, Fumiyo Fukumoto, Xiaoxi Huang, Jiyi Li, Rongbo Wang, and Zhiqun Chen. 2020. DeepMet: A reading comprehension paradigm for token-level metaphor detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 30–39, Online. Association for Computational Linguistics.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mingyu Wan, Kathleen Ahrens, Emmanuele Chersoni, Menghan Jiang, Qi Su, Rong Xiang, and Chu-Ren Huang. 2020. Using conceptual norms for metaphor detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 104–109, Online. Association for Computational Linguistics.

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

Yorick Wilks. 1978. Making preferences more active. *Artificial intelligence*, 11(3):197–223.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Chaohui Yu, Jindong Wang, Yiqiang Chen, and Meiyu Huang. 2019. Transfer learning with dynamic adversarial adaptation network. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 778–786.

Shenglong Zhang and Ying Liu. 2022. Metaphor detection via linguistics enhanced Siamese network. In *Proceedings of the 29th International Conference* *on Computational Linguistics*, pages 4149–4159, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We have discussed the limitations in Section Limitations.

☑ A2. Did you discuss any potential risks of your work?
We believe that our work is only for metaphor detection and linguistic study, so there will not be potential risks. However, we discussed the underlying poisonous languages from the pre-trained language model that we used in Section Ethics Statement.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
We have summarized the main claims in Abstract and Introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
We did not use any AI writing assistants.

B ☑ Did you use or create scientific artifacts?

We used public datasets for our experiments, and open-source libraries for implementation. Please see Section 4.1 Dataset and 4.3 Implementation Details.

☑ B1. Did you cite the creators of artifacts you used?
We have cited the datasets and open-source software in Section 4.1 Dataset and 4.3 Implementation Details.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
MOH-X dataset did not clearly tell the license. TroFi dataset is under GPL policy. VUA All and VUA Verb datasets are under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Huggingface Transformers is under the Apache-2.0 license. DeBERTa is under the MIT license. We use these artifacts for research purposes, which is permitted by the terms of all artifacts.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We use the artifacts for research purposes, which is permitted by their terms.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. The datasets we used in this work are widely used, but we cannot trace how they deal with ethical problems. However, we cannot make changes to the datasets to maintain fair comparisons with the baseline methods.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We only have a brief introduction to the used artifacts in Section 4.1 Datasets. We cited the artifacts, and the original websites or information can be easily found.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

be significant, while on small test sets they may not be.
We have statistical information for the used datasets in Section 4.1 Datasets, including the number of examples, details of the dataset split, and how they were collected.

C ☑ Did you run computational experiments?

The experiments are shown in Section 4 Experiments.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The parameters and computing device are reported in Section 4.3 Implementation Details. All experiments are completed on a single RTX 3090 GPU. The training time on VUA All, VUA Verb, MOH-X, and TroFi are about 3.5h, 18m, 10m, and 70m respectively.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
The information is discussed in Section 4.3 Implementation Details and Appendix A.2 Hyperparameter Search.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Experimental statistics can be seen in Section 5. The computing method is also clarified in Section 4.3 Implementation Details.

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Our method is end-to-end, so we did not use any existing packages.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.