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 Pre-trained 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 Adversarial Multi-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.
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 multimode 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.
### 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 \( D_{\text{MD}} = \{ (x^i_{\text{MD}}, y^i_{\text{MD}}) \}_{i=1}^{n_{\text{MD}}} \) and the BSD dataset \( D_{\text{BSD}} = \{ (x^i_{\text{BSD}}, y^i_{\text{BSD}}) \}_{i=1}^{n_{\text{BSD}}} \), they have \( n_{\text{MD}} \) and \( n_{\text{BSD}} \) labeled training samples respectively. \( x = (\text{[CLS]}, \text{target}, \text{[SEP]}, \text{sentence}, \text{[SEP]}) \). Usually, MD and BSD have different data distributions \( p \), so \( p_{\text{MD}}(x_{\text{MD}}) \neq p_{\text{BSD}}(x_{\text{BSD}}) \). Both \( D_{\text{MD}} \) and \( D_{\text{BSD}} \) will be used to train a multi-task learning model, which will align \( p_{\text{MD}} \) and \( p_{\text{BSD}} \) in a same feature space via adversarial training. Our goal is to minimize the risk \( \epsilon = \mathbb{E}_{(x,y) \sim D_{\text{MD}}[f(x) \neq y]} \). 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 \( Q_\lambda \) (the grey...
part), global task discriminator $Q_g^d$ (the red part) and local task discriminators $Q_{lc}^d$ (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 $v$:

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

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

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

### 3.4.2 Task-specific Classifier

Task-specific classifier $Q_y$ takes semantic discrepancy feature $v$ 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.

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

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

$$L_y^\text{MD} = \frac{1}{|D_\text{MD}|} \sum_{i=1}^{|D_\text{MD}|} L_{CE} \left( \hat{y}_i, y_i \right), \tag{4}$$

$$L_y^\text{BSD} = \frac{1}{|D_\text{BSD}|} \sum_{i=1}^{|D_\text{BSD}|} L_{CE} \left( \hat{y}_i, y_i \right), \tag{5}$$

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) $Q_\lambda$ 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:

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

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

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

### 3.4.4 Global Discriminator

Sentence embedding $h = Q_f(x)$ first goes through GRL, then global discriminator $Q_g^d$ tries to predict which task $h$ belongs to. The training objective of $Q_g^d$ is:

$$L_g^d = \frac{1}{|D|} \sum_{x_i \in D} L_{CE} \left( Q_g^d(Q_f(x_i)), d_i \right), \tag{8}$$

where $D = D_\text{MD} \cup D_\text{BSD}$. $d_i$ is the task label for input $x_i$ ($d = 0$ for MD and $d = 1$ for BSD).

The feature extractor $Q_f$ tries to generate similar features to fool global task discriminator $Q_g^d$, so that $Q_g^d$ 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.

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

---

**Figure 2:** Multi-mode structures of MD and BSD data distributions.
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 $c \in \{0, 1\}$:

$$L_d^i = \frac{1}{|D|} \sum_{c=0}^{C} \sum_{x_i \in D} w_d L_{CE}(Q^i_d(\hat{y}_i^c Q_f(x_i)), d_i),$$

(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 $x_i$, no matter which task it belongs to. We can view it as an attention mechanism. $Q_y$ thinks $x_i$ 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 $Q^i_d$, so that $Q^i_d$ 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:

$$L(\theta_f, \theta_d, \theta_y) = L_y^{MD} + \gamma L_y^{BSD} - \lambda (\beta L_y^g + \gamma L_d^i),$$

(10)

where $\alpha, \beta$ and $\gamma$ are hyper-parameters to balance the loss magnitudes. $\theta_f, \theta_d$ and $\theta_y$ 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:

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

(11)

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

(12)

At the saddle point, $\theta_y$ will minimize classification loss $\mathcal{L}_y$ (combined by $\mathcal{L}_y^{MD}$ and $\mathcal{L}_y^{BSD}$). $\theta_d$ will minimize task discrimination loss $\mathcal{L}_d$ (combined by $\mathcal{L}_d^g$ and $\mathcal{L}_d^i$). $\theta_f$ 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:

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

(13)

$$\theta_y \leftarrow \theta_y - \eta \left( \frac{\partial \mathcal{L}_y^g}{\partial \theta_y} \right),$$

(14)

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

(15)

where $i$ denotes the i-th training sample and $\eta$ is learning rate. The update for $\theta_y$ and $\theta_d$ is the same as SGD. As for $\theta_f$, 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
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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Sent.</th>
<th>#Tar.</th>
<th>%Basic</th>
<th>Avg. Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>SemCor</td>
<td>34,479</td>
<td>130,808</td>
<td>60.83</td>
<td>22.34</td>
</tr>
</tbody>
</table>

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 that 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Epochs</th>
<th>Batch Size</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VUA All</td>
<td>8</td>
<td>64</td>
<td>3e-5</td>
</tr>
<tr>
<td>VUA Verb</td>
<td>5</td>
<td>64</td>
<td>3e-5</td>
</tr>
<tr>
<td>MOH-X</td>
<td>5</td>
<td>32</td>
<td>2e-5</td>
</tr>
<tr>
<td>TroFi</td>
<td>10</td>
<td>64</td>
<td>1e-5</td>
</tr>
</tbody>
</table>

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

Instead of using a fixed constant, the parameter \(\lambda\) 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. \(\lambda\) 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, \(\lambda\) should be small so that the generated feature is not too hard for task discrimination. With training going on, adversarial
training can be strengthened for better knowledge transfer. We choose the best model on the validation set for each metric in **bold**, and the second best in *italic underlined*.

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 is **bold**, and the second best is *italic underlined*.

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.

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 mediocly 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.

<table>
<thead>
<tr>
<th>Genre</th>
<th>Pre.</th>
<th>Rec.</th>
<th>F1</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academic</td>
<td>83.9</td>
<td>83.5</td>
<td>83.7</td>
<td>94.4</td>
</tr>
<tr>
<td>Conversation</td>
<td>66.6</td>
<td>73.9</td>
<td>70.1</td>
<td>95.2</td>
</tr>
<tr>
<td>Fiction</td>
<td>74.5</td>
<td>81.7</td>
<td>77.9</td>
<td>95.7</td>
</tr>
<tr>
<td>News</td>
<td>81.1</td>
<td>76.4</td>
<td>78.7</td>
<td>93.7</td>
</tr>
</tbody>
</table>

Table 6: Performance of four genres in VUA All.

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.

<table>
<thead>
<tr>
<th>POS</th>
<th>Train</th>
<th>Val</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERB</td>
<td>20.917</td>
<td>7.152</td>
<td>9.872</td>
</tr>
<tr>
<td>NOUN</td>
<td>20.514</td>
<td>6.859</td>
<td>8.588</td>
</tr>
<tr>
<td>ADJ</td>
<td>9.073</td>
<td>3.213</td>
<td>3.965</td>
</tr>
<tr>
<td>ADV</td>
<td>6.973</td>
<td>2.229</td>
<td>3.393</td>
</tr>
<tr>
<td>Closed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART</td>
<td>2.966</td>
<td>1.137</td>
<td>1.463</td>
</tr>
<tr>
<td>PRON</td>
<td>6.942</td>
<td>2.230</td>
<td>3.955</td>
</tr>
<tr>
<td>ADP</td>
<td>13.310</td>
<td>4.556</td>
<td>5.300</td>
</tr>
<tr>
<td>DET</td>
<td>10.807</td>
<td>3.541</td>
<td>4.118</td>
</tr>
<tr>
<td>CONJ</td>
<td>3.645</td>
<td>1.369</td>
<td>1.581</td>
</tr>
<tr>
<td>INTJ</td>
<td>734</td>
<td>159</td>
<td>398</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Model</th>
<th>MOH-X (Zero-shot)</th>
<th>TroFi (Zero-shot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepMet</td>
<td>79.9</td>
<td>76.5</td>
</tr>
<tr>
<td>MelBERT</td>
<td>79.3</td>
<td>79.7</td>
</tr>
<tr>
<td>MrBERT</td>
<td>75.9</td>
<td>84.1</td>
</tr>
<tr>
<td>MisNet</td>
<td>77.8</td>
<td>84.4</td>
</tr>
<tr>
<td>AdMul</td>
<td>82.3</td>
<td>85.4</td>
</tr>
</tbody>
</table>

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 $Q_d^L$, 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Pre.</th>
<th>Rec.</th>
<th>F1</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdMul</td>
<td>78.5</td>
<td>78.1</td>
<td>78.3</td>
<td>87.0</td>
</tr>
<tr>
<td>w/o global disc</td>
<td>75.0</td>
<td>77.3</td>
<td>76.2</td>
<td>85.5</td>
</tr>
<tr>
<td>w/o local disc</td>
<td>71.9</td>
<td>80.5</td>
<td>76.0</td>
<td>84.7</td>
</tr>
<tr>
<td>w/o adv</td>
<td>79.3</td>
<td>73.0</td>
<td>76.0</td>
<td>86.2</td>
</tr>
<tr>
<td>DeBERTa_{base}</td>
<td>78.2</td>
<td>71.3</td>
<td>74.6</td>
<td>85.4</td>
</tr>
</tbody>
</table>

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., $\alpha$, $\beta$, and $\gamma$ 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 $\alpha$ is too small, then the model cannot get enough transferred knowledge from BSD. On the contrary, if $\alpha$ is too large, then BSD will dominate the training, leading to poorer performance of MD.

Two adversarial weights $\beta$ and $\gamma$ 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 $\gamma$, 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 $\alpha$, global alignment loss weight $\beta$, local alignment loss weight $\gamma$, learning rate $\eta$, batch size, and total training epoch. We tune each hyper-parameter with the rest fixed. $\alpha$, $\beta$, and $\gamma$ are searched from 0.05 to 0.5, with an interval of 0.05. $\eta$ is searched 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 $\eta$, 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 (2019THZWC38).

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

References


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.*