MetaMapper: Interpretable Metaphor Detection

Stanford CS224N Custom Project

Ziwen Chen
Graduate School of Business
Stanford University
ziwench@stanford.edu

Yining Mao
Department of Electrical Engineering
Stanford University
yiningm@stanford.edu

Abstract

Identifying metaphors has long been a key interest in the natural language processing (NLP) community. However, most prior research on metaphor detection has focused on token-level binary classification without specifying the metaphor's source and target domains, which makes the results less interpretable. Our project aims to address this limitation by employing a dual-branch model based on the cutting-edge MelBERT transformer architecture. We adapted MelBERT's transformer-based, token-level classification framework to a unique domain annotation dataset. Our findings reveal that while the token-level approach is effective in determining the target domain of a metaphor, it struggles to identify the source domain. These results provide fresh insights into the limitations of current metaphor detection techniques and suggest possible avenues for enhancement in the field of metaphor detection research.

1 Key Information to include

- Mentor:
- External Collaborators (if you have any):
- Sharing project:

2 Introduction

The NLP community has long been intrigued by the detection of metaphors. Metaphors play a crucial role in human cognition by helping us organize information and engage in creative thinking. They are powerful because they often connect two distinct and seemingly unrelated domains of meaning, enabling the transfer of knowledge and experience between them (Lakoff and Johnson, 2008). For instance, in the metaphor "You are wasting my time," the target domain is time, and the source domain is money or resources, essentially conveying the idea that "time is money." Moreover, the family of concepts developed around money, triggered by this metaphor, could now be used to describe time. This function of metaphors is commonly referred to as "conceptual metaphor theory."

Until now, most metaphor detection research has focused on identifying metaphorical expressions within a text. Consequently, the majority of metaphor detection studies (Leong et al., 2020; Shutova et al., 2017; Chakrabarty et al., 2021)only label individual words as metaphorical or literal without specifying the source and target domains of the metaphor. Likewise, most annotation datasets and benchmarks are developed solely for token classification accuracy. As a result, when given the sentence "You are wasting my time," current models would recognize "wasting" as metaphorical. However, it is often uncertain whether this correct result is due to the model successfully capturing the domain mapping relationship represented by the metaphor, or if the model, despite its relatively good precision in detecting metaphors, still lacks a clear understanding of the underlying domains. Furthermore, having only token label results makes interpretation more difficult, which further restricts the applicability of metaphor detection algorithms in real-world applications.

In this study, we developed a novel metaphor detection approach using a dual-branch model based on the advanced MelBERT (Choi et al., 2021) transformer architecture. While MelBERT is known for its effectiveness in metaphorical token detection tasks, we applied it to a unique domain annotation dataset (Gordon et al., 2015). Our objectives are twofold: (1) to propose a new metaphor detection method that not only performs token-level metaphor classification but also extracts target and source domains of the detected metaphor, and (2) to explore the potential of token-level metaphor classification architecture in identifying latent metaphorical domains. Consequently, our work connects the previously separate fields of token-level metaphor classification and metaphor domain extraction through a single unified model. Furthermore, our model's performance helps assess whether current metaphor detection techniques genuinely understand the function of metaphor in human cognition.

We discovered that although the MelBERT architecture appears to take into account the target and source domains of metaphors, its performance in domain extraction is imbalanced. While the MelBERT architecture can effectively detect the target domain of metaphors, its capability to identify corresponding source domains is weak. Overall, our findings indicate that current metaphor detection algorithms still struggle to understand the underlying domain mappings. It is possible that addressing the domain understanding issue more effectively could lead to significant improvements in metaphor detection performance.

3 Related Work

Metaphor detection has long been an area of interest within the NLP community. The evolution of metaphor detection has paralleled the development of major NLP tools. Initially, metaphor detection relied on feature and rule-based methods, typically created manually by linguists (Dodge et al., 2015; Shutova et al., 2016). These methods heavily depended on lexical, syntactic, and semantic cues to differentiate between literal and metaphorical language. However, feature and rule-based methods often struggled to detect novel and uncommon metaphors that were not covered by the predefined features and rules. With the advent of deep learning, neural network-based approaches, ranging from word embedding(Shutova et al., 2017), to CNN and LSTM(Wu et al., 2018), to transformer(Aghazadeh et al., 2022), have been employed for metaphor detection tasks. As with other NLP tasks, it has been observed that CNN and LSTM outperform simple word2vec embedding-based methods, while transformer-based approaches outshine the previous shallow neural network architectures.

Language models featuring a transformer architecture, such as BERT, RoBERTa, and GPT, are considered particularly well-suited for metaphor detection tasks. These pre-trained models can capture rich semantic and contextual information, making them ideal for detecting metaphors through fine-tuning. DeepMet (Su et al., 2020) detects metaphors using RoBERTa alongside various linguistic features, including global and local text context and part-of-speech features. Chen et al. propose a multi-task learning framework for metaphor detection. Choi et al. (2021) developed a novel metaphor detection structure by matching contextual and literal meaning generated via the BERT model. These models are all designed to classify tokens in a given input sentence as either metaphorical or literal.

Compared to the token classification task, domain extraction tasks have received less attention. This is partly because there is no shared, relatively large-scale benchmark dataset of metaphorical domains. Until recently, most domain classification tasks were conducted either heuristically(Chmielecki, 2013; Card et al., 2022) or using linguistic features (Ge et al., 2022). Sengupta et al. (2022) utilized BERT to identify the source domain of metaphors, but they focused only on detecting domains given the ground truth metaphor labels. Moreover, their architecture for domain extraction differs from most token-level metaphor classification tasks, making it difficult to unify the two tasks.

4 Approach

Our proposed model architecture is depicted in Figure 1. The architecture is similar to that of MelBERT. Given an input sentence, the model will first generate its embedding and frame information using transformer encoders. The embeddings and frame information are then concatenated to form MIP and SPV, which are used for both token classification and domain classification tasks.

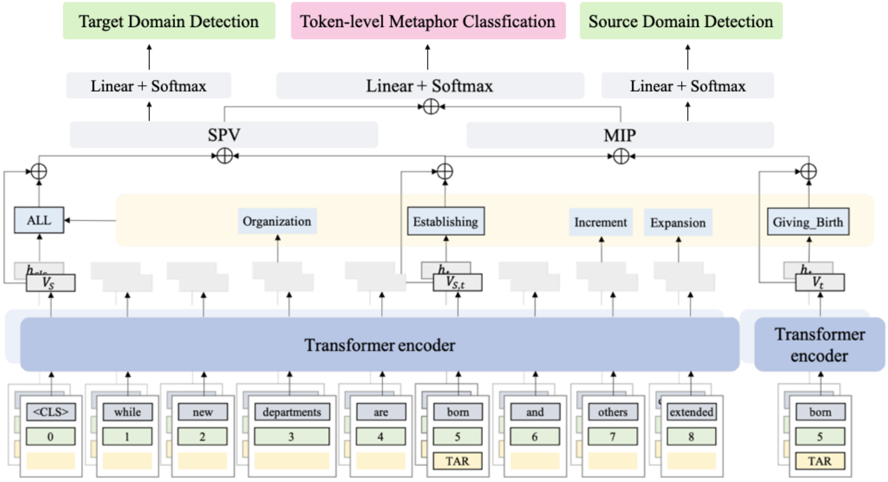


Figure 1: **Model Architecture**

We utilize a dual-branch architecture where each branch consists of a separately pre-trained MelBERT model on the VUA 20 dataset. One branch is designed specifically for metaphor detection, while the other branch is responsible for detecting the source and target domains of the metaphor. When applied for different tasks, the corresponding branch will then be fine-tuned with the given dataset.

4.1 Encoder

An input sentence is processed by two transformer encoders, one surface (sentence) encoder and one background (frame) encoder. The surface encoder transforms each word in a sentence into a set of contextualized embedding vectors, also known as hidden states. The CLS token is a special token that indicates the beginning of a sentence, and its hidden state represents the embedding of the entire sentence.

Similar to the surface encoder, the frame encoder also generates hidden states of the input sentences, but these hidden states are used to train and identify the FrameNet(Baker et al., 1998) labels of the input sentences. For example, in Figure 1, the frame of the word "born" in the sentence "while new departments are born" is "establishing." Although the original MelBERT paper does not include a FrameNet component, it has been added to the current model to make the contrast between the target and source domains more prominent, consistent to some prior practices (Stowe et al., 2020).

4.2 MIP and SPV

The underlying logic of MIP is that a metaphorical word is identified by the gap between the contextual and literal meaning of a word. MIP is implemented by concatenating the contextual embedding of the target token $V_{S,t}$, the isolated embedding V_t, and their corresponding frame information $h_{S,t}$ and h_t. The underlying logic of MIP is that a metaphorical word is identified by its semantic difference from its surrounding words. SPV is implemented by concatenating the contextual embedding of the target token $V_{S,t}$, its sentence embedding V_S, and their corresponding frame information $h_{S,t}$ and h_S. The concatenation of MIP and SPV can be expressed as:

$$h_{MIP} = V_t \oplus V_{S,t} \oplus ht \oplus hS, t$$

$$h_{SPV} = V_S \oplus V_{S,t} \oplus hS \oplus hS, t$$

4.3 Finetuning Process and Loss Function

In the process of finetuning our model on the domain specific dataset created by Gordon et al. (2015), we utilize the binary cross entropy loss to train the metaphor classification branch, and cross entropy loss to train the domain detection branch. Specifically, in each batch, we define the metaphor classification loss to be:

$$\mathcal{L}_{Meta} = BCE(y_{Meta}, \hat{y}_{Meta})$$

Where y_{Meta} and $\hat{y}_{Meta}$ are ground truth and predictions of metaphor classification labels. For the domain detection loss, it is masked with the ground truth of the true metaphor tokens since the target and source domain are all related to the specific metaphor token.

$$\mathcal{L}_{Target} = mask_{Meta}(CE(y_{Target}, \hat{y}_{Target}))$$
$$\mathcal{L}_{Source} = mask_{Meta}(CE(y_{Source}, \hat{y}_{Source}))$$

Where y_{Target} and $\hat{y}_{Target}$, y_{Source} and $\hat{y}_{Source}$ are ground truth and predictions of target and source domains respectively, and the mask is based on the ground truth of metaphor classification. The overall loss of our framework is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{Meta} + \lambda_2 \mathcal{L}_{Target} + \lambda_3 \mathcal{L}_{Source}$$

5 Experiments

5.1 Data

We employ the FrameNet dataset(Baker et al., 1998) to train frame embeddings. FrameNet is founded on the principles of Frame Semantics, which suggests that the meanings of words can be best comprehended in relation to the conceptual structures, or "frames," they invoke. FrameNet has been extensively used in natural language processing (NLP) and computational linguistics research for tasks such as semantic role labeling and information extraction.

For the metaphor token classification task, we utilize the VU Amsterdam Metaphor Corpus (VUA)(Steen, 2010). The VUA dataset has two versions, VUA 18 and VUA 20. Both serve as benchmark datasets for metaphor detection and have been used in shared metaphor detection competitions(Leong et al., 2020). The VUA dataset comprises about 16,000 annotated sentences, each with binary labels (metaphorical vs. literal) for each token in the sentence. We primarily use the VUA dataset to evaluate our model's performance on stand-alone metaphor token classification tasks.

For the metaphor domain classification task, we use the annotation dataset created by Gordon et al. (2015). While this dataset is relatively small, containing only around 1,700 annotated sentences, it is highly valuable due to its inclusion of both metaphorical tokens and their corresponding target and source domains. In total, there are 67 source domains and 14 target domains within the dataset. Our model was designed to predict the categories of the corresponding source and target domains for a given input sentence and metaphorical token. To prepare the data for our model, we preprocessed it to match the format of the VUA20 dataset. This involved augmenting the data to create separate data points for each token in each sentence. By taking this approach, we were able to maximize the amount of data available for training and achieve a more accurate and robust model.

5.2 Experimental details

Pre-training Process. The first step of our experiments is pre-train a standard MelBERT on the VUA20 dataset. We built our model upon the codebase of MelBERT using PyTorch and implemented our model following the architecture and training procedure described in the original paper. Specifically, we used a pre-trained RoBERTa3 with 12 layers, 12 attention heads in each layer, and 768 dimensions of the hidden state. We set the same hyperparameters with MelBERT, which were tuned on VUA-18dev based on F1-score. The max sequence length was set as 150. The batch size was set to 16 instead of 32 in the original paper due to GPU memory limitation. We train the model for 3 epochs with Adam optimizer. We first increased the learning rate from 0 to 3e-5 during the first two epochs and then linearly decreased it during the last epoch. We set the dropout ratio as 0.2. The training takes roughly 5 hours on one GPU with 16 GB memory.

Dual-branch Model Finetuneing Process. Our approach utilizes a dual-branch architecture where each branch consists of a separately encoder based on pre-trained MelBERT model in the previous step. Here each model is finetuned on the annotation dataset created by Gordon et al. (2015). We set the hyperparameters aligned with MelBERT. We finetune the model for 3 epochs with Adam optimizer. We set the learning rate to be 3e-5. The finetuning process takes roughly 3 hours on one GPU with 16 GB memory.

5.3 Evaluation and Results

For the metaphor token classification task, we compared our model with two baseline models: **RoBERTa_BASE** and **RoBERTa_SEQ**. These two are basic adoptions of the RoBERTa model for metaphor detection without the MelBERT infrastructure. As shown in Table 1, the performance is comparable.

Table 1: Metaphor Token Classification Results

Dataset	Model	Metrics		
		Prec	Rec	F1
VUA 18	RoBERTa_BASE	79.2	73.2	75.9
	RoBERTa_SEQ	80.2	73.8	77.0
	Our Model	81.4	73.6	**77.3**
	MelBERT	**81.4**	**74.2**	77.2
VUA 20	RoBERTa_BASE	73.8	67.7	71.1
	RoBERTa_SEQ	75.2	66.7	70.8
	Our Model	**76.7**	66.7	71.4
	MelBERT	75.8	**68.0**	**71.8**

For the domain classification task, we compared our model to the majority baseline, which represents the performance the model would achieve if it always predicts the majority source and target domain from the training dataset. We use this baseline partly because we couldn't find other works that detect both source and target domains based on the Gordon et al. (2015) dataset. Additionally, the dataset is relatively skewed, with many examples concentrated in a few target and source domains. In this case, the majority baseline can provide a useful insight into the label distribution. As shown in Table 2, our model performs well on token classification and target domain classification tasks but struggles with source domain classification tasks. Nonetheless, our model outperforms the majority baseline.

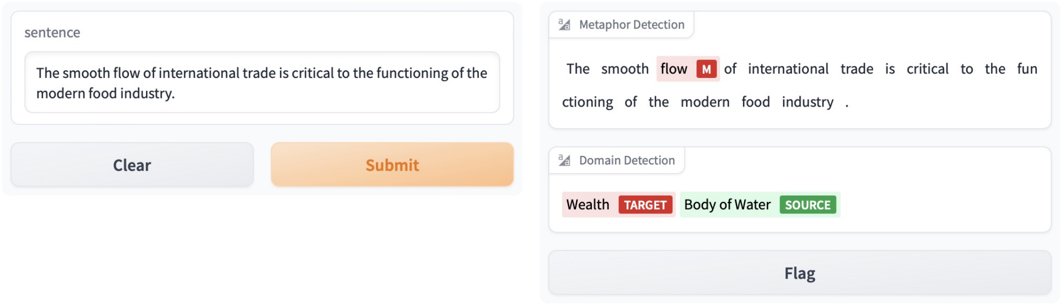
We also created a Hugging Face interface using Gradio, as illustrated in Figure 2. The interface returns both metaphorical tokens and the target and source domains they belong to. In the provided example, the word "flow" is labeled as metaphorical. Moreover, the target domain of this metaphor is "wealth," and the source domain is "body of water."

6 Analysis

For the metaphor token classification task, the main difference between our model and the baseline models is the incorporation of FrameNet information. Initially, we believed that including FrameNet would provide more concrete information about the target and source domains of detected metaphors.

Table 2: Metaphor Domain Classification Performance

		Metrics			
		Acc	Prec	Rec	F1
Our Model	Token Classification	99.1	93.0	83.4	87.9
	Source Domain Classification	18.4	0.8	3.8	1.3
	Target Domain Classification	93.4	71.5	70.4	70.8
Majority Baseline	Source Domain Classification	12.5	0.2	2.0	0.4
	Target Domain Classification	18.7	1.4	7.7	2.4



The smooth flow of international trade is critical to the functioning of the modern food industry.

The smooth flow M of international trade is critical to the fun ctioning of the modern food industry .

Wealth TARGET Body of Water SOURCE

Figure 2: **Hugging face Interface.** The user input their sentence, and our model will output the token-level predicted metaphor detection results along with target and source domains. Our model takes roughly 5 seconds to run the metaphor detection and 5 seconds to run the domain detection on CPU.

Unfortunately, we discovered that many detected metaphors lack frame information due to the sparse nature of the original FrameNet annotation. As a result, adding a FrameNet component does not significantly improve the performance of metaphor token classification. Our model's performance is quite comparable to the original MelBERT model. Generally, frame information, despite its theoretical relevance to metaphor detection, does not offer additional assistance from an engineering standpoint.

For the domain classification task, our model achieves relatively good performance on the target domain but struggles with the source domain. It is important to note that we achieved around 99% accuracy in detecting metaphorical tokens using the same dataset. This means that even though the model successfully identifies the metaphor, it cannot determine the reason behind it. We have considered several potential reasons for the performance discrepancy between target and source domain detection. First, there may not be enough data, as detecting conceptual domains is inherently more latent than surface-level token classification and requires more data for training. Furthermore, the data provided for target domain classification is more abundant than that for source domain prediction. While the model can infer the former based on both sentence embedding and token contextual embedding, it can only estimate the latter based on the isolated embedding of a single token. Lastly, domain classifications are multiclass classification tasks. While there are only around 10 classes in the target domain, the source domain is about seven times more sparse. Combined with the relatively limited data, this also makes source domain classification more challenging. If this is the case, a larger and more balanced dataset should lead to a significant performance improvement.

7 Conclusion

In this study, we developed a novel approach for detecting metaphors in natural language text using a dual-branch model based on the state-of-the-art MelBERT transformer architecture. In addition to MelBERT, we've incorporated a FrameNet component to better support metaphor detection tasks. We discovered that despite FrameNet's theoretical relevance to metaphor detection, it does not offer additional benefits in terms of detection accuracy. Moreover, while the MelBERT architecture can successfully identify metaphorical tokens and their corresponding target domains, it struggles to detect source domains. We believe this could be because domain classification is inherently more latent and challenging than surface-level token classification. Furthermore, our findings reveal that current metaphor detection methods, despite achieving relatively good performance in locating metaphorical tokens, do not possess a solid understanding of why these tokens are metaphorical. We think this could partly explain the existing performance bottleneck in metaphor detection. If NLP researchers want to enhance metaphor detection performance further, they need to develop better solutions for the domain classification problem. While token label classification offers a useful tool, detecting metaphor domains will contribute to a deeper understanding of the cognitive processes and cultural factors that shape the use of metaphorical language.

References

Ehsan Aghazadeh, Mohsen Fayyaz, and Yadollah Yaghoobzadeh. 2022. Metaphors in Pre-Trained language models: Probing and generalization across datasets and languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2050, Dublin, Ireland. Association for Computational Linguistics.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Dallas Card, Serina Chang, Chris Becker, Julia Mendelsohn, Rob Voigt, Leah Boustan, Ran Abramitzky, and Dan Jurafsky. 2022. Computational analysis of 140 years of US political speeches reveals more positive but increasingly polarized framing of immigration. *Proc. Natl. Acad. Sci. U. S. A.*, 119(31):e2120510119.

Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. MERMAID: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4250–4261, Online. Association for Computational Linguistics.

Xianyang Chen, Chee Wee Leong, Michael Flor, and Beata Beigman Klebanov. Go figure! multi-task transformer-based architecture for metaphor detection using idioms: ETS team in 2020 metaphor shared task.

Michał Chmielecki. 2013. Conceptual negotiation metaphors across cultures. *No.*, 3:103–118.

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.

Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. MetaNet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49, Denver, Colorado. Association for Computational Linguistics.

Mengshi Ge, Rui Mao, and Erik Cambria. 2022. Explainable metaphor identification inspired by conceptual metaphor theory. *AAAI*, 36(10):10681–10689.

Jonathan Gordon, Jerry Hobbs, Jonathan May, Michael Mohler, Fabrizio Morbini, Bryan Rink, Marc Tomlinson, and Suzanne Wertheim. 2015. A corpus of rich metaphor annotation. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 56–66, Denver, Colorado. Association for Computational Linguistics.

George Lakoff and Mark Johnson. 2008. *Metaphors We Live By*. University of Chicago Press.

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.

Meghdut Sengupta, Milad Alshomary, and Henning Wachsmuth. 2022. Back to the roots: Predicting the source domain of metaphors using contrastive learning. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 137–142, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170, San Diego, California. Association for Computational Linguistics.

Ekaterina Shutova, Lin Sun, Elkin Darío Gutiérrez, Patricia Lichtenstein, and Srini Narayanan. 2017. Multilingual metaphor processing: Experiments with semi-supervised and unsupervised learning. *Comput. Linguist. Assoc. Comput. Linguist.*, 43(1):71–123.

Gerard Steen. 2010. *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. John Benjamins Publishing.

Kevin Stowe, Leonardo Ribeiro, and Iryna Gurevych. 2020. Metaphoric paraphrase generation.

C Su, F Fukumoto, X Huang, J Li, and others. 2020. DeepMet: A reading comprehension paradigm for token-level metaphor detection. *Proceedings of the*.

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.