IlliniMet: Illinois System for Metaphor Detection with Contextual and Linguistic Information

Hongyu Gong Kshitij Gupta Akriti Jain Suma Bhat
University of Illinois at Urbana-Champaign, IL, USA
{hgong6,kg9,akritij,spbhat2}@illinois.edu

Abstract

Metaphors are rhetorical use of words based on the conceptual mapping as opposed to their literal use. Metaphor detection, an important task in language understanding, aims to identify metaphors in word level from given sentences. We present IlliniMet, a system to automatically detect metaphorical words. Our model combines the strengths of the contextualized representation by the widely used RoBERTa model and the rich linguistic information from external resources such as WordNet. The proposed approach is shown to outperform strong baselines on a benchmark dataset. Our best model achieves F1 scores of 73.0% on VUA ALLPOS, 77.1% on VUA VERB, 70.3% on TOEFL ALLPOS and 71.9% on TOEFL VERB.

1 Introduction

Metaphors are a form of figurative language used to make an implicit or implied comparison between two things that are unrelated (Ortony and Andrew, 1993). They are widely used in natural language, conveying rich semantic information which deviates from their literal meaning. For instance, in the sentence "Tom has always been an early bird waking up at 5:30 a.m.", the phrase "early bird" does not mean a real animal but refers to someone doing something early.

The ubiquity and subtlety of metaphors present challenges to language understanding in natural language processing. Detecting metaphors is the first step towards metaphor understanding, which helps to uncover the meaning more accurately. Metaphor detection has been used in a variety of downstream applications such as sentiment classification (Rentoumi et al., 2012) and machine translation (Koglin et al., 2015).

Some existing approaches to metaphor detection rely on linguistic features such as lexicon based metaphor constructions, lexical abstractness and word categories in WordNet (Dodge et al., 2015; Klebanov et al., 2016). Most of these approaches either do not consider the contextual information or only focus on limited contexts. Others use unigram based features regardless of their contexts (Köper and im Walde, 2017), and still others identify metaphors in the limited context of subject-verb-object triples (Bulat et al., 2017). We note that the contextual information is crucial for metaphor detection. As shown in Table 1, the word "fix" can be used both metaphorically and literally depending on its context.

Table 1: Metaphorical and Literal Usage of Word "Fix"

Metaphor: I think that we need to begin from facts and **fix** important data.
Literal: They couldn't **fix** my old computer, and I had to buy a new one.

Recent studies incorporating contextual information into metaphor detection include unsupervised approaches (Gong et al., 2017) and supervised models (Wu et al., 2018; Stowe et al., 2019). Neural networks such as Long Short Term Memory (LSTM) have achieved state-of-the-art performance in metaphor detection due to their ability to encode contextual information (Gao et al., 2018).

Recently proposed contextualized representation models that have been widely used include Embeddings from Language Models (ELMo) (Peters et al., 2018), Representations from Transformer (BERT) (Devlin et al., 2019) and Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al., 2019). Their use has shown dramatic improvements in the performance of several NLP tasks. These models are pretrained on large text corpora to encode rich contextualized knowledge into the semantic representation of words with deep network structures.

Proceedings of the Second Workshop on Figurative Language Processing, pages 146–153

https://doi.org/10.18653/v1/P17

In this paper, we build our metaphor detection model upon RoBERTa to leverage its strength in capturing contextual information. In addition, we enhance the power of contextualized representation by using linguistic features. Using external resources our model integrates features such as word concreteness, which serves to complement the contextual knowledge in RoBERTa embeddings and aid the task of metaphor detection. This model was our contribution to the Second Shared Task on Metaphor Detection (Leong et al., 2020). Our best model achieves F1 scores of 73.0% and 77.1% on all words and verbs alone respectively for metaphors in the VUA dataset. Our model performance is 70.3% and 71.9% on all words and only verbs respectively in the TOEFL dataset. We make our implementation available for a wider exploration[1].

2 Related Work

Metaphor Detection. Metaphor detection has recently attracted a lot of research interest in the area of natural language processing. The metaphor detection task can be broadly classified into two categories. The first involves predicting whether a given word or phrase is a metaphor (Gong et al., 2017). The second category can be formulated as a sequential labeling task of predicting the metaphorical or literal usage of every word in a sentence (Leong et al., 2018). Our current study falls into the second category.

Feature-based approaches. Many recent works have explored the use of various linguistic features for automatic metaphor detection. These features include word abstractness (Köper and im Walde, 2017), WordNet features, hypernyms and synonyms (Mao et al., 2018), syntactic dependencies and semantic patterns (Hovy et al., 2013), word imageability (Strzalkowski et al., 2013) as well as word embeddings (Köper and im Walde, 2017).

Neural network models. Deep learning models are becoming very popular in various downstream applications of natural language processing owing to the ability to train models in an end-to-end manner without explicit feature engineering. Approaches built upon neural networks have shown great successes in metaphor detection task as well. Different structures of neural networks have been extensively studied to better encode semantic knowledge by capturing metaphorical patterns (Leong et al., 2018).

Sequential models such as Recurrent Neural Networks (RNN) demonstrate strong performance in metaphor detection (Bizzoni and Ghanimifard, 2018). An LSTM is applied to identifying metaphors together with a Convolutional Neural Network (CNN) (Wu et al., 2018). The model utilizes pretrained word2vec word embeddings. Another approach built on sequential models is a Bidirectional LSTM model augmented with pretrained contextualized embeddings (Gao et al., 2018).

A recent work draws inspiration from linguistic theories, and proposes RNN_HG and RNN_MHCA, which are variants of RNN models (Mao et al., 2019). Assuming that pretrained GloVe embeddings carry literal meaning, the model RNN_HG detects metaphors by comparing GloVe embeddings with the contextualized word representations learned by the RNN. The other variant, RNN_MHCA, integrates the multi-head attention mechanism in model hidden states, and enriches word representations with information from broader contexts.

Contextualized representation model. The linguistic theory of Selectional Preference Violation (SPV) states that a metaphorical phrase or word is semantically different from its context (Wilks, 1975, 1978). This suggests the importance of contextual information for metaphor detection. A few pretrained contextualized representation models have been recently proposed and shown to achieve better performance than commonly used sequential models in a variety of language understanding tasks. A few models that encode contextual knowledge include ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). The advantages of these models are that they are trained on a large amount of text to encode rich semantic and contextual information into the representations giving them greater representation power than the models which are only trained on small task-specific datasets. In this work, we build our system upon RoBERTa to take advantage of its contextual representation for metaphor detection.

3 Metaphor Detection

In this section, we will introduce our model design, training and prediction for metaphor detection.

[1] https://github.com/HongyuGong/MetaphorDetectionSharedTask.git

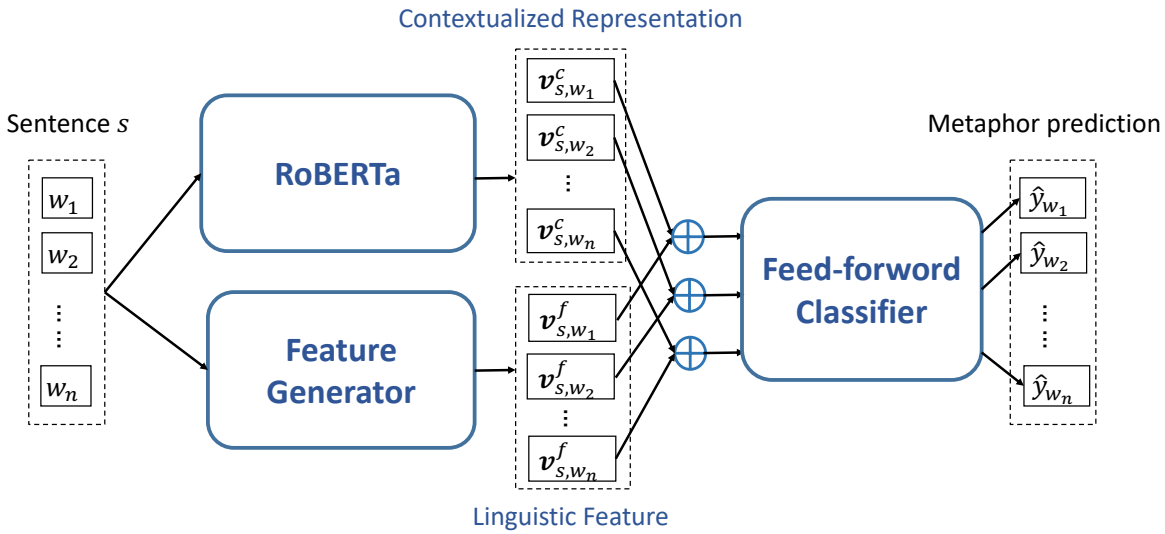

Figure 1: Model structure of IlliniMet system for metaphor detection. Its RoBERTa module learns contextualized representations, and the feature generator generates linguistic features for words. The outputs of RoBERTa and feature generator are combined as the word representations, which are sent to feed-forward classifier for metaphor classification.

3.1 Model

We cast the problem of metaphor detection as a sequence labeling task, where each word in the input sentence is classified as either metaphor or non-metaphor. As shown in Fig. 1, the framework of our IlliniMet model consists of three modules: RoBERTa, a feature generator and a feed-forward classifier. We will discuss each module in detail in the following parts.

RoBERTa. The meaning of words can vary subtly from one context to another, and RoBERTa generates contextualized word representations to capture the context-sensitive semantics of words (Liu et al., 2019). The use of word representations from RoBERTa has resulted in state-of-the-art performance in a variety of language understanding tasks. Given a sentence s consisting of n words $\{w_1, \ldots, w_n\}$, RoBERTa model generates their contextualized representations $\{\mathbf{v}^c_{s,w_1}, \ldots, \mathbf{v}^c_{s,w_n}\}$.

Linguistic features. The second module in our model is a feature generator. Previous works on metaphor detection have shown that linguistic features are useful for detecting metaphors. The features considered in our work are:

- Part-of-speech (POS) feature. We use the part-of-speech tags of the input words as the POS feature (Klebanov et al., 2014). Instead of using a one-hot vector as the POS feature, we create an embedding lookup table for POS tags, where each POS tag is mapped to a vector. All POS vectors are randomly initialized and are tuned during model training. We can obtain the representation of a given tag from the table during model training and prediction.

- Topic feature. The topic feature (Klebanov et al., 2014) is a distribution of a word over 100 topics extracted using Latent Dirichlet Allocation (LDA) (Blei et al., 2003).

- Word concreteness. Word concreteness is obtained from the database of (Brysbaert et al., 2014). Following (Klebanov et al., 2015), we binned words' concreteness rating ranging from 1 to 5 with upward and downward thresholds respectively. A word would be assigned to a bin with the upward threshold a if its concreteness is at least a. Similarly, it is assigned to a bin with the downward threshold b if its concreteness is at most b. Each word is represented with a binary vector indicating the bins in which its rating falls.

- WordNet feature. WordNet, a commonly used linguistic resource, provides the semantic classes of words such as verbs of communication and consumption. We use binary vectors corresponding to these classes as the WordNet feature for words (Klebanov et al., 2016).

- VerbNet feature. The VerbNet database classifies verbs based on their syntactic and seman-

tic patterns such as their frames, predicates, thematic roles and thematic role fillers. We again use binary feature vectors to represent the classes of verbs as in (Klebanov et al., 2016).

- Corpus-based feature. Verbs are clustered into 150 semantic clusters, and are assigned with corresponding one-hot vectors as the corpus-based features (Klebanov et al., 2016).

We note that some words may not have certain features; for instance, nouns do not have VerbNet features. We assign feature vectors of all zeros to those words without corresponding features.

The Feed-forward classifier. Lastly we have a feed-forward network based classifier as the inference module. The model derives word representations from the concatenation of contextualized representations and linguistic features. The concatenated representations are fed to the classifier and used to predict word labels (either metaphor or non-metaphor). We use a one-layer fully-connected feed-forward neural network for classification. For word w denote its RoBERTa embedding by $\mathbf{v}^c_{s,w}$ in sentence s and linguistic feature $\mathbf{v}^f_{s,w}$. The inference module predicts $\hat{\mathbf{y}}_w$, the probability over the two classes (both metaphor and non-metaphor).

$$\hat{\mathbf{y}}_w = \text{softmax}(\mathbf{W}(\mathbf{v}^c_{s,w} \oplus \mathbf{v}^f_{s,w}) + \mathbf{b}), \quad (1)$$

where $\hat{\mathbf{y}}_w$ is a two-dimensional vector, and $\oplus$ is the concatenation operator. The weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$ are both trainable model parameters.

As will be described in Section 4, the datasets we have are imbalanced with many more non-metaphorical words than metaphorical ones. Therefore, we used a weighted cross-entropy loss, and assign less weight to the more frequent label. We denote y_w as the true label of word w, and $\hat{y}_w$ as its predicted probability. Given a set of training sentences S, the training loss L is formulated as follows:

$$L = -\sum_{s \in S} \sum_{w \in s} \alpha_{y_w} \log \hat{y}_w, \quad (2)$$

where α_{y_w} is the weight coefficient of label y_w, and it is the total number of labels divided by the count of the label α_{y_w}.

3.2 Training and Prediction

Training. We divided the training data into train and dev sets with a split ratio of $4:1$, and trained the model to minimize the loss. The model which achieved the best F1 score on the dev set was used for testing. We used the pretrained RoBERTa model with 24 hidden layers in our system. Other model parameters were randomly initialized.

The system was trained end-to-end, and all model parameters including RoBERTa's parameters were tuned during model training. We set the dropout probability as 0.1, and the training epochs as 4 for all datasets. The learning rate was set as $2e-5$. We availed the warmup schedule by linearly increasing the learning rate from 0 to $2e-5$ within the first training epoch. The warmup schedule is a useful technique for tuning a pretrained model, while prevents the large deviation of the model from its pretrained parameters when it is tuned on a new dataset (Devlin et al., 2019).

Prediction. We use an ensemble method for the model prediction. Three models were trained independently with different train/dev splits, and we collect their predictions on the test data. Ensemble methods have been proposed to reduce the variance in predictions made by machine learning and deep learning models (Dietterich, 2000). In our experiments, we decide the word label by the majority vote of the predictions from these three models.

4 Experiment

We empirically evaluate our model for metaphor detection in this section. Precision, Recall and F1 score are the evaluation metrics used in our experiments. We report metaphor detection results on words of any POS tags as well as verbs alone in the test set provided by the shared task [2].

Dataset. Two datasets were used for the evaluation of metaphor detection – the VU Amsterdam Metaphor Corpus (VUA) (Steen, 2010) and the TOEFL dataset (Klebanov et al., 2018). The sentences in these two datasets were manually annotated for the task of metaphor detection at the word level.

The VUA dataset was collected from the BNC in four genres including news, academic, fiction and conversation. It provides $12,122$ sentences for training, and $4,080$ sentences for testing. Around

[2] https://competitions.codalab.org/competitions/22188

Table 2: The Test Performance of Different Models on Metaphor Detection

Dataset	VUA						TOEFL					
Type	ALLPOS			VERB			ALLPOS			VERB		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RoBERTa w. feature (ensemble)	74.6	71.5	73.0	76.7	77.2	77.0	72.6	67.5	70.0	73.1	70.7	**71.9**
RoBERTa w. feature (single)	75.0	69.8	72.3	76.5	75.5	76.0	74.2	63.8	68.6	72.0	69.4	70.7
RoBERTa (ensemble)	74.4	70.3	72.3	76.1	78.1	**77.1**	70.9	69.7	**70.3**	70.8	72.7	71.8
RoBERTa (single)	75.6	68.6	72.0	77.4	75.2	76.3	70.6	67.5	69.0	68.2	70.4	69.3
CNN-LSTM	60.8	70.0	65.1	60.0	76.3	67.2	-	-	-	-	-	-
ELMo-LSTM	71.6	73.6	72.6	68.2	71.3	69.7	-	-	-	-	-	-
RNN_HG	71.8	76.3	**74.0**	69.3	72.3	70.8	-	-	-	-	-	-

11% of VUA tokens are metaphors in the training data.

The TOEFL dataset consists of essays written by non-native speakers of English. It contains 180 train essays with 2,741 sentences and 60 test essays with 968 sentences. In the training partition, 7% of TOEFL tokens are metaphors.

Models. We report and compare the performance of the variants of our model. The methods we discuss in this work include:

1. RoBERTa w. feature (ensemble): our full model with both RoBERTa embeddings and linguistic features. The ensemble method is used to make predictions based on the votes of three separately trained models.

2. RoBERTa w. feature (single): a single full model trained to classify metaphors.

3. RoBERTa (ensemble): the model built on only RoBERTa and the classification layer without using linguistic features. Again, an ensemble method is applied to model predictions.

4. RoBERTa (single): the model has only RoBERTa and the classifier. A single model is trained to make predictions on test data.

Baselines. We also include three strong baselines for an empirical comparison in the task of metaphor detection.

- CNN-LSTM (Wu et al., 2018). This baseline combines CNN and LSTM layers to learn contextualized word embedding, and also includes additional features such as POS and word clusters. It has achieved the best performance on the VUA dataset in the 2018 VUA Metaphor Detection Shared Task (Leong et al., 2018).

- ELMo-LSTM (Gao et al., 2018). Built upon LSTM, this baseline makes use of pretrained contextualized embeddings from ELMo model (Peters et al., 2018).

- RNN_HG (Mao et al., 2019). This is the most recent model on metaphor detection reporting the state-of-the-art results on VUA dataset. It includes a bidirectional LSTM and makes use of GloVe and ELMo embeddings.

Results. Table 2 reports the results of our model variants as well as those of the three baselines discussed above. All variants of our model achieve better performance than the baselines, CNN-LSTM and ELMo-LSTM on both VUA ALLPOS and VUA VERB. The ensemble model of *RoBERTa with feature* falls behind the baseline RNN_HG by 1% in F1 score on VUA ALLPOS, while outperforming it by a large margin of 6.2%.

Ablation analysis. Ablation analysis was performed to compare the performance of our model variants. We can evaluate the effect of linguistic features by comparing our model with and without external features. When the ensemble method is used, the incorporation of external features does not influence the detection of metaphor verbs performance too much on both VUA and TOEFL data. When it comes to metaphor detection of all words, external features improve the F1 score by 0.7%

on VUA data, but degrades the performance on TOEFL data by 0.3%.

The gain in VUA ALLPOS is brought by the linguistic information of external features, which is not captured by the contextualized embeddings. The performance drop in TOEFL ALLPOS might have resulted from a larger search space of model parameters when the external features were added. Considering that the training data of TOEFL is much smaller in size compared tto that of VUA, the TOEFL model is likely to result in sub-optimal parameters after training when more parameters are introduced to the classification layer by the linguistic features.

We also evaluated the effectiveness of the ensemble method by comparing its performance with the performance of the single model. The ensemble model outperforms the single model regardless of whether external features were used. With external features, the ensemble model achieves gains of 0.7% and 1.4% in VUA and TOEFL dataset respectively when evaluated on all words.

Other model designs. We have only reported our best-performing models until now. In order to provide some empirical insights for those interested, we also discuss some designs which we had tried even though those did not yield performance gains. Besides RoBERTa embedding and linguistic features, we tried other features to expand the word representation. Borrowing ideas from (Devlin et al., 2018), we included the concatenation and the average of hidden state vectors of RoBERTa's last four layers. Another idea, which was inspired from (Mao et al., 2019), was to include the context-independent embedding from the bottom layer of RoBERTa. The intuition was that the model could identify metaphors more easily by comparing words' context-independent and context-sensitive embeddings.

Besides expanding word representation, we also experimented with different classification modules. We increased the layers of the feed-forward network, and also tried different activation functions in the feed-forward layer. We did not observe significant performance gains with these modifications to word representations and the inference module.

5 Discussion

In this section, we perform an analysis to explore the strengths and weaknesses of our model. Since we did not have ground truth labels for the test instances, we divided the training data into train, dev and test sets with a ratio of $4:1:1$ for the purpose of error analysis. We trained and tuned our best performing model on the resulting train and dev sets respectively, and evaluated it on the test set.

Model performance on different POS tags. We evaluate the model performance on words of different part-of-speech tags. On the VUA dataset, determiners ("DT"), prepositions and conjunctions ("IN") had the highest F1 scores, while adjectives ("JJ") and plural nouns ("NNS") got the lowest F1 scores among all words. On the TOEFL dataset, the best-performing words were adjectives ("JJ") and verbs ("VB"). Words ranking at the bottom were prepositions and conjunctions ("IN") and nouns ("NN").

The nouns on both datasets received low F1 scores. One explanation is that nouns have a large vocabulary and often have multiple senses. As will be discussed in the error analysis below, the model may not make correct predictions if a noun and its different senses are not included in the training data. Interestingly, prepositions and conjunctions in the VUA dataset got good F1 scores while those in the TOEFL dataset got low scores. Since TOEFL data was collected from written texts of non-native English learners, we conjecture that there is more variety in the usage of prepositions in TOEFL dataset. The corresponding noise may have made it harder for the model to generalize from the training set to the test set.

Error analysis. We look through the examples where our model made wrong predictions, and summarize the patterns of these error examples below.

- Words that are unseen in the training set. Some metaphorical words in the test set do not occur in the training set. Examples are "whip-aerials", "puritans", "half-imagined" and "pandora box". Our model incorrectly classifies these words as non-metaphorical.

- Words that have senses unseen in the training set. We note that some words occur in both training and test sets, but they are used with different senses. For example, the word "wounded" is only used metaphorically in the training set. Our model incorrectly predicts its literal usage as metaphorical usage in the test data.

- Words whose test labels have inter-annotator

disagreement. Words "pack" and "bags" are labeled as metaphorical in the sentence " his job was to convince amaldi to pack his bags because there was a ship waiting at naples to take him to the united states". However, we think these words carry their literal senses in the given context.

- Inaccurate interpretation of contexts. In the long sentence "the 63-year-old head of pembridge investments , through which the bid is being mounted says ...", word "says" with the subject "the head" is not a metaphor. Our model may not capture its subject correctly given the long-distance dependency, which results in a false positive prediction.

6 Conclusion

In this paper, we introduced the IlliniMet system for the task of word-level metaphor detection. Our model leveraged contextual and linguistic information by combining contextualized representation and external linguistic features. We adopted an ensemble approach to reduce the variance of predictions and improve the model performance. The empirical results showed the effectiveness of our model. We also performed an error analysis to gain insights into the model behavior.

Acknowledgments

This work is supported by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network. We would like to thank the anonymous reviewers for their constructive comments and suggestions.

References

Yuri Bizzoni and Mehdi Ghanimifard. 2018. Bigrams and bilstms two neural networks for sequential metaphor detection. In *Proceedings of the Workshop on Figurative Language Processing*, pages 91–101.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.

Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. Modelling metaphor with attribute-based semantics. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 523–528.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Ellen K 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.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Conference on Empirical Methods in Natural Language Processing*.

Hongyu Gong, Suma Bhat, and Pramod Viswanath. 2017. Geometry of compositionality. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Dirk Hovy, Shashank Srivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, and Eduard Hovy. 2013. Identifying metaphorical word use with tree kernels. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57.

Beata Beigman Klebanov, Ben Leong, Michael Heilman, and Michael Flor. 2014. Different texts, same metaphors: Unigrams and beyond. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 11–17.

Beata Beigman Klebanov, Chee Wee Leong, and Michael Flor. 2015. Supervised word-level metaphor detection: Experiments with concreteness and reweighting of examples. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 11–20.

Beata Beigman Klebanov, Chee Wee 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.

Beata Beigman Klebanov, Chee Wee Leong, E Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–106.

Arlene Koglin et al. 2015. An empirical investigation of cognitive effort required to post-edit machine translated metaphors compared to the translation of metaphors. *Translation & Interpreting, The*, 7(1):126.

Maximilian Köper and Sabine Schulte im Walde. 2017. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30.

Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 vua and toefl metaphor detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, Seattle, WA.

Chee Wee 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

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.

Andrew Ortony and Ortony Andrew. 1993. *Metaphor and thought*. Cambridge University Press.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Vassiliki Rentoumi, George A Vouros, Vangelis Karkaletsis, and Amalia Moser. 2012. Investigating metaphorical language in sentiment analysis: A sense-to-sentiment perspective. *ACM Transactions on Speech and Language Processing (TSLP)*, 9(3):1–31.

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

Kevin Stowe, Sarah Moeller, Laura Michaelis, and Martha Palmer. 2019. Linguistic analysis improves neural metaphor detection. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 362–371.

Tomek Strzalkowski, George Aaron Broadwell, Sarah Taylor, Laurie Feldman, Samira Shaikh, Ting Liu, Boris Yamrom, Kit Cho, Umit Boz, Ignacio Cases, et al. 2013. Robust extraction of metaphor from novel data. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 67–76.

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.

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.