Metaphorically Speaking
Detecting Metaphors in Natural Language

Christie Ibaraki, Cole Frank, Deniz Tokmakoğlu
{ibarakicj, cvf, denizt}@uchicago.edu

Abstract

We examine two models for metaphor detection: BiLSTM and RoBERTa. We attempt to replicate and adapt the model architectures implemented in previous works and report our findings. We present error analysis on the VUA dataset to illustrate the strengths and weaknesses of both models. Our results suggest that RoBERTa outperforms the BiLSTM in detecting the metaphoricity of non verb tokens.

1 Introduction

As humans, we rely on metaphors to express ourselves; metaphors allow us to convey meaning that transcends the literal, dictionary definitions of words. Metaphors are deeply embedded in language, yet they can easily go unnoticed. Given the preponderance of metaphorical language, accurate detection of metaphors is essential in language processing applications like machine translation and sentiment analysis.

In this paper we examine and compare existing work on automated metaphor detection. We focus on replicating the bi-directional LSTM proposed by Gao et al. (2018). We reproduce their model results on MOH-X, TroFi, and VUA datasets for both task formulations, and discuss any notable differences in our findings.

We also consider RoBERTa for metaphor detection and adapt the model of Gong et al. (2020). We perform classification on MOH-X and TroFi datasets, and compare these results to the BiLSTM. Finally, we use RoBERTa to sequence the VUA dataset and present error analysis of RoBERTa and BiLSTM sequence labeling results. Our code is publicly available at github.com/cmsc-25700/metaphor-detection. [1]

Task	Example
SEQ	The day thrift turned into a nightmare
CLS	This poses an interesting question.

Table 1: For sequencing (also referred to as tagging), every word in an input sentence is labeled as metaphorical or literal. For classification, only a single word per sentence is labeled.

2 Detection Tasks

We consider the two standard task formulations for metaphor detection: sequencing and classification.

1. **Sequence labeling**

   Classify metaphoricity of every word in a given sentence.

   Given a sentence $x_1, \ldots, x_n$ predict a sequence of binary labels $l_1, \ldots, l_n$.

2. **Classification task**

   Given a word and a sentence, determine whether the use of the word is metaphorical or literal.

   Given a sentence $x_1, \ldots, x_n$ and a target verb index i, predict a binary label l to indicate the metaphoricity of the target x_i.

3 Models

3.1 Bi-directional LSTM

Gao et al. (2018) proposed BiLSTM to encode complete sentences. Prior to this work, most models used restricted forms of context (for example, focusing only on subject-verb-object triples or unigram features). In order to capture richer contextual word representations, Gao et al. used a BiLSTM to encode a sentence with a feedforward neural network for classification. Their model is described below:

[1] github.com/cmsc-25700/metaphor-detection, github.com/cmsc-25700/gong-metaphor-detection

Sentence Encoding

For both tasks, each token x_i in input sentence is represented by pre-trained word embedding w_i. We have also concatenated ELMo (Peters et al. (2018)) vectors e_i to encode contextual information, as described in the Gao paper. The authors posit that these embeddings are useful for word sense disambiguation.

Sequence Labeling Task: Model

Each word is represented as $[w_i;e_i]$.

For each word x_i input word representation $[w_i;e_i]$ to BiLSTM produces h_i, a contextualized representation for each token.

Feedforward neural network takes h_i to predict label l_i for each word x_i.

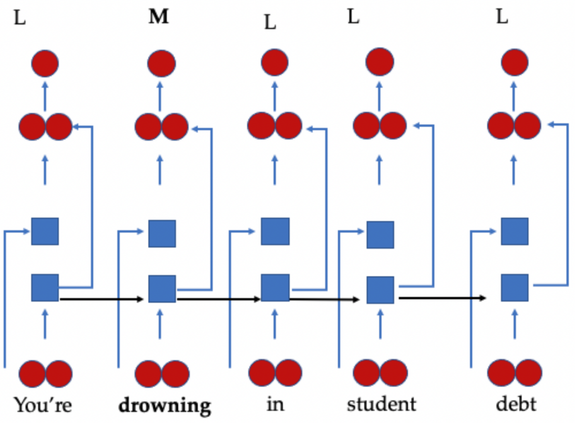

Figure 1: BiLSTM SEQ Model Architecture (adapted from Gao et al. (2018))

Classification Task: Model

For each word, concatenate an index embedding n_i, which indicates whether x_i is the target verb.

Then a word is represented as $[w_i;e_i;n_i]$.

For each word x_i input word representation $[w_i;e_i;n_i]$ to BiLSTM produces h_i, a contextualized representation for each token.

An attention layer is added, computing attention weight a_i for token x_i, and taking c, a weighted sum of LSTM output states where W_a and b_a are learned parameters.

$$a_i = \text{SoftMax}_i(W_a h_i + b_a)$$

$$c = \sum_{i=1}^{n} a_i h_i$$

The feedforward neural network takes input c and outputs label scores for each target verb.

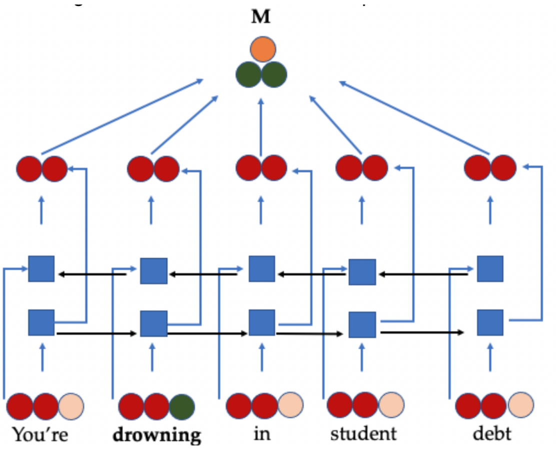

Figure 2: BiLSTM CLS Model Architecture (figure adapted from Gao et al. (2018))

3.2 RoBERTa

Gao et al.'s improvement in performance by adding more contextual information for metaphor detection motivated further research on the incorporation of other types of contextual representation. These other types of contextual representation include using different pre-trained word embeddings, Representations from Transformer (BERT) (Devlin et al. (2019)) and Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al. (2019)).

We focus specifically on the work of Gong et al. (2020) who combine RoBERTa contextual representations with a features-based approach in their IlliniMet model architecture. We selected this model for adaptation, because it outperformed Gao et al. on the VUA sequence labeling task. While Gong et al. only implement their model for sequence labeling, we implement the model (with modifications) on both the sequence labeling and classification tasks. For the sake of simplicity, we omit the majority of the linguistic features included in the original IlliniMet model. Our adaptation of their model has three components:

1. **RoBERTa model**

   In order to capture context, RoBERTa produces contextualized word representations for each token based on the entire sentence. BERT uses a word-piece tokenizer, however our raw data was tokenized at the word level.

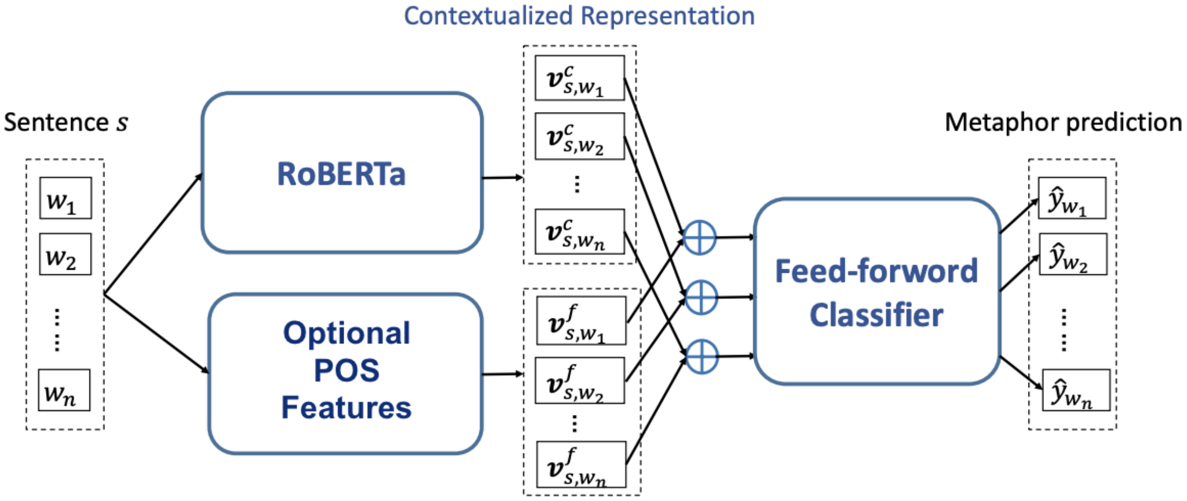

Figure 3: RoBERTa Model Architecture of IlliniMet (figure adapted from Gong et al. (2020))

So, in order to use BERT we first transformed our raw tokens into BERT tokens by passing each raw token x_i into the BERT tokenizer independently. For each sub-word z_i that one of our raw tokens was broken into we attached the metaphoricity label and POS information of the raw token to the first subword and padded to ensure the number of BERT tokens matched the number of labels and POS specifications for each sentence. Then for each sub-word token z_i in the input sentence s the RoBERTa model generates its contextualized representation v^c_{s,z_i}.

2. **(Optional) POS Feature Generator**

   Previous research on metaphor detection suggests that linguistic features can improve model performance. We create an embedding table with one vector for each unique part of speech (POS) that appears in the training data.[2] For each token z_i in an input sentence s the corresponding POS embedding v^f_{s,z_i} is retrieved from the POS embedding table and concatenated to the contextualized representation v^c_{s,z_i}. All of the POS embeddings are randomly initialized to begin and learned during the model training.

3. **Feed-forward Classifier**

   The final component of the model is a one layer fully-connected feed-forward neural network. For each token z_i in input sentence s, this classifier layer takes the concatenation of the RoBERTa contextualized representation v^c_{s,z_i} and (optionally) the POS embedding v^f_{s,z_i} as input. The classifier outputs a 2-dimensional probability vector $\hat{y}_{z_i}$ over the two possible classes (metaphor and non-metaphor):

$$\hat{y}_{z_i} = \text{SoftMax}(W(v^c_{s,z_i} \oplus v^f_{s,z_i}) + b)$$

   Where $\oplus$ is the concatenation operator and the weight matrix W and bias vector are learned during model training. Due to imbalanced classes (in all three datasets there are more non-metaphorical words than metaphorical), the model is trained using weighted cross-entropy loss with weights derived from the prevalence of each class in the training data.

4 Data

For both the BiLSTM and RoBERTa experiments, we use the benchmark datasets presented in Gao et al., 2018. There are two datasets for classifcation (TroFi and MOH-X, Table 2) and one for sequencing (VUA, Table 3). Data are briefly described below.

1. **MOH, MOH-X** (Classification)
   Shorter, simpler sentences (compared to TroFI and VUA, which come from news articles). Higher percentages of metaphor. MOH-X is a subset of MOH where the verb

[2] The VUA dataset includes POS tags for each token. We generate our own POS tags using spaCy for the MOH-X and TroFi datasets.

	n	Perc Metaphor	Uniq Verb	Avg Sentence Len
MOH-X	647	0.49	214	8
MOH	1,603	0.25	436	7.43
TroFi-X	1,444	0.42	857	29.35
TroFi	3,737	0.44	50	28.35
VUA	23,113	0.28	2298	26.11

Table 2: Summary statistics for MOH-X and TroFi data. Data were provided by Ge Gao. We did not find significant differences between our summary statistics and those presented by Gao et al.

	Uniq Tokens	n Tokens	Unique Sentences	Perc Metaphor
TRAIN	13,843	116,622	6,323	0.11
DEV	7,458	38,628	1,550	0.12
TEST	7,200	50,175	2,694	0.12

Table 3: Summary statistics for VUA data. We did not find significant differences between our summary statistics and those presented by Gao et al.

and its argument are extracted from each sentence.

2. **TroFi** (Classification)
   Two times larger than the MOH dataset, but contains only 50 unique verbs.

3. **VUA**
   Contains over 2K unique verbs. Contains annotations for all words in each sentence. Label classes are less balanced (11% metaphors at token level).

5 Results

5.1 Gao et al. Replication

A full comparison of Gao's original BiLSTM results and our replication of those results is in Appendix A. After running the publicly available Gao et al. code [3], we were able to obtain similar performance metrics on all tasks, with the exception of VUA verb classification (see Table 8, rows 1 and 2). We found that for the VUA data, the classification model performs just as strongly as the sequence model, which is interesting since Gao et al. report that the sequence model significantly outperforms the classification model. They use this finding to support the claim that metaphor labels of context words improves the prediction on the target verb, but we were unable to reproduce this result. Additionally we found that although classification model performance on VUA varied significantly across runs, F1 was generally much higher than 53.4 (see Figure 6).

5.2 BiLSTM - BERT Comparison

Our main results are presented in tables 4, 5, 6. Table 4 compares Gao's BiLSTM and our RoBERTa (plus POS features) model's results on the VUA sequence labeling task. RoBERTa outperforms the BiLSTM on all evaluation metrics. Table 5 presents the same results broken down by the part of speech of the target word. Here we see the same pattern in performance and POS for both models: adposition is the easiest to classify (and the most frequently metaphorical in training data), while particles are the hardest to classify.

Table 6 contains the results of two BiLSTM model architectures (CLS and SEQ) and two RoBERTa model architectures (with and without POS features) on the classification task across all three of our data-sets. Here we do not see an improvement in performance for the RoBERTa model over BiLSTM. For the MOH-X data, RoBERTa and BiLSTM have similar performance metrics, while BiLSTM outperforms RoBERTa on TroFi.

	P	R	F1	Acc
Gao et al.	71.6	73.6	72.6	93.1
RoBERTa + POS	79.8	78.1	78.9	94.8

Table 4: Comparison of performance on VUA sequencing task for Gao et al. BiLSTM and RoBERTa. (Gao et al. numbers come from Table 4 in the original paper.)

[3] Original repo: https://github.com/gao-g/metaphor-in-context

	Gao et al.			RoBERTa		
POS	P	R	F1	P	R	F1
VERB	68.1	71.9	69.9	76.7	75.2	76.0
NOUN	59.9	60.8	60.4	77.4	66.0	71.2
ADP	86.8	89.0	87.9	90.3	90.7	90.5
ADJ	56.1	60.6	58.3	68.4	62.1	65.1
PART	57.1	59.1	58.1	65.8	67.1	66.4

Table 5: Comparison of performance on VUA sequencing task by POS for Gao et al. BiLSTM and RoBERTa. (Gao et al. numbers come from Table 5 in the original paper.)

	MOH-X				TroFi				VUA				
Model	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	MaF1
Gao et al. CLS	75.3	84.3	79.1	78.5	68.7	74.6	72.0	73.7	53.4	65.6	58.9	69.1	53.4
Gao et al. SEQ	79.1	73.5	75.6	77.2	70.7	71.6	71.1	74.6	68.2	71.3	69.7	81.4	66.4
RoBERTa	65.7	86.6	74.7	70.4	56.2	90.5	69.3	64.9	—	—	—	—	—
RoBERTa + POS	70.4	92.0	79.8	74.7	56.6	85.9	68.2	64.9	—	—	—	—	—

Table 6: Comparison of performance on classification task for Gao et al. BiLSTM and RoBERTa. (Gao et al. numbers come from Table 6 in the original paper.)

6 Error Analysis

Error analysis of the sequence labeling results for the BiLSTM and RoBERTa models showed that the RoBERTa model tended to do a better job of catching non-verb metaphorical terms. For example consider the sentence in figure 4. The RoBERTa model successfully flags the metaphorical parts of the phrase "throws a spanner in the Whitehall machinery," whereas the BiLSTM model does not catch that the word "machinery" is being used metaphorically.

"The Labour Party Conference: Policy review throws a spanner in the Whitehall machinery"
BERT: throws, spanner, in, machinery
Bi-LSTM: throws, spanner, in

Figure 4: Sample sentence with gold labels and RoBERTa and BiLSTM sequence labeling results. Green words are correctly labeled metaphors and red-words are incorrectly labeled metaphors

Reviewing the errors also highlighted that metaphor tagging is more of an art than a science. Metaphor labeling is subject to the same forms of ambiguity that other forms of linguistic feature tagging like POS-tagging are subject to. Often the line between what is and isn't a metaphor is not obvious. Some words' metaphorical or figurative usage is much more common that their literal usage. Consider the example sentence from the VUA dataset in figure 5. The expert annotator's decision to label the words "after" and "opening" as metaphors is questionable, particularly given the fact that they did not label "closing" as a metaphor. In fact, both models "incorrectly" label "closing" as a metaphor! Our metaphor detection models are only as good as the data they are trained and tested on.

"The FTSE failed to hold its gains yesterday but shortly after opening it was only six points away from last September's previous record closing level."
BERT: failed, hold, gains, after, opening, points, away, from, closing, level
Bi-LSTM: failed, hold, gains, after, only, away, from, closing,

Figure 5: Sample sentence with gold labels and RoBERTa and BiLSTM sequence labeling results. Green words are correctly labeled metaphors and red-words are incorrectly labeled metaphors

7 Discussion

The performance of our models varies with different types of data and classification tasks. Therefore our results do not provide conclusive evidence as to whether BiLSTM or RoBERTa is better suited for metaphor detection.

For datasets that are not labeled on a token level (MOH-X and TroFi), BiLSTM consistently outperforms RoBERTa without POS. For MOH-X, simply including POS features improves BERT's performance significantly (5 point increase in F1). The comparable performance of the two models on the classification task is surprising given that BERT's architecture is generally regarded as superior to BiLSTM. One advantage of of using a pretrained model like RoBERTa is that a huge amount of contextual information is learned by the model during the pre-training phase. It's worth noting that by using ELMo vectors as the inputs to our BiLSTM model we were giving additional information that the BiLSTM architecture alone would not have learned. Conceivably this use of ELMo embeddings neutralized any advantage gained from using pretrained RoBERTa.

For the sequence labeling task with POS features, the RoBERTa model performs significantly better than the BiLSTM model on all POS.

8 Conclusion

In this study we experimented with different models, data-sets, and tasks for metaphor detection. We found that a RoBERTa model augmented with POS features significantly outperforms a BiLSTM model on the metaphor sequence labeling task. We did not find conclusive evidence that either model was consistently better on the sentence (verb) classification task.

The apparent ambiguity in what is and isn't a metaphor discussed in section 6 (Error Analysis) suggests that metaphor detection might not be as well-specified a task as we would like it to be. Perhaps further research could focus on specific types of metaphors that are easier for a human annotator to classify.

Acknowledgements

We would like to thank Chenhao and the TAs, especially Mourad for his willingness to advise and help our group on a Saturday. We would also like to thank Ge Gao for providing the raw data and ELMo vectors.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *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).*

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

Hongyu Gong, Kshitij Gupta, Akriti Jain, and Suma Bhat. 2020. IlliniMet: Illinois system for metaphor detection with contextual and linguistic information. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 146–153, Online. Association for Computational Linguistics.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke. Zettlemeyer. 2018. Deep contextualized word representations. *HLT-NAACL.*

A Appendix: Replication Results

	Gao et al.			Our Replication		
POS	P	R	F1	P	R	F1
VERB	68.1	71.9	69.9	69.7	67.7	68.7
NOUN	59.9	60.8	60.4	69.5	55.8	61.9
ADP	86.8	89.0	87.9	88.3	88.6	88.5
ADJ	56.1	60.6	58.3	67.1	54.3	60.1
PART	57.1	59.1	58.1	57.6	53.7	55.6

Table 7: Comparison of Gao et al.'s BiLSTM model vs. our replication of their model on the VUA sequence labeling task (broken down by POS of the word being labeled, Gao et al. numbers come from Table 5 in the original paper.)

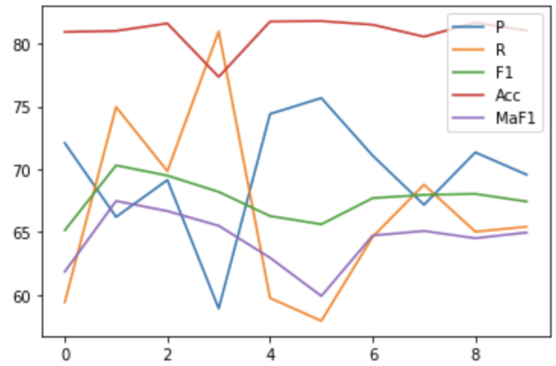

Figure 6: CLS Model performance metrics on VUA test set over multiple (unrelated) runs, using hyperparameters set by Gao et al. (20 epochs, learning rate=0.01).

	MOH-X				**TroFi**				**VUA**				
Model	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	MaF1
Gao et al. CLS	75.3	84.3	79.1	78.5	68.7	74.6	72.0	73.7	53.4	65.6	58.9	69.1	53.4
Replication CLS	75.3	75.4	74.8	75.2	72.4	66.7	69.4	72.1	69.6	66.7	67.6	80.9	64.4
Gao et al. SEQ	79.1	73.5	75.6	77.2	70.7	71.6	71.1	74.6	68.2	71.3	69.7	81.4	66.4
Replication SEQ	74.8	77.2	75.5	75.9	70.4	71.5	70.9	74.4	69.9	66.7	68.3	81.4	65.4

Table 8: Replication of classification task for Gao et al. BiLSTM (Gao et al. numbers come from Table 6 in the original paper.)