Metaphor Detection with Transformer-based Contextual Embeddings

Abstract

Metaphors represent means of conveying certain ideas more artistically and are not meant to be taken literally. Metaphors are highly reliant on symbolism, explaining a concept's meaning by using comparisons or unexpected associations. Because of their practicality, they can be found in a wide array of domains: poetry, articles, novels, all to convey the author's ideas in the most suitable ways. Being of such importance, we propose a solution that applies deep learning techniques for identifying the metaphors. We experiment with different word embeddings, alongside *Recurrent Neural Networks* (*RNNs*) and two different approaches for this certain task: classification (*softmax*) or sequence labeling (*Conditional Random Fields - CRFs*).

1 Introduction

A metaphor's purpose is to outline certain attributes of a concept by comparing it to a seemingly unrelated counterpart. Even though the two notions look like they have nothing in common, the metaphor identifies a series of hidden common aspects that can be highlighted to convey the author's idea. Therefore, it becomes increasingly more difficult to identify the metaphors present in texts by using traditional machine learning techniques. The Second Shared Task on Metaphor Detection, co-located with ACL 2020, intends to improve the state-of-the-art by challenging participants to discover new ways to improve the overall performance in identifying metaphors in different types of texts. The purpose of the competition is to identify word-level metaphors in texts, as well as verb-level. There are two different datasets for training and evaluating the solutions: a subset of the ETS Corpus of Non-Native Written English, annotated with word-level metaphors [1] and the VU Amsterdam Metaphor Corpus (VUA) dataset [2]. Therefore, we experimented with different deep learning solutions, implying different word embeddings, such as BERT, RoBERTa, SciBERT, XLNET, ALBERT, alongside with an RNN based layer, the Bidirectional Long Short-Term Memory (BiLSTM) and a softmax or a CRF layer. We managed to achieve a 0.5517 F1 score on the VUA testing dataset on the all-POS test data and a 0.5328 F1 score on the verb test data.

2 State-of-the-art

Wu et al. [3] proposed a solution that makes good use of deep learning solutions, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Their approach considers the word-level metaphor identification task as a sequence labeling one. In their architecture, they initially apply a series of preprocessing techniques, including verb **lemmatizing** by using the NLTK package, such that the word will be transformed into its base form. Furthermore, the authors added an **embedding layer** that acts as a lookup table for the input words. The word embeddings have the purpose to convert the input sequences into sequences of low-dimension dense vectors. The embeddings the authors opted to use are obtained from the pre-trained word2vec model. Furthermore, to extract local contextual information, the authors added a **CNN** layer. Considering that the task works with word sequences, it was ubiquitous for the authors to also use a **BiLSTM** to detect long-term dependencies. Finally, the **CRF layer** is used to predict the metaphor labels of each word. At the same time, for the final layer, the authors also

experimented with the **softmax function**, included in a dense layer for prediction.

Cirstea et al. [4] (Metaphor Detection) proposed a model based on two different types of features for two types of metaphors: text categorization-type features for metaphors formed with verbs and semantic features for IS-A and OF metaphors. Their first approach consists of **mapping text to a vector of term weights**, which represents a bag-of-words method. The output scores for each term can be computed by either using TF/IDF or as binary features, with labels of 0 and 1. Another approach is represented by **feature selection**, which implies the selection of relevant features and removal of the entries containing low levels of information. Examples of such techniques are represented by document frequency thresholding, chi statistics, term strength, information gain, and mutual information. A third approach amounts to the usage of **semantic features**, including the normalized Google distance, Normalized pointwise mutual information, Knowledge-based measures of similarity and Measures of concreteness.

Gao et al. [5] (Neural Metaphor Detection in Context) proposed a model that uses BiLSTMs in order to encode sentences. For word embeddings, the authors used ELMo (Embeddings from Language Models) vectors [6]. Their solution uses a BiLSTM for producing a contextualized representation for each token. Furthermore, for prediction, the **BiLSTM** is followed by a **feedforward neural network** with the purpose of receiving the representation and predicting a certain label for each word form the input sequence.

3 Neural Architectures

3.1 Contextual Embeddings

We experimented with different contextual embeddings for our architecture. Considering that the state-of-the-art in NLP tasks is represented by Transformer-based architectures, it becomes natural that we needed to experiment with Transformer-based contextual embeddings.

Firstly, **BERT** [7] represents an innovation in the NLP field. It uses the attention mechanism, Transformers, for a bidirectional training manner. Unlike other traditional solutions, BERT, with the benefits of attention, manages to catch long-term dependencies inside sequences, thus increasing the performance of the tasks the model is applied to. At the same time, BERT comes with two versions: base and large, the second one offering more performance at the cost of being more hardware demanding. We experimented with the base version by fine-tuning the model on the combined VUA and TOEFL metaphor datasets, such that the 768-dimensional vectorial representation that BERT base outputs will be adjusted accordingly to the context.

We also experimented with different learning rates and concluded that 5e-5 is recommended for fine-tuning the model with unfrozen weights and 5e-4 for frozen.

After we experimented with BERT, it became mandatory to also experiment with **RoBERTa** [8], considering that the model's purpose is to improve on BERT, by modifying key hyperparameters and removing the next-sentence pre-training objective. At the same time, RoBERTa uses larger learning rates, as well as mini-batches. Furthermore, RoBERTa was trained for a longer period compared to BERT, allowing it to obtain better task performances.

We also experimented with **SciBERT** [9], a BERT version extensively trained on scientific texts, such that we will be able to capture hidden meanings behind rare encounters of metaphors in scientific texts. As expected, the results obtained with SciBERT were not on part with other contextual embeddings, considering that metaphors are a rare occurrence in scientific texts.

Furthermore, **XLNet** [10], similar to BERT, also represented a good option for contextual embeddings. Unlike BERT, XLNet is an autoregressive language model, it uses the context of the current word in order to predict the next word, by either using the forward pr the backward direction. At the same time, BERT is an autoencoder language model, a fact that allows it to go through the input sequence in both directions at the same time. XLNet does not use the [MASK] symbol in pretraining. The differences from BERT continue, XLNet using the Permutation Language Modeling during the pre-training phase.

Moreover, **ALBERT** [11] further improves on BERT and thus offers better contextual embeddings by taking into account the impact of the number of parameters, by lowering it and implicitly allowing better memory consumption.

ALBERT's principle is based on a factorization process of embeddings in order to further improve the hardware usage, memory consumption. One key aspect that differentiates ALBERT from BERT is represented by the pretraining stage, where ALBERT uses a Sentence Order Prediction strategy such that the model will be able to understand the links, the coherence between sentences.

3.2 Conditional Random Fields (CRF)

Our task, identifying metaphors at a word level, can be compared to a Named Entity Recognition (NER) task. As Lample et al. [12] proposed, we can use Conditional Random Fields (CRFs) [13]. The purpose of CRFs is to model tagging decisions jointly, thus improving the performance compared to the situation when we simply treated each tag as a separate entity.

By passing the input sequence

$$X = (x_1, x_2, ..., x_n),$$

through the first component of the neural network, the BiLSTM, we obtain a $n \times k$ matrix of scores, P, where k is the number of tags, in our situation 2 (metaphor or non metaphor) and n is the number of elements in the input sequence. Thus, the element situated on the ith row and the jth column of P can be considered as the score of the jth tag of the ith word from the input sequence, the sentence.

Because we need to find a tag for each word of the input sequence, the model must output a sequence of tags of equal length with the input. Therefore, the output will have the form:

$$y = (y_1, y_2, ..., y_n).$$

Furthermore, Lample et al. defined A as the matrix of transition scores. In this situation, $A_{i,j}$ represents the score of the transition from tag i to tag j.

Therefore, the score is defined as:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i}, y_{i+1} + \sum_{i=1}^{n} P_{i,y_i}$$

Furthermore, it is used a softmax over all possible tag sequences, in order to obtain a probability for a certain sequence y.

$$p(y \mid X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}}.$$

The output sequence that yields the maximum score is obtained by applying:

$$y^* = \arg\max_{\tilde{y} \in Y_X} s(X, \tilde{y})$$

3.3 Softmax layer

On the other hand, we can add a dense layer using a softmax function to the output of the BiLSTM. Because the prediction is made at word-level, it becomes obvious that metaphorical words are in a much smaller number compared to the non-metaphorical counterpart. Therefore, we needed to add class weights in order to treat this imbalance, assigning a larger loss weight to the metaphor class.

4 Evaluation

4.1 Dataset and Preprocessing

There are two different training datasets used for this task, the VUA dataset and the TOEFL dataset. Both of them cover different subjects and use similar annotations for word-level metaphors, by placing an "M_" token in front of the word. Table 1 shows the class distribution among the two datasets, as well as the number of inputs for each.

	Sequence entries	Non-metaphors	Metaphors
VUA	10894	61567	11044
TOEFL	2741	56300	2140

Table 1: Class distribution

Furthermore, we applied some preprocessing techniques such that we removed irrelevant information from the dataset. Firstly, we removed the punctuation and numbers. Secondly, we applied a lowercase transformation to all the tokens, such that we will be able to use the uncased versions of the contextual embedding solutions. Finally, we applied a lemmatization to the tokens, such that every word (especially verbs) will be turned into their base form.

4.1 Experimental Setup

As we found out in the previous section, the class distribution is highly unbalanced, a fact that determines us to apply class weights for the softmax classification task. We do this by simply computing the percentage of positive (metaphor) and negative (non-metaphor) labels in the entire dataset and assigning a bigger loss weight to the positive label. Therefore, our model will avoid treating both classes in a similar manner and thus will grant more importance to the metaphor class, which is more rare.

Furthermore, another important aspect of our experiment is represented by the contextual embeddings. Regardless of the used solution, we fine-tuned the model on the train data with a small learning rate (5e-5) and the weights frozen and saved the model. After that, we unfroze the weights and loaded the previously saved model, this time increasing the learning rate to 5e-4. The results obtained by performing this sequence of steps were better than just fine-tuning the embedding model with just frozen/unfrozen weights.

4.2 Results

Table 2 contains the results for the AllPOS task on the VUA dataset, by using the CRF approach.

	Precision	Recall	F1
BERT+BiLSTM+CRF	0.6164	0.4359	0.5107
RoBERTa+BiLSTM+CRF	0.5420	0.5617	**0.5517**
SciBERT+BiLSTM+CRF	0.6570	0.2699	0.3826
XLNET+BiLSTM+CRF	0.5935	0.4254	0.4956
ALBERT+BiLSTM+CRF	0.5956	0.3869	0.4691

Table 2: AllPOS Results, VUA, CRF

At the same time, we obtained a 0.5328 F1 score for the verb identification task on the VUA dataset, by using the RoBERTa + BiLSTM + CRF solution.

As we can see, the best solution is obtained by using the RoBERTa embeddings, considering the fact that the model was trained on more data compared to BERT. The lowest scores come from SciBERT, partly because of the inclination of the model towards the scientific texts, that usually lack metaphors or artistic language.

	Precision	Recall	F1
BERT+BiLSTM+CRF	0.7185	0.2307	0.3493
RoBERTa+BiLSTM+CRF	0.3754	0.5113	**0.4329**

Table 3: AllPOS Results, VUA, softmax

Table 3 shows the results obtained by replacing the CRF layer with a softmax. We experimented with the best two results from the previous step, the CRF layer. As we can see, the best result for this situation is still yielded by the usage of the RoBERTa embeddings.

	Precision	Recall	F1
BERT+BiLSTM+CRF	0.6147	0.3225	0.4230
RoBERTa+BiLSTM+CRF	0.6146	0.4701	**0.5328**

Table 4: Verb Results, VUA, CRF

5 Conclusion

Finding metaphors proves to be a non-trivial task, considering the fact that they usually tend to express certain ideas in an artistic manner. Therefore, it becomes increasingly more difficult for deep learning solutions to identify them, especially when these metaphors are spread over a wide domain of interests. Different contextual embeddings tend to greatly influence the performance of the solution, considering the data they have been trained on and the overall quantity and quality of information that data conveys. As expected, models trained on large amounts of data, such as RoBERTa, offer the best results because they cover much more topics compared to a simpler model. At the same time, the dimension of output greatly influences the outcome: larger models generate more precise embeddings, fact that helps the solution properly separate different words in different contexts.

References

[1] Beata Beigman Klebanov, Chee Wee Leong, and Michael Flor. 2018. *A Corpus of Non-Native Written English Annotated for Metaphor*. NAACL

[2] Gerard Steen, Aletta Dorst, Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification*. Amsterdam: John Benjamins

[3] Chuhan Wu, Fangzhao Wu, Yubo Chen, Sixing Wu, Zhigang Yuan, Yongfeng Huan, *Neural Metaphor Detecting with CNN-LSTM Model*

[4] Bogdan-Ionut Cirstea, Costin-Gabriel Chiru, *Metaphor Detection*

[5] Ge Gao, Eunsol Choi, Luke Zettlemoyer, *Neural Metaphor Detection in Context*

[6] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, *Deep contextualized word representations*

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoye, Veselin Stoyanov, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*

[9] Iz Beltagy Kyle Lo Arman Cohan, *SCIBERT: A Pretrained Language Model for Scientific Text*

[10] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le, *XLNet: Generalized Autoregressive Pretraining for Language Understanding*

[11] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, *ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS*

[12] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, *Neural Architectures for Named Entity Recognition*

[13] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In Proc. ICML.