# Neural Metaphor Detection with GloVe Embeddings and Bidirectional-LSTMs

Nick Saulnier
Khoury College of Computer Science, MSc Student
Northeastern University
Seattle, Washington

## Abstract

**I present an analysis of current state of the art techniques for detecting metaphorical language usage using GloVe embeddings and bidirectional-LSTM models. A selection of literal and non-literal examples of verb usage, taken from the hand-annotated VU Amsterdam Metaphor Corpus, is used as training data. I demonstrate that test data accuracy scores over 0.76, and F1 scores around 0.53, can be achieved in the literal/non-literal verb classification task with relatively simple model architectures and minimal text preparation.**

## 1. INTRODUCTION

Metaphorical language pervades all modes of discourse, giving humans a means of expressing complex, abstract topics in forms that are accessible and easy to understand. As such, the quality of many natural language processing systems—such as machine translation and question and answering systems—is directly related to the ability of machines to detect and interpret figurative language usage. The goal of this paper is to demonstrate how current state-of-the-art deep learning models perform metaphorical language detection, and to explore the implementation and effectiveness of various techniques using the Keras API.

The Vrije University Amsterdam Metaphor corpus (VUAMC) is the most widely used dataset for training metaphor detection systems (Steen et al., 2010). The corpus has annotations for three types of metaphor: indirect, direct, and implicit. Most of the metaphorical usages in the corpus are indirect metaphors, which are identifiable by contrasting a contextual and a more basic meaning of the target word. The following example is cited on the VUAMC website as an example indirect metaphorical usage of the word 'valuable': "professional religious education teachers like Marjorie B Clark… are doing *valuable* work in many secondary schools". In this context, the word 'valuable' is referring not to its basic, literal meaning—a lot of money—but to the concept of value as it relates to education. With direct metaphors, no contrast is made between a contextual and basic meaning; metaphorical usage is determined by how the word is used, and is often signaled by a word such as 'like'. The following is an example of a direct metaphor: "… he's like a ferret". Implicit metaphors are indicated by semantic and/or grammatical linkage between non-literal usage and a referent that is not immediately identifiable as a direct or indirect metaphor. In the following example, the word 'step' forms an implicit metaphor when referred to by the word 'it': "Naturally, to embark on such a step is not necessarily to succeed in realizing it". The VUAMC corpus contains text from academic works, newspapers, conversation, and fiction. Annotations were made using the MIPVU metaphor identification protocol, with each selected sample having an inter-annotator reliability score over 0.8.

Automatic metaphor identification is often divided into two task formulations: sequence labeling and classification. For the sequence labeling task, each sample sentence is accompanied with a sequence of binary labels indicating the metaphoricity of each word. For the classification task, each sample sentence has a target verb and a binary label indicating the verb's non-literal or literal usage. I chose to work on the classification task, using a subset of the VUAMC corpus consisting of 23,112 literal and non-literal usages of 2,298 unique verbs. Each sample consists of a single sentence, a target verb, the index of the target verb in the sentence, and a binary label. Using a

simple bidirectional-LSTM model with a single hidden layer and an 11-gram text preprocessing technique described by Brooks and Youssef (2020), I was able to achieve an accuracy score of over 76 on the test set, and an F1 score of 52.42.

## 2. Related Work

As noted by Stowe et al. (2021), while syntactic structure alone can indicate non-literal usage, "most computational approaches to metaphor eschew syntax for more semantic features". Word vector representations, in particular GloVe (global vectors for word representation) and ELMo (deep contextualized word representations) embeddings, have become standard tools for boosting model performance on the metaphor identification tasks. These embeddings capture not only semantic characteristics of word usage across multiple linguistic contexts, but also model aspects of syntax (Peters et al, 2021). In one of the most cited investigations of deep word embeddings and their application to metaphor detection models, Gao et al. (2018) were able to improve state-of-the-art performance on the VUAMC sequence labeling task by 7.5 F1, and 2.5 F1 on the verb classification dataset, by concatenating a 50d locally generated index embedding with 1024d ELMo embeddings and 300d GloVe embeddings for each input token. Their model was a "relatively standard BiLSTM model… which operate[s] on complete sentences". Pramanick, Gupta, and Mitra (2018) built a bidirectional-LSTM model augmented with a Conditional Random Field (CRF) layer above the LSTM layer for the VUAMC sequence labeling task. They used word2vec Continuous Bag of Words (CBOW) embeddings with a window size of 8 words in various feature configurations that include: token embeddings, token lemmas, part-of-speech, and lemma-token relational markers. Their results indicate that the models performed best using only token word embeddings as input. Brooks and Youssef (2021) built two bidirectional-LSTM models—a many-to-one model where the outputs of the "forward and backward LSTM cells in the attention layer are concatenated only at the output" for the target word, and a many-many model where the weights are updated in relation to the model's performance on both the target word and the surrounding context words—for the VUAMC verb classification task. Similar to Gao et al., they concatenated 300d GloVe and 1024d ELMo embeddings for each input token. They used an 11-gram with the target verb at the center of the 11-gram to represent each sample, padding the 11-gram window to the left and right as necessary. By creating ensembles of these two architectures, Brooks and Youssef were able to achieve F1 scores of over 0.73 for verb classification.

## 3. Approach—Data Preprocessing and model architecture Experiments

To construct a baseline model that could be extended with a variety of enhancements, I decided to use the preprocessing techniques outlined by Gao et al. and Brooks and Youssef. I began by downloading 1024d ELMo embeddings and 300d GloVe embeddings, using the 840 billion token, 2.2 million vocab GloVe file. I then created used a spaCy part-of-speech tagger to generate POS tags for each sample sentence. For each sentence, I created an 11-gram window with the target verb at the center, padding the windows with UNK symbols to the left and right if the target verb was located at the beginning or end of the sample sentence. I then replaced each token in the sample windows with a 1341d vector created by concatenating GloVe, ELMo, and one-hot POS vectors. I then created a simple bidirectional-LSTM architecture with a single hidden layer with 300 nodes, an input layer, and an output layer that applies a sigmoid activation. I discovered that inputting the sample windows in this format using Keras is extremely difficult and fraught with tensor dimension incompatibility errors between the input, embedding, hidden, and output layers. Upon further investigation, I found that the Gao et al. model is publicly available for use on Github; it uses PyTorch, and I was unable to translate their input scheme to my Keras model. The closest I came was using TensorFlow ragged tensors, but that functionality is purportedly not well-supported and repeatedly crashed my notebook.

Taking what I learned about Keras from these experiments, I decided to scale back my input data and create a single GloVe embedding layer for the model. With this setup, outlined by Jason Brownlee (2021) in his Machine Learning Mastery blog post, an embedding weight matrix for GloVe vectors is

created using the 11-gram window tokens as a vocabulary. The model's embedding layer weights are initialized with the GloVe weight matrix, its vocab size is set to the size of the window samples' vocabulary, and the embedding dimension is set to match the GloVe vector dimension. I used what seemed like sensible hyperparameters when creating the model. The bidirectional-LSTM layers have 300 or 128 nodes, a dropout of 0.3, and a recurrent dropout of 0.2. I set my learning rate to 0.001, used a batch size of 128, and an epoch count of 5. I used the rmsprop optimizer and a binary cross entropy loss function, optimizing for accuracy. Because the dataset is limited in size, I thought it best to train the model using k-fold cross validation, with a fold of 8. I set aside 20% of the dataset to use as a test set. The labels are input as an array of binary values.

## 4. Experiments with Model Architecture

I tested 6 different model configurations, varying the type and number of embedding layers and number of bidirectional-LSTM layers. Several models had Luong-style dot product attention layers that operate on output from the LSTM layers. The first model used a simple setup, with a single bidirectional-LSTM layer with 300 nodes and a 300d GloVe embedding layer. To get a sense of how the 11-gram preprocessing technique pioneered by Brooks and Youssef affects results, I ran this model on both the 11-gram input and the 'raw' text samples, where little preprocessing was performed other than mapping the input tokens to integer values. I retained token case and punctuation for both the windowed and raw input to preserve syntactic structure. The windowed model trained quickly, recording F1 scores over 95.0 by the end of the fifth fold. On the test set, the simple GloVe model with 11-gram input recorded an F1 score of 52.04, precision of 56.68, and an accuracy of 75.41. Recall for this model was low at about 50. The model performed much worse on the 'raw' input, outputting an F1 score of 38.55, recall of 37.11, precision of 44.41, and accuracy of 69.50. Given the disparity in results, I decided to conduct the remaining model experiments using only the 11-gram input.

My next models used custom 300d embedding layers that were trained on the input data. I tried the custom embeddings with and without the GloVe embedding layer. By itself, the custom embedding layer model recorded an F1 score 0.4 points lower than the same model with a single GloVe embedding layer. When I combined both embedding layers in a single bidirectional model, I achieved my highest F1 score of 52.42, indicating that embeddings had the greatest affect on model performance, after accounting for the 11-gram preprocessing.

My other two model experiments involved the addition of a Luong-style Dot-product attention layer after the bidirectional-LSTM layers. For both of these models, I added an additional bidirectional-LSTM layer and used the LSTM outputs as the query and key tensor inputs (using the terminology from the Keras Attention API documentation) to the attention layer. The first model used a single GloVe embedding layer passed through two LSTMs. The second model used a GloVe embedding layer and a custom 300d embedding layer. Interestingly, neither of these models were able to improve the results of the single bidirectional-LSTM models. The model with layers for both embeddings predictably produced the higher F1 score—51.68 vs 50.96.

| **Model #** | **Description** |
|---|---|
| model 1 | Single BiLSTM with Glove embeddings and 11-gram Input |
| Model 2 | Single BiLSTM with GloVe embeddings and 'raw' input |
| model 3 | Two BiLSTM with Glove embedding and attention layer |
| Model 4 | Single BiLSTM with GloVE and custom 300d embeddings |
| Model 5 | Single BiLSTM with custom 300d embeddings |
| Model 6 | Two BiLSTM with GloVE embeddings |
| Model 7 | Two BiLSTM with GloVe and custom 300d embeddings and attention |

Table 1: Model Descriptions

| **Model #** | **Precison** | **Recall** | **F1** | **Accuracy** |
|---|---|---|---|---|
| Model 1 | 56.68 | 50.90 | 52.04 | 75.41 |
| Model 2 | 44.41 | 37.11 | 38.55 | 69.50 |
| Model 3 | 54.95 | 50.51 | 50.96 | 74.48 |
| Model 4 | 54.06 | 53.43 | **52.42** | 74.56 |

| | | | | |
|---|---|---|---|---|
| Model 5 | 48.78 | 51.55 | 48.63 | 71.12 |
| Model 6 | **58.42** | 48.56 | 51.51 | **76.08** |
| Model 7 | 48.40 | **58.82** | 51.68 | 70.99 |

Table 2: Model results, VUAMC Test Data

| Model | F1 – VUAMC Verb CLS (Test) |
|---|---|
| Gao et al. | 58.9 |
| Brooks & Youssef | 73.7 |
| Saulnier | 52.4 |

Table 3: Model performances on verb classification task

## 5. Analysis

To generate data for analysis, I created predictions of the test data using model 1. Of the 1,819 incorrect predictions, 371 of these verbs were never correctly predicted. Predictably, correctly predicted verbs had a much higher average occurrence rate in the overall dataset: verbs correctly predicted at least once occurred an average of 24.65 times. In comparison, verbs that were never correctly predicted occurred an average only of 4.1 times, underscoring the fact that novel metaphors remain very difficult for neural metaphor detection systems to identify. The split between false positives and false negatives was nearly even, with false positives accounting for 52.28% of the incorrect predictions and false negatives accounting for 47.72%.

While the subset of the VUAMC data that I was working with did not include tags to indicate if metaphorical usage was direct, indirect, or implicit, from my manual inspection it looks like most of the incorrectly classified samples are indirect metaphors, the majority of them commonplace usages. However, there is a healthy mix of implicit and direct metaphors present in the misclassified samples. The following is an example of a false negative, with the target verb emboldened: "Further studies, researchers believe, should **concentrate** on pregnant women known to have a zinc deficiency or to be at risk from problems during labour." Here we have the word 'studies' implicitly linked to the metaphorically used 'concentrate'. This is an interesting usage however, because 'concentrate' is metaphorical in multiple senses: we can interpret 'concentrate' as a property of attentive cognition, or as it relates to a property of substances denoting density, strength, or potency (although this usage is a noun, it resonates with the 'zinc deficiency' mentioned later in the sample). In this instance, 'concentrate' also can be read in its literal sense 'to gather', but the first usage is probably more accurate given that 'concentrate' is followed by 'on'. This is a good example of how sense disambiguation—a task closely related to metaphor detection—is not clear-cut, even for humans; while one meaning, often signaled by part of speech, may be more coherent in a given context, usages often encode multiple senses. Here is a simple indirect metaphor, apparently from a work of fiction, that was incorrectly classified: "There the similarity ends, since **judging** from his writings Bagehot was a roaring snob, whereas judging from his heckles the heckler was not." Here is a simple literal usage that was classified as metaphorical: "The feet of the sauropod are small (relatively speaking!) with short, stubby toes, yet animals that **walk** on soft mud tend to have spreading feet to distribute their weight more evenly."

## 6. Conclusion

As Gao et al. note, and many other researchers on the topic have demonstrated, relatively simple RNN architectures with deep embedding layers produce fairly good results on the VUAMC metaphor identification shared task. With a single bidirectional-LSTM model using GloVe and custom 300d embedding layers, I was able to achieve an F1 score of 52.42 and an accuracy score of 76.08 on a selection of VUAMC dataset containing 23,112 verb classification samples and 2,298 unique target verbs. Neural metaphor identification models notably struggle to classify novel metaphors; in my analysis of model prediction data, verbs that were always incorrectly classified appeared in the dataset an average of 4.1 times, while verbs correctly classified at least once appeared an average of 24.65 times. In this vein, future work on metaphor identification should explore methods for generating additional high quality training data.

## REFERENCES


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 Langauge Processing*, pages 607-613, Brussels, Belgium. Association for Computational Linguistics.

Gerard J Steen, Aletta G Dorst, J Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for linguistic metaphor identification. From MIP to MIPVU*, volume 14. John Benjamins Publishing.

Jason Brownlee. 2021. How to Use Word Embeddings Layers for Deep Learning with Keras. *Machine Learning Mastery Blog*. Online.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processings (EMNLP)*, pages 1532-1543, Doha, Qatar. Association for Computational Linguistics.

Jennifer Brooks and Abdou Youssef. 2020. Metaphor detection using ensembles of bidirectional recurrent neural networks. In *Proceedings of the Second Workshop on Figurative Language*, pages 244-249, Online. Association for Computational Linguistics.

Julia Birke. 2005. TroFi Example Base. *SFU Natural Language Laboratory*.

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 262-371, Hong Kong, China. Association for Computational Linguistics.

Malay Pramanick, Ashim Gupta, and Pabitra Mitra. 2018. An LSTM-CRF Based Approach to Token-Level Metaphor Detection. In *Proceedings of the Workshop on Figurative Language Processing*, pages 67-75, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. 2018. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227-2237, New Orleans, Louisiana. Association for Computational Linguistics.