Metaphor Detection using Machine Learning

Amith Korada, Apuroop Sreepuram, Swathi Vangala and Prasanth Kumar Kunta

Abstract

This research focuses on identifying metaphors in text using various machine learning methods. We used a collection of sentences, each marked as either containing a metaphor or not, to test different models. These models range from simpler ones like logistic regression to more complex ones based on neural networks and the BERT language model. We looked at how well these models perform by checking their accuracy and other key measures. Our results show important findings about how metaphors can be detected using computers. They also provide a starting point for more studies in this area. This work is significant as it helps us understand how machines can recognize metaphors, an important part of human language.

1 Introduction

Understanding metaphors in language is a challenging yet fascinating task. Metaphors are a way of expressing ideas by comparing unlike things, and they are common in everyday language. The ability to identify metaphors in text is not only important for understanding human communication but also has applications in fields like artificial intelligence and language processing.

In this study, we focus on detecting metaphors using machine learning, an area of computer science that allows computers to learn from data. Our goal is to explore different machine learning models and see how well they can identify metaphors in sentences. We use a dataset that consists of sentences labeled as either metaphorical or non-metaphorical.

This report will first review previous research in this area, then describe the methods we used in our study. We will present and discuss the results of our experiments with various models. Finally, we will conclude with our findings and suggest directions for future research. This study aims to contribute to the growing field of natural language processing by enhancing our understanding of how machines can recognize metaphors, a key element of human language.

2 Dataset

2.1 Overview

The dataset comprises 1,870 instances, designed to facilitate the computational understanding of metaphorical language. Each instance in the dataset includes a metaphor candidate word, identified by a metaphorID, and a binary label (label boolean) indicating the word's usage context in the corresponding text excerpt.

2.2 Data Fields

MetaphorID: A numerical identifier assigned to commonly used English words that may have both metaphorical and literal meanings. The metaphorID corresponds to one of several preselected words which are potential candidates for metaphorical usage.

Label Boolean: A binary label, with True representing metaphorical usage and False representing literal usage of the word specified by the metaphorID within the text.

Text: The textual excerpt containing the word in question. The excerpts were sourced to include a diverse range of contexts, thereby providing a robust set of examples for metaphor analysis.

3 Pre-Processing

3.1 Text Normalization

Before analysis, the dataset underwent text normalization. This process involved converting all text to lowercase to ensure uniformity and facilitate accurate word matching, regardless of the original casing used in the text.

3.2 Sentence Tokenization

We employed the Natural Language Toolkit (NLTK) for Python to tokenize the text data into individual sentences. This step is crucial as our analysis required the isolation of the sentence containing the metaphor candidate word to accurately determine its context and usage.[5]

3.3 Word-to-MetaphorID Mapping

Each metaphorID was mapped to its corresponding candidate word to link the numerical identifier to the actual word being analyzed. This mapping was essential for the subsequent step of sentence extraction, allowing us to pinpoint the usage of specific words within the text.[2]

3.4 Sentence Extraction

Using the mapped words, we extracted the first sentence from each text excerpt containing the metaphor candidate word. The purpose of this step was to work with the immediate context in which the word was used, which is often sufficient to determine metaphorical usage.

3.5 Data Structuring

The extracted sentences were then appended to the original dataset as a new column. This enriched the dataset by providing direct access to the relevant textual context for each instance, thereby streamlining the analysis process.

4 Machine Learning Models

4.1 Neural Network Fusion: TF-IDF and BERT Features

Our approach integrates TF-IDF (Term Frequency-Inverse Document Frequency) vectorization with BERT (Bidirectional Encoder Representations from Transformers) embeddings to enhance metaphor detection. The combination harnesses the statistical significance of word occurrence (TF-IDF) and the contextual depth of language (BERT), vital for identifying metaphors.The combination of TF-IDF and BERT allows for a more nuanced analysis, capturing both the statistical relevance and the contextual subtleties necessary for effective metaphor detection.[4]

4.1.1 Data Preprocessing and Feature Combination

The text data were first transformed using TF-IDF vectorization, highlighting the importance of specific words within the context of the document. Following this, BERT embeddings were generated for each sentence to capture contextual nuances. These two feature sets were then combined to create a comprehensive representation of each sentence, balancing both word-level significance and sentence-level context.

4.1.2 Neural Network Integration for Enhanced Feature Analysis

To augment our metaphor detection capabilities, we integrated the combined TF-IDF and BERT features into a neural network model. Developed using TensorFlow, the neural network architecture consists of a sequence of dense layers. It begins with layers of 512, 256, and 128 neurons, each using a 'relu' activation function to introduce non-linearity, essential for handling complex patterns in the data. The architecture culminates in an output layer with a 'sigmoid' activation function, tailored for binary classification tasks, such as distinguishing metaphorical from literal sentences.

The model was compiled using a binary cross-entropy loss function, selected for its effectiveness in binary classification problems, and the Adam optimizer, known for its efficiency in handling large datasets and feature spaces. Training was conducted over 10 epochs with a batch size of 32, a configuration chosen to balance computational efficiency with the ability to learn from the rich, combined feature set. This setup aims to maximize the potential of the combined TF-IDF and BERT features, enabling the model to more accurately identify metaphorical language in varied textual contexts

4.2 Fine Tuning with BERT

In addressing the complex task of metaphor detection, our approach leverages the BERT (Bidirectional Encoder Representations from Transformers) model. This choice is motivated by the necessity to comprehend linguistic context and nuances that are pivotal in identifying metaphors.[3]

4.2.1 Model Configuration and Rationale

In our exploration of metaphor detection, we implement a BERT (Bidirectional Encoder Representations from Transformers) based model. The 'bert-base-uncased' variant is employed for its proficiency in contextual understanding of language, a crucial factor in detecting metaphors. We fine-tune this pre-trained model to specifically cater to the nuances of metaphorical language.

4.2.2 Feature Extraction

The heart of our feature extraction lies in the utilization of a transformer-based model, BERT. BERT's architecture is particularly adept at capturing the subtleties of language, including the nuanced meanings that distinguish metaphorical from literal usage. Its bidirectional nature allows for a rich contextual understanding, essential for metaphor detection. By translating sentences into embeddings, we transform complex linguistic constructs into a mathematical space where machine learning algorithms can operate effectively.

4.2.3 Linear Classifier for Decision Making

Once we have the sentence embeddings from BERT, we need to make a decision: Is this sentence metaphorical? Here, a linear classifier comes into play. It's a relatively simple tool that takes the complex data from BERT and decides if the sentence is metaphorical. This step is vital because, while BERT gives us a rich understanding of the sentence, we need a straightforward way to classify it. The linear classifier does just that, acting as a decision-maker that interprets BERT's detailed analysis and provides a clear yes or no answer regarding the presence of a metaphor.

4.3 LinearBERT-CLS- Contextual Metaphor Insight

This model harnesses the power of BERT's CLS token embeddings, enriched with contextual nuances, integrated into a streamlined linear neural classifier. This model represents a novel blend of deep learning and linguistic insight, setting a new standard for understanding metaphorical language in computational linguistics.[6]

4.3.1 Rationale Behind Utilizing the CLS Token

In BERT's world, the CLS token is like a summary of the entire sentence. It's trained to capture the essence of the whole text, making it a compact yet comprehensive representation of the sentence's meaning. Since metaphors often hinge on the overall context rather than individual words, the 'CLS' token is particularly useful. It provides a condensed version of the sentence that retains the crucial elements needed for metaphor detection, allowing for a more accurate classification.

Once we have the sentence represented as a CLS token from BERT, we employ a linear classifier as mentioned in

4.4 FocusBERT: Enhanced Metaphor Detection

This 'AttentionModel' is a special tool for finding metaphors in sentences. It uses BERT, which is good at understanding language, and adds something called an attention mechanism. This mechanism helps the model pay extra attention to the most important words, making it really good at spotting metaphors.[1] 4.2.3.

4.4.1 Attention Mechanism Focused on Target Words

A key innovation in our model is the targeted attention mechanism. This mechanism calculates attention scores based on each word's distance from the target word within a sentence. These scores are computed as the inverse of the position distance, allowing words closer to the target word to have higher attention scores. The idea is to amplify the influence of words around the target word, as they are more likely to be crucial in understanding the metaphorical use. Attention scores are determined using a simple yet effective formula: This formula ensures that words closer to the target word are given more importance, while the influence of distant words is reduced. This focused attention allows the model to hone in on the most relevant parts of the sentence for detecting metaphors.

Upon obtaining weighted sentence embeddings through the attention mechanism, we employ a linear classifier as mentioned in 4.2.3.

4.5 Logistic Regression with class adjustment

Our model aims to classify text excerpts as metaphorical or literal, addressing the challenge of metaphor detection. This is a binary classification task, where the output is True for metaphorical and False for literal usage.[7]

4.5.1 Addressing Class Imbalance

The uneven distribution of metaphorical and literal instances presents a challenge: a model trained on imbalanced data may develop a bias toward the more frequent class. We counteract this with class weights, equipping the logistic regression to emphasize the significance of the rarer class, thereby fostering a more equitable learning process.

4.5.2 Logistic Regression

We extract features as done for the previous model (4.2 Finetuning BERT) and then We employ logistic regression for its interpretability and efficiency.

Table 1: Performance comparison of metaphor detection models: The precision (L / M), recall (L / M), and F1-score (L / M) are reported separately for literal (L) and metaphorical (M) classes

Model	Precision (L / M)	Recall (L / M)	F1-Score (L / M)	Accuracy
Neural Network Fusion: TF-IDF + BERT	0.82 / 0.95	0.85 / 0.94	0.84 / 0.95	92.25%
Fine-tuned BERT	0.88 / 0.97	0.91 / 0.96	0.90 / 0.96	94.38%
LinearBERT-CLS	0.88 / 0.95	0.86 / 0.96	0.87 / 0.95	93.00%
FocusBERT	0.89 / 0.95	0.85 / 0.96	0.87 / 0.95	93.00%
Logistic Regression	0.79 / 0.94	0.84 / 0.92	0.81 / 0.93	90.00%

Despite its simplicity, when augmented with class weights and high-quality features from BERT, it becomes a powerful tool for classification tasks such as ours. Its linear decision boundary is informed by the complex representations of language derived from the embeddings, enabling it to distinguish between metaphorical and literal expressions with finesse.

5 Results

In assessing machine learning models for metaphor detection, four key metrics are crucial: precision (accuracy in identifying metaphors), recall (ability to find all metaphors), F1-score (balancing precision and recall), and accuracy (overall prediction correctness). These metrics collectively gauge each model's effectiveness in metaphor detection.

From the results, we observe that:

TF-IDF + BERT: This model achieved a balance between precision and recall, indicating a well-rounded performance in identifying both literal and metaphorical sentences, with a slight preference towards detecting metaphorical ones.

Fine-tuned BERT: Exhibiting the highest F1-score and accuracy among the models, fine-tuned BERT demonstrates superior performance, particularly in the precision of metaphorical sentence detection.

BERT + Neural Network: This model shows competitive results, especially in recall for metaphorical sentences, suggesting it is effective at identifying most of the metaphorical language present in the test set.

BERT + Attention: The attention mechanism in this model seems to enhance the recall for metaphorical sentences, which is crucial in applications where missing a metaphorical expression could be more detrimental than a false positive.

Logistic Regression: As a more traditional machine learning approach, logistic regression lags slightly behind the deep learning models but still performs reasonably well, considering the complexity of the task.

The results indicate that while all models perform reasonably well, models leveraging BERT, either in its original form or with additional neural network layers and attention mechanisms, tend to outperform the logistic regression model. This suggests that the depth and contextual awareness provided by BERT contribute significantly to the task of metaphor detection.

6 Conclusion

In conclusion, while the BERT-based models have shown promising results in metaphor detection, the imbalance in our dataset poses a concern for potential overfitting, particularly in classes with fewer instances such as metaphorID 1 and 3 with only 14 and 16 instances respectively. Also, the distribution of the label boolean shows a considerable skew, with 1432 instances labelled True for metaphorical usage and only 438 labelled False for literal usage Such imbalances could lead the models to overfit to the metaphorical class and fail to generalize to more evenly distributed real-world data.

Moving forward, addressing the data scarcity and imbalance will be imperative. Techniques like data augmentation, resampling, and synthetic data generation could prove beneficial in creating a more balanced dataset, thereby enhancing the model's ability to generalize. Future work should also prioritize cross-validation and alternative metrics that are sensitive to class imbalances to ensure a thorough evaluation of model performance.

References

[1] Shashwat Aggarwal and Ramesh Singh. 2020. Metaphor detection using deep contextualized word embeddings.

[2] Kathleen Ahrens. 2010. *Mapping Principles for Conceptual Metaphors*, pages 185–207.

[3] 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, Minneapolis, Minnesota. Association for Computational Linguistics.

[4] Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

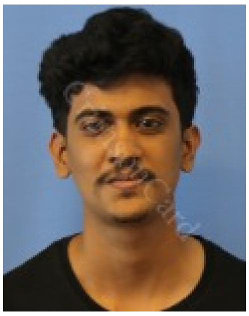
[5] Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

[6] Chang Su, Kechun Wu, and Yijiang Chen. 2021. Enhanced metaphor detection via incorporation of external knowledge based on linguistic theories. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1280–1287, Online. Association for Computational Linguistics.

[7] Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51.

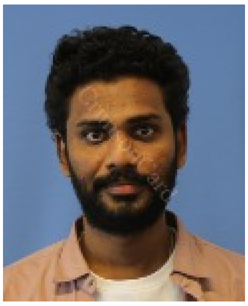
7 Team members

7.1 Apuroop Sreepuram



7.2 Amith Korada

7.3 Prasanth Kumar Kunta



7.4 Swathi Vangala