

Group 13 Metaphor Detection

Pruthvik Elemati, Sai Chandan Akella, Sri Rama Krishna Reddy Dwarampudi, Yashwanth Gajji

1 Background

1.1 Significance of Metaphor Detection in Natural Language Processing

Metaphor detection plays a crucial role in natural language processing (NLP) by contributing to a more nuanced understanding of language. Metaphors are pervasive in everyday communication, often conveying abstract concepts through familiar, concrete terms. Detecting metaphors is essential for machines to comprehend the intended meaning behind words and phrases, enabling more accurate language processing.

Metaphors go beyond literal expressions, adding layers of meaning and cultural context to language. Recognizing metaphors enhances the capabilities of NLP systems to interpret sentiment, context, and the speaker's intended message. This is particularly important in applications such as sentiment analysis, chatbots, and machine translation, where capturing the subtle nuances of language is crucial for delivering contextually accurate results.

1.2 Potential Applications

1.2.1 Sentiment Analysis

Metaphor detection contributes to sentiment analysis by enabling machines to grasp the emotional tone and nuances in text. Understanding metaphorical expressions allows sentiment analysis models to more accurately assess the sentiment behind user-generated content.

1.2.2 Chatbots and Virtual Assistants

In conversational AI, detecting metaphors enhances the naturalness and context-awareness of chatbots and virtual assistants. This capability is vital for creating more engaging and human-like interactions, where machines can respond appropriately to figurative language.

1.2.3 Machine Translation

Metaphors often pose challenges in machine translation as their meaning may not directly align with the literal translation. Metaphor detection helps improve the accuracy of translated content by identifying and handling metaphorical expressions appropriately.

1.2.4 Content Summarization

Metaphor-aware NLP systems can generate more insightful and contextually relevant summaries of content. By understanding metaphors, machines can capture the essence of the text more accurately, providing users with more meaningful summaries.

2 Related Work

The current state of the art results for metaphor detection tasks are all based on the large language models. Most metaphor detection tasks are based on the VU Amsterdam corpus, TroFi dataset, MOH-X and LCC datasets. Other prominent research areas involving metaphors include interpreting the literal meaning from the metaphor and generating a metaphorical sentence given a literal.

For the metaphor detection task, Mao et al. 2019,[6] uses metaphor identification theories using deep neural networks for metaphor detection tasks. Their research achieves SOTA results on VUA, MOH-X and TroFi datasets. With the advent of BERT (Devlin et al., 2019 [4]), a lot of deep neural networks are replaced with this pre-trained model. The pre-trained model is fine-tuned on the given task. Using this and the previous metaphor identification theories, Choi et al., 2021 [1] proposed an architecture which aggregates the meaning of the word in context and without context separately and uses it for identifying metaphors.

Li et al., 2023 [3] identified that BERT does not really encode the actual meaning of a word in its pre-trained embedding and proposed a method for modeling explicit basic meaning of a word. The remaining architecture is similar to Choi et al., 2021 [1]. Since metaphors are also based on the conceptual mapping between domains, Li et al., 2023 [2] used FrameNet to extract frames for the source and target domains. Their research believed that frames from FrameNet could mimic the concept of domains and used the frame embeddings along with the metaphor identification theories to propose a model, which achieved the state of the art results and is also more explainable and interpretable compared to other methods.

[3] introduces the Naïve Bayesian classifier, praising its effectiveness despite assumptions of feature independence. It outlines the classifier's advantages, drawbacks,

and various versions. However, [4] lacks a literature review, which typically offers an overview of prior studies on Naïve Bayesian classifiers, including their historical context, strengths, weaknesses, challenges, etc in machine learning. It proposes using regression analysis to understand the link between these factors and metaphor use.

Study[6] introducing Miss RoBERTa WiLDe (Metaphor Identification using Masked Language Model with Wiktionary Lexical Definitions) for metaphor detection. However, it doesn't explicitly include a section labeled "literature review." The text mainly focuses on the methodology and findings of the proposed model rather than explicitly reviewing existing literature on metaphor identification or related Natural Language Processing (NLP) approaches.

3 Dataset Description

Dataset consists of 1870 rows of 3 columns. Below are the three columns:

- metaphorID - ID of the word used as metaphor. IDs are in the range of [0-6]
- label_boolean - True/False values showing the label whether word is used as metaphor or not. True if word is used as metaphor and False if not used.
- text - Paragraph of sentences in which we have to detect the metaphor.

Dataset contains 1432 positive class labels and 438 negative class labels. Rows with metaphor are four times more than non metaphor rows. There can be chance the models may get more trained for positive class.

When splitting the dataset into training and testing sets, we can use stratified sampling to ensure that the class distribution in both sets reflects the overall distribution in the entire dataset. Choosing appropriate evaluation metrics such as precision, recall, F1-score can provide a more comprehensive view of the model's performance, especially in imbalanced scenarios.

4 Methodology

The current task of metaphor detection is slightly different compared to the previous ones. In the case of other datasets, such as VUA, MOH-X and TroFi, the context is not that long. For the current problem, we are provided with a lot of context. In our methods, we used the entire context for Logistic Regression, Naive Bayes, Random Forrest and Base BERT classifier. But

for the approaching involving contextualized late interactions using metaphor identification theories, based on [1], we have used only the 50 words preceeding the target word and 50 words after the target word. Since this method involves calculating the contextual meaning of the target word, if the context is too long, then it would mean we have to consider longer sequences for last layer hidden representations. To avoid this scenario, we shortened the context.

4.1 Model Architecture

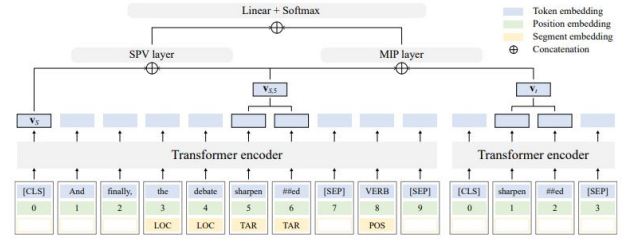


Figure 2: Model architecture of MelBERT. When a target word w_t is split into multiple tokens by BPE, the average pooling is used for the target word.

Figure 1: MelBERT architecture

The MelBERT architecture as proposed in [1], Leverages contextualized late interaction along with metaphor identification theories to perform the task. By late interaction they mean that the sentence and the target word are encoded independently initially so that their individual meanings can be obtained. In this method, they take two sentences as input. First sentence with target word and second only with target word. (similar to bi-encoder network). Given a sentence $S = w_1, \dots, w_n$ with n words and a target word w_t , the model returns a binary output, i.e., 1 if the target word w_t in S is metaphorical or 0 otherwise. RoBERTa is adopted as the backbone model, since it is known to outperform BERT.

4.2 Metaphor Identification Theories

Metaphor Identification Procedure (MIP) - Recognized if literal meaning is different from Contextual meaning. $V_{s,t}$ and V_t are concatenated and passed into linear + Softmax layer, since those two vectors represent contextualized and isolated meaning of word w_t .

Selectional Preference Violation - If the target word is unusual in the context of surrounding words. It is based on the assumption that V_s and $V_{s,t}$ show a semantic gap if w_t is metaphorical. V_s represents the interaction across all pair-wise words in S , but $V_{s,t}$ represents the interaction between w_t and other words in S . In this sense, when w_t is metaphorical, $V_{s,t}$ can be different from V_s by the surrounding words of w_t .

5 Experimentation

5.0.1 Dataset Splitting

Dataset consists of 1870 rows and split into training and test data with 8:2 ratio. 80% of the data is used as training data and the remaining 20% as test data. Used `train_test_split` from `scikit-learn` library to split the dataset.

5.1 Training

Main aim of the training is to build a binary classifier to classify the dataset whether metaphor is detected or not. Various machine learning models like Logistic Regression, Naive Bayes, Random Forest Algorithm and pre-trained NLP models like RoBERTa.

5.1.1 Logistic Regression

Logistic Regression models are relatively simple and offer interpretability. The coefficients assigned to each feature (word) can provide insights into the impact of individual words on the likelihood of metaphorical language. This interpretability can be valuable in understanding the linguistic features associated with metaphor. If there exists linear relationships between certain words and the likelihood of metaphor, Logistic Regression might be effective.

Words in the text are first vectorized using `CountVectorizer` a text feature extraction method provided by `scikit-learn`. `CountVectorizer` breaks the text into individual words or tokens and builds a vocabulary of all unique words (tokens) present in the entire collection and counts the occurrences of each token in the vocabulary. The count is recorded in the corresponding entry of the document-term matrix. The matrix is sparse, as most entries are zero since only a subset of the entire vocabulary appears in each text. The result is a document-term matrix, where each row corresponds to a text row, and each column corresponds to a unique token in the vocabulary. The entries represent the counts of each word in the respective text.

5.1.2 Naive Bayes Algorithm

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem, which describes the probability of an event, based on prior knowledge of conditions that might be related to the event. It is easy to implement, generally requires fewer computational resources and tends to perform well even with small training datasets.

Naive Bayes is well-suited for high-dimensional data, such as text data, where the number of features (words) can be substantial. Despite its "naive" assumption of feature independence, it often performs surprisingly well in practice, making it effective for text classification tasks.

To train the Naive Bayes, we used the same output of tokens from `CountVectorizer` as features. Count of tokens is best useful for Naive Bayes algorithm, which is completely based on counting and finding the posterior probability for a token in a text.

5.1.3 Random Forest Algorithm

Random Forest is robust to noise and outliers in the data. In natural language, metaphorical expressions can vary widely and may introduce noise into the dataset. The ensemble nature of Random Forest helps mitigate the impact of outliers and noisy examples.

Random Forest provides a measure of feature importance, indicating which features (words or linguistic patterns) are most relevant for making predictions. Metaphors often involve non-linear relationships between words, where the combination of certain words may indicate metaphorical usage. Random Forest, being an ensemble of decision trees, can capture complex non-linear patterns in the data more effectively than linear models.

Features for Random forest classifier are the token generated by `TfidfVectorizer` (Term Frequency-Inverse Document Frequency Vectorizer), another feature extraction method provided by `scikit-learn`. It first tokenizes the texts and counts the occurrences of each word in each document. This is the term frequency (TF) part. The inverse document frequency is calculated for each word in the entire text set. IDF is a measure of how unique or rare a word is across all documents. Words that appear in many documents have lower IDF weights, while words that are rare have higher IDF weights.

5.1.4 Baseline BERT Classifier

In this approach, we take a pre-trained BERT model and fine-tune it on the metaphor detection task by adding a linear and a softmax layer on top of BERT. This approach is used as a baseline for comparison against other complex models like MelBERT. BERT gives good contextualized representations and it could be leveraged to perform the classification. It also allows us to proceed with the classification task without much feature engineering.

5.1.5 MelBERT

This model, described in the [1], achieved state-of-the-art performance on metaphor detection tasks by fine-tuning BERT with a late interaction mechanism based on linguistic theories of metaphor identification.

5.2 Result Evaluation

With an accuracy of 0.8 a precision of 0.83 means that when the Logistic Regression model predicts a sentence as metaphorical, it is correct approximately 83% of the

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.8	0.83	0.91	0.87
Naive Bayes	0.81	0.88	0.86	0.87
Random Forest	0.76	0.77	0.76	0.70
Baseline BERT	0.91	0.95	0.92	0.94
MelBERT model	0.95	0.96	0.97	0.96

time. With a recall of 0.91, the model is able to capture about 91% of actual metaphorical sentences. This high recall suggests that the model is effective at identifying a significant portion of the metaphorical instances in the dataset.

The above results suggest that Naive Bayes is a competitive choice with an accuracy of 0.81 which is better than Logistic Regression. While precision is 0.88 which beat the Logistic Regression but recall is 0.86 which is lower than Logistic Regression.

Random Forest model shows a good but slightly lower performance compared to Logistic Regression and Naive Bayes with 0.76 accuracy. It falls slightly behind in terms of precision, recall, and F1-Score as well compared to other models.

MelBERT showcase outstanding performance across all key metrics. Accuracy of 95% suggests a highly accurate model in distinguishing between metaphorical and non-metaphorical sentences. This high F1-Score reflects a robust overall performance, striking a good balance between minimizing false positives and false negatives. MelBERT, being a transformer-based model, likely benefits from its ability to capture contextual information and complex patterns in language.

6 Future Work

In future work we can enhance the diversity and size of the labeled dataset through data augmentation techniques. This can involve generating additional metaphorical and non-metaphorical examples to improve the model’s generalization. We can experiment with fine-tuning the hyperparameters of the existing models, exploring the use of more advanced pre-trained language models. Predictions of multiple models can be combined using ensemble methods such as stacking or bagging can improve the performance.

7 Conclusion


BERT based classifiers outperform traditional machine learning techniques. After experimenting with various machine learning techniques, we have finally observed that fine-tuning BERT with metaphorical identification theories on top of it performs the best. This method not only leverages the contextual embeddings it obtained from its training, but also makes use of the theories which could help in classification. This shows that along with contextualized representations, incorporating linguistic theories into model will greatly benefit the model. Even without any metaphor theories, if we were to just fine-tune BERT, its performance beats all the other techniques.

Traditional machine learning algorithms fail to model the complex representations between words and do not take the context into account, giving a more static representation. Metaphor identification requires capturing linguistic structures which Logistic regression, random forest and Naive Bayes fail to do so. Though some of the linguistic features could be incorporated by manual feature engineering, it requires extensive domain knowledge and therefore, for tasks like these, BERT based model are a good choice, both in terms of performance and time taken to build models.

8 References

1. Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, Jongwuk Lee. "MelBERT : Metaphor Detection via Contextualized Late Interaction Theories."
2. Yucheng Li, Shun Wang, Chenghua Lin, Frank Guerin, Loic Barrault : "FrameBERT : Conceptual Metaphor Detection with Frame Embedding Learning."
3. Yucheng Li, Shun Wang, Chenghua Lin, Frank Guerin : "Metaphor Detection via Explicit Basic Meaning Modeling"
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova : "BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding."
5. Mao, Rui, Xiao Li, Mengshi Ge, and Erik Cambria. "MetaPro: A computational metaphor processing model for text pre-processing." *Information Fusion* 86 (2022): 30-43.
6. Rui Mao, Chenghua Lin, Frank Guerin "End-to-End Sequential Metaphor Identification Theories Inspired by Linguistic Theories"

9 Team

Pruthvik Elemati	Sai Chandan Akella
	
Sri Rama Krishna Reddy Dwarampudi	Yashwanth Gajji
	