IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

METAPHOR DETECTION BY USING DEPENDENCY PARSER

[1]Mansi Kortikar, [2]Roshni Kanse, [3]Samruddhi Mithbawkar, [4]Iffat Kazi

[1]Student, [2]Student, [3]Student, [4]Professor

[1]Department of Computer Engineering,

[1]SNDT Women's University, Mumbai, India

Abstract: A metaphor is a figure of speech used to express a specific meaning or idea by drawing comparisons between two entities or ideas that are not necessarily related to one another. Without using the terms "like" or "as," metaphors use implicit comparisons. Instead, they substitute a word or phrase that doesn't directly relate to the topic at hand but instead makes an analogy or comparison. Understanding and processing metaphors is a difficult topic in natural language processing. Consequently, the current need for metaphor classification involves an innovative approach. In order to conduct tasks like entity linking and relation extraction, we employ a method known as Knowledge Graph Embedding to represent the data in a knowledge graph as numerical vectors. When examining metaphors, we notice a structural pattern where the source and target are linked by a relationship that depends on a specific attribute. The target, the attribute, and the source make up the metaphor triple that describes this pattern. We use dependency parsing to extract collocations of concepts and attributes. In our method, we jointly embed the metaphor Knowledge Graph and the Concept-Attribute collocations. After that, the goal of metaphor identification can be seen as representation-enhanced concept pair classification. We also work on prediction of the possible attributes when given two concepts i.e. target and source from knowledge graph. By improving metaphor detection accuracy, this technique will fill a research gap.

Index Terms – Natural Languages, Computational Linguistics, NLP

I. Introduction

A metaphor is a literary or rhetorical figure of speech that describes a subject or idea by comparing it to something else, which may be unrelated or dissimilar. It is a comparison between two things that highlights a particular similarity or aspect that they share, without using the words "like" or "as" and it is made without using a direct comparator. The term metaphor derives from the ancient Greek word "metaphora", which translates to "to transfer" or "to carry over". In some senses, a metaphor does the exact same purpose by bringing a shared trait or attribute between two things or concepts of various kinds. A widely established paradigm for comprehending metaphors is conceptual metaphor. Metaphor identification approaches based on word embeddings have become popular [2][4] as they do not rely on hand-crafted knowledge for training. Metaphors typically involve two conceptual domains, which are the source domain and the target domain. The source domain is the domain from which the metaphorical expression draws its meaning, while the target domain is the domain to which the meaning is applied. For example, in the metaphor "He has a heart of stone," the source domain is the physical object "stone," while the target domain is the abstract concept of "emotional coldness". Two kinds of representations that are important to understand metaphor are the linguistic representation and the conceptual representation. The linguistic representation refers to the actual words or phrases used to convey the metaphorical expression. It includes the words, grammar, syntax, and structure of the language used to express the metaphor. The conceptual representation, on the other hand, refers to the mental concepts or images that are evoked by the metaphorical expression. It involves the mental connection between the source and target domains and how they are mapped onto each other. Phrase-level models [2][4] are likely to fail in the metaphor identification task if important contexts are excluded. Some of the objectives can be-

1. The main objective of a metaphor detection project is to create a platform that can perform classification of metaphorical and non-metaphorical (literal) sentences.
2. To predict possible attributes when given two concepts (i.e. target and source).
3. To address the research gap by increasing the accuracy of this classification problem.

II. Methods

2.1. Dataset Creation

Our dataset is the collection of metaphorical and nonmetaphorical sentences. The independent variables or inputs of the dataset are sentences and dependent variables or inputs are the labels to each sentence stating whether respective sentence is metaphorical or not. Dataset is created on manual basis and also with the help of web scraping from websites related to english figure of speech. There are total 1815 instances in our dataset out of which 1558 are metaphorical sentences and 257 are non-metaphorical sentences.

2.2. Part-of-Speech Tagging

We firstly did tokenization of sentences and then POS tagging is applied on same instances. Part-of-speech (POS) tagging is to be applied to the dataset sentences, a technique used in natural language processing (NLP) to label each word in a text with the appropriate part of speech. It describes their syntactic category or part of speech such as noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. POS tagging is an important activity in natural language processing since it aids in understanding a sentence's meaning and analysing its structure. In the sentence "The cat sat on the mat", for instance, the words "cat" and "sat" are nouns, "on" is a preposition, and "the" is a determiner. A linguist can tag POS manually, or machine learning algorithms trained on annotated corpora of text can tag POS automatically. Part-of speech tagging is carried out via the NLTK wrapper module using a Stanford POS tagger that is installed locally.

2.3. Dependency Parsing

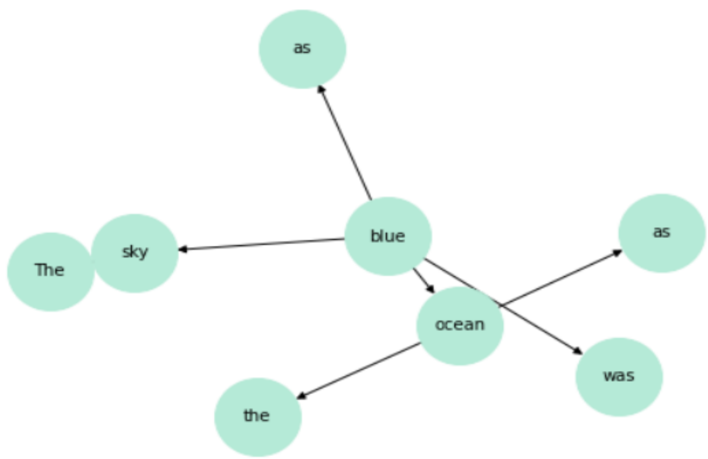

Figure 1. Visualization of Dependency Parsing

The dataset is to be subjected to dependency parsing, a procedure used to examine the grammatical structure of a sentence and identify related terms as well as the nature of the relationships between them. This relations can also be visualized using some of the available tools in python. Utilising Stanford CoreNLP and NLTK, dependency parsing is carried out. Java-based NLP tools are available from CoreNLP. This Java library can be used with NLTK to parse dependencies in Python. Python is explicitly pointed to the location of your Java installation in this as well. On the CoreNLP website, you can get the Stanford CoreNLP zip file and Stanford CoreNLP model jar file. Again the paths to jar file and model jar file are explicitly mentioned in the code. Then the parsing of each sentence in the dataset is done which is obtained in the form of triple ((head word, head POS), relation, (dependent word, dependent POS)). Above figure is the dependency parsing pattern obtained for the sentence "The sky was as blue as the ocean".

2.4. Concept- Attribute Collocations

Collocations of ideas and their corresponding qualities in text or data are referred to as concept-attribute collocations. An attribute is a quality or feature that describes a concept, whereas a concept is a generic idea or thought. Concept attribute collocation is done with the help of dependency parsing. It is basically extraction of relations from dependency parsing patterns. Here adjective-noun and noun-verb relations are to be extracted. We have used following patterns(relations) for the same.

Noun ← Verb
Noun ← Adjective
Noun ← Verb → Adjective
Adjective ← Noun

2.5. Knowledge Graph

A knowledge graph is a sort of database that displays knowledge as a graph of nodes and the connections between them (known as edges). Entities are commonly represented as nodes in a knowledge graph, and relationships are represented as labelled edges connecting the nodes. It offers a structured and meaningful representation of knowledge that is used to query and process. We expressed metaphor as a triple consisting of a target, attribute, and source. Thus, the construction of the knowledge graph involves creating a triple metaphor consisting of a target, attribute, and source which is done using python script. A well-liked open-source graph database management system called Neo4j is used to create knowledge graphs. In our database, nodes are concepts which are basically nouns and relationships are attributes which can be adjectives or verbs. So, the target and source in the metaphor triples are concepts and described by the attribute as a relation.

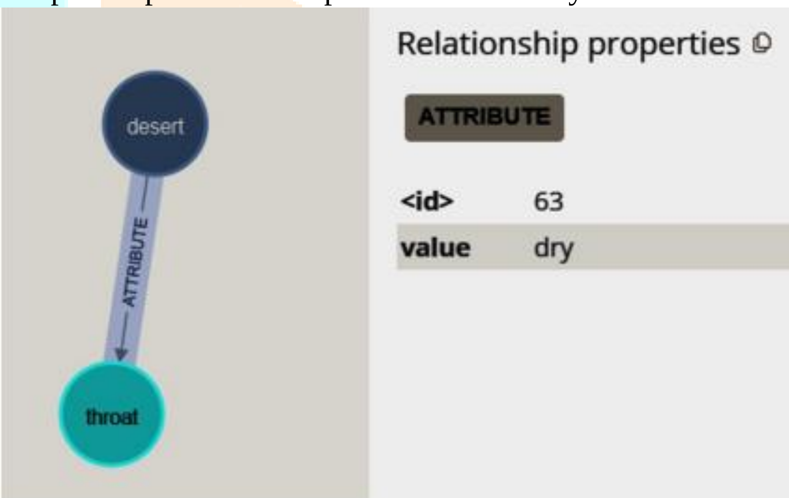

Figure 2. Knowledge Graph Triples (Throat, Desert, Dry)

2.6. Knowledge Graph Study

Queries on knowledge graph are performed in order to filter nodes and relationships and display only specific nodes and relationships. Following are some examples for the same:

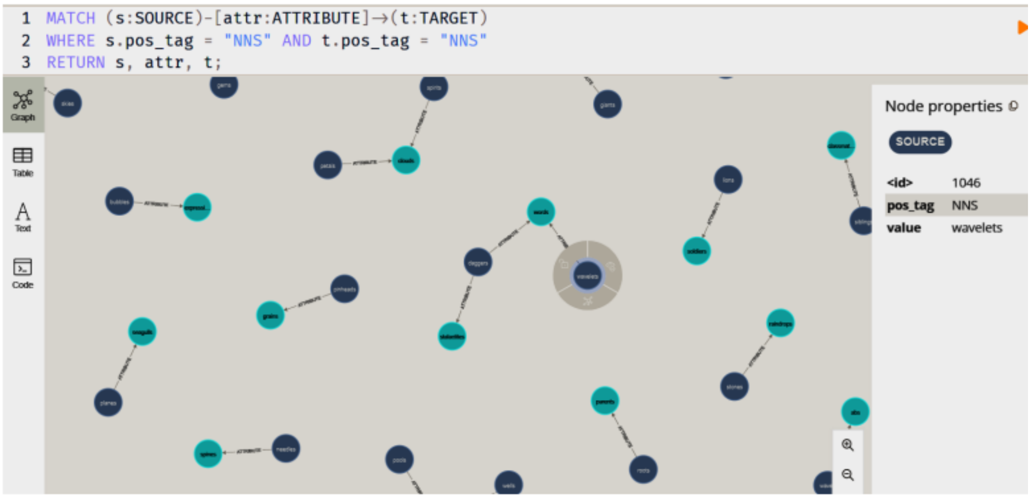

Figure 3. Knowledge Graph Study Query 1

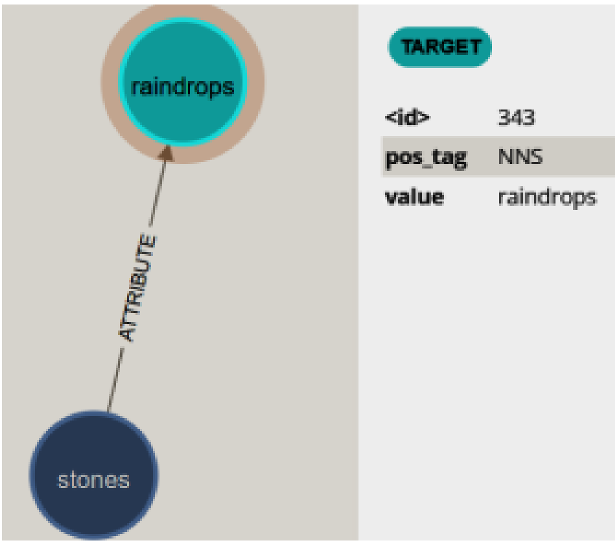

Figure 4. Knowledge Graph Query 1 Instance

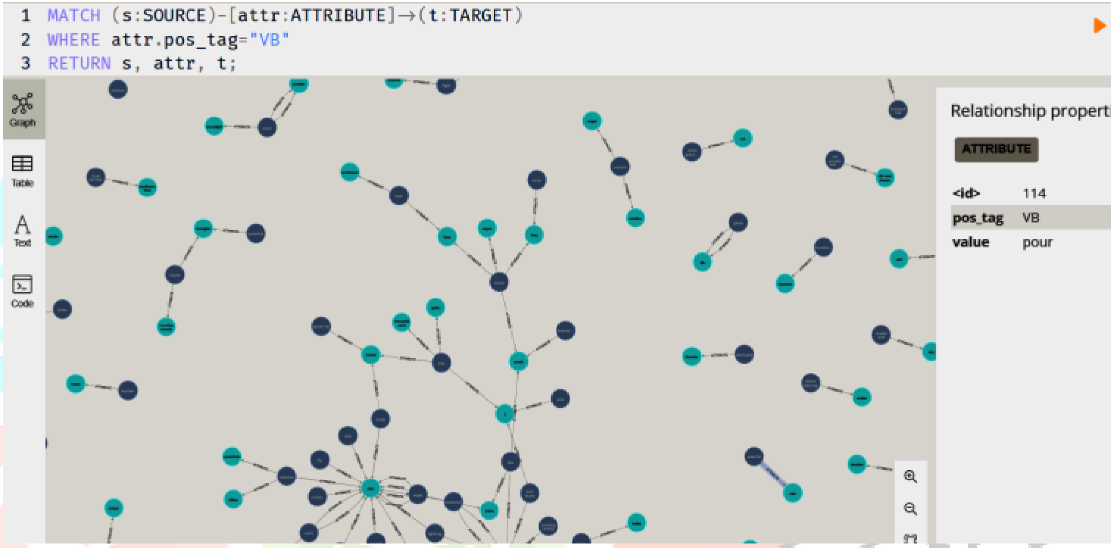

Figure 4. Knowledge Graph Study Query 2

2.7. Shared Embedding

Shared embedding, sometimes referred to as joint embedding, is a method used in machine learning and natural language processing that entails encoding numerous entities or concepts in a single embedding space. Each entity is represented as a vector in shared embedding, which places like or related entities close to one another and dissimilar ones far apart in a high-dimensional space. Using methods like neural networks, which can record intricate links and interactions between things, the embedding vectors are learned from the data.

III. Resolution

We overcome the problem of limited coverage by creating own data set of two thousand entities. To increase the accuracy we put forth a methodology for shared embedding of metaphor Knowledge Graph and Concept-Attribute collocations. Constructed knowledge graphs and extracted concept and attribute collocations are provided as input to the model. After model construction, classifier is built which detects metaphorical sentences with more accurately.

IV. Results and Discussions

4.1. Comparative Analysis of Bi-LSTM and BERT model results

Bi-LSTM model has better accuracy than BERT which is 97.7961 percent and it means that it is able to properly predict the results for the majority of the test dataset instances. An accurate model has well learned the correlations and patterns in the training data. After locating the best fit word with Context2Vec, we identify the metaphoricity of a target word with the same method, so that we can also apply it for metaphor interpretation[2].

4.2. Evaluation Metrics

The Bi-LSTM model is producing good positive predictions, as seen by its high F1-score, good recall and good precision. It means that it is effective in locating positive instances and reducing the false positive predictions.

4.3. Confusion Matrix • Out of 363 records, Bi-LSTM has made 355 true predictions and 8 incorrect predictions. It has given prediction ”yes” for 313 times, and ”No” for 50 times. Whereas the actual ”Yes” was 317, and actual ”No” was 46 times. • Out of 363 records, BERT has made 345 true predictions and 18 incorrect predictions. It has given prediction ”yes” for 308 times, and ”No” for 55 times. Whereas the actual ”Yes” was 324, and actual ”No” was 39 times.

It can be seen from confusion matrices that Bi-LSTM model is performing better than BERT model.

4.3.1. One of the key advantages of our system is that we have created own dataset with 10 times more instances. This means that our system is more accurate and robust in detecting metaphors. Because of less instances in dataset of previous work they can't able to achieve more accuracy which was overcome by our system.

4.3.2. Our system's making use of a Bi-LSTM model is also impressive. This architecture can capture long-term dependencies and handle variable length inputs, making it well-suited for detecting metaphors. Additionally, incorporating dependency parsing was help our system to identify relationships between words and phrases in a sentence, which is useful for detecting metaphors because they often involve a comparison between two seemingly unrelated things :

4.3.2.1. Another unique feature of our system is the use of acknowledge graph. By leveraging external knowledge, our system is potentially improve its ability to detect metaphors. Finally, the use of shared embeddings can help to capture the semantic relationships between words in a sentence, which is particularly useful for detecting metaphors as they often involve a shift in meaning.

4.3.2.2. In metaphorical sentences, there is a hidden relational attribute between the two concepts. Our system uses a knowledge graph query to predict all such possible attributes.

V. Conclusion

In this project, a new metaphor processing technique based on KG embedding is presented. In order to score metaphor relations and simultaneously incorporate metaphor triples and concept-attribute collocations, we present a new scoring function for metaphor relations and an unique joint model. We see a metaphor as an attribute-dependent concept mapping. Under the same framework, the primary metaphor processing activity i.e. metaphor detection might be carried out. Nominal metaphors are primarily the focus of this work. In comparison to earlier metaphor detection techniques, the suggested novel approach i.e. the joint embedding model has significantly enhanced the performance of metaphor detection. Another potential avenue for research in metaphor detection is to focus on the generation of novel metaphors, which could have applications in creative writing and marketing to convey complex ideas in a more easily digestible manner.

References

1) Yucheng Li, Shun Wang, Chenghua Lin, Frank Guerin. Metaphor detection via explicit basic meaning modelling .In Proceedings of association of computational linguistics ACL July 2023.

2)

2) Rui Mao, Chenghua Lin, and Frank Guerin. Word embedding and wordnet based metaphor identification and interpretation. In Proceedings of the 56th annual meeting of the association for computational linguistics. Association for Computational Linguistics (ACL), 2018.

3) Timour Igamberdiev and Hyopil Shin. Metaphor identification with paragraph and word vectorization: An attention-based neural approach. In Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, 2018.

4) Yulia Tsvetko, Leonid Boytsov, Anatole Gershman, Eric Nyberg, Chris Dyer. Metaphor Detection with CrossLingual Model Transfer. In Languages Technolo- gies Institute Carnegie Mellon University Pittsburgh, PA 15213 USA, June 23-25 2014.

5) Ekaterina Shutova, Lin Sun, and Anna Korhonen. Metaphor identification using verb and noun clustering. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1002 1010, 2010.

6) Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. Literal and metaphorical sense identification through concrete and abstract context. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 680–690, 2011.

7) Yair Neuman, Dan Assaf, Yohai Cohen, Mark Last, Shlomo Argamon, Newton Howard, and Ophir Frieder. Metaphor identification in large texts corpora. PloS one, 8(4): e62343, 2013

8) Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. Semantic signatures for example-based linguistic metaphor detection. In Proceedings of the First Workshop on Metaphor in NLP, pages 27–35, 2013.