Literary Device Detection

1st Ajay Anil Kumar
Computer Science and Engineering
PES University
Bengaluru, India
lonesurvivor1234@gmail.com

2nd Eswar Sai Reddy C V
Computer Science and Engineering
PES University
Bengaluru, India
es.reddycv@gmail.com

3rd Rudra Narayan Samanta
Computer Science and Engineering
PES University
Bengaluru, India
rudra.samanta9357@gmail.com

***Abstract*—This document is a exemplary and insistence for Literary Device Identification. Literary device identification is an important task in Natural Language Processing, which plays major role in performing various tasks, such as Text Classification, Text summarizing, etc. This paper proposes different methods to identify Literary Devices in sentences, namely Personification, Metaphor and Oxymoron. Personification is a literary device which is used to attribute human (animate) characteristics to something that is not human (inanimate). Metaphor is a literary device that describes something by saying it's something else. It pulls comparisons between two unrelated ideas. Oxymoron is a literary device in which two words with opposite meanings are used together to create a new meaning.**

***Index Terms*—Literary Devices Identification, Literary Device Detection, Personification, Metaphor, Oxymoron**

I. Introduction

Literary devices are techniques that allow a writer to convey a deeper meaning that extends beyond what is written on the page. They are a type of figurative language or non-literal language, which is a way of expressing oneself that does not use a word's strict or realistic meaning. Figurative language is common in comparisons and exaggerations and is typically used to add creative flourish to written or spoken language or to explain a complicated idea. For example in the sentence "He has a heart of stone" The writer is trying to express the cruel or stern nature of a character.

Figurative language is frequently used in discourse, and figurative expressions play an important role in communication. However, the area of figurative language hasn't been studied extensively in NLP, and it remains an open question to what extent modern language models can interpret nonliteral phrases. In this paper, we propose methods for recognising three literary devices: personification, metaphor, and oxymorons. These models can be used in various NLP applications like text summarization and meaning extraction, sentiment analysis and translation.

II. Related Works

A. Personification

Personification is not widely explored as other literary devices like Simile, Metaphor, etc. Because of this very less amount of work is done on personification till date. Rodas, Fernando Sánchez [8] Proposed personification can be identified by using Named Entity Recognition. Different tools like Spacy, VIP are explored to perform Named Entity Recognition. Spacy performed well compared to VIP. NER is used along with POS Tagging to identify personifications from the sentences. Keh, Sedrick Scott, et al. [5] Identified potential personifications from sentences through Dependency Parsing using Spacy. Only possible sentence segments are extracted and additional processing is required to obtain the personifications. Simon, Gábor [11] discussed the types of patterns which can be noticed in personification and mentioned different patterns in which personification occurs. These patterns are helpful in identifying the potential Personifications in given sentences and identify personifications from them.

B. Metaphor

Various attempts have been made for metaphor detection in recent years, mainly utilizing deep neural networks. The work by Do et al. [3] used only a Multilayer Perceptron with one hidden layer. Another work using deep learning architecture is done by Bizzonni et al. [12]. They compared the performances between two deep learning architectures for the task of metaphor detection. One is a bi-LSTM model and another one is modified model based on the previously mentioned work by Do et al. [3]. Swarnkar et al. [4] proposed that metaphors can be found when the contrast between a word's general meaning and its contextual meaning is found. Then they created their deep learning architecture based on that claim. Razali et al. [7] used a CNN architecture to extract features from the text and further combined these features with other emotional and cognitive features for classification.

C. Oxymoron

Identification of Oxymoron in sentences is challenging task as it's difficult to spot the words which are antonyms in a sentence. Techniques like identifying word similarity fail in identifying oxymorons as the words used in oxymorons can be replaced with each other in sentences, which makes them have similarity index same as synonyms. Cho, Won Ik, et al [2] used GloVe embeddings for identifying oxymorons. The difference between two word vectors is taken and cosine similarity is applied on different word pairs (synonyms and antonyms). The classification of oxymorons was done by setting thresholds for cosine similarity. Silveira, Natalia [10] Stated Antonym words appear in windows length of five words in the sentences. Different types of word representations like bag-of-word vectors, TF-IDF, etc. are used and declared to be non-efficient. SVM was used to identify if two words are antonyms. Batita, Mohamed Ali, and Mounir Zrigui [1] tried to identify antonyms in Arabic WordNet (AWN). Sketch Engine is used to perform the analysis. Pattern identification is done to extract potential opposites in the sentence. Word pairs are extracted from most occuring patterns and antonyms are identified from the obtained word pairs. La Pietra, Marta, and Francesca Masini [6] proposed a method of identifying oxymoron in Italian corpora. Sketch Engine was used to identify oxymorons.

III. Methodology

A. Personification

The task of identification of personification in text is divided into three different tasks, namely noun and verb pair identification, noun and word pair filtering (animate or inanimate) and inanimate word identification. We created our own dataset based on contents from multiple online sources and the dataset created by Keh, Sedrick Scott, et al. The dataset has around 800 and 1150 sample sentences of personification and non-personification sentences respectively. [1]. Spacy is used to perform dependency parsing on the sentences and potential noun and verb pairs are extracted from the sentences. Multiple set of rules are established for this process. Custom word embeddings are generated based on lemmas of nouns (person, animal, artifact, act, etc.) and verbs (motion, possession, creation, completion, etc.), sorted based on their occurrence in sentences (high to low). For the purpose of Inanimate noun detection, word embeddings of size 26 (possible noun - type appearances) are made for each word and different classification models are trained to classify words into animates and inanimates. For the purpose of Noun and Verb pair detection word embeddings of noun and verb are combined to make a single word embedding, which is used to train models to identify the possibility of a noun and verb appearing together. Multiple classification models are trained for the purpose of classification. A custom Deep Learning multi-layer Sequential model is built to perform the classification tasks. All models are combined to create an ensemble model which classifies the input based on the outputs of all models (voting is used to decide the final response). Potential noun and verb pairs are identified from sentences which are then filtered out based on the chance of them appearing together and the pairs with inanimate nouns are removed to finally obtain personifications from input data.

B. Metaphor

The dataset used for identification of metaphors is the MOH-X dataset [5] There are a total of 647 instances in this dataset out of which 49 percent are metaphorical. The data is preprocessed to remove stopwords and convert the texts into lowercase using the NLTK library. Then each word in a sentence is converted to their correspondind fasttext embeddings which has a embedding size of 300. A pre-trained fasttext embedding is used as fasttext returns vectors even for out of vocabulary words. We find that the maximum number of words in an instance is 8, so all the other sentences are padded with 0 vectors to increase their length to 8. CNN has proved to be excellent for sentence classification tasks and NLP tasks in general, following the work of Yoon Kim [5]. A CNN architecture with 3 convolution layers with filters (128 64, 32) are used to extract 32 features from each sentence. The input layer is of shape (None,8,300) where 8 is the number of words in each sentence and 300 is the embedding size of each word. The kernel size is set to 1. ReLu Activation is applied to each of these layers convert all negative vectors to zero. After each convolution layer, a 0.2 dropout layer is added to prevent overfitting. Finally a Maxpooling layer is added at the end along with a fully connected layer with 32 neurons. Finally the features extracted from this CNN architecture are fed into an SVM model for classification.

C. Oxymoron

Oxymoron identification is done by using pre-trained word embeddings. Two different types of word embeddings, FastText word embeddings and different type of GloVe word embeddings are used to perform this classification. The dataset used for this classification is collected from multiple sources including general oxymoron examples, synonyms and antonyms. Final dataset contains 7448 synonym pairs and 7448 antonym pairs. Word embeddings for the words in synonym and antonym word pairs are taken from FastText and GloVe embeddings and difference between word embeddings is computed. The difference in word embeddings along with label (0 for synonym and 1 for antonym) are passed to classification models to train and classify the words into synonyms and antonyms. All word combinations are taken from given sentence and the word combinations which have both positive or both negative type of words are removed as these words can't be opposites (because if one of the words is positive the other word should be negative for the words to be antonyms). Duplicates are removed from set of pairs and synonym-antonym classification is done on all pairs in the sentence. Different models are trained to classify if two words are synonyms or antonyms. The length of word embeddings ranges from 25 to 300. An ensemble model is made where outcomes of all models are taken and most recurring outcome is categorized as final output. All models are trained with all available word embeddings.

IV. Results

A. Personification

Table 1 shows the results obtained from different approaches for the task 'identification of personification in sentences'. We can see that classifying noun and verb pair together gave more accuracy than noun and verb pair combined with inanimate nouns and using inanimate noun only gave very bad results. so, based on the observations we can state that taking noun and verb pair and using the combined vector gives best results for the task - Personification identification.

Different BERT models are trained with different types of inputs - total sentence, Noun and Verb together forming mini

TABLE I
PERSONIFICATION - VALIDATION METRICS

Classification Method	Evaluation Metrics			
	Accuracy	Precision	Recall	F1-Score
Inanimate only	0.539	0.452	0.719	0.555
Noun-Verb pair only	0.853	0.903	0.708	0.794
Both	0.793	0.898	0.543	0.677

sentence, Noun only, Verb only. All models didn't perform well in terms of personification detection so they're not considered.

B. Metaphor

We see in Table II that the CNN model outperforms other deep neural networks in identification of metaphors.

TABLE II
METAPHOR - COMPARING RESULTS OBTAINED WITH PREVIOUS WORK

Author Name	Evaluation Metrics		
	Precision	*Recall*	*F1*
Bizzoni et al.	.595	.680	.635
Swarnkar et al.	.529	.708	.605
Ours	.742	.722	.732

C. Oxymoron

Oxymoron detection results are shown in Table 3. Top 3 Models trained with top 4 Embeddings are shown in the tables along with their Evaluation Metrics. From Table 3 we can conclude that best result is obtained when Glove 42B 300D embedding is used with SVC Model. The accuracy of the models is more than the accuracy obtained in the paper which used Word Embedding.

TABLE III
OXYMORON - VALIDATION METRICS — NOTE: RF - RANDOM FOREST

Word Embeddings and Model	Evaluation Metrics			
	Precision	Recall	accuracy	F1
glove 42B 300d - RF	0.805	0.675	0.756	0.734
glove 42B 300d - SVC Poly	0.766	0.630	0.720	0.692
glove 42B 300d - SVC	0.828	0.766	0.804	0.796
glove 6B 200d - RF	0.779	0.641	0.732	0.703
glove 6B 200d - SVC Poly	0.709	0.642	0.692	0.674
glove 6B 200d - SVC	0.793	0.745	0.777	0.768
glove 6B 300d - RF	0.757	0.647	0.727	0.697
glove 6B 300d - SVC Poly	0.740	0.644	0.717	0.689
glove 6B 300d - SVC	0.805	0.725	0.781	0.763
glove 42B 300d - RF	0.805	0.675	0.756	0.734
glove 840B 300d - RF	0.829	0.672	0.768	0.742
glove 840B 300d - SVC Poly	0.766	0.594	0.709	0.669
glove 42B 300d - SVC	0.848	0.757	0.812	0.800
FastText - RF	0.792	0.674	0.750	0.728
FastText - SVC Poly	0.547	0.913	0.582	0.684
FastText - SVC	0.740	0.780	0.755	0.759

V. CONCLUSION AND FUTURE WORK

In this paper different methodologies for identifying literary devices are proposed and decent results are obtained. The main limitation which we faced is lack of data. There is lot of room to explore and try different approaches for same tasks. Detailed description for each type of literary device is mentioned below.

A. Personification

The proposed method to identify Personification gave decent results with very less number of false positives, which is a good thing. It is observed that the current approach doesn't consider the scenarios where the noun is addressed by another word, like pronoun in a sentence. To overcome this NER should be performed and places where a noun is being referred indirectly should be considered while using pattern extraction for the data. This will help to cover more sentences during Personification detection. It is observed that non-personifications are detected better than personification. This implies that the model needs more fine tuning and this can be done by changing the length of word embeddings and configuring the types of lemmas which are considered for classification.

B. Metaphor

The proposed method is shown to significantly improve the results of other models that use only deep learning. CNN has shown to be a powerful method to extract features from small amount of data. For future work, separate features can be extracted from the data, such as word similarity and topic similarity. These features could further increase accuracy of metaphor detection. Future work also includes exploring the contextual feature of each sentence to better understand the metaphors.

C. Oxymoron

The proposed method is shown to give better results than the papers which used word embeddings before (81 percent vs 66 percent). It is observed that the more the length of word embedding the better the results. From this we can understand that only few columns of word embeddings play major role towards the oxymoron detection. Experimenting with different subsets of word embeddings and checking for correlation between word embeddings and prediction class may improve the model's performance.

REFERENCES

[1] Batita, Mohamed Ali, and Mounir Zrigui. "The enrichment of arabic wordnet antonym relations." Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Revised Selected Papers, Part I 18. Springer International Publishing, 2018.

[2] Cho, Won Ik, et al. "Detecting oxymoron in a single statement." 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA). IEEE, 2017.

[3] E.-L. Do Dinh and I. Gurevych, "Token-level metaphor detection using neural networks," in Proceedings of the Fourth Workshop on Metaphor in NLP, 2016, pp. 28–33.

[4] K. Swarnkar, A. K. Singh, Di-lstm contrast: A deep neural network for metaphor detection, in: Proceedings of the Workshop on Figurative Language Processing, 2018, pp. 115–120.
[5] Keh, Sedrick Scott, et al. ”PINEAPPLE: Personifying INanimate Entities by Acquiring Parallel Personification data for Learning Enhanced generation.” arXiv preprint arXiv:2209.07752 (2022).
[6] La Pietra, Marta, and Francesca Masini. ”Oxymorons: a preliminary corpus investigation.” Proceedings of the Second Workshop on Figurative Language Processing. 2020.
[7] M. S. Razali, A. A. Halin, Y. -W. Chow, N. M. Norowi and S. Doraisamy, ”Deep Learning Metaphor Detection with Emotion-Cognition Association,” 2022 International Conference on Digital Transformation and Intelligence (ICDI), Kuching, Sarawak, Malaysia, 2022, pp. 8-14, doi: 10.1109/ICDI57181.2022.10007398.
[8] Rodas, Fernando Sánchez. ”Using Named Entity Recognition to Identify Personification Constructions in an English¡¿ Spanish Intermodal Corpus of the EP Committee on Petitions.” (2020).
[9] S. Mohammad, E. Shutova, P. Turney, Metaphor as a medium for emotion: An empirical study, in: Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, 2016, pp. 23–33.
[10] Silveira, Natalia. ”Learning to identify antonyms.” URL: http://cs229.stanford.edu/proj2013/Silveira-LearningToIdentifyAntonymy.pdf (: 05.02.2019).
[11] Simon, Gábor. ”Identification and Analysis of Personification in Hungarian.” (2022).
[12] Y. Bizzoni, M. Ghanimifard, Bigrams and bilstms two neural networks for sequential metaphor detection, in: Proceedings of the Workshop on Figurative Language Processing, 2018, pp. 91–101.