Sarcasm Identification and Detection in Conversion Context using BERT

Kalaivani A, Thenmozhi D
Department of Computer Science and Engineering,
SSN College of Engineering (Autonomous),
Affiliated to Anna University, Tamilnadu, India.
kalaiwind@gmail.com, theni_d@ssn.edu.in

Abstract

Sarcasm analysis in user conversion text is automatic detection of any irony, insult, hurting, painful, caustic, humour, vulgarity that degrades an individual. It is helpful in the field of sentimental analysis and cyberbullying. As an immense growth of social media, sarcasm analysis helps to avoid insult, hurts and humour to affect someone. In this paper, we present traditional Machine learning approaches, Deep learning approach (RNN-LSTM) and BERT (Bidirectional Encoder Representations from Transformers) for identifying sarcasm. We have used the approaches to build the model, to identify and categorize how much conversion context or response is needed for sarcasm detection and evaluated on the two social media forums that is Twitter conversation dataset and Reddit conversion dataset. We compare the performance based on the approaches and obtained the best F1 scores as 0.722, 0.679 for the Twitter forums and Reddit forums respectively.

1 Introduction

Social media have shown a rapid growth of user counts and have been object of scientific and sentiment analysis as in (Kalaivani A and Thenmozhi D, 2018). Sarcasm occurs frequently in user-generated content such as blogs, forums and micro posts, especially in English, and is inherently difficult to analyze, not only for a machine but even for a human. Sarcasm Analysis is useful for several applications such as sentimental analysis, opinion mining, hate speech identification, offensive and abusive language detection, advertising and cyber bullying.

(Debanjan Ghosh et al., 2018) performed to identify how much context is needed to find the conversion context is sarcastic or not and analysed the verbal irony tweets using LSTM with more different attention mechanism and still facing the problem with the usage of slangs, rhetorical questions, usage of numbers and usage of non-vocabulary tweets. In recent years, several research works are performed in sarcasm detection in the Natural Language Processing community (Aditya Joshi at el., 2017).

In Figurative Language 2020 Task 2: shared task on sarcasm detection in social media forums. It focuses to identify the given conversion text is sarcastic or not and find how much context is helpful for sarcasm identification have modelled either the given instance may be isolated or combined. It focuses on two social media forums that are Twitter conversion dataset and Reddit conversion dataset (Khodak et al., 2017). For both the datasets, the organizer provides the context and response that is the response is reply to the context and the context is a full dialogue conversation thread. The computational task is to detect and identify the sarcasm and to understand how much conversation context is needed or helpful for sarcasm detection.

The challenges of this shared task include: a) small dataset is hard to train the complex models; b) the characteristics of the language on social media forums difficulties such as non-vocabulary words and ungrammatical context c) how much conversion text to detect sarcasm and the usage of slangs, rhetorical questions, Capitalized words, numbers, Abbreviations, pro-longed words, hashtags, URL, Repetitions of Punctuations, Contractions, Continuous words without spaces.

We address the problem in hash tags, continuation of words without spaces, URL and to classify which context is helpful to find sarcasm. To address the problem, we pre-processed the text by using Machine learning libraries like NTLK, Gensim and classified by using different traditional machine learning techniques, deep learning technique and finally we obtained the

Proceedings of the Second Workshop on Figurative Language Processing, pages 72–76

https://doi.org/10.18653/v1/P17

best result by using BERT models. The tasks are independently evaluated by macro-F1 metrics.

2 Related Work

(Aniruddha Ghosh and Tony Veale, 2016) used neural network semantic model to capture the temporal text patterns for shorter texts. As an example, in this model classified "I Just Love Mondays!" correctly as sarcasm, but it failed to classify "Thank God It's Monday!" as sarcasm, even though both are similar at the conceptual level. (Keith Cortis et al., 2017) performed in the SemEval-2017 shared task to detect the sentiment, humour and to predict the sentiment score of companies' stocks in the smaller texts.

(Raj Kumar Gupta and Yinping Yang, 2017) performed in the shared task of SemEval-2017 Task 4 to detect sarcasm by used the SVM Based classifier and developed the CrystalNest to analyse the features combining sarcasm score derived, sentiment scores, NRC lexicon, n-grams, word embedding vectors, and part-of-speech features.

(David Bamman and Noah A. Smith, 2015) used the predictive features and analysed the utterance on Twitter based on the properties of author, audience and environment features. (Mondher Bouazizi and Tomoaki Otsuki, 2016) used the pattern-based approach to detect sarcasm and analysed the four features such as sentiment-related features, punctuation-related features, syntactic and semantic features, pattern-related features and classification done by the classifiers such as Random Forest, Support Vector Machine, k Near-est Neighbours and Maximum Entropy.

(Meishan Zhang et al., 2016) used the bi-directional gated recurrent neural network and discrete model to detect sarcasm and analyse the local and conceptual information and perform the process in Glove word embedding. (Malave N et al., 2020) used the context-based evaluation based on the data and to determine the user behaviour and context information to detect sarcasm. (Yitao Cai et al., 2019) used the multi-modal hierarchical fusion model to detect the multi-modal sarcasm for tweets consisting of texts and images in Twitter.

3 Data and Methodology

In our approach, we have used Twitter and Reddit dataset given by Figurative Language processing 2020 shared task on sarcasm detection. The dataset is given with columns namely, label, context and response where the response is the reply of context and the context is the full conversion dialogue and it is separated as C1, C2, C3 etc. C2 is the reply of the C1 context and C3 is the reply of C2 context respectively. Both the datasets consists of the labels namely SARCASM and NOT_SARCASM. In the Twitter dataset, the train data has 5000 conversion tweets in that 2500 sarcasm tweets and 2500 not sarcasm tweets and the test data has 1800 tweets.

In the Reddit dataset, the train data has 4400 conversion tweets in that 2200 sarcasm tweets and 2200 non sarcasm tweets and the test data have 1800 tweets. we have the pre-processed the text to removal of @USER, URL and the pro longed words like "ohhhhhh" and replace the words like F * * king as Fucking, replace the question tags like Didn't as Did not, removal of hashtags and separate the words into the continuous space less sentence. Tweet tokenizer is used to tokenize the word and to get the vocabulary words.

Figure 1: Sarcastic words

Figure 2: Not Sarcastic Words

We have employed the traditional machine learning techniques, Recurrent Neural Network with LSTM (RNN-LSTM) and BERT. In the

machine learning approach, first, we have used the utterance of combined context and response (CR) for detecting the sarcasm and then pre-processed data using Gensim libraries to remove the hashtags, punctuation, white spaces, numeric content, stop words and then convert into lower text. We have used the word cloud to identify and categorize the most sarcastic words and non-sarcastic words which are appeared in sarcasm message and not sarcasm message as shown below in Figure 1 and Figure 2.

Models	Combined Context and Response (CR)		Response (R)	
	Doc2Vec	Tfidf	Doc2Vec	Tfidf
LR	0.513	**0.7296**	0.509	**0.7132**
RF	0.513	0.6764	0.527	0.7038
XGB	0.534	0.6876	0.533	0.6928
SVC	0.507	**0.7212**	0.506	0.7016
NB	0.505	**0.7394**	0.512	**0.7106**

Table 1: Accuracies of the models based on the feature extraction of the utterance of combined and isolated text – Twitter data

Models	Combined Context and Response (CR)		Response(R)	
	Doc2Vec	Tfidf	Doc2Vec	Tfidf
LR	0.5061	**0.552**	0.497	**0.597**
RF	0.4947	0.539	0.505	0.564
XGB	0.4965	**0.565**	0.500	0.582
SVC	0.5029	0.538	0.493	0.587
NB	0.4977	0.549	0.493	**0.595**

Table 2: Accuracies of the models based on the feature extraction of the utterance of combined and isolated text – Reddit data

We have performed Doc2Vec transformer and Tfidf Vectorizer for feature extraction and classified by using the Logistic Regression (LR), Random Forest Classifier (RF), XGBoost Classifier (XGB), Linear Support vector machine (SVC), Gaussian Naïve Binomial (NB). By using Tfidf Vectorizer, we got the 28761 features for 5000 tweets. Table 1 presents the cross validation accuracies of the different machine learning classifiers in the Twitter data as mentioned above. Table 2 presents the cross validation accuracies of the models based on the feature extraction in the Reddit data.

In Twitter data, we have chosen the scores which are above 0.70 from the cross validation accuracies of the machine learning techniques. Based on the cross validation scores, we have obtain the best accuracies score in SVM, logistic regression and NB classifiers of the combined context text (CR) in Tfidf vectorizer and the best accuracies score in Logistic regression and Gaussian NB models of the isolated response (R) text in Tfidf vectorizer. In Reddit data, we have chosen the scores which are above 0.55 from the cross validation accuracies of the machine learning techniques. Based on the cross validation scores, we have obtain the best accuracies score in logistic regression and XGBoost Classifier of the combined text (CR) in Tfidf vectorizer and the best accuracies score in Logistic regression and Gaussian NB models of the isolated response text (R) in Tfidf vectorizer. In both the dataset, the result shows Doc2Vec transformer is not performed well because of non-grammatical sentences and Tfidf Vectorizer performs well when compared with the Doc2Vec transformer in dialogue conversion thread.

In the RNN-LSTM Method, we have used the combined context text with response to perform the pre-process using NLTK libraries, tokenize the word by using the word tokenizer and lemmatize the word after that to remove the stop words. Finally, we have obtained the train data has 325382 words total, with a vocabulary size of 32756, max sentence length is 568 and the test data has 30782 words total, with a vocabulary size of 8824, Max sentence length is 467. We used the Word2Vec embedding model for the embedding the words and obtain the 32668 unique tokens. We have evaluated using the RNN-LSTM and trained the deep learning models with a batch size 128 and dropout 0.2 for 5 epochs to build the model. We got the accuracy is 0.4890 which is low when compared with the machine learning approach.

In the BERT model, Google research team releases BERT (Devlin et al., 2018) and achieve good performance on many NLP tasks. We have used the combined context text, isolated context, and isolated response to perform the model. We have used the Bert uncased model for training the model, batch size is 32, learning rate is 2e-5, and number of train epochs is 3.0. Warmup is a period of time where the learning rate is small and gradually increases usually helps training. Warmup proportion is 0.1 and the model configuration is checkpoints is 300, summary steps is 100. We got the accuracy is 0.77 score. We have compared over all cross validation accuracies scores, BERT performs good than the machine learning approaches and deep learning technique.

Type	Precision	Recall	F1 score
BERT(CR)	0.672	0.673	0.671
BERT(C)	0.695	0.701	0.693
BERT(PCRW)	0.704	0.705	0.703
BERT(PCW)	0.703	0.703	0.703
BERT(PC1RW)	0.677	0.678	0.677
BERT(PC1W)	0.689	0.690	0.689
RNN-LSTM(CR)	0.361	0.361	0.361
BERT(R)	**0.722**	**0.722**	**0.722**
BERT(PC2R)	0.658	0.685	0.645
BERT(PR)	0.706	0.706	0.706
SVM(CR)	0.646	0.647	0.646
NB(CR)	0.672	0.672	0.672
NB(R)	0.632	0.632	0.632
LR(R)	0.642	0.643	0.642

Table 3: Results for Twitter Dataset

4 Results

We have evaluated the test data of Twitter and Reddit dataset which is shared by Figurative Language processing 2020 shared task organizers. The performance is evaluated by using the metrics as precision, recall and F1 score. We have chosen the classifiers to predict the test data based on the performance of the cross validation of training data. We have performed to predict the test data by using various combinations of Conversion context and response that are CR represents the combined context of sentences with response, C represents the combined full context of sentences without response, PCRW represents the processed combined context of meaningful words and response, PCW represents the combined full context of meaningful words without response,

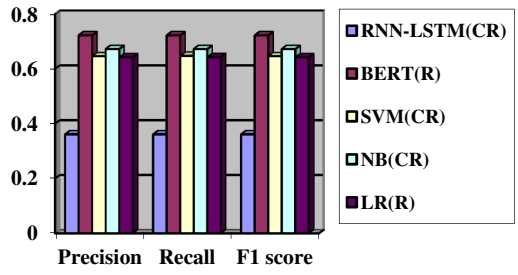

Figure 3: Results analysis for Twitter Dataset

Type	Precision	Recall	F1 score
BERT(C)	0.587	0.589	0.585
BERT(CR)	0.493	0.492	0.477
BERT(R)	**0.679**	**0.679**	**0.679**
BERT(PR)	0.638	0.638	0.637
LR(CR)	0.526	0.526	0.526
LR(R)	0.563	0.564	0.563
NB(R)	0.557	0.557	0.557
SVC(R)	0.551	0.551	0.550
XGB(R)	0.539	0.543	0.528
SVC(CR)	0.516	0.516	0.516
XGB(CR)	0.544	0.544	0.544

Table 4: Results for Reddit Dataset

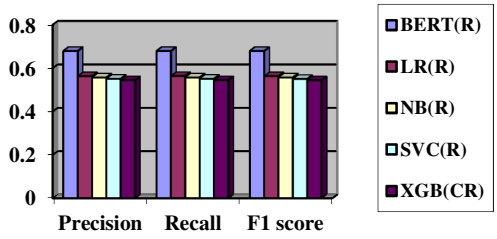

Figure 4: Results analysis for Reddit Dataset

PC1RW represents the processed isolated first context of meaningful words and response, PC1W represents the isolated first context of meaningful words without response, R represents the response, PC1R represents the processed second context with response, PR represents the processed response. The results of the approaches are presented in the Table 3 shows the response text from conversion dialogue by using BERT have higher performance than others for the shared task of the Twitter dataset and the Table 4 shows BERT response text from conversion dialogue thread performs well for the shared task of the Reddit dataset. The best results have obtained by using BERT model with the isolated response(R) text for both the Twitter and Reddit dataset respectively. We have noticed that the BERT performs well in continuous conversion dialogues or continuous sentences with previous dialogues compared with the meaningful words from conversion context. In both the dataset, the RNN-LSTM performs poor than the SVM, NB and LR because of the smaller dataset. The machine learning approach performs better with the smaller dataset. But the BERT model performs

well for the response text of both the Twitter and Reddit dataset with the non-grammatical sentences even the data size is small. Figure 3 shows the chart representations of the performance analysis of the different methods in the Twitter data. Figure 4 shows the chart representations of the performance analysis of the different methods in the Reddit data.

5 Conclusion

We have implemented traditional machine learning, deep learning approach and BERT model for identifying the sarcasm from Conversion dialogue thread and to detecting sarcasm from social media. The approaches are evaluated on Figurative Language 2020 dataset. The given utterance of combined text and isolated text are preprocessed and vectorized using word embeddings in deep learning models. We have employed RNN-LSTM to build the model for both the datasets. The instances are vectorized using Doc2Vec and TFIDF score for traditional machine learning models. The classifiers namely Logistic Regression (LR), Random Forest Classifier (RF), XGBoost Classifier (XGB), Linear Support vector machine (SVC), Gaussian Naïve Binomial (NB) were employed to build the models for both the Twitter and Reddit datasets. BERT uncased model with isolated response context gives better results for both the datasets respectively. The performance may be improved further by using larger datasets.

References

Joshi, A., Bhattacharyya, P., and Carman, M. J. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5), 73.

Ghosh, D., Fabbri, A. R., and Muresan, S. 2018. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4), 755-792.

Khodak, M., Saunshi, N., and Vodrahalli, K. 2017. A large self-annotated corpus for sarcasm. arXiv preprint arXiv:1704.05579.

Aniruddha Ghosh, and Tony Veale. 2016. Fracking Sarcasm using Neural Network", *research gate publication*, Conference Paper. DOI: 10.13140/RG.2.2.16560.15363.

Keith Cortis, Andre Freitas, Tobias Daudert, Manuela Hurlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News", *Proceedings of the 11th International Workshop on Semantic Evaluations* , pages 519–535, Association for Computational Linguistics.

Raj Kumar Gupta, and Yinping Yang. 2017. CrystalNest at SemEval-2017 Task 4: Using Sarcasm Detection for Enhancing Sentiment Classification and Quantification, ACM.

David Bamman and Noah A. Smith. 2016. Contextualized Sarcasm Detection on Twitter, *Association for the Advancement of Artificial Intelligence (www.aaai.org)*.

Mondher Bouazizi And Tomoaki Otsuki (Ohtsuki),. 2016. A Pattern-Based Approach for Sarcasm Detection on Twitter, *IEEE. Translations and content mining*, Digital Object Identifier 10.1109/ACCESS.2016.2594194

Kalaivani A and Thenmozhi D. 2019. Sentimental Analysis using Deep Learning Techniques, *International journal of recent technology and engineering*, ISSN: 2277-3878.

Meishan Zhang, Yue Zhang, and Guohong Fu,. 2016. Tweet Sarcasm Detection Using Deep Neural Network, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460.

Malave N., and Dhage S.N. 2020. Sarcasm Detection on Twitter: User Behavior Approach. In: Thampi S. et al. (eds) Intelligent Systems, Technologies and Applications. *Advances in Intelligent Systems and Computing, vol 910. Springer, Singapore.* DOI https://doi.org/10.1007/978-981-13-6095-4_5.

Yitao Cai, Huiyu Cai and Xiaojun Wan. 2019. Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515 Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.