

Sarcasm Detection Methods in Deep Learning: Literature Review



Shruti Kulkarni and Aparna Biswas

Abstract Sarcasm is rapidly becoming prevalent in all forms of communication today. Each person has a different way of exploiting and understanding sarcasm. It is difficult for even humans to interpret sarcastic texts, as it depends on a lot of things like perspective, context and tone. This makes it a challenging task to train a machine to distinguish sarcastic text from non-sarcastic text. As there are no concrete rules upon which a model to detect sarcasm can be built, we must resort to promising and upcoming techniques to do so. In this paper, we have reviewed the works done in sarcasm detection using deep learning in combination with the natural language processing techniques. We have also proposed an outline of our own system based on current gaps in literature and challenges in the field.

Keywords Sarcasm · Neural network · NLP · Deep learning

1 Introduction

With the latest advances into artificial intelligence, there is an increasing need to widen the boundaries of natural language processing tasks. Sentiment analysis is one of the ways in which machines can achieve better understanding of human speech and make sense out of it. Human language is a symbolic categorical system while we need to train a machine to think like a brain, which requires continuous patterns of activation. Sarcasm detection is an area of NLP which many researchers are trying to find a solution to. Till date, most approaches to sarcasm have treated the task primarily as a text categorization problem. Sarcasm, however, can be expressed in very subtle ways and requires a deeper understanding of human language that standard text categorization techniques cannot grasp.

S. Kulkarni (✉) · A. Biswas
College of Engineering, Pune, India
e-mail: shrutikul22@gmail.com

A. Biswas
e-mail: aparna.comp@coep.ac.in

In this paper, we have reviewed works dealing with sarcasm detection, concentrating on methods including neural network models and other deep learning techniques. The paper has been arranged as follows.

2 Datasets

There is a surge of data volume due to increase in the usage of social media. This provides us with authentic user-generated data in unstructured format. There are APIs that facilitate smooth scraping of data from Twitter like tweep [1] and Twitter API [2].

For Twitter-based datasets, two approaches to obtain annotations have been used. The first is manual annotation. Ellen Riloff [3] introduced a dataset of tweets which were manually annotated as either sarcastic or not. Maynard and Greenwood [4] conducted a careful study of sentiment classification and used sarcastic tweets as an impact on the aforementioned classification. The second technique to create datasets is the use of hashtag-based supervision. This method is more popular than manually annotating the tweets, as it relies on user perspective more than human identification of sentiment. By providing a hashtag, the author of the tweet is giving additional information that aides in determining the labelling of dataset. There is also the factor of more automation in this category, as it is easier to classify text as sarcastic or non-sarcastic on the basis of well-established sarcasm-indicative hashtags such as #sarcasm, #not, #lol etc. [5] have used this technique to supervise tweets. They have employed a method which ignores such hashtags at the beginning of the tweets, and only include descriptive trailing hashtags for their segregation. For example, “#sarcastic behavior is a form of passive-aggression” will not be labelled as sarcastic.

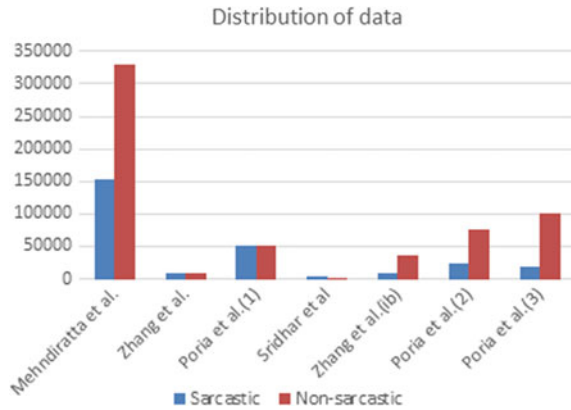
In case of Reddit-based datasets like SARC [6], there is an inbuilt provision of adding a ‘/s’ at the end of a sarcastic post. This enables the author to provide annotations. So, the labelling of this dataset was largely based on this self-annotated method.

The data from the following figure gives the statistics regarding the amount of data used to train and test the models as reported in previous works. Here, we can see that even though results are satisfactory, the data used may or may not be sufficient to generalize those results on any other model.

3 Approaches

Zhang et al. [2] used a bi-directional gated recurrent neural network for capturing the syntactic and semantic information and to extract contextual features automatically from the tweets and used a pooling neural network. They obtained an accuracy of 87.25% with an F1-score of 77.37 on unbalanced dataset. These results changed after

Fig. 1 Data distribution survey



the dataset used was balanced with an accuracy of 79.29% with F1-score of 79.36. This actively illustrates that deep learning models benefit the performance.

Poria et al. [7] have applied pre-trained convolutional neural network models for sentiment, emotion and user personality on three different datasets. These datasets are balanced, imbalanced and heavily imbalanced respectively, with figures for the number of tweets provided in Fig. 1 for each dataset. As F1-score is the metric used for natural language processing tasks, results have been evaluated using F1-score for each dataset. They have applied the models both individually and as a combination, with the combined model giving the highest F1-score of 90.34. The third dataset also provided with a high score of 93.30, but there is a possibility of overfitting in such a scenario because dataset is highly imbalanced.

Mehndiratta et al. [1] have used a deep convolutional neural network on Twitter dataset with approximately 1000–10,000 iterations of single hidden layer CNN. Their efficiency of their algorithm increased with the number of iterations, reaching a maximum of 89.9 from an average of 81.90 at the highest point. The salient feature of their work is that they have removed certain drawbacks of using Twitter data using text processing algorithms.

Lakshmanan and Anjana [8] have used different machine learning algorithms like Random Forest, Support Vector Machine, Naïve Bayes and Simple CART on Reddit dataset. They have filtered subreddits/pol (politics),/sarcasm for minimizing the data volume. Amongst all the algorithms, simple CART showed best results for detection for sarcasm at an accuracy of 66%.

Table 1 provides a survey of important works in sarcasm detection using deep learning and some baseline methods.

Table 1 Survey of previous works

Authors	Year	Dataset	Approach	Accuracy
Mehndiratta et al.	2017	Twitter	Deep CNN	81.9
Zhang et al.	2016	Twitter	Bi-directional gated Recurrent NN	87.25 F: 77.37
Poria et al.	2017	Twitter	Deep CNN, Pre-trained emotion, sentiment, personality models	F: 90.34
Lakshamanan et al.	2018	Reddit	RF, SVM, NB, Simple CART	66 (max)
Sridhar et al.	2016	Facebook	Hybrid model	82
Zhang et al. (balanced)	2016	Twitter	Bi-directional gated Recurrent NN	79.29 F: 79.36

4 Challenges

Deep learning is a breakthrough technology which has given us state-of-the-art systems. The systems designed using deep learning models provide better results than the previous works including baseline models. The work in deep learning has been made possible due to the integration of faster and stronger processors in the personal computers, adding to the convenience of the average researcher to explore this field. In sarcasm detection too, there is an upsurge in performances of systems using neural network models as a base. A lot of work has been done using commonly available datasets, with more concentration on compiling specific systems which give higher accuracy. However, there is no significant work with the use of context and user modelling which gives us an important research gap. There is also a lack of a generalized model which works for multiple datasets having different schema and post styles.

To sum up everything that has been stated, we have summarized some challenges which the areas in which future work are also can be done:

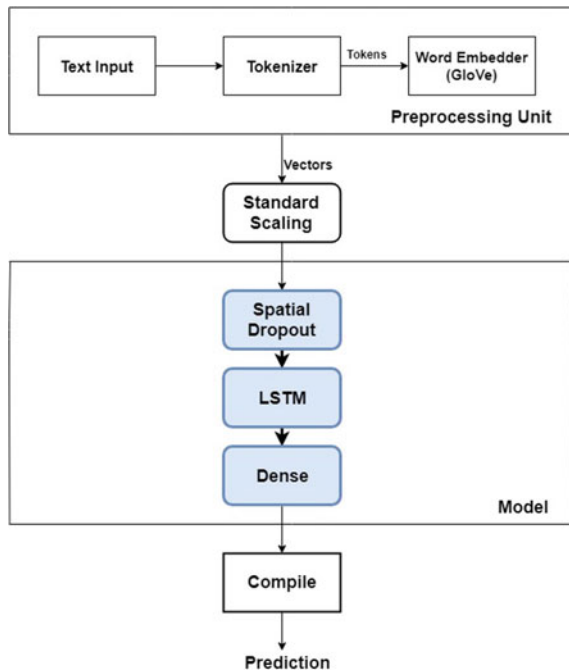
- (1) Larger corpora: Until recently, there was a lack of a standardized dataset which was large and suitable for training a deep learning model effectively. As a result, the extent of work in this area is limited.
- (2) Use of contextual information: The inclusion of context embeddings in any language processing task has given better quality of solutions. According to Silvio et al. [9], we can see that the embeddings can be exploited and applied.
- (3) Generalized model: According to research, there is a saturation in the performance of any algorithm as datasets get bigger than a certain optimized amount. As larger models tend to overfit and smaller models underfit, there is a need to find a balance between the two by mixing datasets, varying the complexity of models and trial-and-error of the former two as a combination.
- (4) User profiling as a feature: Identifying sarcasm in text is a challenging task for humans, making it even more so for machines. Hence, user perspective and history, which form a profile, are helpful in sarcasm detection.

5 Proposed System

Upon analyzing the trends and studying the gaps in the current research, we found that working on large volumes of data requires expertise and specialized algorithms to handle it. This is due to the fact that the main memory, where the data is to be stored, is expensive and hence limited in machine. So, to make our systems cost-effective, we need to perform specialized algorithms on our data which have the capability to use the available memory effectively without losing the results obtained. There is very limited work done in the field of sarcasm detection where large amount of training data has been used. Also, there is a need of a more generalized system to predict sarcasm in long as well as short text posts. While defining our problem statement, we have focused on these two issues.

Our proposed system will be based on the following problem definition: Using sequential deep learning models to effectively detect sarcasm, combined with natural language processing concepts. We are putting forth the idea of a system which includes the GloVe project for word embeddings, recurrent neural network with LSTM. Figure 2 shows the block diagram of our proposed system.

Fig. 2 Block diagram of proposed system



References

1. Mehndiratta P, Sachdeva S, Soni D (2017) Detection of sarcasm in text data using deep convolutional neural networks. *Scalable Comput Pract Exper* 18(3):219–228
2. Zhang M, Zhang Y, Fu G (2016) Tweet sarcasm detection using deep neural network. In: International conference on computational linguistics: Technical Papers COLING, Osaka, Japan, pp 2449–2460
3. Riloff E, Qadir A (2013) Sarcasm as contrast between positive sentiment and negative situation. *EMNLP*, pp 704–714
4. Maynard DG, Greenwood MA (2014) Who cares about sarcastic Tweets? Investigating the impact of sarcasm on sentiment analysis. In: *LREC*, pp 4238–4243
5. Kingma Da (2014) A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
6. Khodak M, Saunshi N, Vodrahalli K (2018) A large self-annotated corpus for sarcasm. [arXiv:1704.05579v4](https://arxiv.org/abs/1704.05579v4)
7. Poria S, Cambria E, Hazarika D, Vij P (2016) A deeper look into sarcastic Tweets using deep convolutional neural networks. arxiv.org/abs/1610.08815
8. Lakshmanan B, Anjana A (2018) Sarcasm detection in reddit. *Int J Recent Eng Res Dev (IJRERD)* ISSN: 2455–8761
9. Joshi A, Tripathi V, Bhattacharyya P, Carman MJ (2016) Harnessing sequence labeling for sarcasm detection in dialogue from TV series ‘Friends’. In: *CoNLL*