Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018)
IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN:978-1-5386-2842-3

Sarcasm Detection using Recurrent Neural Network

Saurabh Porwal[1], Gaurav Ostwal[2], Anagha Phadtare[3], Mohini Pandey[4]
Department of Computer Engineering.
PVG's College of Engineering & Technology
Pune - 411009
(porwalsaurabh12@gmail.com[1], gauravostwl6@gmail.com[2], ana.phadtare@gmail.com[3], mohinipandey179@gmail.com[4])

Prof. Manisha V. Marathe
Department of Computer Engineering
PVG's College of Engineering & Technology
Pune - 411009
(mvm_comp@pvgcoet.ac.in)

***Abstract*—Sarcasm is a sharp and often ironic utterance that is meant to convey contempt or mock. In today's world, one of the challenging problems for opinion mining task is sarcasm detection. Many researchers are exploring the properties of sarcasm like semantic properties, syntactical properties, lexical feature etc, to design algorithms for sarcasm detection.We aim at using a recurrent neural network (RNN) model for sarcasm detection because it automatically extracts features required for machine learning approaches. Along with the recurrent neural network, this model also uses long short-term memory (LSTM) cells on tensorflow to capture syntactic and semantic information over Twitter tweets to detect sarcasm. Finally, we present the result of this model and a statistical overview of the dataset.**

Index Terms—Long Short-Term Memory (LSTM), Machine Learning (ML), Natural Language Processing (NLP), Neural Network (NN), Recurrent Neural Network (RNN), Sarcasm, Sarcasm Detection, TensorFlow, Tweets, Twitter.

I. INTRODUCTION

In recent years, social media sites such as Twitter have gained massive popularity and importance. These sites have evolved into large ecosystems where user express their ideas uninhibitedly. Many companies use their social media presence for marketing, consumer assistance and after-sales service. These unique ecosystem are helpful for companies to tap into public opinion on their products or services. As the social media data is increasing rapidly in terms of volume, companies rely on various tools to analyze data and to provide customer service. Tasks such as sentiment analysis, content management, and extraction of relevant messages for the company's customer service representatives to consider and respond. However, these tools lack the sophistication to decipher more nuanced forms of language such as sarcasm or humor, in which the meaning of a message is not always obvious and explicit. This imposes an extra burden on the social media team - already inundated with customer messages to identify these messages and respond appropriately. This gives us the reason of detecting sarcasm so as to eliminate the intentional ambiguity.

Our goal in this study is to tackle the difficult problem of sarcasm detection on Twitter. While sarcasm detection is inherently challenging, the nature and style of content on Twitter further complicate the process of detecting it as twitter is more informal in nature with an evolving vocabulary of abbreviations and slang words.

In this paper, we exploit a deep neural network for sarcasm detection using Tensorflow. We build the neural network model which contains recurrent neural networks and LSTM (Long Short Term Memory) which automatically extracts the features avoiding the overhead of exclusively extracting the features. We are training and testing the model on our self-designed twitter dataset. Results on a tweet dataset show that the neural model achieves significantly better accuracies compared to the previous studies done on sarcasm detection using linguistic approach [1,2,3,4,5], demonstrating the advantage of the automatically extracted neural features.

II. RELATED WORK

Sarcasm has been widely studied by behavioral scientists, psychologists and linguists for many years. Kreuz and Caucci [6] studied lexical features for sarcasm detection, finding that words, such as punctuation and interjections, are effective for the task. Carvalho et al. [7] demonstrated that oral or gestural expressions represented by emoticons and special keyboard characters are useful indicators of sarcasm. Both Kreuz and Caucci [6] and Carvalho et al. [7] rely on unigram lexical features for sarcasm detection. More recently, Lukin and Walker [8] extended the idea by using n-gram features as well as lexicon-syntactic patterns. Tsur et al. [9] applied features based on semi-supervised syntactic patterns extracted from sarcastic sentences of Amazon product reviews. Davidov et al. [10] further extracted these features from sarcastic tweets. Riloff et al. [11] identified a main type of sarcasm, namely contrast between a positive and negative sentiment, which can be regarded as detecting sarcasm using sentiment information. There has been work that comprehensively studies the effect of various features (Gonz´alez-Ib´anez et al. [13]; Gonz´alez - Ib´anezet al. [2]; Joshi et al., [12]). Recently, contextual information has been exploited for sarcasm detection (Wallace et al. [16]). In particular, contextual features extracted from history tweets by the same author has shown great effectiveness for tweet sarcasm detection (Rajadesingan et al., [15]; Bamman and Smith, [14]). We consider both the contextual features and traditional lexical features from history tweets under a unified neural network framework. Our observation is consistent with prior work: both sources of features are highly effective for sarcasm detection (Rajadesingan et al. [15]; Bamman and Smith [14]).

Although very limited work has been done on using neural networks for sarcasm detection, neural models have seen increasing applications in sentiment analysis, which is a closely-related task. Different neural network architectures have been applied for sentiment analysis, including recursive auto-encoders (Socher et al. [17]), dynamic pooling networks (Kalchbrenner et al. [18]), deep convolutional networks (dos Santos and Gatti [19]) and neural CRF (Zhangetal. [20]). This line of work gives highly competitive results, demonstrating large potentials for neural networks on sentiment analysis. One of the important reason is the power of neural networks in automatic feature induction and this motivates us for working further on neural network for detecting sarcasm.

III. WORD EMBEDDING

When it comes to learn and process huge amount of text it becomes really difficult, because computers convert every information into bits (ones and zeros). To convert text into bits or numbers we need to use word embedding.

“What is word embedding?”, It is the collective name for a set of language modeling technique where words or phrases from the vocabulary are mapped to vectors of real numbers. This feature learning techniques in natural language processing (NLP) is called word embedding.

We have used word embedding and created a vocabulary of more than 3,00,000 unique words. The collection of word embedding is used to convert the text into vectors of 300 size also called “embed size”. These vectors are then fed to the neural network model, as neural network can only process numeric information and not natural languages.

Word embedding also serves as a set of grammar rules or syntax because the occurrence of the parts of speech. This helps the model to learn and change the vector information for particular words according to their occurrences. Example, the probability of ‘Royal’ followed by ‘King’ or ‘Queen’ is comparatively high then any others words, so this information is updated in the vector table by the model as it learns.

IV. MORE ABOUT MODEL

The model consists of Recurrent Neural Network (RNN) to traverse the data in both direction forward and towards the next node in the same layer to convey the information about the previous occurrences words to know the context of the sentence. Example, “I was born and brought up in France, so I am fluent in French.” Here we want the context of born in France to be conveyed to the later nodes in the same layer so it possible for the model to predict the occurrence of French in the later half of the text. The information about the context is developed and known to the model with help of recurrent neural network.

It is difficult to detect features for sarcasm, as there are no specific rules or semantic to deliver sarcasm it changes from people to people. Sarcasm even depends on situation and it is sometimes time dependent. So its meaning may change with situation and time. Sarcasm is also context based most of the times, so a statement may be considered as sarcasm for only those people who are aware of the context and may even confuse people who are not aware of the context.

Due to this contextual, situation, way of delivery and time based nature of sarcasm it becomes very difficult to manually or explicitly extract features for sarcasm. So we needed a technology which could implicitly extract features for sarcasm. Using neural networks solves this problem by implicitly extracting features. One more important benefit of neural networks is that it works on activation function, so the gravitas of activation function changes for each node and in this way it learns to detect sarcasm.When using neural networks training your neural network on a variety and feature rich dataset becomes important.

Now the question is, “Why Long Short Term Memory (LSTM) Cells?”. Basically when the data travels in the same layer it becomes really important to decide that the gravitas of which words is more. Remember those whose gravitas is more and forget those with less gravitas. This task of what to remember and forget information is done by LSTM cells. They are analogous to gates in electronics which stores the information and when many gates are places in a particular sequence they act as counters and various different applications. Long short term memory works in the same way, they store the required information and forget the remaining.

The whole model is implemented on Tensorflow, a friendly machine learning library which can be easily programmed using python programming language. Due to the high flexibility provided by Tensorflow the model code is written in and is trained on Amazon Web Service (AWS). Till now we have trained on 2000 tweets dataset, we are yet to utilize the benefits of high performance cloud architecture of AWS to the full extent.

Another important aspect of our model is that we have used 2 recurrent neural network layers, with each layer consists of 256 LSTM cells. Increasing the neural network layers increases the computations and decreasing it, affects the accuracy. So deciding the appropriate number of layers of neural networks plays an important.

The model is currently deployed as a web application for demonstration purpose using python flask, a light weight web framework in python. Other deployment strategies are discussed in future scope section.

V. RESULTS

Inspite of such a huge amount of data we trained our model on 1000 sarcastic and non-sarcastic tweets each due to hardware constraints, still the model performed really well. During testing the model gave more than 90% accuracy with just 10 epochs. The model is tested on different data set and following are the results.

Dataset containing 99 sarcastic and non-sarcastic tweets each, total 198 tweets

True Positives : 91

True Negatives : 89

False Positives : 10

False Negatives : 8

Precision = tp/(tp+fp) = 0.901

Recall = tp/(tp+fn) = 0.92

Accuracy = (tp+tn)/(tp+tn+fp+fn) = 0.91

According to the following results it is clear that the model gives 91% accuracy on unknown dataset, and that is better than all the discrete model alogrithms used for sarcasm detection. As we train this model on huge datasets more accurate the predictions will be, as the accuracy increases.

VI. Future Scope

As we know that sarcasm is context based it becomes difficult to interpret and predict. In areas where huge human generated data is used like product reviews or company reviews, the sarcasm is misinterpreted by the traditional sentiment analysis tools and software. That leads erroneous report generation this may lead to defamation of company or false product review statistics. To avoid this we can use this model as a filter, to filter out sarcastic sentences from non-sarcastic ones and passing only non-sarcastic sentences to the sentiment analysis tools and software.

Sarcasm detection model can also be implemented as an application programming interface (API). API makes the access to this model really easy by passing sentence as an input and returning the output as sarcasm or non-sarcasm.

VII. Conclusion

Above information gives the details about the design and implementation of our sarcasm detection model. As the sarcasm depends on so many different factors the above results are better than the previous discrete sarcasm detection models.

References

[1] O. Tsur, D. Davidov, and A. Rappoport. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In ICWSM, 2010.

[2] R. Gonza´lez-Iba´n˜ez, S. Muresan, and N. Wacholder. Identifying sarcasm in twitter: A closer look. In ACL (Short Papers), pages 581–586. Citeseer, 2011.

[3] C. Liebrecht, F. Kunneman, and A. van den Bosch. The perfect solution for detecting sarcasm in tweets# not. WASSA 2013, page 29, 2013.

[4] E. Riloff, A. Qadir, P. Surve, and Silva. Sarcasm as contrast between a positive sentiment and negative situation. In EMNLP, pages 704–714. ACL, 2013.

[5] P. Rockwell. The effects of cognitive complexity and communication apprehension on the expression and recognition of sarcasm. Hauppauge, NY: Nova Science Publishers, 2007.

[6] Roger J Kreuz and Gina M Caucci. 2007. Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on computational approaches to Figurative Language, pages 1–4.

[7] Paula Carvalho, Lu´ıs Sarmento, M´ario J Silva, and Eug´enio De Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, pages 53–56.

[8] Stephanie Lukin and Marilyn Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In Proceedings of the Workshop on Language Analysis in Social Media, pages 30–40.

[9] Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In ICWSM.

[10] Dmitry Davidov,OrenTsur,and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 107–116.

[11] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In EMNLP, pages 704–714.

[12] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In Proceedings of the 53rd ACL, pages 757–762.

[13] Roberto Gonz´alez-Ib´anez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In Proceedings of the 49th ACL, pages 581–586.

[14] David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In Ninth International AAAI Conference on Web and Social Media.

[15] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, pages 97–106.

[16] Byron C. Wallace, Do Kook Choe, and Eugene Charniak. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In Proceedings of the 53rd ACL, pages 1035– 1044.

[17] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the EMNLP.

[18] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In Proceedings of the 52nd ACL, pages 655–665. Association for Computational Linguistics.

[19] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, pages 69–78.

[20] Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015. Neural networks for open domain targeted sentiment. In Proceedings of the EMNLP, pages 612–621.