Sentiment Analysis for Sarcasm Detection using Deep Learning

Vialli L. K. Gonsalves[1], Sherica L. Menezes[2]

[1]Student, Dept. of Computer Engineering, Goa College of Engineering, Farmagudi, Ponda, Goa, India
[2]Project Guide, Assistant Professor, Dept. of Computer Engineering, Goa College of Engineering, Farmagudi, Ponda, Goa, India

---***---

Abstract - *Detection of sarcasm in texts is useful to understand the thorough meaning of the message as to whether the message is positive or negative and hence classify it accordingly. It is normally a tough procedure as a negative message is portrayed as a seemingly positive one and hence due to this, there is an error in the classification of such messages. Unlike sentiment analysis, the borders of sarcasm are not well defined. This paper aims to address the difficult task of sarcasm detection by using some Deep learning techniques like LSTM, GRU, and Bi-LSTM. The aim is to use these techniques on the same dataset and compare their performance. The one with the best-observed performance is chosen as the main model and the final classification task is done using this model*

Key Words: Sarcasm, Detection, Classification, LSTM, Bi-LSTM, GRU

1. INTRODUCTION

The technique to analyze and determine the sentiment behind a piece of text is called Sentiment analysis. It uses a combination of machine learning principles and natural language processing (NLP) to achieve this. Using sentiment analysis, we can recognize whether the sentiment behind a block of text is positive, negative, or neutral. It is a powerful technique in Artificial intelligence(AI) that has an important application in business.

For example, you can use it to analyze customer feedback. After acquiring that feedback or reviews through various mediums, you can run sentiment analysis algorithms on those text pieces to calculate the sentiment score and based on that understand your customers' attitude towards the product.

Sentiment analysis is useful when there is a large amount of unstructured data, and to then classify that data by automatically tagging it. Net Promoter Score (NPS) surveys are a technique that is used extensively to gain knowledge and understanding of how a customer perceives a service or product. In human social interaction, sarcasm—which can be both constructively humorous and negatively offensive—plays a significant role.

In social media, sarcasm is frequently utilized to convey a negative opinion using positive or amplified positive language. Because of this, sentiment analysis models can be readily tricked by sarcasm unless they are specially created to consider this possibility. Sarcasm identification is a crucial component of sentiment analysis because of this deliberate ambiguity. It is claimed that sarcasm detection is a binary classification issue.

Without a thorough understanding of the situation's context, the targeted topic, and the surroundings, sarcasm identification in sentiment analysis is exceedingly challenging. Both a human and a machine may find it challenging to comprehend. It might be challenging to adequately train sentiment analysis algorithms since sarcastic phrases frequently use a wide range of terms.

2. RELATED WORK

It is beneficial for the user to understand the sentiment of the message that is received to have a proper understanding of it.

Avinash kumar[1] aims to identify sarcastic remarks in a given corpus; they introduce the multi-head attention-based bidirectional long-short memory (MHA-BiLSTM) network. The experiment's findings show that BiLSTM performs better than feature-rich SVM models and benefits from a multi-head attention mechanism.

Liyuan Liu[2], in their work, to improve classification performance, a new deep neural network called A2Text-Net created to emulate face-to-face speech. This network incorporates auxiliary variables like punctuation, part of speech (POS), numerals, emoji, etc. The outcomes of the experiment show that our A2Text-Net technique outperforms traditional machine learning and deep learning algorithms in terms of classification performance.

Md shaifullah razali[3] identifies sarcasm in tweets by combining hand-crafted contextual data with deep learning extracted features. Convolutional Neural Network (CNN) feature sets are retrieved from the architecture and then mixed with meticulously designed feature sets. These individually designed feature sets are developed based on their corresponding contextual justifications. Each feature set has been created expressly to detect sarcasm in mind.

Ibrahim Abu-Farha[4] now proposes ArSarcasm, a dataset for detecting sarcasm in Arabic that was produced by reannotating existing datasets for sentiment analysis in Arabic. 10,547 tweets total in the sample, with 16% of them being ironic. The data was tagged for sentiment and dialects in addition to sarcasm. The shift in sentiment labeling based on annotators' biases serves as evidence in our examination of the extremely subjective character of these tasks.

Deepak Jain[5] in this study suggests utilizing deep learning to identify sarcasm in tweets that move between English and Hindi, the native language of India. The suggested model combines a softmax attention layer with a convolution neural network for real-time sarcasm detection, together with a bidirectional long short-term memory component. Real-time mash-up tweets are taken from popular trending political (#government) and entertainment (#cricket, #bollywood) posts on Twitter to assess the performance of the suggested approach. 3000 sarcastic and 3000 non-sarcastic bilingual Hinglish (Hindi English) tweets make up the randomly selected dataset.

Rolandos Alexandros Potamias[6]. In their paper "A transformer-based approach to irony and sarcasm detection" To solve the identification of the aforementioned FL forms' problems, we use cutting-edge deep learning approaches. We offer a neural network methodology that significantly expands on our earlier work. It is based on a recently proposed pre-trained transformer-based network architecture and is further improved by the use of a recurrent convolutional neural network. Data preparation is minimized in this configuration. Results show that the suggested methodology outperforms all previous methodologies and published research, even by a significant margin, and achieves state-of-the-art performance under all benchmark datasets.

Himani Srivastava[7] in their paper "A Novel Hierarchical BERT Architecture for Sarcasm Detection" offer a novel deep learning-based method that hierarchically uses the supplied contexts to determine if a statement is sarcastic or not.

For our investigations, we used datasets from Twitter and Reddit1, two online discussion forums. Experimental and error analysis demonstrate that the hierarchical models outperform their sequential counterparts by fully utilizing history to achieve a better representation of contexts.

Lu Ren et al.[8] in their paper "Sarcasm Detection with Sentiment Semantics Enhanced Multi-level Memory Network" Using emotion semantics, we suggest a multi-level memory network to capture the characteristics of sarcastic statements. The first-level memory network in our model is used to represent the sentiment semantics, and the second-level memory network is used to represent the contrast between the sentiment semantics and the context of each sentence. Additionally, we enhance the memory network in the absence of local information using an upgraded convolutional neural network. The experimental outcomes on the Twitter dataset and Internet Argument Corpus (IAC-V1 and IAC-V2) show how well our model performs.

Le Hoang son et al.[9] in their paper "Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network" sAtt-BLSTM convNet, a deep learning model that uses global vectors for word representation (GLoVe) to create semantic word embeddings, is based on a hybrid of soft attention-based bidirectional long short-term memory (sAtt-BLSTM) and convolution neural network (convNet). Punctuation-based auxiliary features are combined with the convNet together with the feature maps produced by the sAtt-BLSTM. Using balanced (tweets from benchmark SemEval 2015 Task 11) and unbalanced (about 40000 random tweets using the Sarcasm Detector tool with 15000 sarcastic and 25000 non-sarcastic words) datasets, the robustness of the proposed model is examined. The proposed deep neural model with convNet, LSTM, and bidirectional LSTM with/without attention is compared using training- and test-set accuracy metrics, and it is found that the novel sAtt-BLSTM convNet model outperforms others with a higher sarcasm-classification accuracy of 97.87 percent for the Twitter dataset and 93.71 percent for the random-tweet dataset.

3. METHODOLOGY

1) Data extraction

For this research, a text dataset provided by Kaggle named "News Headlines Dataset For Sarcasm Detection" was used. Two news websites provided the content for this News Headlines dataset for sarcasm detection. TheOnion tries to produce satirical interpretations of current events, thus we gathered all the headlines from the categories News in Brief and News in Photos (which are sarcastic). From HuffPost, we pull actual (and unsarcastic) news headlines.

Compared to the current Twitter datasets, this new dataset has the following advantages:

- There are no spelling errors or colloquial language because news headlines are prepared by professionals in a proper manner. This lessens the sparsity and raises the likelihood of discovering pre-trained embeddings.

- Furthermore, compared to Twitter datasets, we receive high-quality labels with significantly less noise because TheOnion's main objective is to provide satirical news.

- The news headlines we gathered are standalone, unlike tweets that are replies to other tweets. This would make it easier for us to identify the true sarcastic components.

Each record consists of three attributes:

- is_sarcastic: is a boolean value with 1 as sarcastic and 0 as not
- headline: the headline of the news article
- article_link: link to the original news article. Useful in collecting supplementary data

2) Pre-processing

Today, every industry has access to a significant amount of unstructured data in the form of text, audio, videos, etc. This information can be utilized to assess a variety of variables, which can then be applied to further make decisions or predictions. But in order to achieve better results, the raw data must be organized or summarized.

This is where natural language processing (NLP), a subfield of data science, comes into play. NLP helps analyze, organize, or summarize text data so that meaningful information may be extracted from it and applied to business choices. Text preprocessing refers to the technique of preparing textual data by cleansing it and formatting it appropriately for inputting into a model.. The following is done :

- Remove emojis, symbols, and pictures
- Remove unnecessary characters
- Remove flags(IOS) and map symbols
- Remove abbreviations
- Convert all text to lowercase
- Remove duplicates

3) Visualization

Using the word cloud, we can visualize the frequency of words occurring in the chosen dataset. The larger the word is, the greater its frequency.

The technique of Word Cloud is employed to visualize textual data, where the size of each word indicates its frequency or significance. This technique can be used to emphasize noteworthy textual elements in the data. Word clouds are widely used for analyzing data from social network websites.

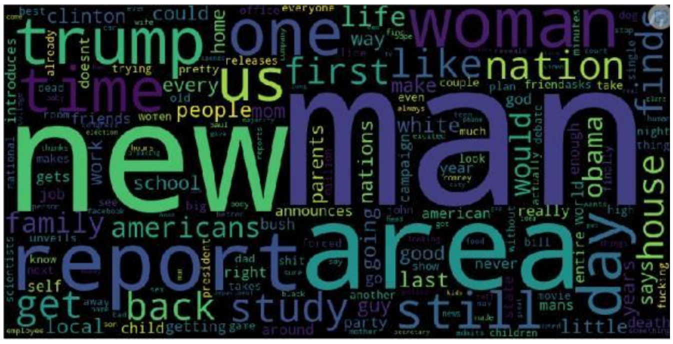

Fig 1 : Dataset Visualization

As we can see in the image, the words such as “new”, “man”, “report”, “area” are the largest in size and hence through this we can say that they are frequently occuring in the dataset.

Where words such as “friends”, “mom”, “mother”, “another” are the smallest in size so we can conclude that these words are rarely present in the dataset.

4) Train-test split

When your model is deployed in production, you don't want it to overlearn from the training data and perform poorly. You require a method to judge how well your model generalizes. To avoid overfitting and to conduct an effective model evaluation, you must divide your input data into training, validation, and testing subsets.

To train a CNN network we need training and validation data, the dataset in Kaggle has only a training dataset. So I have chosen some part of training data as validation data, and the remaining images as the training data. I have chosen a split size of 20% to separate the validation images from the training images.

5) Loading GloVe model

GloVe stands for global vectors for word representation. It is an unsupervised learning algorithm developed by Stanford for generating word embeddings by aggregating global word-word co-occurrence matrices from a corpus. The resulting embeddings show interesting linear substructures of the word in vector space.

6) Building the model

The model used for this project is the Bi-LSTM model. Bidirectional long-short-term memory(Bidirectional LSTM) is the process of making any neural network have the sequence information in both directions backward (future to past) or forward(past to future). After studies and comparison with other models such as LSTM and GRU, it was found that Bi-LSTM performed the best on text-based datasets and hence this model was used for sarcasm detection.

Various parameters were changed and tested to see which combination performs the best. Parameters such as epochs, activation function, dropout and optimizer were taken into consideration and changed so as to find the ideal setup or combination for the chosen model.

a)Epoch : In machine learning, one entire transit of the training data through the algorithm is known as an epoch. The epoch number is a critical hyperparameter for the algorithm. It specifies the number of epochs or full passes of the entire training dataset through the algorithm's training or learning process.

b)Activation function : The activation function of a node defines the output of that node given an input or set of inputs. A standard integrated circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input. This is similar to the linear perceptron in neural networks.

c)Dropout : Dropout refers to data, or noise, that's intentionally dropped from a neural network to improve processing and time to results.

The challenge for software-based neural networks is they must find ways to reduce the noise of billions of neuron nodes communicating with each other, so the networks' processing capabilities aren't overrun. To do this, a network eliminates all communications that are transmitted by its neuron nodes not directly related to the problem or training that it's working on. The term for this neuron node elimination is dropout.

d)Optimizer : Optimizers are used to update weights and biases i.e. the internal parameters of a model to reduce the error.

7) Training

After choosing the configurations of the model, the model is trained on the data that was initially chosen for the purpose of training(train-test split). The number of epochs chosen was 25 for this.

```
Epoch 23/25
 - 25s - loss: 0.3004 - acc: 0.8691 - val_loss: 0.2695 - val_acc: 0.8849
Epoch 24/25
 - 25s - loss: 0.2953 - acc: 0.8736 - val_loss: 0.2652 - val_acc: 0.8897
Epoch 25/25
 - 25s - loss: 0.2951 - acc: 0.8719 - val_loss: 0.2659 - val_acc: 0.8893
 - 25s - loss: 0.3004 - acc: 0.8691 - val_loss: 0.2695 - val_acc: 0.8849
Epoch 24/25
 - 25s - loss: 0.2953 - acc: 0.8736 - val_loss: 0.2652 - val_acc: 0.8897
Epoch 25/25
 - 25s - loss: 0.2951 - acc: 0.8719 - val_loss: 0.2659 - val_acc: 0.8893
```

Fig 2. Training

4. CONCLUSION

Bi-LSTM was used in this project to detect sarcasm. The model was configured in such a way as to obtain the ideal values in terms of accuracy and loss. For an ideal model, the best case would be to have a high accuracy value and a low loss value. The values obtained using the Bi-LSTM model were 22.1% and 91.68% for loss and accuracy respectively which was comparatively higher as compared to the other models such as LSTM and GRU.

Model	Loss	Accuracy
LSTM	31.2%	86.68%
GRU	27.12%	88.84%
Bi-LSTM	22.1%	91.68%

Fig 3: Results

As can be seen, compared to the other models used namely GRU and LSTM. The performance of the chosen model i.e Bi-LSTM is better than that of the others.

5. REFERENCES

[1] Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti, "Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM" IEEE Access, Jan-2020

[2] Liyuan Liu, Jennifer Lewis Priestley, Yiyun Zhou, Herman E. Ray, Meng Han, "A2Text-Net: A Novel Deep Neural Network for Sarcasm Detection" 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)

[3] Md Saifullah Razali, Alfiian Abdul Halin, Lei Ye, Shyamala Doraisamy,Norris Mohd Norow, " Sarcasm

Detection Using Deep Learning With Contextual Features" IEEE Access, May-2021

[4] Ibrahim Abu-Farha and Walid Magdy, "From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset" Language Resources and Evaluation Conference (LREC 2020)

[5] Deepak Jain, Akshi Kumar, Geetanjali Garg, "Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN" Elseveir Volume-91, June 2020

[6] Rolandos Alexandros Potamias, Georgios Siolas, Andreas - Georgios Stafylopatis, "A transformer-based approach to irony and sarcasm detection" Neural Computing and Applications (2020)

[7] Himani Srivastava, Surabhi Kumari, Saurabh Srivastava, Vaibhav Varshney, "A Novel Hierarchical BERT Architecture for Sarcasm Detection"

[8] Lu Ren, Bo Xu, Hong fei Lin, Xikai Liu, Liang Yang, " Sarcasm Detection with Sentiment Semantics Enhanced Multi-level Memory Network" Elsevier, Volume 401, August 2020

[9] Le Hoang Son, Akshi Kumar, Saurabh Raj Sangwan, Ashika Arora, Anand Nayyar, Mohamed Abdel-Basset, "Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network" IEEE Access Feb-2019

[10] News Headlines Dataset For Sarcasm Detection | Kaggle