Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 93–102

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Sarcasm Detection Using Bidirectional Encoder Representations from Transformers and Graph Convolutional Networks

Anuraj Mohan[a], Abhilash M Nair[a,*], Bhadra Jayakumar[a,*], Sanjay Muraleedharan[a]

[a]*Department of Computer Science and Engineering, NSS College of Engineering, Palakkad, Kerala, India*

Abstract

The Internet has become a crucial space for customer feedback and the budding of various ideologies across different cultures. But some people provide their opinions whose sole meaning is different from what their figurative meanings imply. This sophisticated form of sentiment expression through mockery or irony is called sarcasm. Sarcastic form of texts has taken a prominent place in social media and their growth has become exponential in recent years through messages and posts. Sarcastic comments, tweets, or feedback can be misleading in data mining activities and may result in wrong predictions since the boundaries of the sarcastic context are not well defined. There is a need to model new systems that can precisely define the boundaries of the sentence and detect sarcasm in them. Different methods of classification have been used for sarcasm detection in various works including deep learning, neural networks, and other architectures. This paper focuses on combining the capabilities of both Bidirectional Encoder Representations from Transformers (BERT) and Graph Convolutional Networks (GCN) for detecting sarcastic content from text. A BERT-GCN architecture is proposed which takes as input, the graph, and text representations and learns the complex structural and semantic patterns in the text, thereby detecting sarcastic content. The efficiency of BERT-GCN is compared with various baseline methods.

Keywords: Sarcasm Detection, Graph Convolutional Network, BERT, Transformers, Word Embedding.

1. Introduction

The modern world contains a massive load of unstructured, unclassified, and unsorted data. The usage of these data as well as extracting the information out of them is an urgent requirement, as it can help to speed up the processing in minimal time and cost. Text classification is one such classical problem existing in machine learning. Text Classification refers to classifying/sorting texts based on sentiments/emotions obtained from that sentence. In the present world, text classification is widely required, and different models are used to detect the meaning and the sentiment aspect of a sentence. But, a new problem arises when people started to use sarcastic statements to comment on things. Sarcasm

* Corresponding author
E-mail addresses: abhilashmnair20@gmail.com (Abhilash M Nair)., bhadrajayakumarsandhya@gmail.com (Bhadra Jayakumar).

1877-0509
10.1016/j.procs.2022.12.405

is a humorous way of representing a negative situation in positive words. When it comes to sarcasm, the task becomes more complex as sarcasm is deeper than text classification and it includes the meaning, aspect, and context in which the sentence is used. As the context is relevant in sarcasm detection, the trained model should be able to extract the words, the context, and the actual meaning of the sentence with minimal errors. As sarcasm is widely used in the present scenario for communication as well as for other sectors including social media analysis, and review systems as well, the feedback of the users plays a critical role in systems like product review. It is essential to understand what the user actually tries to convey to the outside world with his/her review. Hence sarcasm detection becomes a task of higher complexity and importance.

Various neural network-based approaches have been already applied to this particular problem. A hybrid neural network architecture [13] with an attention mechanism model is proposed which emphasizes providing the various aspects that tell what actually makes a sentence sarcastic. A supervised learning model is developed to detect sarcasm on Facebook [3]. The major contributions include considering user interaction and a supervised learning model for detecting sarcasm. A deep neural network-based sarcasm detection technique on news headlines using CNN and LSTM [12] is proposed, but this requires more training time for LSTM and text tagging is difficult in CNN as adjacent words might not be related. Another approach uses the graph-based neural network based on SenticNet [2] with the support of additional graph structures of the sentence according to the specific aspect. Similarly, a graph network-based approach [10] with the affective knowledge of individual words from SenticNet common knowledge database to improve the dependency word graph of sentences is proposed. Also, a hybrid neural network model is proposed which uses a GCN to extract global information from the sentences along with a Bi-LSTM [6] network to get the feature sequence which is eventually concatenated and passed into a conventional classifier for prediction. In this work, the strengths of both BERT[4] and GCN [8] are combined to improve the detection of sarcasm in texts. The adjacency and dependency graphs are created from the text words with the help of SenticNet, and the text features are learned using BERT. The graph structures and the features generated by BERT are then passed to a Graph Convolution Network. Context representations thus obtained will be updated based on the outputs of the GCN, and are then given to a softmax classifier to get the probability of whether the statement is sarcastic or not.

Sections 2 and 3 of the paper provide the problem definition and related works respectively. Section 4 provides some preliminary content about the concepts used in the work. Section 5 describes the system design. Section 6 and section 7 detail the experimental setup and results respectively. The paper concludes with section 8 which presents the conclusion and future scope.

2. Problem Definition

Given a set of statements, $S = \{s_1, s_2, s_3 ... s_n\}$, a defined set of classes $C = \{0, 1\}$ where 0 and 1 denotes non-sarcastic and sarcastic labels respectively, and a training set which contains the labeled statements, the task is to learn a classifier, $f : S -> C$ i.e. learning a function that can predict the labels of an unknown test statement.

3. Related Works

Sarcasm Detection is a comparatively new hurdle in text mining and classification tasks. Different approaches have been used by considering the aspects of the text, sentences, and words. Various approaches comprising LSTM and its variants, Neural Networks, and its variants are used to solve this problem.

In the work done by Poria et al [5], a model called CASCADE (ContextuAl SarCasm Detector) is proposed using a mixed method of both content and context-based mode to identify sarcasm in online social media conversations and forums. A sequential model is developed that takes knowledge of the context of a sentence to categorize it as sarcastic or non-sarcastic. To determine whether the text may contain a sarcastic context in near future, Simple Exponential Smoothing (SES) is used in the integration layer, which is a predictor of time series data that does not change without trend or season. SCUBA model [14] (Sarcasm Classification Using a Behavioural modeling Approach) captures emotional differences, compares present and past tense, readability, status, vocabulary, grammar skills, structural diversity, and the place where the message is delivered for direct separation. Instead of focusing entirely on the content and context of the tweet, this approach refers to the user behavior model as an important part of identifying the paradox in their comments. Misra et. al. [13] presented an in-depth reading model that is consistent with the attention-grabbing

approach that provides insight into what makes a sentence mockery. In the proposed model, the content is provided in an LSTM module with a focus to re-evaluate the encoding context. Similar to the model proposed for the Twitter platform, an approach was done for the social networking site, Facebook. This model emphasizes both the content of the post and how the user interacts with it. A sarcasm detection model using BERT embedding and GloVe embedding as features was proposed to achieve state-of-art results [7]. An approach to defining both the affective and contextual meaning of sentences for carrying out sarcasm detection was developed by Babanejad et al [1]. The proposed system is of two mixed models, ACE1 and ACE2, to identify irony, where each model takes one document as input and assesses whether the document is satirical. An approach to combine LSTM models and pre-trained BERT is done to compare the efficiency of the models [15]. The model implemented all LSTM models and used the existing PyTorch-pre-trained-BERT package for sarcastic detection in text with GloVe for word embeddings. LSTM-based models used were Vanilla LSTM, Conditional LSTM, and LSTM with sentence-level attention. A BiLSTM-based model is proposed [9] which uses word-level vector encoding of the embedded layer, CNN, and news headlines. In this method, a deep neural network-based model was proposed which combines both the advantages of CNN and LSTM layers which are trained with word embeddings obtained from the word vector encodings of news headlines to determine the sarcastic probability of a headline. A sentence-level multi-head attention layer takes the whole hidden state as input and multiplies it by word vector, and then sends it to the output hyperbolic function. Auxiliary feature concatenation extracts semantic, sentimental, and punctuation-based handicraft features for a given comment. For understanding the relationship between words in a sentence and their implied context, an affective knowledge about the content is to be understood. An affective graph representation approach[11] is implemented in the architecture using graph convolutional networks and deep learning model, to understand the sarcastic context in a sentence. A dependency graph formulated using the word connections in a sentence along with an affective sentiment graph constructed using the common knowledge database is fed alternatively to a graph convolutional network which learns the sarcastic context of the words in the sentences to enhance the predictions.

4. Preliminaries

4.1. SenticNet

SenticNet, [2] as shown in Fig.1, is a sentiment analysis framework and common knowledge database, for performing tasks such as word polarity detection and emotion recognition by considering both denotative and connotative information provided by the words and their multiword expressions instead of relying on co-occurrence frequencies.

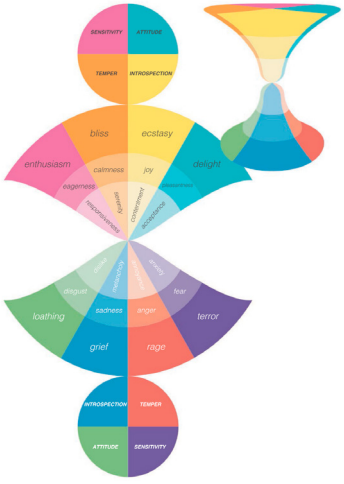

Fig. 1: SenticNet [2]

4.2. Bidirectional Encoder Representation from Transformers

Bidirectional Encoder Representation from Transformers (BERT) [4] is an open-source NLP framework based on transformers developed by Google which can be fine-tuned for various specific applications that can be used to understand the meaning of ambiguous languages by predicting surrounding texts. The basic BERT architecture is shown in Fig. 2.

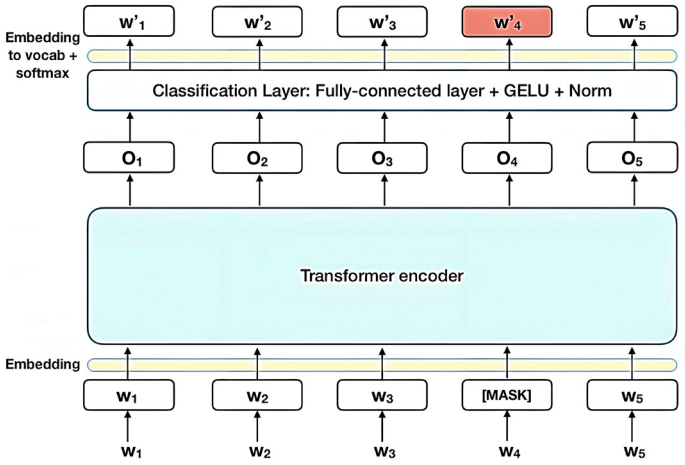

Fig. 2: BERT [4]

4.3. Graph Convolutional Networrks

Graph Convolution Networks (GCN) [8] are neural networks that work on the graph domain. Words in a language can be connected to form graphs and hence applying GCN over such graphs can learn the global context of the words within the text. However, GCN only considers the global lexicon information and might fail to capture local lexical information (such as word co-occurrence and order), which is very crucial in understanding the interpretation of a sentence. Fig. 3 depicts the semi-supervised method using C channels as input to F feature maps in the output which is layered given by Y labeled numerically.

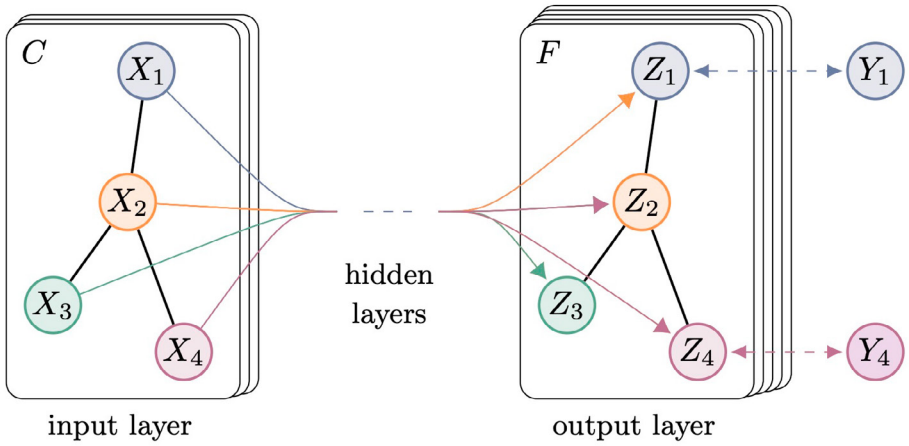

Fig. 3: Graph Convolution Network [8]

5. Methodology

Our proposed model as shown in Fig. 4. aims to develop an efficient system that can detect whether the statement is sarcastic or not from a given dataset by leveraging the capabilities of both BERT and GCN. This is because graphs can represent the longer dependencies between words and the transformer models have the ability to recognize the context of longer sentences and even reply-response contexts. In the case of sarcastic statements, the aim is to learn the context and the content for generating accurate results. Combining graph embeddings and word embeddings can generate a better semantic representation which helps in achieving better accuracy in sarcastic text classification.

The outline of the overall process is as follows. An affective graph is created using a common knowledge base, SenticNet, and the spaCy library. A dependency graph is created from the adjacency matrix which defines the relationship between words in the sentence. The dependency and affective graphs are created from a labeled sarcastic dataset. The tokenized dataset formed with the BERT tokenizer is given as input to the pre-trained BERT. The word embeddings are produced as the output and are fed as input features to the Graph Convolution Network. The affective graph and dependency graph are given alternatives to the graph convolution network to produce the combined representations which are then passed to a softmax classifier to predict the corresponding label.

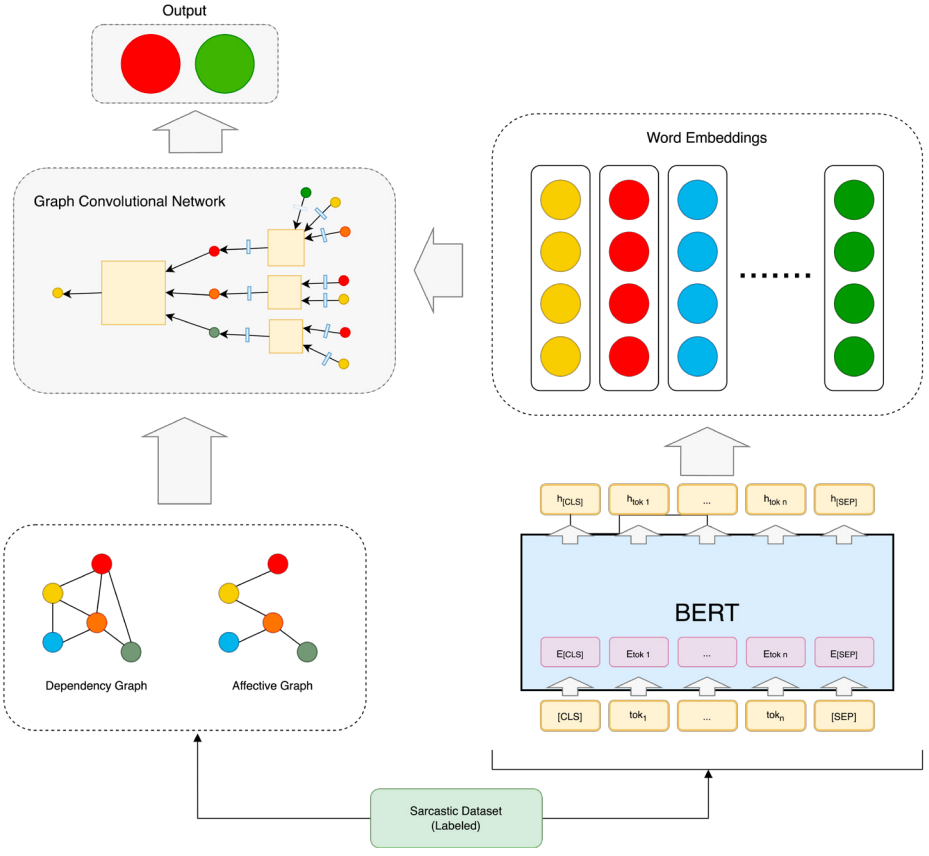

Fig. 4: System Architecture

5.1. Deriving Affective and Adjacency Graph

Our aim is to convert sarcastic statements into their equivalent graph format without losing their meaning and sentiment. Inspired from [15], a dependency adjacency graph, **D** is created to form an $n \times n$ matrix, where n is the number of words in the sentences (Eq. 1). Hence a word-to-word connection in a sentence is formed. The dependency

graph is created from the dependency tree **T** formed from the text.

$$\mathbf{D_{i,j}} = 1 \qquad if \;\; \mathbf{T}(w_i, w_j) \tag{1}$$

where $\mathbf{D_{i,j}}$ is the dependency of words w_i and w_j in the sentence. As the affective and dependency information has to be maintained, an undirected graph is constructed, $\mathbf{D_{i,j}} = \mathbf{D_{j,i}} = 1$, and a self-loop is also set $\mathbf{D_{i,i}} = 1$.

From the dependency graph D thus formed, an affective graph is built by creating a sentiment-based graph with the help of SenticNet. Let S be a sentence with n words, represented by $S = \Sigma w_i$, the affective graph is built on the affective scores of words obtained from an external affective common sense knowledge and is calculated as given in Equation. (2).

$$\mathbf{A_{i,j}} = |S(w_i) - S(w_j)| \tag{2}$$

where $S(w_i) \in [1, 1]$ represents the overall affective score of the word w_i obtained from SenticNet and is set as $S(w_i) = 0$ if w_i is not held in the repository.

The usage of both graphs enhances the information in the graph structure as the semantic scores inferred from the affective graph and the word relations obtained from the dependency graph produces a difference in the sentimental values that are used to understand the sarcastic probabilities in the given textual data.

The dependency and affective graphs obtained from the sample sentence, "I just love it when people don't text me" is depicted in Fig. 5 and 6 respectively, and are discussed below.

An affective dependency graph is created for each sentence in the dataset with the words as nodes and the connection between the words as the edges of the graph. As described earlier, the graph is undirected and contains self-loops to preserve the context of each word. The graph is defined from the tree structure of the sentence. The semantic graph is created by adding the weights of the sentiment value of each word with respect to the existing knowledge base. The weights provide a flow of sentiment value over the words and finally provide the overall sentiment for the given sentence which is later examined for sarcastic probability.

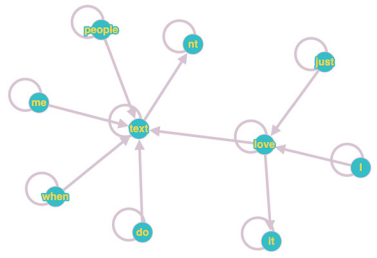

Fig. 5: Dependency Graph

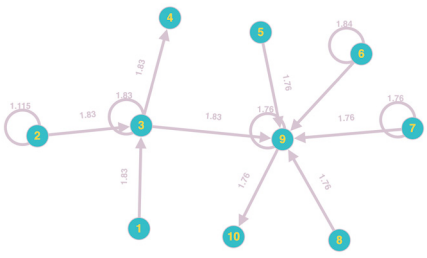

Fig. 6: Affective Graph

5.2. Representative Learning from Context using BERT

Each word needs to be given attention and relevant word features must be extracted from the sentences. This feature extraction is performed by the BERT architecture which creates the embedding for each word in the sentence, forms sentence embedding by performing $S = \Sigma w_i$ of its corresponding word embeddings, and the mapping is done by checking it with the lookup table of the vocabulary size of the BERT model.The obtained embedding matrix $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \ldots \mathbf{x_n}]$ fed into the BERT to encode the input sentences into its corresponding vector representations. (Eq. 3)

$$\mathbf{H} = \{\mathbf{h_1}, \mathbf{h_2}, \mathbf{h_3}, ..., \mathbf{h_n}\} = BERT(\mathbf{x}) \tag{3}$$

where **h** represents the hidden representations of **x** at each time interval.

5.3. Predictions from Graph Convolution Networks

The output of the BERT is then passed to a graph convolution network along with the affective graph and dependency graph created in the initial stage. The graphs are passed alternatively to the graph convolution network so that

the representation of each node is updated at a given GCN layer, according to the hidden representations obtained from its neighborhoods. Each node thus learns its weights from the information flow of neighbor nodes and previous output representation as depicted in Eq. (4).

$$\mathbf{g^l} = ReLU(\mathbf{H}ReLU(\mathbf{A}\mathbf{g^{l-1}}\mathbf{W_a^l} + \mathbf{b_a^l})\mathbf{W_d^l} + \mathbf{b_d^l}) \tag{4}$$

where $\mathbf{g^{l-1}} \in R$ is the latent node representations generated from each GCN layer, and the input to the first graph convolutional layer is the initial context representation learned by BERT given by $\mathbf{H}$.

The dependency graph and affective graph are created from the sarcastic dataset. The word embeddings and the graphs are then given to a graph convolution network for the sarcastic probability predictions. The BERT Word embedding used here is of 300 dimensions, BERT with 768 BERT layers and the GCN consists of 4 layers.

5.4. Predicting the Outputs

The predictions made by the architecture are the normalized output of the GCN network. The GCN network uses the BERT word features along with the dependency learned from the word graph formed. The output of the final GCN layer is passed into a softmax normalized activation layer to record the sarcastic probability distribution to form the predictions.

$$\hat{\mathbf{y}} = softmax(\mathbf{W_o}\mathbf{g^l} + \mathbf{b_o}) \tag{5}$$

where $\hat{\mathbf{y}}$ is the predictions obtained about the sarcastic probability of the sentence and $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters.

5.5. Learning Objective

The system uses cross-entropy loss criterion with the standard gradient descent algorithm to train the model. The formulation is given by:

$$\min_{\Theta} L = -\sum_{k=0}^{N} \mathbf{y} log \hat{\mathbf{y}} \tag{6}$$

where $\mathbf{y}$ represents the ground-truth values and $\hat{\mathbf{y}}$ indicates the predicted probabilities of the given instance of the input.

6. Experimental Setup

The experiment was conducted using publically available datasets from Kaggle [1] and Riloff's website [2] for sarcasm. The hardware infrastructure used was a system with an Intel i9 processor and 16 GB RAM and the software tools used are Pytorch, Scikit-learn, and NetworkX.

6.1. Dataset Description

The datasets prepared for this experiment are described in this section. The dataset consists of sarcastic sentences from news articles and Reddit forums. For the sake of computational cost, the existing datasets were reduced using Python scripts and were used in our model. An overview of the dataset used is given in Table.1

6.1.1. News Headline Dataset

The News Headlines dataset is used to train the BERT model. The dataset is made by collecting news headlines from two news websites; theonion.com and huffingtonpost.com. Most of the existing n works use Twitter datasets, which have been proven inefficient due to their noisy labeling and language usage. *TheOnion* aims at producing

[1] https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection
[2] http://www.cs.utah.edu/~riloff/publications_chron.html

Table 1: Statistics of various datasets used

Dataset	Train		Test	
	Sarcastic	Non Sarcastic	Sarcastic	Non Sarcastic
Riloff	282	1051	35	113
Headlines	2516	2504	570	410

sarcastic versions of current affairs, which are collected by scraping, and the collection of real (and non-sarcastic) news headlines is performed from HuffPost.

6.1.2. Riloff Dataset

The dataset contains 35,000 tweets extracted from Twitter. It is an unbalanced dataset with fewer sarcastic tweets and more non-sarcastic statements. The approach is intended sarcasm as the tweet is labeled sarcastic or not by the author.

6.2. Evaluation Measures

6.2.1. Accuracy

Accuracy is defined as the number of predictions made correctly by the system to the total number of sentences in the system.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{7}$$

6.2.2. F1 score

F1 score can be defined as twice the product of accuracy and recall by the sum of precision and recall

$$F1score = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{8}$$

7. Experimental Results

A comparison was done with 3 different models (Table 2) including the GCN, Sequential, and LSTM Models. It can be clearly seen that our model, the BERT-GCN Model outperforms other models and shows 90.7% accuracy with an f1 score of 89.6% in the Headline dataset and an accuracy of 88.3% with an f1 score of 87.7% in the Riloff dataset.

Table 2: Performance comparison with baselines

Model	Headlines		Riloff	
	Accuracy (%)	F1 Score (%)	Accuracy (%)	F1 Score (%)
Sequential	79	36	81	43
LSTM	56	36	78	75
GCN	87.2	86.8	86.5	76.35
BERT-GCN	**90.7**	**89.6**	**88.3**	**87.7**

Graphs plotted between accuracy and number of epochs for two datasets show that the model shows an accuracy of 90.t% with the Headline dataset and an accuracy of 88.3% with the Riloff dataset, which is the highest with the number of epochs is 20 and early stopping. A comparative analysis of the effect of the number of epochs with respect to the loss and accuracy was performed. The effect of the number of GCN layers with respect to performance is also studied.

7.1. Effect of Training

The plot showing the improvement in accuracy for both datasets with respect to the number of epochs is shown in Fig. 7. A considerable efficiency increase in the evaluation of the model can be observed with the balanced sarcastic news headlines dataset. It can also be noted that (Fig. 8) the model is having higher loss values for a lesser number of epochs and as the number of epochs increases the loss becomes lower. Hence, an early stopping condition is attained with a patience level set as 5 during the tuning of hyperparameters at 20 epochs, which exhibits the best performance of the system.

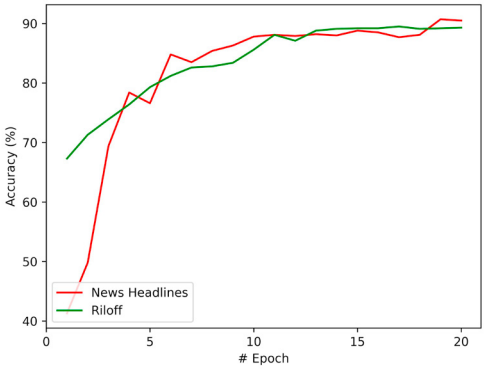

Fig. 7: Accuracy vs Number of Epochs

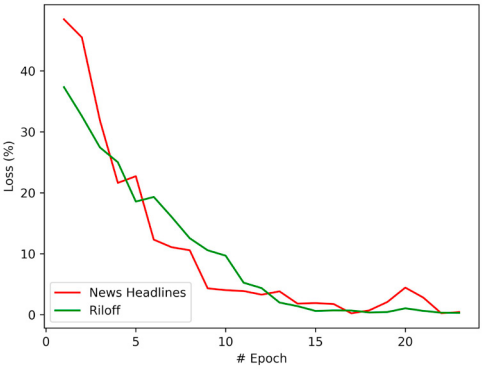

Fig. 8: Loss vs Number of Epochs

7.2. Accuracy vs GCN Layer Size

The changes in performance while varying the number of layers of the graph convolution network are given in Fig. 9. It is observed that the model produced a steady increase in accuracy up to three layers, peaked when the number of layers was four, and there is a drop in the accuracy with the increasing number of layers. Hence, the number of GCN layers in this experiment is chosen as four.

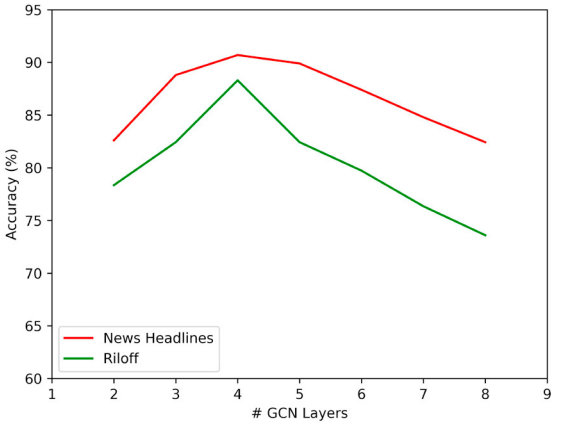

Fig. 9: Accuracy vs Number of GCN Layers

8. Conclusion and Future Work

Recent studies have shown that BERT architecture can give better accuracy even for small datasets for many problems related to text mining. Also, many studies proved that deep semantic information about the text can be obtained using graph-based methods, and can provide global information features and structural relationships which can improve the performance of sarcasm detection from text. In this work, a method to combine both the capabilities of text-based and graph-based approaches is proposed, and a BERT-GCN architecture for sarcasm detection is developed. By conducting experiments with the News-Headline dataset and Riloff Dataset, the performance of the method is assessed and a comparison is made with various baseline models. As a scope of future work, attention mechanisms can be incorporated into the graph neural network that can learn the other aspects of sarcastic sentences, and thus form a representation that depicts the relation of the nearby words to the word in consideration. This allows for the removal of the equal weightage given to each word by its neighboring words and may provide an improved result by only taking the predominant connections in the graph structure. Also, variations of transformer models need to be employed to further extend the evaluation of the proposed system.

References

[1] Babanejad, N., Davoudi, H., An, A., Papagelis, M., 2020. Affective and contextual embedding for sarcasm detection, in: Proceedings of the 28th International Conference on Computational Linguistics, pp. 225–243.
[2] Cambria, E., Hussain, A., 2015. Senticnet, in: Sentic Computing. Springer, pp. 23–71.
[3] Das, D., Clark, A.J., 2018. Sarcasm detection on facebook: A supervised learning approach, in: Proceedings of the 20th International Conference on Multimodal Interaction: Adjunct, pp. 1–5.
[4] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .
[5] Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., Mihalcea, R., 2018. Cascade: Contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805.06413 .
[6] He, S., Guo, F., Qin, S., 2020. Sarcasm detection using graph convolutional networks with bidirectional lstm, in: Proceedings of the 2020 3rd International Conference on Big Data Technologies, pp. 97–101.
[7] Khatri, A., et al., 2020. Sarcasm detection in tweets with bert and glove embeddings. arXiv preprint arXiv:2006.11512 .
[8] Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 .
[9] Kumar, A., Narapareddy, V.T., Srikanth, V.A., Malapati, A., Neti, L.B.M., 2020. Sarcasm detection using multi-head attention based bidirectional lstm. Ieee Access 8, 6388–6397.
[10] Liang, B., Su, H., Gui, L., Cambria, E., Xu, R., 2022. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. Knowledge-Based Systems 235, 107643.
[11] Lou, C., Liang, B., Gui, L., He, Y., Dang, Y., Xu, R., 2021. Affective dependency graph for sarcasm detection, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1844–1849.
[12] Mandal, P.K., Mahto, R., 2019. Deep cnn-lstm with word embeddings for news headline sarcasm detection, in: 16th International Conference on Information Technology-New Generations (ITNG 2019), Springer. pp. 495–498.
[13] Misra, R., Arora, P., 2019. Sarcasm detection using hybrid neural network. arXiv preprint arXiv:1908.07414 .
[14] Rajadesingan, A., Zafarani, R., Liu, H., 2015. Sarcasm detection on twitter: A behavioral modeling approach, in: Proceedings of the eighth ACM international conference on web search and data mining, pp. 97–106.
[15] Xu, L., Xu, V., 2017. Sarcasm detection, Standford University.