IJICTR
International Journal of Information & Communication Technology Research

Volume 16- Number 3 – 2024 (45 -53)

A Deep Learning Approach for Sarcasm Detection on Twitter

Mohammad Javad Shayegan*
Department of Computer Engineering
University of Science and Culture
Tehran, Iran
shayegan@usc.ac.ir

Sara Kojouri Haraj
Department of Computer Engineering
University of Science and Culture
Tehran, Iran
sarakojouri68@yahoo.com

Received: 24 November 2023 – Revised: 7 January 2024 - Accepted: 14 April 2024

Abstract—**Sarcasm is a form of speech in which a person expresses his opinion implicitly. We may encounter a seemingly positive sentence in sarcasm, but the speaker has a contrary opinion. Sarcasm can be recognized in spoken language based on body language and the tone of voice. However, the lack of these features makes it difficult to recognize sarcasm in text. In recent years, Twitter has attracted much attention and has become a popular platform for sharing opinions and viewpoints. It is also common for people to use sarcasm on Twitter as an indirect means of expressing their opinions. The presence of sarcasm in the text makes it difficult to recognize the sentiment. Thus, it is necessary and inevitable to have solutions that can detect sarcasm. This study aims to provide a solution for detecting sarcasm on Twitter using deep learning approaches. This study used two Twitter datasets containing balance and imbalance data for modeling. The main idea of this research is to use additional features such as sentimental features, subjectivity, number of hashtags, and punctuation along with features that deep learning algorithms automatically extract. The impact of each feature is reported in the paper. In this research, GRU-Capsule based neural network has been used. According to the results, the proposed model has improved accuracy by 5% for balanced data and by 2% for imbalanced data.**

Keywords: component, sarcasm, sarcasm detection, deep learning, sentiment analysis

Article type: Research Article

Publisher: ICT Research Institute

[DOI: 10.61186/itrc.16.3.45]

I. INTRODUCTION

The increasing growth of social networks has caused people to use this platform to express their ideas and opinions. The data generated in social networks are personal and are very suitable for data analysis. Companies use these data to improve their position in the market [1]. Twitter is one of the social networks that enjoy global attention. The number of published tweets on Twitter is around 6,000 tweets per second [2]. Unlike other social networks, Twitter has its limitations and features, including that its character length is limited to a maximum of 280 characters. Limiting the number of characters leads to summarization or using slang terms. Also, the tendency of individuals not to express their opinions

* Corresponding Author

explicitly due to personal desire, personality traits, or considerations has led to opinions being expressed in other literary forms, such as humor and sarcasm. Unlike emotion [3-5] and traditional sentiment analyis [6, 7], sarcasm detection in a standalone textual input such as tweets is a complex task due to its below-the-surface semantics [8].

Sarcasm is a form of verbal humor used to humiliate or ridicule[9]. Detection of sarcasm is one of the biggest challange in natural language processing [10]. In computational models sarcasm detection is often performed without regard contextual features [11]. Due to this, sentiment analysis on Twitter is more difficult than on other web platforms, including websites and blogs, because it is much more difficult to discern the true meaning of sarcasm and to determine the correct polarity of sentences.

Recognizing sarcasm in spoken language is easier than text because of body language and tone of speech. Sentiment analysis algorithms are not able to detect sarcasm [12]. Therefore, research on sentiment analysis alone cannot meet the needs of sarcasm detection. Due to the large amount of sarcastic content present in sources such as Twitter, conventional text sentiment analysis methods alone are not effective in assessing real sentiment. sarcasm detection is a binary classification problem that both feature-rich traditional models and deep learning-based models are used to detect it [13].

Due to the many gaps in the accurate detection of sarcasm, in this study, deep learning methods in addition to sentimental characteristics, subjectivity, number of hashtags and punctuation marks have been used to detect sarcasm on Twitter. The remainder of this paper is structured as follows: Related works are presented in Section 2 and the research method in Section 3. Section 4 discusses the important findings of this study, and Sections 5 and 6 review the results and future works.

II. Related work

Prior to reviewing past works, it should be noted that this study primarily evaluated articles that addressed sarcasm directly. There are many studies in the field of sentiment analysis, for example, aspect-based sentiment analysis [14-20] and attention-based sentiment analysis [19, 21-23], which require considerable discussion. This section therefore only discusses previous works that focused on sarcasm.

There are two ways to distinguish sarcastic tweets from non-sarcastic tweets. The first method is manual annotation of the text, in which people label the text as sarcastic or non-sarcastic after reviewing it [9]. In the second method, tweets are collected using a tag that the tweet's author uses when publishing the tweet. Special tags such as #sarcasm, #not, etc., are used in this method to collect tweets that indicate something contrary to the content that was expressed by the user.

Poria et al. [24] proposed a framework that uses the convolutional neural network to learn sarcastic features from a set of sarcasm by extracting sentimental features. They also used character-based features for the first time in their work. They used CNN to extract sentimental attributes and SVM for final classification. Experiments were performed on three datasets which CNN-SVM performed better than SVM. Bouazizi et al. [25] examined sarcasm for four categories of characteristics related to sentiments, syntactic and semantic, punctuation, and pattern-related. The weight ratio of positive and negative words in a sentence was calculated, and then the sentimental weight of the sentence was determined. In addition to words, emojis and hashtags were also used to calculate the sentimental weight of the sentence. Their observations showed that the random forest classifier could perform better than the three support vector machine (SVM), k-nearest neighbor, and maximum entropy classifications.

Agrawal et al. [26] proposed the approach of emotional word embedded and word representation to identify sarcasm. Using the remote observer, they labeled the words into two categories: sentiment or emotion. They then introduced two architectural models: one for capturing sentiment information along with binary dimensions such as positive and negative (AWES-senti), and the other for decoding a stronger range of emotions and finding sentiment class (AWES-emo). They then used a long-term, short-term, two-way recurrent neural network model for training. Features were examined on two types of long and short text. It was observed that AWES-senti on short text documents and AWES-emo on long documents have better results than in previous works. Joshi and Parbhune [27] examined the use of the embedded word to record context incompatibility in the absence of sentimental words. They added a similarity score between two words to the set of features of their previous work to study its function on sarcasm recognition. They found the pairs of words with the most and the least similarity in a sentence and used the similarity score as a feature to sarcasm detection. They reported RNNs had more performance than naïve Bayes in detecting sarcasm.

Kumar et al. [28] introduced multi-head attention based on a deep neural network to detect sarcasm. They believe that the attention mechanism plays an important role in deep learning networks that capture explicit and latent contexts. The authors introduced a multi-head attention-based two-way long-term short-term memory network that makes good use of manually extracted features. Their model was able to help increase the performance of sarcasm recognition by identifying different parts of the sentence. Ræder et al. [29] used the features of pos tagging, punctuation, sentiment, and unigrams as features for recognizing sarcasm. They trained their model with logistic regression and SVM.

Ghosh and Veale [30] presented an in-depth learning-based architecture for sarcasm detection. They used multiple layers of CNN, LSTM, and DNN to detect sarcasm in their dataset. In addition, they used recurrent SVM to detect sarcasm [31]. They used the

[DOI: 10.61186/itrc.16.3.45]

properties of BOW, POS, sentiment, and hashtag to train SVM. The authors added the context tweet and the author attribute as attributes to their in-depth model and increased the detection accuracy on their data set. Later, Xiong et al. [32] identified sarcasm by working the Ghosh and Veale datasets with a self-matching network that is suitable for detecting inconsistencies between words. They used a BiLSTM that accepts the first hidden state as output. Xiong et al. [32] introduced two models called SMSD and SMSD-BiLSTM. In the first model, by entering the feature vector generated by the self-matching network to the predictive layer, they identify the sarcasm. This model did not perform well for sarcasm detection because it could not generate sentence synthesis information necessary for it [32]. Joshi et al. [33] employed word embeddings to capture context incongruity in the absence of sentiment words. They used semantic similarity by word vector similarity scores (as given by Word2Vec). Table 1 summarizes the related works.

In previous works, especially in combined works, the hashtag number feature has not been used to detect less sarcasm.

TABLE I. SUMMARY OF RELATED WORKS

No	Method	Achievements	Reference
1	CNN-SVM with sentimental, emotional, and personality-based features	CNN-SVM performs better than SVM.	[24]
2	Using the characteristics of sentiments, syntactic and semantic, punctuation and pattern-related.	Random forest classification is better than SVM and k-nearest neighbor and maximum entropy.	[25]
3	Word embedding and word display	AWES-senti is suitable for short text documents and AWES-emo is suitable for long text documents	[26]
4	The similarity score between two words plus the characteristics of the previous work	The word embedding feature alone is not sufficient to detect sarcasm and should be used in conjunction with other features.	[33]
5	Naïve Bayes and the RNN network	The recurrent neural network is more suitable than the naïve Bayes network because it can store sentence information.	[27]
6	multi head attention based on Bilateral Deep Neural Network (BiLSTM)	The use of multi-head attention has increased the accuracy of sarcasm detection on the BiLSTM network.	[28]
7	Regression logistic network method and SVM	According to their proposed method, SVM was able to	[29]
	using the features of punctuation, sentiment and unigrams to detect sarcasm	detect sarcasm better than regression logistics	
8	Using multiple layers of CNN, LSTM and DNN to detect sarcasm. They also used an SVM to detect sarcasm. BOW, POS, sentiment and hashtag properties were used.	Increased the accuracy of sarcasm detection by deepening the number of layers	[30]
9	added the author's tweets context and attributes to their previous work	They used two added features and other previous features and increased the accuracy of the sarcasm detection.	[31]
10	Using self-matching network and BiLSTM to detect sarcasm.	It was observed that the self-matching network alone is not sufficient to detect sarcasm	[32]

Besides sentiment, subjectivity, and punctuation, we also used the hashtag attribute in this study. As well, the impact of each attribute on sarcasm recognition was examined separately.In addition, we used a capsule network that can accept the vector as input. Capsule networks dissipate useful features better in the network and perform better than other methods for detecting sarcasm.

III. RESEARCH METHOD

The overall research approach is summarized in Fig. 1, and we will describe each of the sections below.

In this study, two datasets of balance and imbalance data have been used to train the model. In a balanced dataset, the number of sarcastic and non-sarcastic texts is equal, while in an imbalanced dataset, the number is not equal. For the imbalanced data set, the same as [29, 34, 35], the data collected by [36] was used. In this method, using the Twitter API, sarcastic data was collected using #sarcasm. Non-sarcastic datasets include tweets that do not contain this hashtag.

For the balanced data set, the data set collected by Ghosh and Veale [30] was used.

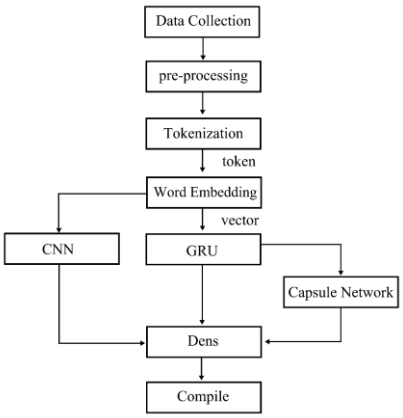

Figure 1. Overall research approach

[DOI: 10.61186/itrc.16.3.45]

They also collected sarcasm data using #sarcasm, #sarcastic and #ironie, and tweets that do not contain these hashtags are collected as non-sarcastic tweets. Table 2 shows the summary of datsets.

TABLE II. SUMMARY OF DATASETS SUMMARY OF RELATED WORKS

Non-Sarcastic Tweets	Sarcastic Tweets	dataset
24000	24000	balance
28000	30000	imbalance

A. Preprocessing

Normalization is the first step in data preprocessing. Mentions, retweets, URLs and non-ASCII letters were removed using regular expressions. In addition, the hashtags #sarcasm, #sarcastic, and #ironie were removed. In the case of other hashtags, only the hashtag character was removed. Observations indicate that users use these hashtags in a sentence, thus removing them would result in incomplete tweets.The use of acronyms is widespread on social networks. A dictionary of acronyms commonly used in English was created and replaced in the text using a function. This was also done for abbreviations.

One of the most important steps in preprocessing is Tokenization. The goal of tokenization is to find words in a sentence. Tokenize was performed using the Word Tokenizer function. Stemming and lemmatization are two important steps in natural language preprocessing. The lemmatization method was used for stemming. WordNetLemmatizer was used for this purpose. To remove the pause words, the most repetitive words were found in the sentences with a regular phrase, and among them, the words that did not change the sentiment of the sentence were removed.

Sarcasm could not be detected well in short sentences. Thus, sentences shorter than seven words were excluded from the dataset. This study uses the 200-dimensional Twitter GloVe vector, which has been trained on 2 billion tweets.

B. Features

Deep learning algorithms are able to extract local features automatically [37]. In addition to these local features, some features were used to train higher layers to represent a larger group of words in sentences. Sentimental characteristics, subjectivity and the number of punctuation marks and the number of hashtags are among the characteristics that have been considered in this research to identify sarcasm.

One of the signs of sarcasm in the text is the excessive use of punctuation marks, emoticons, repetition of a letter more than twice. In this study, the number of punctuation marks used in the sentence has been considered as one of the signs of the presence of sarcasm in the text and as a feature for recognizing sarcasm. Exclamation marks (!), Question marks (?), Quotation marks (:), commas (,) and punctuation marks were used as marking features.

The TextBlob library, which is used to process textual data, was used to apply sentimental attributes and subjectivity. Sentimental characteristics play an important role in recognizing sarcasm. In sarcastic sentences, the feeling changes from positive to negative or vice versa. But sentic patterns positively recognize real sentiments. For example, in the sentence "I love the pain of breaking", considering the presence of the word "love", the feeling of a positive sentence is considered, but it is not really so, and this sentence contains sarcasm. Sarcastic sentences are often expressed subjectively in which the person ironically expresses dissatisfaction with a problem. So, this feature can help us identify sarcasm.

C. Model Training

Deep text learning models can process text as well as time series and data sequences. The most basic algorithms for sequence processing are convolutional networks and RNNs. Convolutions are faster than RNN for text classification and time series prediction [38]. These networks are able to learn hierarchical features.

- Model training with CNN

  One-dimensional convolution is suitable for learning the properties of the text and considers each sequence as a general sentence [38]. One-dimensional convolution recognizes local patterns in a sequence and a pattern learned in a particular situation in a sentence in a different situation [38].

  The dropout rate, which was used to randomly remove units in the CNN neural network, was chosen by trial and error as 0.5. Fully connected architecture is used to classify the data into two categories, sarcastic and non-sarcastic. The output of the fully connected layer passes through three dense layers. Finding the right number of layers in a dense layer is a hyperparameter. Increasing the number of dense layers increases recall, but increasing it too much also causes the network to overfit [38]. After trial and error, the values of 32, 16, and 1 were considered as the dimensions of the first, second and third dense layer output space, respectively. Relu activity function was considered for one-dimensional convolution and two dense layers.

  The sigmoid activation function was used for the last dense layer that performs the classification task. Adam optimization function was used to evaluate the model. To measure the recall of the model during the training process, we need a parameter; the loss function does this. For the classification of sarcasm, we used ‘Binary crossentropy, which performs the work of binary classification, as a loss function (based on [30]). Accuracy, recall, F1 and sensitivity criteria were defined and used to evaluate the model.

- Model training with GRU-Capsule based network

  GRU and capsule networks were used to train the model. GRU was created to solve the problem of recurrent networks that are unable to store past information and can not maintain long-term dependencies [39]. The GRU settings obtained by trial and error are summarized in Table 3.

TABLE III. GRU-CAPSULE HYPERPARAMETERS

[DOI: 10.61186/itrc.16.3.45]

units	activation function	dropout	Recurrent_ dropout	Return_ sequence
145	Relu	0.5	0.5	True

Capsule networks accept the vector as input. Using GRU neural networks, different properties are extracted from the input dataset and fed to the encapsulated networks as a vector. Capsule networks are a group of neurons whose activity vector represents the various parameters of an entity. The size of this vector indicates the probability of recognizing the feature and its orientation of the sampling parameters. The architecture intended for the training is taken from the architecture presented by [39] in Fig. 2.

GRU encrypts sentences and extracts hidden vectors. Encrypted sentences in the form of hidden vectors are the input to the capsule. H=[h_1 , h_2, …,h_{N_s}] are the hidden vectors of a sentence and N_s is the number of words encoded by GRU. The average of the hidden vectors [39] is:

$$v_s = \frac{1}{N_s}\sum_{i=1}^{N_s} h_i \qquad (1)$$

Operation in the capsule network is performed in the following four steps[35]:

Multiplication of input vector matrix by weight coefficients: In this step, the input vector, which is output from GRU, and we showed with h_i, is multiplied by the weight matrix w.

$$u_{j|i} = h_i W_{ij} \qquad (2)$$

The product of vector $u_{j|i}$, which represents the spatial relationship between the low-level and high-level properties.

The numerical weighting of vectors created in the previous step: Dynamic routing is used for weighting. This is the point of difference with the convolutional network in which pooling is used. The low-level capsule uses a dynamic routing mechanism to detect which output of the high-level capsule it produces is closest to the output of the capsule and sends the output to that capsule.

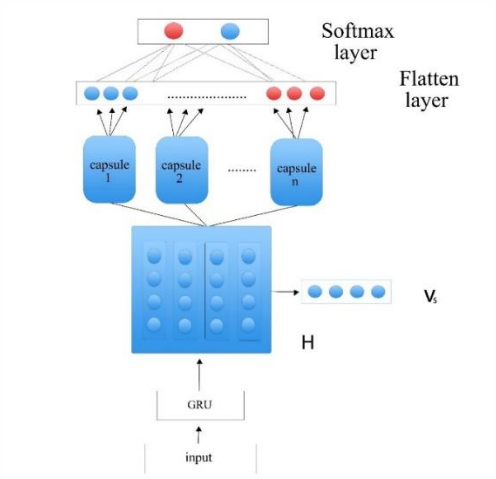

Figure 2. GRU-Capsule architecture (taken from the architecture of [39])

The numerical weighting of vectors created in the previous step: Dynamic routing is used for weighting. This is the point of difference with the convolutional network in which pooling is used. The low-level capsule uses a dynamic routing mechanism to detect which output of the high-level capsule it produces is closest to the output of the capsule and sends the output to that capsule.

Addition of weighted input vectors: The vectors that have been weighted in the previous step are added together. The total input to the S_j capsule is the sum of the weights around the predicted vectors $u_{j|i}$.

$$s_j = \sum_i u_{j|i}\, c_{ij} \qquad (3)$$

Compression of received vectors: Each capsule will have an output vector that indicates the probability of corresponding properties with the length of the vector. Capsules receive hidden vectors as input, and dynamic routing operations perform weight determination. The weighted input vectors are added together and the activation function is applied to them. The activator function is called 'Squash', which takes a vector and converts it into a vector with a length of one, but without changing the vector's direction.

$$V_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \qquad (4)$$

v_j is the output vector of capsule j and s_j is the input vector of capsule j.

The Flatten layer is sent to the dense layer. In addition, the four mentioned features will be combined with the features extracted by the Gru-Capsule based network architecture and sent to the dense layer. Table 4 lists some settings of capsule network hyperparameters obtained by trial and error.

TABLE IV. HYPERPARAMETERS FOR CAPSULE NETWORK

Number of Capsule	Capsule Dimension	Routing	dropout	adam
10	32	3	.5	.01

IV. RESULTS

The proposed networks were implemented on two sets of balanced and imbalanced datasets, the results of which are described below. The results are then compared with previous work and analyzed at the end.

A. Results of the imbalanced dataset

Convolution networks can detect local patterns in a sequence and can be used to identify patterns in text [38]. Sarcasm is better detected by convolution networks than other machine learning algorithms [40]. This is consistent with the obtained results. The values obtained for F1 and precision using the CNN architecture on the test data are 80% and 81%, respectively.

However, in other machine learning methods, the values obtained by the SVM algorithm and logistic regression [29] were reported as 79% and 56%, respectively. The results are shown in Table 5.

[DOI: 10.61186/itrc.16.3.45]

The value of f1 reported in the proposed method was better than both the SVM algorithm and logistic regression by [29] and Naïve Bayes by [41]. This is because of CNN's ability to detect hidden patterns in the text.

Fig. 3 shows that CNN was more capable of detecting sarcasm in the text than other machine learning algorithms.

A convolutional neural network is able to automatically extract key properties from the training data, and, after performing convolution operations on the data, form a global feature vector.To evaluate the performance of the GRU-Capsule based network and the effect of each feature, each feature was first injected separately into the softmax layer and the accuracy of the model was calculated.

Then all these features were injected into the model and the accuracy of the model was calculated. Also, the accuracy of the model was measured alone and without any features. Table 6 shows the values of different criteria. These results are the average of three runs on the dataset.

TABLE V. COMPARISON OF BASE METHOD AND CNN ON IMBALANCED DATASETS

Model	F1	Precision
SVM [8]	79	-
Logistic regression [8]	71	-
Naïve Bayes[15]	56	78
CNN's proposed approach	80	81

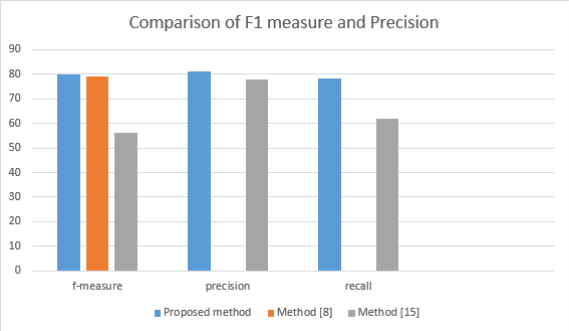

Figure 3. Comparison of evaluation metrics

Table 6 shows the effect of each feature for improving the sarcasm detection method. For instance, the hashtag feature has had the highest improvement effect on F1. This matter indicates that people use more hashtags to convey their meaning in sarcastic texts. This is because people often use abbreviations in their sarcasm and use hashtags to convey the meaning of their words to others.

Fig. 4 shows a comparison of F1 and precision values on the proposed method and the base method. As can be seen, the accuracy of the model increased when we used all features together with the GRU+capsule network.

TABLE VI. COMPARISON OF METRICS PER FEATURE ON AN IMBALANCED DATASET

Model	F1	Recall	Precision	Accuracy
gru+capsule	81.70	80.08	79.66	79.70
gru+capsule+ sentimental feature	83.70	81.80	81	79.40
gru+capsule+subj ectivity feature	83.82	84.39	78.74	79.49
gru+capsule+ number of punctuation marks feature	82.07	83.11	80.16	81.03
gru+capsule+ hashtag number feature	84.28	83.40	79.58	80.01
gru+capsule + all features	**86.32**	**82.46**	**82.13**	**81.70**

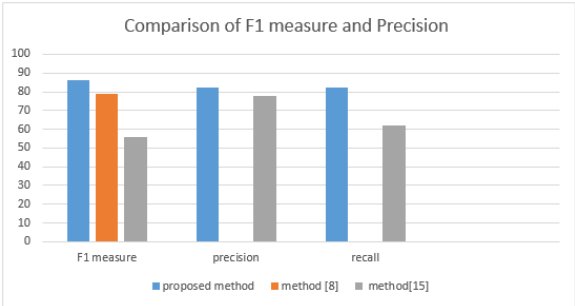

Figure 4. Comparison of GRU-Capsule based network with other methods

B. *Results of the balanced dataset*

The results of training in the convolutional network on a balanced dataset are described in Table 7.

TABLE VII. COMPARISON OF BASIC AND CNN METHODS ON BALANCED DATASETS

Model	F1	recall	precision	accuracy
Recursive SVM [31]	73.2	72.1	74.3	-
CNN+LSTM+DNN [29]	73.2	73.2	73.1	-
SMSD [30]	74.42	72.56	76.39	**80.09**
SMSD-BiLSTM [30]	74.40	68.80	78.65	78.37
CNN proposed approach	**77.69**	**75.72**	**80.90**	78.66

The convolutional networks have a higher detection accuracy than compared models because they are able to detect patterns within sequences. Furthermore, we injected the features derived from the GRU-Capsule-based network, along with four categories of sentimental attributes, punctuation marks, hashtags, and subjectivity, into the softmax layer to detect sarcasm. The results of this experiment on a balanced dataset are listed in Table 8.

The results of Table 8 indicate that, compared to the mentioned methods, the proposed approach is more accurate at detecting sarcasm. Table 9 presents the impact values of each feature on the balanced dataset.

[DOI: 10.61186/itrc.16.3.45]

According to Table 9, with employing the features in the network, the accuracy of the model has increased around 5.55%. A comparison between the proposed approach and other approaches is shown in Fig. 5.

As Table 6 examines the impact of each feature in detecting sarcasm on an imbalanced dataset, Table 9 tests the same approach for a balanced dataset. As seen in Table 6, the hashtag feature again had a greater impact on the F1 measure.

On the basis of various tests conducted on the datasets, it was determined that convolution networks performed better than other approaches. Instead, convolutional networks have the problem of ignoring some features for Max polling. In addition, the output of the convolutional network is numerical. Capsule networks were able to solve this problem.

This is also evident in the experiments performed. Other features were considered to enrich the capsule networks and increase the accuracy of sarcasm detection.

By injecting sentiment features, punctuation, hashtag and subjectivity and calculating the accuracy of the model, it was observed that each of these features alone can increase the accuracy of the model. By combining these features, we were able to increase the model's accuracy on a balanced data set by up to 5% and improve sarcasm detection. Also, the value of F1 on the balanced data set increased by 5.45% and the values of precision and recall increased by 3.43% and 9.42%, respectively. For imbalanced data sets, accuracy, recall, precision, and f1 measure increased by 2%, 2.38%, 2.47%, and 4.62%.

TABLE VIII. COMPARISON OF BASIC AND NETWORK METHODS BASED ON GRU-CAPSULE ON A BALANCED DATASET

Model	F1	recall	precision	accuracy
CNN+CNN [31]	73/2	72.1	74.3	-
LSTM+LSTM [31]	73.2	73.2	73.1	-
CNN+CNN+LSTM +LSTM+ DNN [29]	84.1	85.2	**84.8**	-
SMSD [30]	74.42	72.56	76.39	80.09
SMSD-BiLSTM[30]	74.40	68.80	78.65	78.37
GRU-Capsule proposed approach	**85.75**	**90.84**	81.99	**85.15**

TABLE IX. TABLE 1. COMPARISON OF THE IMPACT OF EACH FEATURE ON A BALANCED DATASET

Model	F1	Recall	Precision	Accuracy
gru+capsule	80.30	81.42	78.56	79.60
gru+capsule+ sentimental feature	82.21	85.10	80.30	81.99
gru+capsule+subj ectivity feature	81.99	82.75	79.21	81.22
gru+capsule+ number of punctuation marks feature	81.97	83.50	78.94	81.24
gru+capsule+ hashtag number feature	82.48	84.33	80.22	82.20
gru+capsule + all features	**85.75**	**90.84**	**81.99**	**85.15**

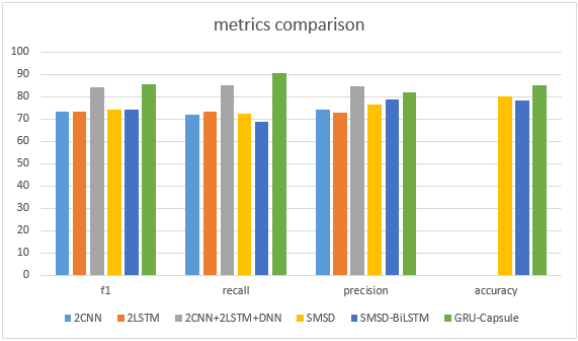

Figure 5. Comparison of different criteria on GRU-Capsule based model and other methods

CONCLUSION

This study aimed to investigate the effect of sentimental characteristics, subjectivity, the number of hashtags, and punctuation marks to detect sarcasm on the GRU-capsule network. In this method, we used two types of balanced and imbalanced data sets and calculated the impact percentage of each of these features. Finally, the effect of using all the features together was examined. Capsule networks were created to solve the problem of aggregation in convolutional networks.In this study, it was observed that these networks try to identify features more accurately by purposefully exchanging outputs with each other. As the nature of capsule networks was to solve the problem of convolutional networks, in this study, these networks were able to perform better than convolutional networks.

It was also observed that the capsule and convolutional networks perform much better than other machine learning models. To increase the detection accuracy, we added features to the network. It was observed that the addition of these features increases the accuracy of sarcasm detection. We used all of these features together to reach maximum detection. The values of accuracy, recall, precision, and F1 measure on the imbalanced data set increased by 2%, 2.38%, 2.47%, and 4.62%, respectively. For the balanced data set, the values of accuracy, recall, precision, and F1 increased by 5.55%, 9.42%, 2.47%, and 4.62%, respectively.

Sarcasm detection is a relatively new field in natural language processing and sentiment analysis. Using other features, such as calculating the feelings of the emojis and comparing it with the tweet text, can be one of the signs of sarcasm in the text. One of the future tasks is to focus more on emojis to recognize sarcasm. Detecting sarcasm in non-English language tweets and measuring their effectiveness can also be part of future work.

[DOI: 10.61186/itrc.16.3.45]

Ethics declarations

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors**.**

Conflicts of interest: The authors declare that they have no conflict of interest.

REFERENCES

[1] J. Aquino, "Transforming social media data into predictive analytics," *CRM Magazine*, vol. 16, no. 11, pp. 38-42, 2012.

[2] D. Sayce. "The Number of tweets per day in 2022." https://www.dsayce.com/ (accessed October 6, 2022.

[3] M. S. Akhtar, A. Ekbal, and E. Cambria, "How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes]," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 64-75, 2020.

[4] S. Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, "All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework," *IEEE transactions on affective computing*, 2019.

[5] B. Zhang, E. M. Provost, and G. Essl, "Cross-corpus acoustic emotion recognition with multi-task learning: Seeking common ground while preserving differences," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 85-99, 2019.

[6] C. Clavel and Z. Callejas, "Sentiment Analysis: From Opinion Mining to Human-Agent Interaction," *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 74-93, 2016, doi: 10.1109/TAFFC.2015.2444846.

[7] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis," *Cognitive Computation*, vol. 10, no. 4, 2018.

[8] M. Bedi, S. Kumar, M. S. Akhtar, and T. Chakraborty, "Multi-modal sarcasm detection and humor classification in code-mixed conversations," *IEEE Transactions on Affective Computing*, 2021.

[9] E. Riloff, Qadir, A., Surve, P., De Silva, L., Gilbert, N. and Huang, R., "Sarcasm as contrast between a positive sentiment and negative situation," *In Proceedings of the 2013 conference on empirical methods in natural language processing*, vol. 704-714, 2013.

[10] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1-22, 2017.

[11] A. Kumar and G. Garg, "Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets," *Journal of ambient intelligence and humanized computing*, pp. 1-16, 2019.

[12] P. Katyayan and N. Joshi, "Sarcasm Detection approaches for English language," in *Smart Techniques for a Smarter Planet*: Springer, 2019, pp. 167-183.

[13] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm detection using multi-head attention based bidirectional LSTM," *Ieee Access*, vol. 8, pp. 6388-6397, 2020.

[14] Z. Hu, Z. Wang, Y. Wang, and A.-H. Tan, "MSRL-Net: A multi-level semantic relation-enhanced learning network for aspect-based sentiment analysis," *Expert Systems with Applications*, p. 119492, 2023.

[15] M. Pontiki *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *ProWorkshop on Semantic Evaluation (SemEval-2016)*, 2016: Association for Computational Linguistics, pp. 19-30.

[16] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[17] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813-830, 2015.

[18] J. Su, S. Yu, and D. Luo, "Enhancing aspect-based sentiment analysis with capsule network," *IEEE Access*, vol. 8, pp. 100551-100561, 2020.

[19] H. Wu, C. Huang, and S. Deng, "Improving aspect-based sentiment analysis with Knowledge-aware Dependency Graph Network," *Information Fusion*, vol. 92, pp. 289-299, 2023.

[20] G. Zhao, Y. Luo, Q. Chen, and X. Qian, "Aspect-based sentiment analysis via multitask learning for online reviews," *Knowledge-Based Systems*, p. 110326, 2023.

[21] P. Bhuvaneshwari, A. N. Rao, Y. H. Robinson, and M. Thippeswamy, "Sentiment analysis for user reviews using Bi-LSTM self-attention based CNN model," *Multimedia Tools and Applications*, vol. 81, no. 9, pp. 12405-12419, 2022.

[22] T. Ghosh, M. H. Al Banna, M. J. Al Nahian, M. N. Uddin, M. S. Kaiser, and M. Mahmud, "An attention-based hybrid architecture with explainability for depressive social media text detection in Bangla," *Expert Systems with Applications*, vol. 213, p. 119007, 2023.

[23] G. Khodabandelou, H. Moon, Y. Amirat, and S. Mohammed, "A fuzzy convolutional attention-based GRU network for human activity recognition," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105702, 2023.

[24] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," *arXiv preprint arXiv:1610.08815*, 2016.

[25] M. Bouazizi and T. O. Ohtsuki, "A pattern-based approach for sarcasm detection on twitter," *IEEE Access*, vol. 4, pp. 5477-5488, 2016.

[26] A. Agrawal and A. An, "Affective representations for sarcasm detection," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1029-1032.

[27] A. M. Joshi and S. S. Prabhune, "Sarcasm Detection on Twitter Data: Generative Versus Discriminative Model," in *Advanced Computing Technologies and Applications*: Springer, Singapore, 2020, pp. 247-254.

[28] A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," *IEEE access*, vol. 7, pp. 23319-23328, 2019.

[29] J. G. C. M. Ræder and B. Gambäck, "Sarcasm Annotation and Detection in Tweets," in *International Conference on Computational Linguistics and Intelligent Text Processing*, 2017: Springer, pp. 58-70.

[30] A. Ghosh and T. Veale, "Fracking sarcasm using neural network," in Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, 2016, pp. 161-169.

[31] A. Ghosh and T. Veale, "Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 482-491.

[32] T. Xiong, P. Zhang, H. Zhu, and Y. Yang, "Sarcasm detection with self-matching networks and low-rank bilinear pooling," in *The World Wide Web Conference*, 2019, pp. 2115-2124.

[33] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, and M. Carman, "Are word embedding-based features useful for sarcasm detection?," *arXiv preprint arXiv:1610.00883*, 2016.

[34] A. Abirami and V. Gayathri, "A survey on sentiment analysis methods and approach," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*, 2017: IEEE, pp. 72-76.

[35] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *arXiv preprint arXiv:1710.09829*, 2017.

[36] M. Cliche. "The sarcasm detector." http://www.thesarcasmdetector.com/. (accessed.

[37] S. Tam, R. B. Said, and Ö. Ö. Tanriöver, "A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification," *IEEE Access*, vol. 9, pp. 41283-41293, 2021.

[38] F. Chollet, Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG, 2018.

[39] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1165-1174.

[40] I. Touahri and A. Mazroui, "Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied

[DOI: 10.61186/itrc.16.3.45]

sarcastic features," *Knowledge-Based Systems*, vol. 227, p. 107232, 2021.

[41] C.-C. Peng, M. Lakis, and J. W. Pan, "Detecting sarcasm in text," *cs229. stanford. edu/proj2015/044 report. pdf*, 2015.

Mohammad Javad Shayegan is an Associate Professor at the Department of Computer Engineering at the University of Science and Culture, Tehran, Iran. He received his Ph.D degree in Information Technology and Multimedia Systems from the University Putra Malaysia in 2008. He is the founder and program chair for the IEEE International Conference on Web Research and Editor-in-Chief for Intrenational Journal of Web Research. His research intereset includes data and web mining, distributed systems and e-business

Sara Kojouri Haraj is received master of software engineering from department of computer engineering at the University of Science and Culture, Tehran, Iran. Her research interest include data mining and machine learning.

[DOI: 10.61186/itrc.16.3.45]