Contents lists available at ScienceDirect

Technology in Society

journal homepage: http://www.elsevier.com/locate/techsoc

Short communication

A pragmatic and intelligent model for sarcasm detection in social media text

Mayank Shrivastava, Shishir Kumar [*]

Department of Computer Science and Engineering, Jaypee University of Engineering and Technology, Guna, India

ARTICLE INFO

Keywords:
Sarcasm
Irony
BERT
Text analysis
Computational
Intelligent

ABSTRACT

The world has now become an ecumenical village because of the Internet. Online platforms like e-commerce sites, search engines, social media have convoluted with the general routine of daily life. Social sites such as Twitter and Facebook have a user population larger than most of the countries, due to which communication is now largely shifted to text-based communication from verbal communication. This research investigates a common yet crucial problem of sarcasm detection in text-based communication. To prevent this problem a novel model has been proposed based on Google BERT (Bidirectional Encoder Representations from Transformers) that can handle volume, velocity and veracity of data. The performance of the model is compared with other classical and contemporary approaches such as Support Vector Machine, Logistic Regression, Long Short Term Memory and Convolutional Neural Network, BiLSTM and attention-based models which have been reported to be used for such tasks. The proposed model establishes its competence by evaluation on different parameters such as precision, recall, F1 score and accuracy. The model is built with the hope that it may help not only the government but also the general public to build a safer and technologically advanced society.

1. Introduction

The advent of social media has instigated a flow of information at an unprecedented level. Online presence is the new norm of society. Facebook with its 2.4 billion monthly active users generates a large amount of information [1]. Twitter shares a hefty amount of data generation, producing more than 6000 tweets every second. Sites like Twitter, Facebook are just the tip of a huge iceberg of data generation. Due to the growth in popularity, social media are now well accepted as a source of news. However, society has witnessed multiple incidences where misinterpreted words led to maleficent development.

Words articulated together may available in various forms such as satire, irony and sarcasm. Irony is a figure of speech that is contrary to what is said in literal [2] whereas, sarcasm is irony used in the service of mocking something or someone. Although effective examination of words may suggest irony and sarcasm, the detection is a perplexing task. It is mostly contingent on the modulation of voice and other nonverbal cues like faces and hand gestures and as a result, the detection becomes difficult in text-based communication [3] which is deprived of both. As the means of communication in today's era are largely shifted toward text-based communication, the detection gets troublesome. Henceforth, a practical model is required that can effectively detect sarcasm in text.

Misinterpreted sarcastic words impart a great toll on progressive society. Although, in the past years, various researches have been surfaced which provided quite satisfactory results; nonetheless the progressive researches in the field of NLP (Natural Language Processing) demand the extension and applicability of new models in this field. This study aims to utilize such advances to propose a pragmatic model built with transformers that can be applied at the very premise of data generation.

The main research objectives of this paper are:

- Discussion of various sarcasm/irony datasets available for computational applications.
- Designing and analysis of a practical text-based sarcasm detection model.
- Analysis of the role of hyperparameters.

The rest of this paper is organized as follows: Section 2 discusses the background of sarcasm and various datasets, detection mechanisms and techniques. Section 3 briefly introduces BERT followed by the proposed model in section 4. In section 5, experiment settings including the used

[*] Corresponding author.
E-mail addresses: email2shrivastava@gmail.com (M. Shrivastava), dr.shishir@yahoo.com (S. Kumar).

https://doi.org/10.1016/j.techsoc.2020.101489
Received 20 April 2020; Received in revised form 5 November 2020; Accepted 9 December 2020
Available online 14 December 2020

dataset, evaluation metrics and baseline models are discussed. Section 6 presents the results and analyses in both a quantitative and a qualitative manner. Section 7 concludes the paper with possible future extensions.

2. Background

2.1. Sarcasm

In the year 1775, Dr. S. Johnson [4] defined the term 'sarcasm' as 'a mode of speech in which the meaning is contrary to the words.' Kreuz and Glucksberg [5] discussed the similarities and dissimilarities of sarcasm and irony. Sarcasm is targeted to a specific victim whereas irony does not.

Rajadesingan et al. [6] investigated sarcasm used to portray inverse sentiments, express emotions, even to express high functional address or an expression of conversance. A written form of sarcastic expression may involve prosodic variations. Riloff et al. [7] observed sarcasm expressed in contrast situations like a positive expression followed by a negative situation.

Sarcasm is conveyed in a variety of forms. Various pieces of literature are available which employed other formats of communication in the detection of sarcasm as well. Multimodal approaches have been observed using visual along with textual modes in [8–11]. Mishra and Bhattacharya [12] proposed gaze-based technology to understand sarcasm from data.

Recent trends in the detection of sarcasm have been observed in different languages like Arabic [13–15], Dutch [16], English [7,17,18], French [19], Italian [20], Spanish [21] and so on. Refer to Table 1.

2.2. Detection mechanism

The section discusses the detection mechanisms of sarcasm and irony. The study of reviewed literatures suggested two mechanisms viz. linguistic detection and computational detection.

Further descriptions are categorized in the following subsections.

2.2.1. Linguistic detection

Earlier methodologies involve detection based on linguistic. Kreuz and Glucksberg [5] detected sarcasm as a function of polarity and victim availability. In the same study, it was identified that positive sentences tend to be more sarcastic as compared to negative sentences. A study conducted by Rockwell [22] explored the possibility of detection of sarcasm using verbal and nonverbal cues. Cheang and Pell [23] studied the detection of sarcasm on recorded voices by non-native English speakers as compared to native English speakers. The investigations from literature incline that the early methods of detection of sarcasm were largely linguistic instead of computational. Sooner these studies laid the groundwork for computational detection of sarcasm and irony.

Table 1
Details of sarcasm/irony detection datasets.

S. no.	Source	Modality	Language
1	Karoui et al. [13]	Textual	Arabic
2	Ghanem et al. [14]	Textual	Arabic
3	Abu Farha & Magdy [15]	Textual	Arabic
4	Liebrecht et al. [16]	Textual	Dutch
5	SemEval 2018 Task 3 [17]	Textual	English
6	Riloff et al. [7]	Textual	English
7	iSarcasm [18]	Textual	English
8	Karoui et al. [19]	Textual	French
9	Gianti et al. [20]	Textual	Italian
10	Ortega-Bueno et al. [21]	Textual	Spanish
11	Rockwell [22]	Audio	English
12	Cheang& Pell [23]	Audio	English
13	Schifanella et al. [8]	Textual, Visual	English
14	Cai et al. [9]	Textual, Visual	English
15	Sinha et al. [11]	Textual, Visual	English
16	Mishra & Bhattacharyya [12]	Textual, Visual	English

2.2.2. Computational detection

The availability of a large amount of data facilitates the computational detection of sarcasm. The known study of computational detection of sarcasm was commenced first by Tepperman et al. [24]. In the study, the phrase "yeah right" was used to mark sarcastic comments. It was based on dialogue transcripts and reported an accuracy of 87% and an F1 score of 70. Davidov et al. [25] used a dataset of tweets and product reviews from the eCommerce website, Amazon, and claimed an accuracy of 89.6% and an F1 score of 54.5. In another study, Davidov et al. [26] used semi-supervised learning techniques in detecting sarcasm based on hashtags and smileys from tweets. In the contemporary period, Twitter emerged as a goldmine for researchers in the field of Natural Language Processing. Soon the applications of computational detection had drawn the attention of various researchers. Zhang et al. [27] used recurrent neural networks to detect sarcasm. Son et al. [28] investigated the usage of CNN, LSTM, BiLSTM. The use of attention has also been discussed in [28]. The mechanism was further extended to multimodal forms like visual and textual [10,29].

Industry 4.0 with its social media sites like Twitter, Facebook and Instagram allowed researchers to access an abundant data repository required for the computational models. This resulted in a wide variety of techniques used for the detection of sarcasm.

2.3. Techniques used

Even though a large number of approaches are predicated on both linguistically and computationally developed techniques handling miscellanea of data, this paper fixates on computationally developed techniques based on textual data.

In the subsections, approaches of automatic detection of sarcasm and irony, primarily based on text data, are discussed. The classifications are based on characteristic operations.

- Rule-based

One of the foremost techniques is rule-based. This technique is mostly impressed by linguistic models where certain rules are selected which can imply the presence of sarcasm in the given data. A rule like selecting keywords such as 'Yeah right!' in spoken dialogues is used by Tepperman et al. [24]. One of the most used approaches is using hashtag keywords and smileys in tweets [25,26,30]. One of the interesting approaches is to generate a parse tree to detect negative phrases in a positive sentence. This approach is developed by Bharti et al. [31]. The success of this approach relied upon the availability of certain words in most of the detection cases. However, the nature of sarcasm is often implicit incongruity which often remains undetected by rule-based techniques. To address such problems, feature-based techniques have been developed.

- Features-based

In general, data is divided into training data and testing data. The feature-based approaches involve the extraction of features from the training data and using these to detect the gist of the testing data. These methods are extensions of rule-based approaches that can detect various forms of sarcasm as well. One of the successful pioneer approaches is Bag-Of-Words. According to this approach, the words of the sample sentence are bound together and labeled [32].

Liebrecht et al. [16] suggested a bi-gram and tri-gram approach. Bi-gram approach (N = 2) and Tri-gram approach (N = 3) later developed into N-gram. N-gram based approach uses adjacent N words, where N is a count of words to be used together.

However, both of these techniques suffered a major drawback as different orders of the same set of words may deliver different meanings.

For example,

1. A good book on boring guys.
2. A boring book on good guys.

These two sentences contain a similar set of words with different meanings.

- Machine Learning-based

Natively, computers are computing machines. To harness this computing power; researchers utilize the concept of mathematical models in classification tasks. Researchers have used rule-based and feature-based to develop linear and multilinear classifiers for these tasks e.g. Naïve Bayes, a linear classifier model based on Bayesian probabilities. McCullum et al. [32] used naïve bayes for text classification. Similarly, Jain et al. [30] used naïve bayes and support vector machine (SVM) to extract emotions from a multilingual dataset. Gonzalez-Ibanez et al. [33] employed SVM with sequential minimal optimization and logistic regression to classify data from Twitter into sarcastic/non-sarcastic classes using positive and negative sentiments. Ptacek et al. [34] developed a language independent model on a dataset extracted from Twitter (Czech) and evaluated two machine learning approaches, Maximum Entropy and SVM, at sarcasm detection.

- Deep Learning-based

Prior to this approach, models followed a shallow architecture. With the introduction of deep architectures, the paradigm shifted. These approaches are able to extract multiple features at once. Collobert et al. [35] proposed the use of neural network architecture. A simple neural network outperformed contemporary techniques like POS tagging and named-entity relationship (NER) in [35]. This was still advocated as a shallow approach. But soon after that various deeper model based on derivatives of the neural network like Convolutional Neural Network (CNN) [36,37], Recurrent Neural Network (RNN) [27,38], Long Short Term Memory (LSTM) [17,39], Bi-LSTM [17,40], BiLSTM with Attention [28], Bidirectional Encoder Representations from Transformers (BERT) [41] are employed to same or similar tasks.

Due to the wide applicability of deep learning-based approaches with adequate performance, a deep learning approach is selected in this study to identify sarcasm. The proposed model is based on BERT (Bidirectional Encoder Representations from Transformers) [41]. BERT employs transfer learning which is accumulated in a bidirectional manner. The evaluation of the vanilla model reported outperforming other similar models such as OpenAI GPT and Embeddings from Language Models (ELMo) in various benchmark datasets such as General Language Understanding Evaluation (GLUE) and Stanford Question Answering Dataset (SQuAD).

3. Bidirectional Encoder Representations from Transformers (BERT)

BERT was developed by Devlin et al. [41] from Google AI Language in late 2018. It uses the concept of transformers as introduced by Vaswani et al. [42]. Transformers use attention which allows longer sequences to be processed by using encoders-decoders. Empirically, it is established bidirectional nature allows the model to understand the features from data efficiently. The model was developed around this establishment. The framework was introduced as a procedure with two phases; pre-training and fine-tuning.

During the former phase, Data is tokenized before pre-training using WordPiece embeddings. BERT is then pre-trained on BookCorpus [43] and Wikipedia (English). In the pre-training phase, two unsupervised methods are used, Masked Language Model (MLM) and Next Sentence Prediction (NSP).

a. In the case of MLM, a word is masked in a sentence at 80% of the time. 10% of the time the word was replaced with a different word and in the remaining time, the sentence remains unchanged. This data is fed to the model which then converges with suitable weights by identifying masked word. In this process, the model understands the dynamics of language. The idea is to make the model grasp the gist as humans perceive language. BERT analyzes each word with respect to its neighbor words to understand the meaning and context of the word. For example, the phrase 'Mahatma Gandhi' may have a different meaning in different contexts as in Sentence 1 and 2.
Sentence 1: Mahatma Gandhi was a great leader from India.
Sentence 2: I live at the Mahatma Gandhi Road.
Here in sentence 1, the phrase Mahatma Gandhi was used as the name of a person from India but in sentence 2, the same phrase is used to address a place. Humans understand the difference by analyzing the context of the sentence. BERT Transformer does the same using self-attention.
Technically, the model predicts the masked word based on the context of the sentence by analyzing non-masked words. For this task, the sentence is initially passed through the embedding thus creating an output vector of the vocabulary, the output vector is then fed into a stack of transformers. The classification layer on the top of it uses a softmax function to predict the masked words.

b. In the second phase of NSP, the two sentences are fed into the system with a label. The system uses the label to understand if the first sentence semantically precedes the second sentence. Here the input sequence is processed by a stack of transformers followed by a classification layer which creates a vector shaped [fx]. The probability of the next sentence is calculated using the softmax function.

The two processes are performed serially together. The tokens used in pre-training are [MASK], [CLS], [SEP] and IsNext/NotNext. The sample inputs are illustrated in the examples.

Input = [CLS] It is a bright [MASK] day [SEP] We should play [MASK] at beach.

Label = IsNext.

Input = [CLS] It is a bright [MASK] day [SEP] I love my lazy [MASK] dog.

Label = NotNext.

The fine-tuning process is inexpensive as compared to the pre-training process. BERT uses bidirectional cross attention. This enables sentence pair to be used in multiple types of tasks such as Multi-Genre Natural Language Inference (MLNI) [44], Question Natural Language Inference (QNLI) [45], Corpus of Linguistic Acceptability (CoLA) [46] and so on.

The multi-layered pre-trained model is available in 8 variants as shown in Table 2.

4. Proposed model

The proposed model is divided into three modules viz. Data Preparation module, Model Fine Tuning module and Classification module as

Table 2
Variants of BERT developed by Devlin et al. [41].

Model	Layers	Parameters
BERT-Large, Uncased (Whole Word Masking)	24	340 M
BERT-Large, Cased (Whole Word Masking)	24	340 M
BERT-Base, Uncased	12	110 M
BERT-Large, Uncased	12	340 M
BERT-Base, Cased	12	110 M
BERT-Large, Cased	24	340 M
BERT-Base, Multilingual Cased	12	110 M
BERT-Base, Chinese	12	110 M

shown in Fig. 1. These modules represent three stages of the proposed model for sarcasm detection. Furthermore, a detailed description is provided below.

i. Data Preparation Module

The available data may not always be ordered. Often the data is raw and required to be represented in a certain fashion. In the data preprocessing, the raw data is first normalized into different columns. The architecture of BERT requires two sentences separated by a special token [SEP] as discussed in section 3. The model is designed in a fashion where it requires data in four divisions which are Tweet id, Text A, Text B, Label. Text A is the primary sentence or in this case, a tweet, whereas Text B is a secondary sentence. Since the NSP module requires a secondary sentence therefore a generic statement is used. In the proposed model, NSP (Next Sentence Prediction) is modified as a classification task and therefore original label *IsNext/NotNext* is replaced with classification labels 0 and 1 (*NotSarcasm/Sarcasm*). The formatted content is then written into an output TSV file (output_tsv) which is to be used in the next module as input data.

Original tweet:

Tweet id	Tweet	Label
1	Yay for another work at 4am day :neutral_face:	1

Transformed tweet:

Tweet id	Text A	Text B	Label
1	Yay for another work at 4am day :neutral_face:	A	1

ii. Model Fine Tuning Module

This module uses a data file to provide a fine-tuned module that can classify sarcastic data into sarcastic and non-sarcastic classes. This is a two-part process that utilizes a) BERT tokenizer, b) BERT Pre-trained model.

a. BERT Tokenizer

The module uses BERT tokenizer which takes the output of the previous module i.e. output_tsv as the input. The input sequences are then converted into a tokenized vector. The first token is [CLS] as discussed in section 3.

b. BERT Pre-trained Model

The tokenized vectors are fed into the pre-trained model. The pre-trained model then fine-tunes on the training data. The resultant of this module is a fine-tuned model for sarcasm detection.

To perform this part, BERT-base cased model is used. It contains 12 transformer layers and is trained on 110 million parameters. The pre-trained model provides a task specific model based on a generic framework of language understanding.

iii. Classification module

This is the final module that detects sarcasm. The process of classification takes pre-processed data as the input. On top of the previous module, a feed-forward single-layer neural network is added as described in [47]. The remaining process closely follows [41]. This module outputs sarcastic/non-sarcastic labeled data.

5. Experiment settings

The experiments were performed at the Google Colab environment which provided a shared Tesla K80 GPU with 12 GB of GPU RAM. Various combinations of hyperparameters were evaluated in training the model. A detailed study is provided in section 6.1. Details about the employed dataset, evaluation metrics and used baseline models are discussed in this section.

5.1. Dataset

The experiments were conducted on the dataset published in

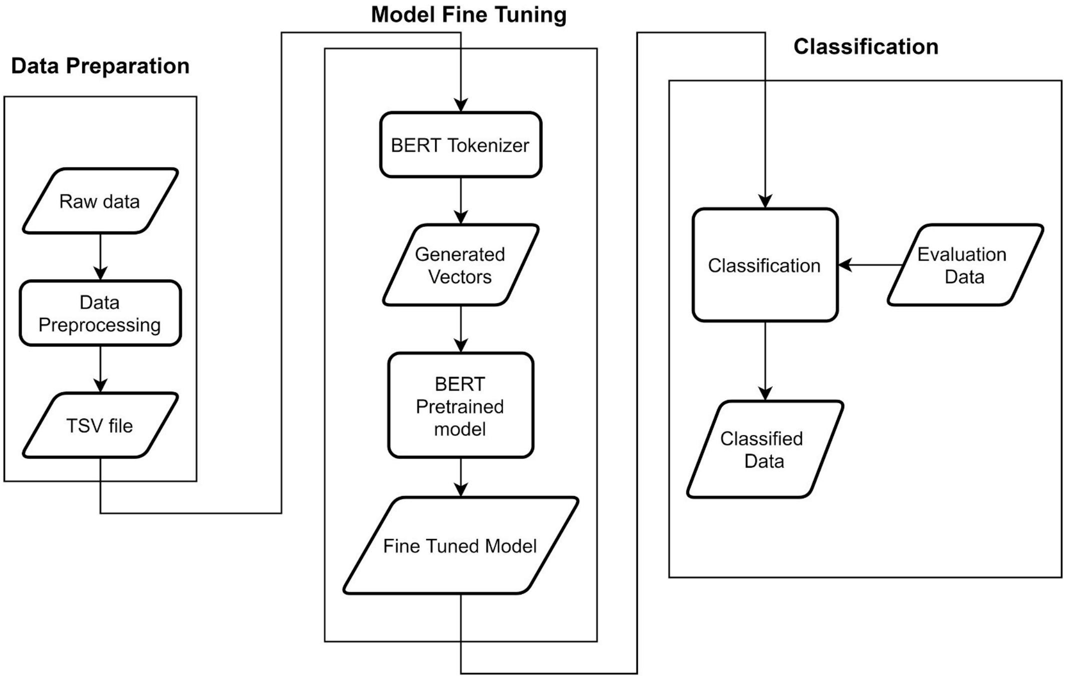

Fig. 1. The proposed framework of a pragmatic and intelligent model to detect sarcastic content.

SemEval 2018 Task 3 [17]. SemEval 2018 task 3 corpus was a part of the competition organized by the LT3 (Language and Translation Technology) team at the Faculty of Arts and Philosophy at Ghent University. The corpus is a collection of over 3000 tweets from 2676 unique users between December 01, 2014 to January 04, 2015. Annotation of the entire corpus was performed linguistically by three non-native English speakers. Each one annotated one-third of the corpus where the authors ensured specific keywords like #irony not to be present in the corpus before annotation. To ensure reliability the Fleiss Kappa score [48] of annotations was evaluated. The inter annotation scheme was ensured by testing annotators using a test sample from [49].

Kappa score is an agreement score used between three or more annotators. The reported kappa score of 0.72 is categorized as a substantial agreement among annotators. A sample of tweets from the original dataset [17] is shown in Table 3.

Table 3
Sample tweets from SemEval 2018 Task 3 dataset (0: Non-Sarcasm, 1: Sarcasm).

Tweet	Annotation
@samcguigan544 You are not allowed to open that until Christmas day!	0
Yay for another work at 4am day:neutral_face:	1
i feel like whole life is about waiting waiting and waiting	0
Hey there! Nice to see you Minnesota/ND Winter Weather	1

5.2. Evaluation metrics

As the tweets in the original dataset are classified in binary (0: non-sarcastic, 1: sarcastic), for this study we have selected the same evaluation metrics as reported in the official task. As per the reviewed literature, accuracy and F1 score have been used as metrics of evaluation in various cases such as [17,28]. Accuracy measures correct predictions but when there is an uneven distribution in positive and negative labeled cases, accuracy tends to neglect cases that are wrongly predicted. In such cases, F1 score is applicable as it balances between precision and recall.

Due to the absence of detailed scores like precision, recall and accuracy in every case, they have been reported as per the availability in the reviewed literature.

5.3. Baseline models

SemEval 2018 task 3 is one of the benchmark datasets in the domain of sarcasm detection. Various researchers have established their proposed models based on this dataset. In this study, we have compared the proposed model with state-of-the-art models (SOA).

- Logistic regression (LR) and SVM are established as important classification algorithms in sarcasm detection. The implementation of these approaches is based on [33] and [34] respectively.
- Mukherjee et al. [50] established their model for sarcasm detection on the premise of feature detection and Bayesian probabilities.
- For comparison with deep learning approaches we have used the reported results with CNN, LSTM, CNN-LSTM, BiLSTM from Zhang et al. [51].
- In the same paper by Zhang et al. [51], three models viz. AABi-LSTM, SABi-LSTM and STBi-LSTM were proposed and evaluated on the same dataset. The models were based on transfer learning-based approaches to using sentiment knowledge to improve the attention mechanism of recurrent neural models for capturing hidden patterns for incongruity.
- Results of the official competition of SemEval 2018 Task 3 [17] have also been compared.

For extensive testing, a comparison on the subset of the dataset with CNN based on [52], LSTM based on [53] along with LR and SVM have been performed.

6. Results and analysis

6.1. Quantitative analysis

The quantitative performances of various models are shown in three tables Table 4, Table 5 and Table 6. Detailed analysis is provided in the following subsections.

a. In Table 4, we have compared the proposed model with official entries of SemEval 2018 Task 3.

In the official competition, 43 teams participated in the task as reported by Hee et al. [17]. When compared, the proposed model stands among the best results of the competition. The results of the top 5 teams and the supplied baseline approaches are shown in Table 4. The metric used in the official competition was the F1 Score. However, scores of the other metrics are supplied as well.

Although the proposed model could not outperform all the competitors, it showed a significant lead in performance against the two supplied baseline approaches (by 18.22% and 86.69% respectively). Refer to Fig. 2.

b. In Table 5, we have compared models on a subset of the original dataset of SemEval 2018 Task 3.

To scrutinize the performance, a subset of the original dataset has been prepared which contains even lesser data than its parent dataset. From the original dataset of 3834 tweets, a random sample of 3000 tweets has been extracted as training data and the rest kept as testing data. We tested classic approaches like LR, SVM and plain vanilla DL architectures like CNN, LSTM and BiLSTM. The results show that with changed training and testing data the various models provided mixed results in different metrics. The trends in performances of various approaches remain similar to that of the original dataset, however, the proposed model has reported better performance even with a smaller dataset as shown in Fig. 3.

c. In Table 6, we have compared the proposed model with state-of-the-art models and approaches presented in various reviewed literatures.
   - The proposed model on comparison with rule-based and feature-based approaches like Logistic Regression, SVM and Bayesian Probabilities [33,34,50] observed a lead of 7.58%, 5.51% and 74.1% respectively.
   - Similar patterns have been observed in comparison with deep learning approaches. The results are evaluated in terms of F1 Score where performances are better than CNN (12.26%), LSTM (9.39%), BiLSTM (8.56%) and a hybrid deep approach like CNN-LSTM (13.86%).
   - The proposed model also performed better in comparison to models developed in [51] by at least 0.93%.

Table 4
Comparison of performance of the proposed model with official entries of SemEval 2018 Task 3.

Team	Accuracy	Precision	Recall	F1 Score
UCDCC	**0.797**	**0.788**	0.669	**0.724**
THU NGN	0.735	0.63	0.801	0.705
NTUA-SLP	0.732	0.654	0.691	0.672
WLV	0.643	0.532	0.836	0.65
NLPRLIITBHU	0.661	0.551	0.788	0.64
Unigram SVM BL [17]	0.635	0.532	0.659	0.589
Random Baseline [17]	0.503	0.373	0.373	0.373
Proposed Model	0.68877	0.58574	**0.85852**	0.69637

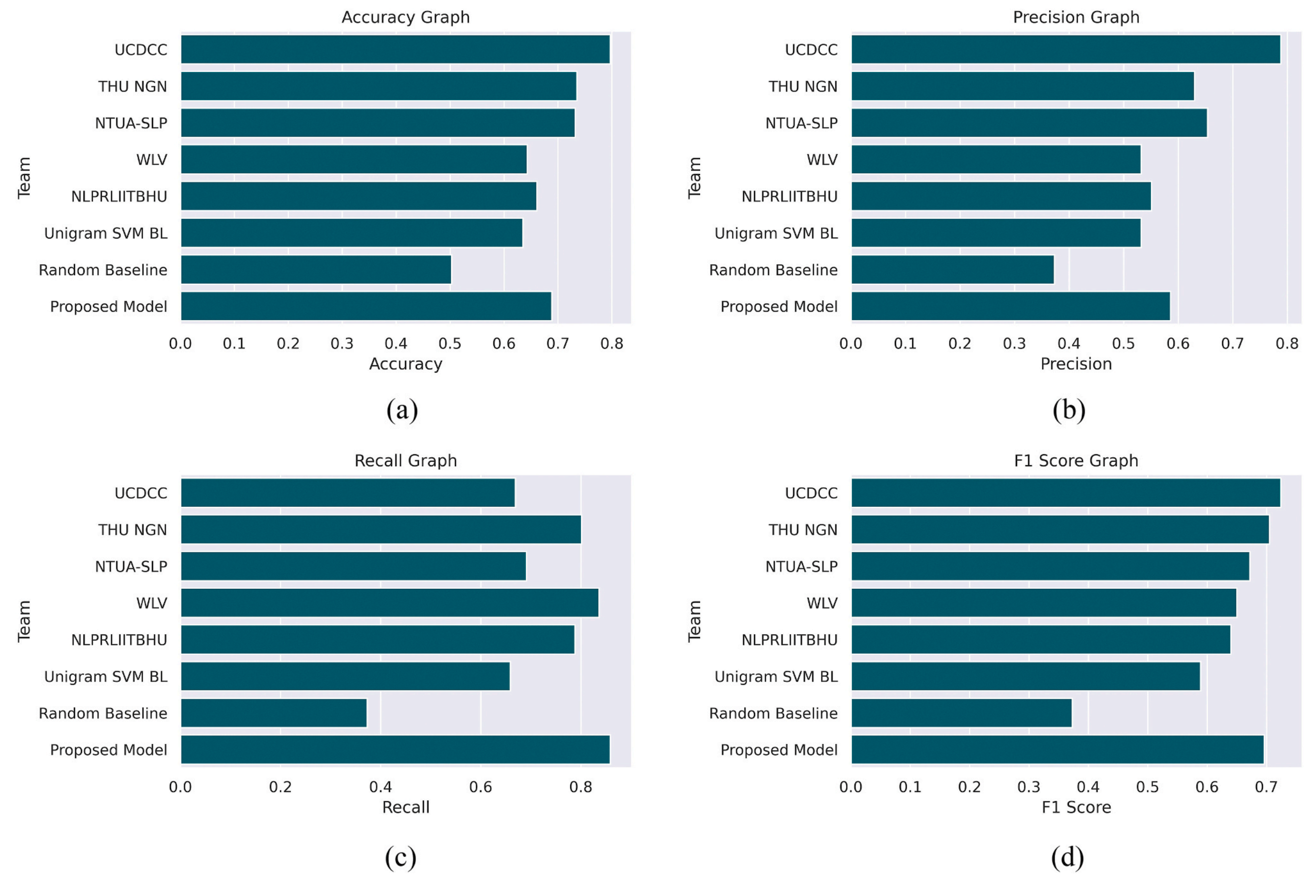

Fig. 2. Performance comparison of the proposed model with official entries in the competition of SemEval 2018 Task 3 (a) Accuracy, (b) Precision, (c) Recall, (d) F1 Score.

Table 5
Comparison of performance of models on a subset of the original dataset of SemEval 2018 Task 3.

Model	Accuracy	Precision	Recall	F1 Score
LR [33,34]	0.656	0.559	0.621	0.588
SVM [34]	0.480	0.486	0.437	0.460
LSTM [53]	0.595	0.606	0.593	0.600
Bi-LSTM [28]	0.610	0.640	0.580	0.609
CNN [52]	0.621	0.626	0.709	0.665
Mukherjee et al. [50]	0.604	0.606	0.633	0.619
Proposed Model	**0.706**	**0.687**	**0.725**	**0.705**

Table 6
Comparison of performance of the proposed model with SOA models from reviewed literature.

Model	F1 Score
CNN [52]	62.03
LSTM [53]	63.66
Bi-LSTM [28]	64.15
CNN-LSTM [54]	61.16
AABi-LSTM [51]	67.86
SABi-LSTM [51]	65.33
STBi-LSTM [51]	69.00
LR [33,34]	64.73
SVM [34]	66.00
Mukherjee et al. [50]	40.00
Proposed Model	**69.64**

- However, contradictory results have displayed where the primitive approaches like LR and SVM have outperformed deep learning approaches like CNN, LSTM, BiLSTM and hybrid deep approaches like CNN-LSTM, AABi-LSTM, SABi-LST. Refer to Fig. 4.

6.2. *Qualitative analysis*

a. Role of Hyperparameters

The proposed model is based on a deep learning approach. These models are prone to over-fit especially when poorly configured. Such models are required to be tuned up during training by adjusting hyperparameters which affect the outcomes.

To address the issue, the selection of hyperparameters was initially based on the reviewed literature [41,47]. Devlin et al. [41] suggested a likely range of hyperparameters.

Batch size: 16, 32.
Learning rate: 2e-5, 3e-5, 5e-5.
Number of epochs: 2, 3, 4.

Devlin et al. also suggested that selection hyperparameters should be task-oriented. Further investigations suggested that a set of hyperparameters may not result in better performance for every metric as shown in Table 7.

Furthermore, examining with a variety of permutations resulted in an optimal set of hyperparameters for the proposed model which is batch size of 16, maximum sequence length of 128, learning rate as 2e-5 and training epochs as 2. The tests were conducted on the subset dataset as described in section 6.1 b. Even though the proposed model performed comparatively better in the computed scenarios, the observations suggest a decisive factor can also be chosen metric as different cases require a different approach to handle the problem. In addition,

Table 7
Performance testing of the proposed model with different hyperparameters.

Version	Hyperparameters				Precision	Recall	F1 Score	Accuracy
	Batch Size	Maximum Sequence Length	Learning Rate	Epochs				
1	8	64	2.00e-05	2	0.6811224	0.6742424	0.677665	0.6887255
2	8	64		10	0.661215	0.7146465	0.6868932	0.6838235
3	8	128		2	0.6657609	0.6186869	0.6413613	0.6642157
4	8	128		10	0.6666667	0.6919192	0.6790582	0.682598
5	8	256		2	0.6991643	0.6338384	0.6649007	0.689951
6	8	256		10	0.6871921	0.7045455	0.6957606	0.7009804
7	16	64		2	0.6931217	0.6616162	0.6770026	0.6936275
8	16	64		10	0.6967419	0.7020202	0.6993711	**0.7071078**
9	16	128		2	0.6831683	0.6969697	0.69	0.6960784
10	16	128		10	0.679803	0.6969697	0.6882793	0.6936275
11	16	256		2	0.7027027	0.5909091	0.6419753	0.6801471
12	16	256		10	0.6801008	0.6818182	0.6809584	0.689951
13	24	64		2	0.7088949	0.6641414	0.6857888	0.7046569
14	24	64		10	0.6771845	0.7045455	0.6905941	0.6936275
15	24	128		2	0.6717172	0.6717172	0.6717172	0.6813725
16	24	128		10	0.6775701	0.7323232	0.7038835	0.7009804
17	24	256		2	0.5357711	**0.8510101**	0.657561	0.5698529
18	24	256		10	0.6826196	0.6843434	0.6834805	0.692402
19	32	64		2	0.7116788	0.4924242	0.5820896	0.6568627
20	32	64		10	0.6866029	0.7247475	**0.7051597**	0.7058824
21	32	128		2	**0.7164634**	0.5934343	0.6491713	0.6887255
22	32	128		10	0.685567	0.671717	0.6785714	0.6911765
23	32	256		2	0.7039275	0.5883838	0.6409904	0.6801471
24	32	256		10	0.6706161	0.7146465	0.6919315	0.6911765

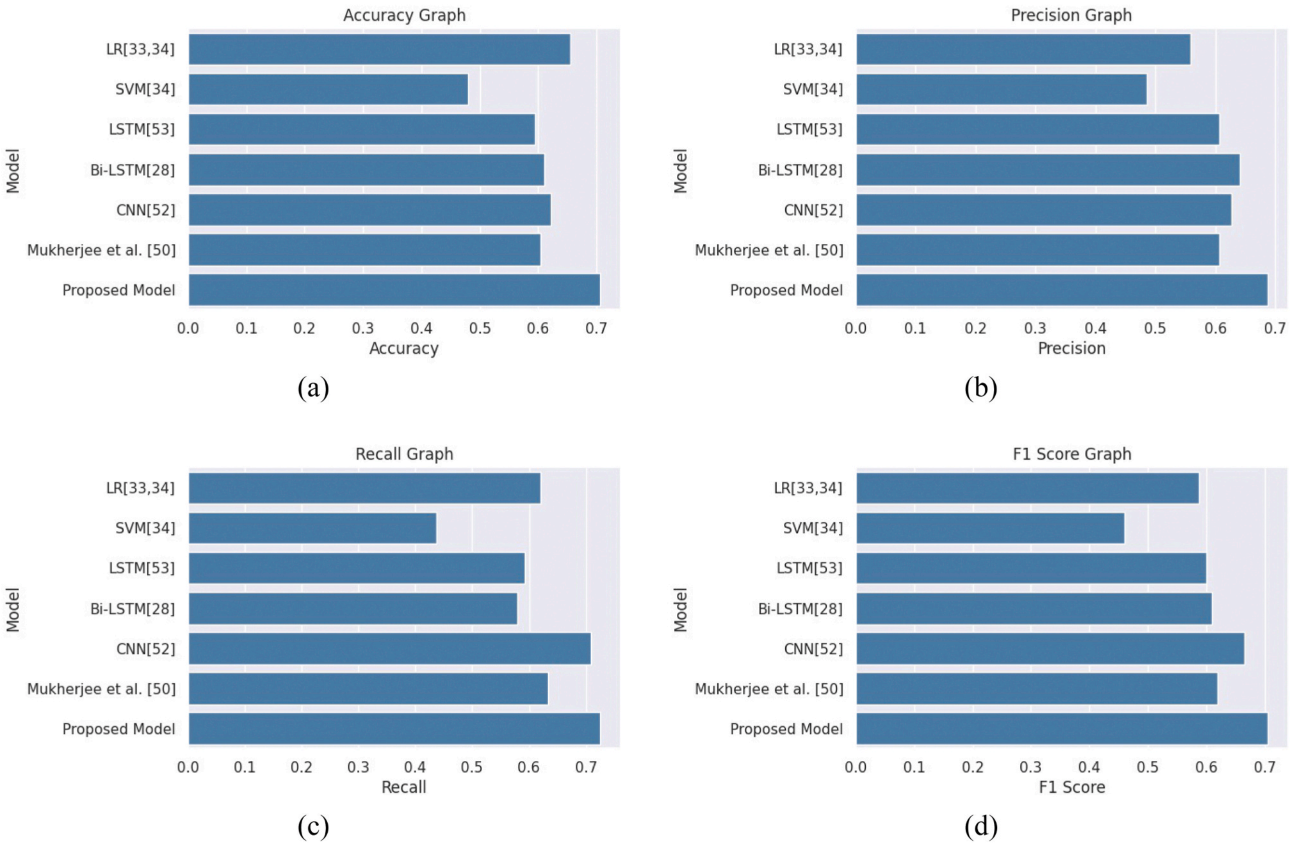

Fig. 3. Performance comparison of the proposed model with classic and DL models on a subset dataset of SemEval 2018 Task 3 (a) Accuracy, (b) Precision, (c) Recall, (d) F1 Score.

the varied results with the different hyperparameters during training also suggest the importance of hyper tuning.

b. Predictions

On a closer peek at the test dataset, tweets displayed sarcasm in a variety of forms [55] like pattern-based features, prosodic occurrences, linguistic features, polarity features. The proposed model successfully detected sarcasm in pattern-based (e.g. a followed *#not*), prosodic based (e.g. *reallllly, noooo*) even without an explicit declaration. Features

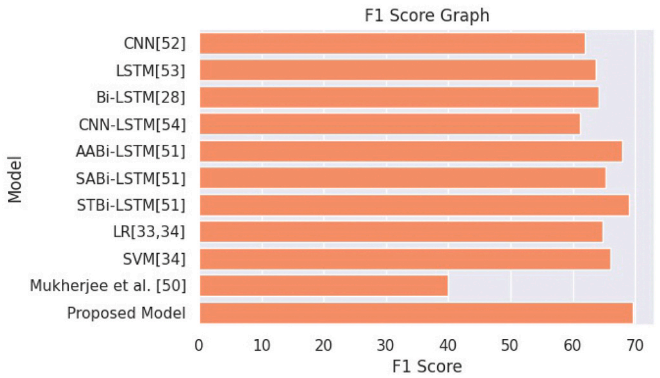

Fig. 4. Performance comparison of the proposed model with SOA models from reviewed literature.

Table 8
Sample of tweets along with their original annotation and prediction.

Tweet	Original annotation	Prediction
@KimKardashian actually cropped out her daughter in this pic for a selfie:see-no-evil_monkey: #topmum http://t.co/lfxHIlkZEV	Sarcastic	Non-sarcastic
@nypost I should sue my parents	Sarcastic	Non-sarcastic
home alone aka turn on all the lights #scaredycat	Non-Sarcastic	Sarcastic
I work in front of laptop for 8 h a day and suffer terrible dry eye. #bloodshot #eyerony	Non-Sarcastic	Sarcastic
So fucking excited to be the 5th wheel for another New Year's Eve!!!! Can't blame anybody but myself for being super picky re: men	Sarcastic	Sarcastic
Who ever thought moving in the middle of December would be so peaceful?	Sarcastic	Sarcastic

based on polarity were also well classified in the experiments which suggest the effective detection of implicit and explicit incongruity by the proposed model. However, the model fails to detect sarcasm where past knowledge about the tweet is required or predicaments are unknown. A few examples of classification are shown in Table 8.

7. Conclusions and future work

The role of technology in society is self-evident. Social media, which was once a medium informal communication, has evolved into one of the most common formal sources of news. However, to understand what literally meant by the OP (original poster), identification of sarcasm is evidently quite essential.

The detection of sarcasm in online platforms is still an arduous task. Researchers from different parts of the world have been developing datasets and models as a means to overcome the problem. Then again English is one of the most used languages over the Internet. To oversee the exponentially generated data in English, society demands a sophisticated system that can detect sarcasm in such contents. The proposed model exploits the advances of NLP and presents itself as an effective solution even in case of context incongruity. In addition, the study investigated the role with different hyperparameters and the importance of scenario oriented hyper tuning.

The proposed model uses a variant of BERT. In the future, this model can be extended to attain multilingualism. The investigation suggests the possibility of sarcasm in multimodal datasets. A pragmatic multimodal model is also one of the possible extensions of the model which can deal with audio and visual data. The proposed approach uses one of the pre-trained models. The pre-training is expensive; however, with the availability of resources development of a task-oriented pre-trained model is a plausible extension of the proposed model. In the future, the other variants of BERT can be tested for a similar task such as BERT-Large which contains 24 layers and 340 million parameters. Finally, despite the diversity and complexity of languages, the study of the recent technologies presents a feasible solution to the problem of computational detection of sarcasm in ever-growing data.

Author statement

This is a part of PhD thesis work. The reported work is performed by Mayank Shrivastava under the supervision of Prof. Shishir Kumar.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.techsoc.2020.101489.

References

[1] J. Koetsier, Why 2020 Is A Critical Global Tipping Point for Social Media, Forbes. Com, 2020 accessed July 25, 2020, https://www.forbes.com/sites/johnkoetsier/2020/02/18/why-2020-is-a-critical-global-tipping-point-for-social-media/#71934b592fa5.

[2] G. Vlastos, Socratic irony, class, Far E. Q. 37 (1987) 79–96, https://doi.org/10.1017/S0009838800031670.

[3] S. Peters, Why is sarcasm so difficult to detect in texts and emails? (n.d.)accessed September 24, 2020, https://theconversation.com/why-is-sarcasm-so-difficult-to-detect-in-texts-and-emails-91892.

[4] S. Johnson, A dictionary of the English language, (1755), accessed September 22, 2020, https://archive.org/details/dictionaryofengl01johnuoft.

[5] R.J. Kreuz, S. Glucksberg, How to Be sarcastic: the echoic reminder theory of verbal irony, J. Exp. Psychol. Gen. 118 (4) (1989) 374–386, https://doi.org/10.1037/0096-3445.118.4.374.

[6] A. Rajadesingan, R. Zafarani, H. Liu, Sarcasm detection on twitter, in: Proc. Eighth ACM Int. Conf. Web Search Data Min. - WSDM '15, ACM Press, New York, New York, USA, 2015, pp. 97–106, https://doi.org/10.1145/2684822.2685316.

[7] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, R. Huang, Sarcasm as contrast between a positive sentiment and negative situation, EMNLP 2013-2013 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf (2013) 704–714.

[8] R. Schifanella, P. de Juan, J. Tetreault, L. Cao, Detecting sarcasm in multimodal social platforms, in: Proc. 2016 ACM Multimed. Conf. - MM '16, ACM Press, New York, New York, USA, 2016, pp. 1136–1145, https://doi.org/10.1145/2964284.2964321.

[9] Y. Cai, H. Cai, X. Wan, Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model, in: Proc. 57th Annu. Meet. Assoc. Comput. Linguist., Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 2506–2515, https://doi.org/10.18653/v1/P19-1239.

[10] D. Das, A.J. Clark, Sarcasm detection on Facebook, in: Proc. Int. Conf. Multimodal Interact. Adjun. - ICMI '18, ACM Press, New York, New York, USA, 2018, pp. 1–5, https://doi.org/10.1145/3281151.3281154.

[11] A. Sinha, P. Patekar, R. Mamidi, Unsupervised approach for monitoring satire on social media, in: Proc. 11th Forum Inf. Retr. Eval, ACM, New York, NY, USA, 2019, pp. 36–41, https://doi.org/10.1145/3368567.3368582.

[12] A. Mishra, P. Bhattacharyya, Predicting readers' sarcasm understandability by modeling gaze behavior, in: 30th AAAI Conf. Artif. Intell. AAAI 2016, 2018, pp. 99–115, https://doi.org/10.1007/978-981-13-1516-9_5.

[13] J. Karoui, F.B. Zitoune, V. Moriceau, SOUKHRIA: towards an irony detection system for Arabic in social media, Procedia Comput. Sci. 117 (2017) 161–168, https://doi.org/10.1016/j.procs.2017.10.105.

[14] B. Ghanem, J. Karoui, F. Benamara, V. Moriceau, P. Rosso, IDAT at FIRE2019: overview of the track on irony detection in Arabic tweets, in: Proc. 11th Forum Inf. Retr. Eval, ACM, New York, NY, USA, 2019, pp. 10–13, https://doi.org/10.1145/3368567.3368585.

[15] I. Abu Farha, W. Magdy, From Arabic sentiment analysis to sarcasm detection: the ArSarcasm dataset, in: proc. 4th work. Open-source arab. Corpora process. Tools, with a shar. Task offensive lang. Detect., European Language Resource Association, Marseille, France, 2020, pp. 32–39. https://www.aclweb.org/anthology/2020.osact-1.5.

[16] C. Liebrecht, F. Kunneman, A. Van den Bosch, The perfect solution for detecting sarcasm in tweets #not, in: Proc. 4th Work. Comput. Approaches to Subj. Sentim. Soc. Media Anal., Association for Computational Linguistics, 2013, pp. 29–37. http://www.aclweb.org/anthology/W13-1605.

[17] C. Van Hee, E. Lefever, V.·eronique Hoste, SemEval-2018 task 3: irony detection in English tweets, Proc. 12th Int. Work. Semant. Eval (2018) 537–540, https://doi.org/10.18653/v1/s18-1087.

[18] S. Oprea, W. Magdy, iSarcasm: A Dataset of Intended Sarcasm, 2019 accessed July 22, 2020, http://arxiv.org/abs/1911.03123.

[19] J. Karoui, B. Farah, V. Moriceau, N. Aussenac-Gilles, L. Hadrich-Belguith, Towards a Contextual Pragmatic Model to Detect Irony in Tweets, in: Proc. 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. (Volume 2 Short Pap, Association for Computational Linguistics, Stroudsburg, PA, USA, 2015, pp. 644–650, https://doi.org/10.3115/v1/P15-2106.

[20] A. Gianti, C. Bosco, V. Patti, Annotating irony in a novel Italian corpus for sentiment analysis, in: proc. 4th Int. Work. Corpora Res. Emot. Sentim. Soc. Signals, 2012, pp. 1–7.

[21] R. Ortega-Bueno, F. Rangel, D.I. Hernández Farıas, P. Rosso, M. Montes-Y-Gómez, J.E. Medina-Pagola, Overview of the task on irony detection in Spanish variants, In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019), co-located with 34th Conference of the Spanish Society forNatural Language Processing (SEPLN 2019) (2019). CEUR-WS. org. ISSN: 16130073.

[22] P. Rockwell, Lower, slower, louder: vocal cues of sarcasm, J. Psycholinguist. Res. 29 (2000) 483–495, https://doi.org/10.1023/A:1005120109296.

[23] H.S. Cheang, M.D. Pell, The sound of sarcasm,, Speech Commun. 50 (2008) 366–381, https://doi.org/10.1016/j.specom.2007.11.003.

[24] J. Tepperman, D. Traum, S. Narayanan, "Yeah right": sarcasm recognition for spoken dialogue systems, in: INTERSPEECH 2006 9th Int. Conf. Spok. Lang. Process. INTERSPEECH 2006-ICSLP, 2006, pp. 1838–1841.

[25] D. Davidov, O. Tsur, A. Rappoport, Semi-supervised recognition of sarcastic sentences in twitter and Amazon, in: CoNLL 2010-Fourteenth Conf. Comput. Nat. Lang. Learn. Proc. Conf., Association for Computational Linguistics, 2010, pp. 107–116.

[26] D. Davidov, O. Tsur, A. Rappoport, Enhanced sentiment learning using twitter hashtags and smileys, in: COLING '10 Proc. 23rd Int. Conf. Comput. Linguist. Posters, Association for Computational Linguistics Stroudsburg, PA, USA ©2010, 2010, pp. 241–249. https://pdfs.semanticscholar.org/61ba/d001868aaf56c5ea69a1ece963342ea7281a.pdf?_ga=2.42263717.1008999599.1557379277-1671378736.1544503149.

[27] M. Zhang, Y. Zhang, G. Fu, Tweet sarcasm detection using deep neural network, in: COLING 2016-26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap, 2016, pp. 2449–2460.

[28] L.H. Son, A. Kumar, S.R. Sangwan, A. Arora, A. Nayyar, M. Abdel-Basset, Sarcasm detection using soft attention-based bidirectional Long short-term memory model with convolution network, IEEE Access 7 (2019) 23319–23328, https://doi.org/10.1109/ACCESS.2019.2899260.

[29] Y. Diao, H. Lin, L. Yang, X. Fan, Y. Chu, K. Xu, D. Wu, A multi-dimension question answering network for sarcasm detection,, IEEE Access 8 (2020) 135152–135161, https://doi.org/10.1109/ACCESS.2020.2967095.

[30] V.K. Jain, S. Kumar, S.L. Fernandes, Extraction of emotions from multilingual text using intelligent text processing and computational linguistics, J. Comput. Sci. 21 (2017) 316–326, https://doi.org/10.1016/j.jocs.2017.01.010.

[31] S.K. Bharti, K.S. Babu, S.K. Jena, Parsing-based sarcasm sentiment recognition in Twitter data, in: Proc. 2015 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2015, 2015, pp. 1373–1380, https://doi.org/10.1145/2808797.2808910.

[32] A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, AAAI/ICML-98 work. Learn, Text Categ 752 (1998) 41–48, 10.1.1.46.1529.

[33] R. González-Ibáñez, S. Muresan, N. Wacholder, Identifying sarcasm in Twitter: a closer look, in: ACL-HLT 2011-Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol (2011) 581–586.

[34] T. Ptáček, I. Habernal, J. Hong, Sarcasm detection on Czech and English twitter, in: COLING 2014-25th Int. Conf. Comput. Linguist. Proc. COLING 2014 Tech. Pap, 2014, pp. 213–223.

[35] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural Language processing (almost) from scratch, J. Mach. Learn. Res. 12 (August) (2011) 2493–2537. http://arxiv.org/abs/1103.0398.

[36] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proc. 52nd Annu. Meet. Assoc. Comput. Linguist, Association for Computational Linguistics, 2014, pp. 655–665.

[37] X. Ouyang, P. Zhou, C.H. Li, L. Liu, Sentiment analysis using convolutional neural network, in: 2015 IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secur. Comput. Pervasive Intell. Comput, IEEE, 2015, pp. 2359–2364, https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.349.

[38] A. Timmaraju, V. Khanna, Sentiment analysis on movie reviews using recursive and recurrent neural network architectures, CS224N Proj (2015) 2–7. https://cs224d.stanford.edu/reports/TimmarajuAditya.pdf.

[39] V.K. Jain, S. Kumar, P. Mahanti, Sentiment recognition in customer reviews using deep learning, Int. J. Enterprise Inf. Syst. 14 (2018) 77–86, https://doi.org/10.4018/IJEIS.2018040105.

[40] G. Xu, Y. Meng, X. Qiu, Z. Yu, X. Wu, Sentiment analysis of comment texts based on BiLSTM, IEEE Access 7 (2019) 51522–51532, https://doi.org/10.1109/ACCESS.2019.2909919.

[41] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proc. NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 4171–4186. http://arxiv.org/abs/1810.04805.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 5998–6008. http://arxiv.org/abs/1706.03762.

[43] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning Books, Movies Towards, Story-like visual explanations by watching movies and reading books, in: 2015 IEEE int. Conf. Comput. Vis., IEEE, 2015, pp. 19–27, https://doi.org/10.1109/ICCV.2015.11.

[44] A. Williams, N. Nangia, S. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: proc. 2018 conf. North Am. Chapter assoc. Comput. Linguist. Hum. Lang, Technol. 1 (2018) 1112–1122, https://doi.org/10.18653/v1/N18-1101. Long Pap., Association for Computational Linguistics, Stroudsburg, PA, USA.

[45] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuad: 100,000+ questions for machine comprehension of text, in: EMNLP 2016-Conf. Empir. Methods Nat. Lang. Process. Proc, 2016, pp. 2383–2392.

[46] A. Warstadt, A. Singh, S.R. Bowman, Neural network acceptability judgments, Trans. Assoc. Comput. Linguist 7 (2019) 625–641, https://doi.org/10.1162/tacl_a_00290.

[47] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune BERT for text classification?, in: lect. Notes comput. Sci. (Including subser. Lect. Notes artif. Intell. Lect. Notes bioinformatics). https://doi.org/10.1007/978-3-030-32381-3_16, 2019, 194-206.

[48] J.L. Fleiss, Measuring nominal scale agreement among many raters, Psychol, Bull. (Arch. Am. Art) 76 (5) (1971) 378–382, https://doi.org/10.1037/h0031619.

[49] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, A. Reyes, SemEval-2015 task 11: sentiment analysis of figurative language In twitter, in: Proc. 9th Int. Work. Semant. Eval. (SemEval 2015, Association for Computational Linguistics, Stroudsburg, PA, USA, 2015, pp. 470–478, https://doi.org/10.18653/v1/S15-2080.

[50] S. Mukherjee, P.K. Bala, Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering, Technol. Soc. 48 (2017) 19–27, https://doi.org/10.1016/j.techsoc.2016.10.003.

[51] S. Zhang, X. Zhang, J. Chan, P. Rosso, Irony detection via sentiment-based transfer learning, Inf. Process. Manag. 56 (2019) 1633–1644, https://doi.org/10.1016/j.ipm.2019.04.006.

[52] Y. Kim, Convolutional Neural Networks for Sentence Classification, in: Proc. 2014 Conf. Empir. Methods Nat. Lang. Process, Association for Computational Linguistics, Stroudsburg, PA, USA, 2014, pp. 1746–1751, https://doi.org/10.3115/v1/D14-1181.

[53] Y.-H. Huang, H.-H. Huang, H.-H. Chen, Irony detection with attentive recurrent neural networks, in: lect. Notes comput. Sci. (Including subser. Lect. Notes artif. Intell. Lect. Notes bioinformatics). https://doi.org/10.1007/978-3-319-56608-5_45, 2017, 534-540.

[54] A. Ghosh, T. Veale, Magnets for sarcasm: making sarcasm detection timely, contextual and very personal, in: proc. 2017 conf. Empir. Methods nat. Lang. Process., Association for Computational Linguistics, Stroudsburg, PA, USA, 2017, pp. 482–491, https://doi.org/10.18653/v1/D17-1050.

[55] M. Abulaish, A. Kamal, M.J. Zaki, A survey of figurative Language and its computational detection in online social networks,, ACM Trans. Web 14 (2020) 1–52, https://doi.org/10.1145/3375547.