Journal of Computer and Knowledge Engineering, Vol. 7, No. 1, 2024. (49-58)

Ferdowsi University of Mashhad

Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

Information and Communication Technology Association of Iran

A Hybrid Content and Context-Based Method for Sarcasm Detection*

Research Article

Zahra Keivanlou-Shahrestanaki[1], Mohsen Kahani[2], Fattane Zarrinkalam[3]

DOI: 10.22067/cke.2023.85399.1109

Abstract: With the growing use of social media, figurative language has become very common on social media platforms. Given its complexity, figurative language can confuse natural language processing systems and lead to incorrect results. To address this issue, researchers have developed methods to detect humor, jokes, irony, and especially sarcasm. To date, most studies have used deep learning methods to identify sarcasm. Some studies have also incorporated context such as previous posts or conversations to improve the accuracy of sarcasm detection. But the context that can be highly effective in detecting the sarcasm of posts is the characteristics of the writer of the posts. So, the present paper aims to develop a hybrid approach that combines content and context features to better identify sarcastic posts. i.e., this study additionally proposes a deep learning method to model the content of tweets and suggests a multi-dimensional method that considers the user's writing style and personality traits as context features. Several experiments were used to evaluate the effectiveness of the proposed method. The results indicated that the proposed method outperformed baseline methods in sarcasm detection.

Keywords: social media, sarcasm detection, deep learning, content-based features, context-based features.

1. Introduction

Doing simple tasks can sometimes be a bit of a challenge. For instance, choosing a mobile phone can be overwhelming due to the vast variety of models available with different brand names and features. However, a recommender system can make the selection process easier. It is important to note that a good recommendation system relies on user opinions and feedback. For example, if a user says, "the battery of this phone is excellent, it can last for half an hour," the system will consider it positive feedback for the phone's battery. Such comments can cause inaccuracies in the results of a recommender system. This is also true for opinion mining systems used in elections. It can lead to incorrect results. Therefore, the presence of such sentences can disrupt the performance of any natural language processing system.

Despite such anomalies, the web, especially social media is one of the largest platforms for people to express their opinions and share information. Governments and organizations rely on this data to understand public opinion on products, services, and social and political events. However, the informal and figurative language used on social media can make it challenging to determine the true meaning behind people's expressions. Therefore, it is crucial for natural language processing to accurately detect informal and figurative statements [1].

When people express their opinions and feelings about various topics such as politics, social issues, products, films, or books, using figurative language, the literal meaning of their statements may be different from their intended meaning. This makes it challenging to understand what they truly mean. In many cases, people use sarcasm to express important issues in a more engaging way. This form of language can attract more attention and have a greater impact on the audience. In some cases, humor, jokes, irony, or sarcasm may be the only way to criticize the current situation in closed political spaces. Unfortunately, current natural language processing systems struggle to understand figurative language [2].

Sarcasm is a commonly used form of figurative language. According to the Free Dictionary, sarcasm is a type of verbal irony that is used to express contempt or mockery [2]. Because of its figurative nature, sarcasm can pose a challenge for sentiment analysis. Sarcasm statements often express negative feelings while appearing to be positive, making them challenging to detect. As a result, researchers have identified automatic sarcasm detection as a complex issue that requires further exploration. This is because sarcasm can be expressed in subtle ways, making it a difficult problem for computational approaches to solve [3].

At the same time, as many people use sarcasm to express their opinions on social media, it is inherently difficult for machines and even for humans to analyse these posts. Sarcastic posts can have a significant effect on sentiment analysis. For example, the comment "having an expensive

* Manuscript received: 2023 November 16, Revised, 2023 December 17, Accepted, 2023 December 31.

1 Ph.D Student Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.

2 Corresponding author, Professor of Computer Engineering, Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, **Email**: kahani@um.ac.ir

3 Assistant Professor of Computer Engineering, School of Engineering, University of Guelph, Canada.

phone with short battery life feels great" is a sarcastic sentence that uses positive words like "feels great" to express negative feelings about battery life. This kind of sentence is challenging for machines to understand correctly. As a result, it may be considered a positive opinion instead of a negative one about the phone. Therefore, it is crucial to detect sarcasm to improve the performance of natural language analysis systems. Sarcasm detection can be expressed as a binary classification task to predict whether a given sentence is sarcastic or not. This has the potential to aid opinion mining, sentiment analysis, recommender systems, summarization systems, and many other natural language processing applications [4].

In past years, a lot of research has been done in this field. Many recent studies have employed deep learning methods to detect sarcasm. However, since detecting sarcasm solely based on text is a complex process, considering information beyond the text can significantly enhance the performance of these methods. Although several attempts have been made to detect sarcasm using context, most of them have relied on specific contexts like previous user posts and conversations. But the context that can be highly effective in detecting the sarcasm of posts is the characteristics of the writer of the posts. Therefore, the primary aim of this research is to gather information beyond the text about the characteristics of the writer of the posts and then use a combination of content and user context to improve the accuracy of sarcasm detection.

What makes this study innovative is the use of a combination of writing style modelling and user personality as context features. Apart from gathering information beyond the text to model writing style and user personality, this paper proposes a hybrid method based on tweet content and a multi-dimensional modelling of new user contexts.

In other words, we attempt to address sarcasm detection by leveraging behavioral traits intrinsic to users expressing sarcasm by evaluating the personality of users and their writing style. We identify such features using the user's past tweets.

More specifically, in this paper we aim to answer the following research questions:

RQ1. Does a deep learning model that considers both content and the user’s writing style as context in detecting sarcasm, improve the performance of the sarcasm detection task?

RQ2. Does a deep learning model that considers both content and the user’s personality as context in detecting sarcasm, improve the performance of the sarcasm detection task?

RQ3. Does a deep learning model that considers both content and the user's writing style and personality as context in detecting sarcasm, improve the performance of the sarcasm detection task?

RQ4. How does the proposed content and context-based hybrid method perform, as opposed to baseline sarcasm detection methods?

To answer the above research questions, the paper is structured as follows: In the second section, we provide an overview of previous research related to the topic. The third section outlines our proposal, detailing the method introduced in the study. In the fourth section, we evaluate the proposed method from various perspectives to determine whether it has improved the results. Finally, we conclude our study and discuss future research directions.

2. Related works

Automatic sarcasm detection is a widely researched topic. Researchers view it as a typical issue in text classification. Presently, studies on automatic sarcasm detection can be categorized into three groups: rule-based, statistical, and deep learning methods [2].

2.1. Rule-based methods

These approaches aim to detect sarcasm by analyzing specific evidence. In other words, rule-based classifications search for a combination of positive expressions and negative situations within a sentence. The set of expressions indicating a negative situation and positive expressions starts with an initial set, using an iterative algorithm, which is then expanded [2].

The authors of studies [5] and [6] were among the first to identify sarcasm in social media. They introduced unexpected and contradictory concepts that are frequently used in sarcastic expressions.

The authors of [1] propose that the sentiment conveyed through hashtags is a key factor in detecting sarcasm. Hashtags are frequently used by Twitter users to emphasize sarcasm. Hence, if the sentiment conveyed by the hashtag does not align with the text of the tweet, it is predicted to be sarcastic.

The researchers in [7] present a rule-based classification algorithm. The algorithm uses parse-based word generation to create parse trees from sentences and identify situational expressions that contain sentiment. The algorithm works by predicting if a sentence is sarcastic if a negative phrase occurs in a positive sentence.

The authors in [8] also proposed a rule-based classification system that searched for a sentence containing both a positive verb and a negative status phrase. They extract a collection of negative state expressions using a well-structured iterative algorithm that commences with a set of positive verbs and then progressively expands both collections (i.e., positive verbs and negative state expressions). The authors conducted experiments with different rule settings, including restricting the order of the verb and the state expression.

Most rule-based studies have utilized similar methods, but have often failed to produce good results due to the use of simplistic approaches. However, rule-based methods are not universally applicable and can only function under certain circumstances. So, we have limited ourselves to introducing some of them. The next section introduces statistical methods.

2.2. Statistical methods

Sarcasm detection techniques using statistical methods differ based on the features and learning algorithms employed [2]. The features used can be categorized as either content-based or context-based. In content-based approaches, a classifier is trained to detect sarcasm using lexical, linguistic cues, and syntactic patterns. However, context-based approaches use information beyond the text to detect sarcasm [9].

Authors in [10] and [11] employed features that are based on linguistic content, such as exclamation marks, emoticons, and quotation marks, to detect sarcasm. Additionally, researchers in [12] and [13] used syntactic patterns and lexical signs that are associated with sarcasm. It is important to note that this class of sarcasm detection tasks uses only the input text and does not consider any contextual information.

The popularity of methods based on contextual features has increased in recent years, especially with the rise of social media platforms. As these platforms often contain grammatical errors, slang, and informal language, using contextual information can help in identifying sarcasm more accurately [11].

Authors in [14] and [15] utilized sentiments and the emotional information present in the input text as contextual clues. On the other hand, some researchers, like those in [16] and [17], employed the user's previous posts to identify sarcastic statements. However, as demonstrated by the authors in [11], contextual information, while helpful in improving the performance of the model, is not necessary for detecting sarcasm.

Although many of these methods have used machine learning algorithms such as SVM, neural networks, etc., to detect sarcasm, they have not yielded satisfactory results. Therefore, with the emergence and expansion of the use of deep learning methods in various fields, they have also become popular in sarcasm detection.

2.3. Deep learning methods

Deep learning methods have achieved notable success in the field of natural language processing (NLP). Consequently, recent studies have employed deep neural networks to detect sarcasm. However, some researchers also regard content-based statistical approaches, such as recurrent neural networks, as deep learning methods.

There are different approaches to learning and classifying meaningful features in studies such as Kareei [18], [19], [20], and [21]. Some use deep learning, while others use statistical methods that require manual feature extraction. Although statistical approaches are easy to interpret, they tend to have low performance. Meanwhile, deep learning methods achieve high performance but are not easily interpretable [9].

Word embedding, which involves mapping words to real-valued vectors [22], is crucial in the success of Recurrent Neural Networks (RNNs) and other deep learning neural network architectures for detecting sarcasm [23]. Combining word embedding with Convolutional Neural Networks (CNNs), as introduced by [19] and [24], has been shown to achieve good performance.

Attention-based recurrent neural networks have also achieved good results when combined with pre-trained Word2Vec word embeddings [25] and contextual information [26].

In [27], researchers introduced a multi-objective attention-based network LSTM (MHA-BiLSTM) to identify sarcastic comments in a given dataset. The results showed that this mechanism increased the performance of BiLSTM and also outperformed feature-rich SVM models.

In [23], authors implemented a deep learning classification algorithm. Their method employed an attentional bidirectional LSTM model using GloVe's pre-trained word embeddings, resulting in improved accuracy.

The study presented in [9] proposes a deep learning-based architecture for sarcasm detection that uses a self-attention mechanism to achieve better performance in different datasets and also provides interpretability of the model.

In [18], researchers proposed a Convolutional Neural Network (CNN) method that incorporates user behavior as a contextual feature along with textual content features. Likewise, the authors of [15] employed pre-trained convolutional neural networks to extract emotions, and sentiment for developing mechanisms to detect sarcasm.

In a study conducted by researchers [17], they used a two-way gated recurrent neural network (RRN) to extract semantic and syntactic information from tweets. They also employed a convolutional neural network to automatically extract textual features from previous tweets. Another study [28] used a bi-directional short-term memory (BiLSTM) to understand the language and psychological concepts related to the user. In [7], authors used a multidimensional attentional recurrent network that calculated the similarity between any two words in a tweet to obtain inconsistent information.

In [29], a self-adaptive network was presented that utilized a modified joint attention mechanism and syntactic information based on a bidirectional LSTM encoder to obtain inconsistent information. Meanwhile, [30] used LSTM and its bidirectional variant Bi-LSTM, along with GloVe for word representation to capture the semantic embeddings of words and context. In [31], a hybrid model of BiLSTM and soft attention model was proposed. In [32], a three-layer stacked Bi-LSTM framework and a three-gram-based weighted word embedding model with a focus on cue words were introduced for sarcasm detection.

Similar to other areas of NLP, recently, significant progress has been made in sarcasm detection using pre-trained linguistic models such as BERT[33], XLNET[34], and GPT3[35]. For instance, in [36] researchers employed Bi-LSTM and attention mechanism to extract effective sentence features, which were then fed into the BERT model for sarcasm detection. In [37], researchers proposed a BERT model for detecting sarcasm, with added adversarial perturbations at the embedding level for increased generalizability.

In [38], researchers proposed a model to detect sarcasm. The model comprises four main components: (1) a text-mode representation that uses BERT as a text encoder; (2) an image-mode representation that uses a pre-trained Vision Transformer (ViT) [39] as an image encoder; (3) a graph Modal that creates a cross plot for each multimodal instance; and (4) a multimodal fusion combining text and image representations by graph convolutional networks (GCN) and attention mechanism. Later, researchers in [40] investigated a transfer learning framework that leverages the correlation between sarcasm and emotions. This framework considers emotion classification and recognition as auxiliary tasks to inject knowledge into the main task of detecting category usage sarcasm. Furthermore, authors in [41] used auxiliary information, such as the news title at the sentence level, to learn the context of sarcastic sentences. They also included auxiliary information on large-scale BERT for improved performance.

In [42], researchers used three pre-trained language models based on a transformer, including RoBERTa [43], XLMRoBERTa [44], and DeBERTa [45]. The researchers utilized ensemble learning, incorporating hard and soft voting between the three models, and demonstrated that the DeBERTa model outperformed the other two pre-trained language models.

In a study conducted by researchers [46], they proposed a new model, which consists of novel stance-centered graph attention networks (SCGAT) and BERT for sarcasm detection.

The study presented in [47] proposed a new framework that leverages the ConceptNet knowledge base to incorporate prior knowledge and determine image–text relatedness through sample-level and word-level cross-modal semantic similarity detection to identify sarcasm.

Also, studies such as [48] and [49] have used auxiliary tasks

such as detecting emotions and sentiments to help detect sarcasm.

Over the years, significant progress has been made in sarcasm detection using deep learning techniques, with promising results. However, it appears that not enough attention has been given to contextual features. To enhance the efficacy of these approaches, we suggest combining deep learning methods with contextual information. We propose a hybrid model that incorporates both content and context for sarcasm detection.

3. The Proposed Approach

Our aim in this section is to develop a sarcasm detection model for tweets, which combines tweet content modeling and user modeling as a context. The general framework of the proposed model is illustrated in Figure 1.

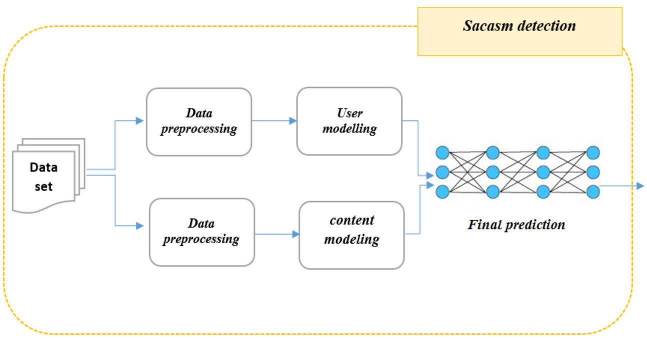

Figure 1. the overview of proposed sarcasm detection framework

This model aims to determine whether a tweet is sarcastic by analyzing the tweet's text and the recent tweets of each user.

Figure 1. shows that the model consists of four key components: data pre-processing, content modeling, user modeling, and a deep learning network for the final prediction. The first step involves preprocessing the data used in this model, including the original tweets, for content modeling and the recent tweets of each user for user modeling. The model comprises of two convolutional neural networks. The first network is trained for user modeling, while the second network is trained for content modeling of tweets. The users' recent tweets serve as the input to the first network, which generates embeddings based on users' stylistic features and personalities. The text of users' tweets and their tags are used as input to the second network, creating a content-based model. Finally, both models are combined, and the desired model is based on tweet content, the users' writing style, and personality. The model architecture is explained in detail in the following section.

3.1. Data pre-processing

Before the collected data are used in model training, a series of pre-processing should be applied on them. This makes the model understand the concept of data more and perform better. It should be noted that data preprocessing was done for both the tweet and user's previous posts.

The processing performed on the text of all tweets is as follows:

1. Normalization and conversion of tweet text into token: With this pre-processing, the words in the text are separated. We call each of these words a token and then they are normalized.
2. Converting all the letters of each token to lowercase letters: If all the words are in the same state, it is more suitable for modeling. Because it is no longer necessary to check the same words written in small and capital letters separately. As a result, all words are converted to lowercase form.
3. Eliminating monosyllabic words: words consisting of only one letter often do not add much meaning to the text; Therefore, these words can be deleted.
4. Removing symbols and links: these items do not have much effect on the overall meaning of the sentences, and their frequency can cause the model to be wrong. Hence, by removing them, the model can be trained more easily.
5. Removing stop words: These words are words that are used frequently in sentences but do not produce valuable information; Like (The) first English words; Therefore, by removing them, the processing of the model during training will be less.
6. Correcting the spelling of words: Words that have misspellings are corrected using a spell checker.

3.2. User modelling

In this step, we try to increase the accuracy of predictions by extracting data beyond the existing text. What has a significant effect on the sarcasm of a text is the background characteristics of the writer. In other words, by getting more information about the writer, more accurate predictions can be made. For this reason, in this phase, we try to improve the performance of our proposed approach by modeling the user. Our goal of user modeling is to obtain a vector representation of the hidden aspects of the user that shows the sarcastic tendencies of the user, which can map similar users in the embedding space close to each other. The findings show that modeling the user and his tendencies is very efficient to perform this task [8]. In fact, in this phase, we want to look at the user from several different perspectives, i.e. writing style and personality, and model the user's characteristics. In order to obtain a user embedding vector that can indicate the user's tendencies in using sarcastic sentences. The purpose of learning these background features is to obtain discriminating information to detect sarcasm. In fact, it seems that there is useful information in the user's previous posts that can be effective in detecting sarcasm. In other words, considering that various aspects of the user's characteristics may be effective in detecting the sarcastic tendencies of the user, we want to present a multidimensional modeling of various aspects of the user. Since different aspects cannot be modelled at the same time, there is a need to create separate spaces for different dimensions. Then, model each aspect separately, the resulting embedding vectors are integrated with Cross Correlation Analysis (CCA) method as a single embedding [15].

Therefore, to model the user, we model the characteristics of their writing style and personality using the recent tweets of the users. In the rest of this section, we will first explain how to model the writing style and then model the user's personality.

3.2.1. Modeling the writing style of users

Each person has his own writing style, which is reflected in his writings. These styles are generally influenced by features such as gender, vocabulary, syntactic effects, etc. and show behavioral patterns that help to detect sarcasm. We use this motivation to learn the stylistic characteristics of users by collecting and modeling their recent tweets.

First, after collecting the text of each user's recent tweets and performing pre-processing for each user's tweets, we create a document using a special separator <END>. Then, an unsupervised representation learning method of Paragraph Vector is applied to this document [54]. This method creates a vector of fixed size for each user. There are several reasons for choosing the ParagraphVector method. Apart from its ability to effectively encode the user's writing style, it has the advantage of being applied to variable lengths of text. In addition, it has been shown that ParagraphVector performs well for sentiment classification tasks. And considering that the sarcastic orientation of the sentence is associated with the expression of emotions; This method can also work well for classifying sentences as sarcastic or not.

Now we describe the function of this method:

Each user document and all the words inside them are first mapped to unique vectors; So that each vector corresponds to a column in the matrix $D \in R^{ds\times Nu}$ and $W_s \in R^{ds\times|V}$ It is shown. Here, ds is the size of the embedding vector and |V| shows the size of words.

Then we implement the bag of words method [55] for each of these documents. Indeed, given a document d_i for user U_i that contains a sequence of n_i words (w_1, w_2,…, w_n), we calculate the average logarithm of the prediction probability of each word in a sliding text window of size K_s. This probability is equal to:

$$p = \frac{1}{n_i}\sum_{t=k_s}^{n_i-k_s} \log\, p\left(w_t|d_i, w_{t-k_s}, \dots, w_{t+k_s}\right) \quad (1)$$

To predict a word in a window, we take the average of all the vectors of words close to it in terms of content along with the document vector d_i and use a neural network with softmax prediction:

$$p\left(w_t|d_i, w_{t-k_s}, \dots, w_{t+k}\ \right) = \frac{e^{y_{w_t}}}{\sum_1 e^{y_i}} \quad (2)$$

Here, y = [y_1, …, $y_{|V|}$] is the output of the neural network. For example:

$$y = U_d h\left(d_i, w_{t-k_s}, \dots, w_{t+k}\ \ ; D, W_s\right) + b_d \quad (3)$$

where $b_d \in R_{|V|}p$, $U_d \in R_{|V|}\times d_s$ are the parameters and the function h( ) represents the average of the vectors d_i, w_(t-k_s) and..., w_(t+k_s) taken from D and Ws. Hierarchical softmax is used for faster training [56]. Finally, after training, D learns the users' document vectors that represent their stylistic features.

3.2.2. User personality modeling

Personality discovery from text has many applications of natural language processing such as product recognition, mental health diagnosis, etc. Personality diagnosis, which is described as a combination of multiple characteristics, helps to identify a person's behaviour, thought patterns. To model users' personality associations with the sarcastic nature of their tweets, we embed personality traits into user embeddings.

Personality assessment is a field that involves the automatic classification of people's personality traits that can be compared with gold standard labels. In this context, we use a semantic vector approach to personality assessment, which consists of vectors that represent personality dimensions and automatically measures the similarity between these vectors and texts written by human subjects.

We utilize five personality feature ns used by [15], i.e., Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. To create the final feature vector, we concatenate the feature vector of each personality dimension. Therefore, the personality model is ultimately a 5-dimensional feature vector (one dimension for each of the five personality features).

3.3. Tweet content modeling

In this section, we used a convolutional neural network to model the content of tweets, which we will describe below. Each user tweet is a text and consists of n words:

$$S = [w_1, \ldots, w_n] \tag{4}$$

Each word w_i is represented as an embedded word $w_i \in R^{dem}$ using pre-trained FastText embeddings. Then, a single-layer neural network is modeled based on this input sequence (S). First, a convolutional layer with three filters $F_{[1,2,3]} \in R^{dem \times h[1,2,3]}$ of heights $h_{[1,2,3]}$ is applied respectively. This produces a feature vector m_k of size $R^{|S|-h_k+1}$ where $k \in \{1,2,3\}$. Each input $m_{k,j}$ is obtained as follows:

$$m_{k,j} = \alpha\left(F_k \,.\, S_{[j:j+hk-1]} + b_k\right) \tag{5}$$

Here, $b_k \in R$ is the b_{ais} and α is a nonlinear activation function. M feature maps are created from each F_k filter, and a total of 3 feature maps are provided as output. Following this, a "max summation" operation is performed along each feature map; Therefore, for all feature maps M computed from F_k, the output O_k is computed. Then the outputs of these 3 feature maps

are combined and make the final output:

$$O = [O_1 \oplus O_2 \oplus O_3] \in R^{3M} \tag{6}$$

where ⊕ denotes concatenation. Finally, O is predicted on a density layer with d_p neurons followed by a final prediction layer.

3.4. Final prediction

After extracting the tweet text representation and retrieving the user embedding for the author of each tweet, we concatenate both vectors to form a unified representation. Finally, we give this representation to the output layer of the neural network, which has two neurons and uses *softmax*, to make the final prediction. In fact, this method gives a probability about whether a tweet is sarcastic or not. Then, this probability estimate is used to calculate the categorical cross-entropy, which we use as a loss function.

4. Expriment

Here, we evaluate the proposed method for detecting sarcasm based on content and context. First, we introduce the dataset used and the common evaluation criteria used for this task. Next, we present the baseline method used to compare the results obtained from the proposed method with those of prior studies. Later, we conduct several experiments based on the introduced dataset to evaluate the proposed method. Finally, the results are analyzed and compared with baseline.

4.1. Data Set

The SPIRS dataset [53]: SPIRS is a publicly available (https://github.com/bshmueli/SPIRS) dataset containing 15,000 sarcastic and 15,000 non-sarcastic tweets. This dataset was created using a new data collection method called reactive monitoring, which collects both intended and perceived sarcasm. The collection method is up-to-date and has several advantages over traditional data collection methods. The dataset has reserved 20% of the total data for testing and the rest for training. Being relatively large in size and up-to-date, this dataset was selected for the study. Table 1 shows some samples from SPIRS dataset.

Table 1, some samples from SPIRS dataset

	tweets_ids	tweets_texts	tweets_tag
1	95812024 455134000	Watched the first Paddington tonight because I want to see the sequel at some point given the rave reviews it	Sarcastic
2	105974290 146200000	BernieSanders Everyone @ should be able to be treated for their illness, does that mean you'll have to €contribute foâ	Sarcastic
3	104302007 467041000	I do love how #banking works- As a customer you get charged for every infraction, no matter how minor, but €when aâ	Sarcastic

4.2. Configuration of the proposed method

In this method, meta-parameters are adjusted using the validation set through random search [50]. To optimize the parameters, the Adam optimizer [51 90] is used, which starts with the initial learning rate of $\mathbf{10^{-3}}$. Early stopping technique is used to accelerate model training. Also, for ParagraphVect modeling, the open-source implementation provided by *Gensim*, which is publicly available, is used.

4.3. Evaluation criteria for sarcasm detection

We consider sarcasm detection as a binary classification problem and based on [42] [56] evaluate it using precision, recall, and F1.

4.4. The baseline model

Bert-baseline[49]: In this study, the authors collected the SPIRS dataset to perform a sarcasm detection task. They tested the performance of three baseline models, namely CNN (100 filters with a kernel size of 3), BiLSTM (100 units), and BERT pre-trained language model. The results showed that the BERT baseline model [33] performed better than the other models on the SPIRS dataset. Therefore, we selected it as the baseline model for sarcasm detection.

4.5. Evaluation and comparison of the results of the proposed method

In this section, we analyze the performance of our hybrid method, which combines content and context-based approaches, for detecting sarcasm. The results of this analysis are reported in Table 2. More specifically, we aim to answer the research questions mentioned in section 1.

To answer the first research question (RQ1), we conducted a comparison between the hybrid model based on content and writing style as context, and the method without the use of contextual data, i.e., the content-based model. The results showed that the proposed model performed better in terms of precision, recall, and F1, as opposed to the content-based model. These observations indicate that the proposed model has a higher accuracy rate in distinguishing between sarcastic and non-sarcastic tweets. This can be attributed to the model being trained simultaneously on both content-based data and writing style as contextual information.

Table 1. Comparison of the proposed methods with the basic line

Methods	Precision	Recall	F1
BERT baseline	0.70	0.77	0.73
Content	0.63	0.68	0.65
Content and writing style	0.67	0.75	0.70
Content and personality	0.66	0.73	0.69
Content, writing style, and personality	0.73	0.79	0.75

To address the second research question (RQ2), we compared the performance of a hybrid model that incorporated both content and personality assessment as context with a basic content-based model that did not utilize contextual data at all. The results showed that the hybrid model outperformed the content-based model in terms of precision, recall, and F1 score, indicating that it was able to distinguish between sarcastic and non-sarcastic tweets more accurately than the other method. This superiority can be attributed to the model's combined training using both content-based data and personality assessments as contextual data.

To answer the third research question (RQ3), we compared the results of two models: the hybrid model that included both content data and personality and writing style as contextual data; and the content-based model without the use of any contextual data. As the results show, the hybrid model performed much better than the content-based model in terms of precision, recall, and F1. This indicates that the proposed model, which uses both content and writing style and personality as context, can classify tweets more accurately than the content-based method. Furthermore, comparing this model with the previous methods, which used only one context data, showed that using two contexts at the same time yields better results. The model's superiority may be attributed to its training with content-based data and the simultaneous use of writing style and personality as contextual data.

In sum, these observations confirm our idea to improve sarcasm detection with a hybrid learning model based on both content and context.

To address the fourth research question (RQ4), we presented the results of the Bert-baseline sarcasm detection model in Table 2. and compared them with our proposed models. The results revealed that the hybrid models that utilized either writing style or personality as contextual data were weaker than the baseline model in terms of precision, recall, and F1. This indicates that using a hybrid framework for utilizing content and only one context leads to an improvement in the quality of sarcasm detection, as opposed to the content-based model, albeit with weaker results than the baseline model. However, better results were obtained by simultaneously modeling the user's writing style and personality. In other words, these findings demonstrate that the combined approach based on the content and modeling of the user's writing style and personality as context outperforms the previous proposed methods and baseline. Nevertheless, it appears that better results can be attained by incorporating other contextual information, such as users' relationships.

5. Conclusion and future directions

Sarcasm detection has been a widely researched topic. What seems to affect the detection of a sarcastic statement is context. Previous studies have used various fields as context, with the most commonly used being the user's prior posts and the conversation. Our paper presents a model for the automatic detection of sarcasm in tweets. To determine whether a tweet is sarcastic or not, our model analyzes the user's writing style and personality along with the content of the tweet. It does so by modeling the recent tweets of the user as well as the text of the desired tweet.

Several experiments were conducted to evaluate the proposed method. The results of these experiments showed that using context not only achieved better results than the content-based method, but also combining content with the user's writing style and personality as context achieved better results than the baseline methods in the field.

The paper concludes by suggesting some research directions that can be explored in future studies. First and foremost, we seek to improve the accuracy of our sarcasm detection model by adding other contextual data. Collecting additional information about the user can make sarcasm detection more precise. To this end, using a multidimensional modeling approach that takes into account various aspects of the user, such as users' profile information and relationships in the network is recommended. This can enhance the performance of the sarcasm detection method and provide more accurate results.

6. References

[1] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in *Lrec 2014 proceedings*, ELRA, 2014.

[2] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Comput. Surv. CSUR*, vol. 50, no. 5, pp. 1–22, 2017.

[3] Silviu Oprea and Walid Magdy. isarcasm: A dataset of intended sarcasm. Silviu Oprea, Walid Magdy, isarcasm: A dataset of intended sarcasm, 2019, arXiv preprint arXiv:1911.03123, 2019.

[4] 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.

[5] A. Reyes, P. Rosso, and D. Buscaldi, “From humor recognition to irony detection: The figurative language of social media,” *Data Knowl. Eng.*, vol. 74, pp. 1–12, 2012.
[6] A. Reyes, P. Rosso, and T. Veale, “A multidimensional approach for detecting irony in twitter,” *Lang. Resour. Eval.*, vol. 47, pp. 239–268, 2013.
[7] S. K. Bharti, K. S. Babu, and S. K. Jena, “Parsing-based sarcasm sentiment recognition in twitter data,” in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pp. 1373–1380, 2015.
[8] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, “Sarcasm as contrast between a positive sentiment and negative situation,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 704–714, 2013.
[9] R. Akula and I. Garibay, “Interpretable multi-head self-attention architecture for sarcasm detection in social media,” *Entropy*, vol. 23, no. 4, p. 394, 2021.
[10] P. Carvalho, L. Sarmento, M. J. Silva, and E. De Oliveira, “Clues for detecting irony in user-generated contents: oh...!! it’s" so easy";-,” in *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pp. 53–56, 2009.
[11] R. González-Ibánez, S. Muresan, and N. Wacholder, “Identifying sarcasm in twitter: a closer look,” in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 581–586, 2011.
[12] O. Tsur, D. Davidov, and A. Rappoport, “ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews,” in *Proceedings of the International AAAI Conference on Web and Social Media*, pp. 162–169, 2010.
[13] D. Davidov, O. Tsur, and A. Rappoport, “Semi-supervised recognition of sarcasm in Twitter and Amazon,” in *Proceedings of the fourteenth conference on computational natural language learning*, pp. 107–116, 2010.
[14] B. C. Wallace and E. Charniak, “Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1035–1044, 2015.
[15] S. Poria, E. Cambria, D. Hazarika, and P. Vij, “A deeper look into sarcastic tweets using deep convolutional neural networks,” *ArXiv Prepr. ArXiv161008815*, 2016.
[16] A. Rajadesingan, R. Zafarani, and H. Liu, “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, 2015.
[17] M. Zhang, Y. Zhang, and G. Fu, “Tweet sarcasm detection using deep neural network,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers*, pp. 2449–2460, 2016.
[18] S. Amir, B. C. Wallace, H. Lyu, and P. C. M. J. Silva, “Modelling context with user embeddings for sarcasm detection in social media,” *ArXiv Prepr. ArXiv160700976*, 2016.
[19] 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*, pp. 161–169, 2016.
[20] L. Liu, D. Zhang, and W. Song, “Modeling sentiment association in discourse for humor recognition,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 586–591, 2018.
[21] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *ArXiv Prepr. ArXiv13013781*, 2013.
[23] R.-A. Potamias, G. Siolas, and A. Stafylopatis, “A robust deep ensemble classifier for figurative language detection,” in *Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24-26, 2019, Proceedings 20*, Springer, pp. 164–175, 2019.
[24] L. Kumar, A. Somani, and P. Bhattacharyya, “‘ Having 2 hours to write a paper is fun!’: Detecting Sarcasm in Numerical Portions of Text,” *ArXiv Prepr. ArXiv170901950*, 2017.
[25] Y.-H. Huang, H.-H. Huang, and H.-H. Chen, “Irony detection with attentive recurrent neural networks,” in *Advances in Information Retrieval: 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings 39*, Springer, pp. 534–540, 2017.
[26 38] S. Zhang, X. Zhang, J. Chan, and P. Rosso, “Irony detection via sentiment-based transfer learning,” *Inf. Process. Manag.*, vol. 56, no. 5, pp. 1633–1644, 2019.
[27] Tay, Y., Luu, A. T., Hui, S. C., & Su, J. (2018, July). Reasoning with Sarcasm by Reading In-Between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1010-1020).
[28] 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*, pp. 482–491, 2017.
[29] 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*, pp. 2115–2124, 2019.
[30] A. Kumar and G. Garg, “Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets,” *J. Ambient Intell. Humaniz. Comput.*, pp. 1–16, 2019.
[31] 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.

[32] A. Onan and M. A. Toçoğlu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, vol. 9, pp. 7701–7722, 2021.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv Prepr. ArXiv181004805*, 2018.

[34] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[35] T. Brown *et al.*, "Language models are few-shot learners," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020.

[36]N. Babanejad, H. Davoudi, A. An, and M. Papagelis, "Affective and contextual embedding for sarcasm detection," in *Proceedings of the 28th international conference on computational linguistics*, pp. 225–243, 2020.

[37] A. Kumar, V. T. Narapareddy, P. Gupta, V. A. Srikanth, L. B. M. Neti, and A. Malapati, "Adversarial and auxiliary features-aware bert for sarcasm detection," in *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, pp. 163–170, 2021.

[38] B. Liang *et al.*, "Multi-modal sarcasm detection via cross-modal graph convolutional network," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, pp. 1767–1777, 2022.

[39] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv Prepr. ArXiv201011929*, 2020.

[40] E. Savini and C. Caragea, "Intermediate-task transfer learning with BERT for sarcasm detection," *Mathematics*, vol. 10, no. 5, p. 844, 2022.

[41] Z. Wen *et al.*, "Sememe knowledge and auxiliary information enhanced approach for sarcasm detection," *Inf. Process. Manag.*, vol. 59, no. 3, p. 102883, 2022.

[42] M. Yuan, Z. Mengyuan, L. Jiang, Y. Mo, and X. Shi, "stce at semeval-2022 task 6: Sarcasm detection in english tweets," in *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pp. 820–826, 2022.

[43] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *ArXiv Prepr. ArXiv190711692*, 2019.

[44] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," *ArXiv Prepr. ArXiv191102116*, 2019.

[45] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," *ArXiv Prepr. ArXiv211109543*, 2021.

[46] Zhang Y, Ma D, Tiwari P, Zhang C, Masud M, Shorfuzzaman M, Song D. Stance-level Sarcasm Detection with BERT and Stance-centered Graph Attention Networks. ACM Transactions on Internet Technology. 2023 May 18;23(2):1-21.

[47] Yue T, Mao R, Wang H, Hu Z, Cambria E. KnowleNet: Knowledge fusion network for multimodal sarcasm detection. Information Fusion. 2023 Dec 1;100:101921.

[48] Tan YY, Chow CO, Kanesan J, Chuah JH, Lim Y. Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning. Wireless personal communications. 2023 Apr;129(3):2213-37.

[49] Vitman O, Kostiuk Y, Sidorov G, Gelbukh A. Sarcasm detection framework using context, emotion and sentiment features. Expert Systems with Applications. 2023 Dec 30;234:121068.

[50] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, PMLR, 2014, pp. 1188–1196.

[51] Kadriu A, Abazi L, Abazi H. Albanian text classification: Bag of words model and word analogies. Business Systems Research: International journal of the Society for Advancing Innovation and Research in Economy. 2019 Apr 18;10(1):74-87

[52] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *International workshop on artificial intelligence and statistics*, PMLR, pp. 246–252, 2005.

[53] B. Shmueli, L.-W. Ku, and S. Ray, "Reactive supervision: A new method for collecting sarcasm data," *ArXiv Prepr. ArXiv200913080*, 2020.

[54] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *J. Mach. Learn. Res.*, vol. 13, no. 2, 2012.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv Prepr. ArXiv14126980*, 2014.

[56] A.-C. Băroiu and Ș tefan Trăuș an-Matu, "Automatic sarcasm detection: Systematic literature review," *Information*, vol. 13, no. 8, p. 399, 2022.