Bi-ISCA: Bidirectional Inter-Sentence Contextual Attention Mechanism for Detecting Sarcasm in User Generated Noisy Short Text

Prakamya Mishra
Shiv Nadar University
pm669@snu.edu.in

Saroj Kaushik
Shiv Nadar University
saroj.kaushik@snu.edu.in

Kuntal Dey
Accenture Technology Labs
kuntal.dey@accenture.com

Abstract

Many online comments on social media platforms are hateful, humorous, or sarcastic. The sarcastic nature of these comments (especially the short ones) alters their actual implied sentiments, which leads to misinterpretations by the existing sentiment analysis models. A lot of research has already been done to detect sarcasm in the text using user-based, topical, and conversational information but not much work has been done to use inter-sentence contextual information for detecting the same. This paper proposes a new state-of-the-art deep learning architecture that uses a novel Bidirectional Inter-Sentence Contextual Attention mechanism (Bi-ISCA) to capture inter-sentence dependencies for detecting sarcasm in the user-generated short text using only the conversational context. The proposed deep learning model demonstrates the capability to capture explicit, implicit, and contextual incongruous words & phrases responsible for invoking sarcasm. Bi-ISCA generates state-of-the-art results on two widely used benchmark datasets for the sarcasm detection task (Reddit and Twitter). To the best of our knowledge, none of the existing state-of-the-art models use an inter-sentence contextual attention mechanism to detect sarcasm in the user-generated short text using only conversational context.

1 Introduction

Sentiment analysis is one of the most important natural language processing (NLP) applications. Its goal is to identify, extract, quantify, and study subjective information. The sudden rise in the usage of social media platforms as a means of communication has led to a vast amount of data being shared between its users on a wide range of topics. This type of data is very helpful to several organizations for analyzing the sentiments of people towards products, movies, political events, etc. Understanding the unique intricacies of the human language remains one of the most important pending NLP problems of this time. Humans regularly use sarcasm as a crucial part of the day-to-day conversations when venting, arguing, or maybe engaging on social media platforms. Sarcastic remarks on these platforms inflict problems on the existing sentiment analysis systems in identifying the true intentions of the users.

The Cambridge Dictionary[1] describes sarcasm as an irony conveyed hilariously or amusingly to criticize something. Sarcasm may not show criticism on the surface but instead might have a criticizing implied meaning. Such a figurative aspect of sarcasm makes it difficult to be detected in the modern micro texts [10]. Several linguistic research has been done to analyze different aspects of sarcasm. Kind of responses evoked because of comments has been considered a major indicator of sarcasm [7]. Wilson [33] states that circumstantial incongruity between a comment and its corresponding contextual information plays an important role in implying sarcasm.

Previous research works have used policy-based, statistical, and deep-learning-based methods for detecting sarcasm. The use of contextual information like conversational context, author personality features, or prior knowledge of the topic, have proved to be very useful. Khattri et al. [17] used sentiments of the author's historical tweets as context. Rajadesingan et al. [25] used personality features like the author's familiarity with twitter, language (structure and word usage), and the author's familiarity with sarcasm (history of previous sarcastic tweets) for consolidating context. Bamman and Smith [2] explored the use of historical terms, topics, and sentiments along with profile information as the author's context. They also exploited the use of conversational context like the immediate previous tweets in the thread. Joshi et al. [16] demonstrated that concatenation of preceding comment with the objective comment in a discussion forum led to an increase in the precision score.

Overall in recent years a lot of work has been done to use different types of contextual information for sarcasm detection but none of them have used inter-sentence dependencies. In this paper, we propose a novel Bidirectional Inter-Sentence Contextual Attention mechanism (Bi-ISCA) based deep learning neural network for sarcasm detection. The main contribution of this paper can be summarised as follows:

- We propose a new state-of-the-art deep learning architecture that uses a novel Bidirectional Inter-Sentence Contextual attention mechanism (Bi-ISCA) for detecting sarcasm in short texts (short texts are more difficult to analyze due to shortage of contextual information).

- Bi-ISCA focuses on only using the conversational contextual comment/tweet for detecting sarcasm rather than using any other topical/personality-based features, as using only the contextual information enriches the model's ability to capture syntactical and semantical textual properties responsible for invoking sarcasm.

- We also explain model behavior and predictions by visualizing attention maps generated by Bi-ISCA, which helps in identifying significant parts of the sentences responsible for invoking sarcasm.

The rest of the paper is organized as follows. Section 2 describes the related work. Then section 3, explains the proposed model architecture for detecting sarcasm. Section 4 will describe the datasets used, pre-processing pipeline, and training details for reproducibility. Then experimental results are explained in section 5 and section 6 illustrates model behavior and predictions by visualizing attention maps. Finally we conclude in section 7.

2 Related Work

A diverse spectrum of approaches has been used to detect sarcasm. Recent sarcasm detection approaches have either mainly focused on using machine learning based approaches that leverage the use of explicitly declared relevant features or they focus on using neural network based deep learning approaches that do not require handcrafted features. Also, the recent advances in using deep learning for preforming natural language processing tasks have led to a promising increase in the performance of these sarcasm detection systems.

A lot of research has been done using bag of words as features. However, to improve performance, scholars started to explore the use of several other semantic and syntactical features like punctuations [31]; emotion marks and intensifiers [23]; positive verbs and negative phrases [27]; polarity skip grams [26]; synonyms & ambiguity[3]; implicit and explicit incongruity-based [16]; sentiment flips [25]; affect-based features derived from multiple emotion lexicons [8].

---

[1]https://dictionary.cambridge.org/

Every day an enormous amount of short text data is generated by users on popular social media platforms like Twitter[2] and Reddit[3]. Easy accessibility of such data sources has enticed researchers to use them for extracting user-based and discourse-based features. Hazarika et al. [13] utilized contextual information by making user-embeddings for capturing indicative behavioral traits. These user-embeddings incorporated personality features along with the author's writing style (using historical posts). They also used discourse comments along with background cues and topical information for detecting sarcasm. They performed their experiments on the largest Reddit dataset SARC [18]. Many have only used the target text for classification purposes, where a target text is a textual unit that has to be classified as sarcastic or not. Simply using gated recurrent units (GRU) [5] or long short term memory (LSTM) [14] do not capture in between interactions of word pairs which makes it difficult to model contrast and incongruity. Tay et al. [30] were able to solve this problem by looking in-between word pairs using a multi-dimensional intra-attention recurrent network. They focused on modeling the intra-sentence relationships among the words. Kumar et al. [19] exploited the use of a multi-head attention mechanism [32] which could capture dependencies between different representations subspaces in different positions. Their model consisted of a word encoder for generating new word representations by summarizing comment contextual information in a bidirectional manner. On top of that, they used multi-head attention for focusing on different contexts of a sentence, and in the end, a simple multi-layer perceptron was used for classification.

There has not been much work done in conversation dependent (comment and reply) approaches for sarcasm detection. Ghaeini et al. [9] proposed a model that not only used information from the target utterance but also used its conversational context to perceive sarcasm. They aimed to detect sarcasm by just using the sequences of sentences, without any extra knowledge about the user and topic. They combined the predictions from utterance-only and conversation-dependent parts for generating its final prediction which was able to capture the words responsible for delivering sarcasm. Ghosh et al. [11] also modeled conversational context for sarcasm detection. They also attempted to derive what parts of the conversational context triggered a sarcastic reply. Their proposed model used sentence embeddings created by taking an average of word embeddings and a sentence-level attention mechanism was used to generate attention induced representations of both the context and the response which was later concatenated and used for classification.

Among all the previous works, [9] and [11] share similar motives of detecting sarcasm using only the conversational context. However, we introduce a novel Bidirectional Inter-Sentence Contextual Attention mechanism (Bi-ISCA) for detecting sarcasm. Unlike previous works, our work considers short texts for detecting sarcasm, which is far more challenging to detect when compared to long texts as long texts provide much more contextual information.

3 Model

This section will introduce the proposed Bi-ISCA: Bidirectional Inter Sentence Contextual Attention based neural network for sarcasm detection (as shown in Figure 1). Sarcasm detection is a binary classification task that tries to predict whether a given comment is sarcastic or not. The proposed model uses comment-reply pairs for detecting sarcasm. The input to the model is represented by $U = [W_1^u, W_2^u, ..., W_n^u]$ and $V = [W_1^v, W_2^v, ..., W_n^v]$, where U represents the comment sentence and V represents the reply sentence (both sentences padded to a length of n). Here, $W_i^u, W_j^v \in \mathbb{R}^d$ are $d-$dimensional word embedding vectors. The objective is to predict label y which indicates whether the reply to the corresponding comment was sarcastic or not.

3.1 Intra-Sentence Word Encoder Layer

The primary purpose of this layer is to summarize intra-sentence contextual information from both directions in both the sentences (comment & reply) using Bidirectional Long Short Term Memory Networks (Bi-LSTM). A Bi-LSTM [28] processes information in both the directions using a forward LSTM [14] $\overrightarrow{h}$, that reads the sentence $S = [w_1, w_2, ..., w_n]$ from w_1 to w_n and a backward LSTM $\overleftarrow{h}$ that reads the sentence from w_n to w_1. Hidden states from both the LSTMs are added to get the final hidden state representations of each word. So the hidden state representation of the t_{th} word (h_t)

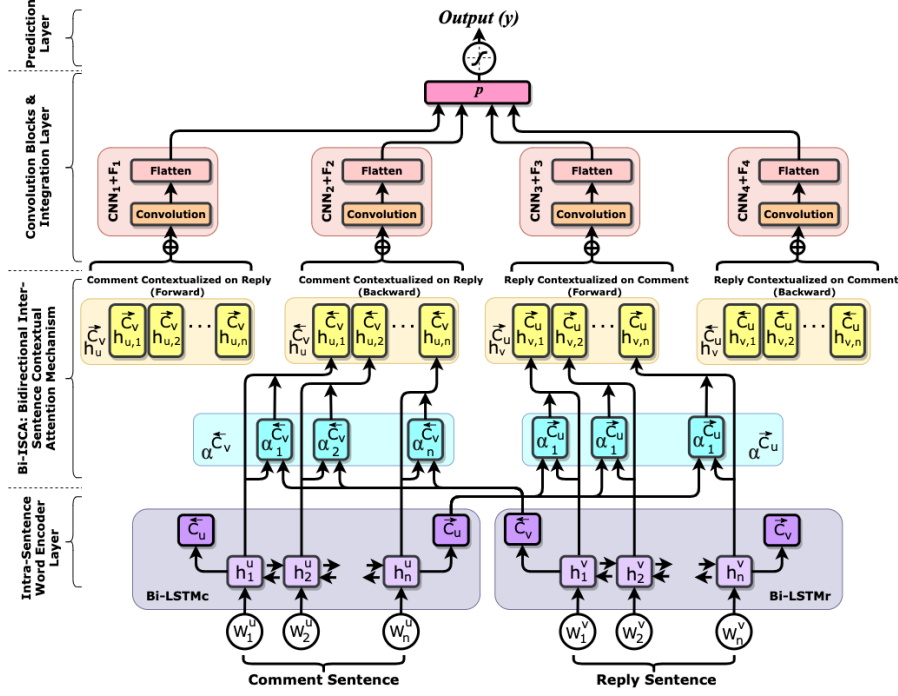[2]www.twitter.com/
[3]www.reddit.com/

Figure 1: Bi-ISCA: Bi-Directional Inter-Sentence Contextual Attention Mechanism for Sarcasm Detection.

can be represented by the sum of t_{th} hidden representations of the forward and backward LSTMs $(\overrightarrow{h_t}, \overleftarrow{h_t})$ as show in equations below.

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t, \overrightarrow{h_{t-1}}); \overleftarrow{h_t} = \overleftarrow{LSTM}(w_t, \overleftarrow{h_{t-1}}) \tag{1}$$

$$h_t = \overleftarrow{h_t} + \overrightarrow{h_t} \tag{2}$$

This Intra-Sentence Word Encoder Layer consists of two independent Bidirectional LSTMs for both comment ($BiLSTM_c$) and reply ($BiLSTM_r$). Apart from the hidden states, both these Bi-LSTMs also generate separate (forward & backward) final cell states represented by $\overleftarrow{C}$ & $\overrightarrow{C}$. The comment sentence U is given as an input to $BiLSTM_c$ and the reply sentence V is given as an input to $BiLSTM_r$. The outputs of both the Bi-LSTMs are represented by the equations 3 and 4.

$$\overrightarrow{C_u}, h^u, \overleftarrow{C_u} = BiLSTM_c(U) \tag{3}$$

$$\overrightarrow{C_v}, h^v, \overleftarrow{C_v} = BiLSTM_r(V) \tag{4}$$

Here, $\overrightarrow{C_u}, \overrightarrow{C_v} \in \mathbb{R}^d$ are the final cell states of the forward LSTMs corresponding to $BiLSTM_c$ & $BiLSTM_r$; $\overleftarrow{C_u}, \overleftarrow{C_v} \in \mathbb{R}^d$ are the final cell states of the backward LSTMs corresponding to $BiLSTM_c$ & $BiLSTM_r$; $h^u = [h_1^u, h_2^u, ..., h_n^u]$ and $h^v = [h_1^v, h_2^v, ..., h_n^v]$ are the hidden state representations of $BiLSTM_c$ & $BiLSTM_r$ respectively, where $h_i^u, h_j^v \in \mathbb{R}^d$ and $h^u, h^v \in \mathbb{R}^{n \times d}$.

3.2 Bi-ISCA: Bidirectional Inter-Sentence Contextual Attention Mechanism

Sarcasm is context-dependent in nature. Even humans sometimes have a hard time understanding sarcasm without having any contextual information. The hidden states generated by both the Bi-LSTMs ($BiLSTM_c$ & $BiLSTM_r$) captures the intra-sentence bidirectional contextual information in comment & reply respectively, but fails to capture the inter-sentence contextual information between them. This paper introduces a novel Bidirectional Inter-Sentence Contextual Attention

4

mechanism (Bi-ISCA) for capturing the inter-sentence contextual information between both the sentences.

Bi-ISCA uses hidden state representations of U & V along with the auxiliary sentence's cell state representations ($\overrightarrow{C}$ & $\overleftarrow{C}$) to capture the inter-sentence contextual information. At first, the attention mechanism captures four sets of attentions scores namely, ($\alpha^{\overrightarrow{C_u}}, \alpha^{\overleftarrow{C_u}}, \alpha^{\overrightarrow{C_v}}, \alpha^{\overleftarrow{C_v}} \in \mathbb{R}^n$). These sets of inter-sentence attention scores are used to generate new inter-sentence contextualized hidden representations. Then ($\alpha^{\overrightarrow{C_u}}, \alpha^{\overleftarrow{C_u}}$) are calculated using the hidden state representations of $BiLSTM_r$ along with the forward and backward final states ($\overrightarrow{C_u}, \overleftarrow{C_u}$) of $BiLSTM_c$ (as shown in equations 5 & 6), similarly ($\alpha^{\overrightarrow{C_v}}, \alpha^{\overleftarrow{C_v}}$) are calculated using the hidden state representations of $BiLSTM_c$ along with the forward and backward final states ($\overrightarrow{C_v}, \overleftarrow{C_v}$) of $BiLSTM_r$ (as shown in equations 7 & 8). In the equations below ($\bullet$) represents a dot product between two vectors.

$$\alpha^{\overrightarrow{C_u}} = [\alpha_1^{\overrightarrow{C_u}}, \alpha_2^{\overrightarrow{C_u}}, ..., \alpha_n^{\overrightarrow{C_u}}]; \alpha_i^{\overrightarrow{C_u}} = \overrightarrow{C_u} \bullet h_i^v \tag{5}$$

$$\alpha^{\overleftarrow{C_u}} = [\alpha_1^{\overleftarrow{C_u}}, \alpha_2^{\overleftarrow{C_u}}, ..., \alpha_n^{\overleftarrow{C_u}}]; \alpha_i^{\overleftarrow{C_u}} = \overleftarrow{C_u} \bullet h_i^v \tag{6}$$

$$\alpha^{\overrightarrow{C_v}} = [\alpha_1^{\overrightarrow{C_v}}, \alpha_2^{\overrightarrow{C_v}}, ..., \alpha_n^{\overrightarrow{C_v}}]; \alpha_i^{\overrightarrow{C_v}} = \overrightarrow{C_v} \bullet h_i^u \tag{7}$$

$$\alpha^{\overleftarrow{C_v}} = [\alpha_1^{\overleftarrow{C_v}}, \alpha_2^{\overleftarrow{C_v}}, ..., \alpha_n^{\overleftarrow{C_v}}]; \alpha_i^{\overleftarrow{C_v}} = \overleftarrow{C_v} \bullet h_i^u \tag{8}$$

In the next step, the above calculated sets of inter-sentence attention scores $\alpha^{\overrightarrow{C_u}}, \alpha^{\overleftarrow{C_u}}$) are multiplied back with the hidden state representations of $BiLSTM_r$ to generate two new set of hidden representations $h_v^{\overrightarrow{C_u}}, h_v^{\overleftarrow{C_u}} \in \mathbb{R}^{n \times d}$ of the reply sentence namely, reply contextualized on comment (forward) & reply contextualized on comment (backward) respectively (as shown in equations 9 & 10). Similarly $\alpha^{\overrightarrow{C_v}}, \alpha^{\overleftarrow{C_v}}$ are multiplied back with the hidden state representations of $BiLSTM_c$ to generate two new set of hidden representations $h_u^{\overrightarrow{C_v}}, h_u^{\overleftarrow{C_v}} \in \mathbb{R}^{n \times d}$ of the comment sentence namely, comment contextualized on reply (forward) & comment contextualized on reply (backward) respectively (as shown in equations 11 & 12). In the equations below ($\times$) represents multiplication between a scalar and a vector.

$$h_v^{\overrightarrow{C_u}} = [h_{v,1}^{\overrightarrow{C_u}}, h_{v,2}^{\overrightarrow{C_u}}, ..., h_{v,n}^{\overrightarrow{C_u}}], ; h_{v,i}^{\overrightarrow{C_u}} = \alpha_i^{\overrightarrow{C_u}} \times h_i^v \tag{9}$$

$$h_v^{\overleftarrow{C_u}} = [h_{v,1}^{\overleftarrow{C_u}}, h_{v,2}^{\overleftarrow{C_u}}, ..., h_{v,n}^{\overleftarrow{C_u}}], ; h_{v,i}^{\overleftarrow{C_u}} = \alpha_i^{\overleftarrow{C_u}} \times h_i^v \tag{10}$$

$$h_u^{\overrightarrow{C_v}} = [h_{u,1}^{\overrightarrow{C_v}}, h_{u,2}^{\overrightarrow{C_v}}, ..., h_{u,n}^{\overrightarrow{C_v}}], ; h_{u,i}^{\overrightarrow{C_v}} = \alpha_i^{\overrightarrow{C_v}} \times h_i^u \tag{11}$$

$$h_u^{\overleftarrow{C_v}} = [h_{u,1}^{\overleftarrow{C_v}}, h_{u,2}^{\overleftarrow{C_v}}, ..., h_{u,n}^{\overleftarrow{C_v}}], ; h_{u,i}^{\overleftarrow{C_v}} = \alpha_i^{\overleftarrow{C_v}} \times h_i^u \tag{12}$$

3.3 Integration and Final Prediction

The proposed model uses Convolutional Neural Networks (CNN) [21] for capturing location-invariant local features from the newly obtained contextualized hidden representations $h_u^{\overleftarrow{C_v}}, h_u^{\overrightarrow{C_v}}, h_v^{\overleftarrow{C_u}}, h_v^{\overrightarrow{C_u}}$. Four independent CNN blocks ($CNN_1, CNN_2, CNN_3, CNN_4$) are used, corresponding to each of the newly obtained contextualized hidden representations. Each CNN block consists two convolutional layers. Both the convolution layer consist of k filters of height h. The role of these filters is to detect particular features at different locations of the input. The output c_i^l of the lth layer consists of k^l feature maps of height h. The ith feature map (c_i^l) is calculated as:

$$c_i^l = b_i^l + \sum_{k^{l-1}}^{j=1} K_{i,j}^l * c_j^{l-1} \tag{13}$$

In the above equation, b_i^l is a bias matrix and $K_{i,j}^l$ is a filter connecting j^{th} feature map of layer $(l-1)$ to the ith feature map of layer (l). The output of each convolution layer is passed through a activation function f. The proposed model uses *Leaky*ReLu as its activation function.

$$f = \begin{cases} a * x, & \text{for } x \geq 0; a \in \mathbb{R} \tag{14} \\ x, & \text{for } x < 0 \tag{15} \end{cases}$$

5

Table 1: Statics of the SARC dataset and FigLang 2020 workshop Twitter dataset.

			No. of comment-reply pairs		Avg. no. of words per comment		Avg. no. of words per reply	
			Sarcastic	Non-Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic	Non-Sarcastic
Training set	Reddit	Balanced	81205	81205	12.69	12.67	12.19	12.21
		Imbalanced	16303	81205	12.69	12.65	12.15	12.21
	Twitter	Balanced	3496	3496	24.97	24.97	24.25	24.25
Testing set	Reddit	Balanced	9058	9058	12.71	12.64	12.14	12.22
		Imbalanced	1747	9058	12.73	12.69	12.20	12.21
	Twitter	Balanced	874	874	24.97	24.97	24.25	24.25

For each of the CNN blocks, the corresponding contextualized hidden representations are first concatenated ($\oplus$) and then given as input. The outputs of all the CNN blocks are flattened ($F_1, F_2, F_3, F_4 \in \mathbb{R}^{dk}$) and concatenated to generate a new vector ($p \in \mathbb{R}^{4dk}$), where d represents the dimension of the hidden representation and k represents number of convolutional filters used. This concatenated (p) vector is then given as input to a dense layer having $4dk$ neurons and is followed by the final sigmoid prediction layer.

$$F_1 = CNN_1([h_{u,1}^{\overrightarrow{C_v}} \oplus h_{u,2}^{\overrightarrow{C_v}} \oplus .... \oplus h_{u,n}^{\overrightarrow{C_v}}]) \tag{16}$$

$$F_2 = CNN_2([h_{u,1}^{\overleftarrow{C_v}} \oplus h_{u,2}^{\overleftarrow{C_v}} \oplus .... \oplus h_{u,n}^{\overleftarrow{C_v}}]) \tag{17}$$

$$F_3 = CNN_3([h_{v,1}^{\overrightarrow{C_u}} \oplus h_{v,2}^{\overrightarrow{C_u}} \oplus .... \oplus h_{v,n}^{\overrightarrow{C_u}}]) \tag{18}$$

$$F_4 = CNN_4([h_{v,1}^{\overleftarrow{C_u}} \oplus h_{v,2}^{\overleftarrow{C_u}} \oplus .... \oplus h_{v,n}^{\overleftarrow{C_u}}]) \tag{19}$$

$$p = [F_1 \oplus F_2 \oplus F_3 \oplus F_4] \tag{20}$$

$$\hat{y} = \sigma(Wp + b), \quad W \in \mathbb{R}^{4dk}; b \in \mathbb{R} \tag{21}$$

The proposed model uses the binary cross-entropy as the training loss function as shown in equation 22. Here (L) is the cost function, $\hat{y}_i \in \mathbb{R}$ represents the output of the proposed model, $y_i \in \mathbb{R}$ represents the true label and $N \in \mathbb{N}$ represents the number of training samples.

$$L = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i) \tag{22}$$

4 Evaluation Setup

4.1 Dataset

This paper focuses on detecting sarcasm in the user-generated short text using only the conversational context. Social media platforms like Reddit and Twitter are widely used by users for posting opinions and replying to other's opinions. They have proved to be of a great source for extracting conversational data. So the experiments were conducted on two publicly available benchmark datasets (Reddit & Twitter) used for the sarcasm detection task. Both the datasets consist of comments and reply pairs.

SARC[4] **Reddit** [18] is the largest dataset available for sarcasm detection containing millions of sarcastic/non-sarcastic comments-reply pairs from the social media site Reddit. This dataset was generated by scraping comments from Reddit containing the \s (sarcasm) tag. It contains replies, their parent comment (acts as context), and a label that shows whether the reply was sarcastic/non-sarcastic to their corresponding parent comment. To compare the performance of the model on a different dataset (latest), the proposed model was also evaluated on the Twitter dataset provided in the **FigLang**[5] **2020 workshop** [12] for the "sarcasm detection shared task". This consists of sarcastic/non-sarcastic tweets and their corresponding contextual parent tweets. The sarcastic tweets were collected using hashtags like #sarcasm, #sarcastic, and #irony, similarly non-sarcastic tweets were collected using hashtags like #happy, #sad, and #hate. This dataset sometime contains more than one contextual parent tweet, so in those cases, all of the contextual tweets are considered independently with the target tweet.

---

[4]https://nlp.cs.princeton.edu/SARC/2.0/
[5]sites.google.com/view/figlang2020

In both the datasets, replies are the target comment/tweet to be classified as sarcastic/non-sarcastic, and their corresponding parent comment/tweet acts as context. Both the datasets constitute of comments/tweets of varying lengths, but because this paper only focuses on detecting sarcasm in the short text, only the short comment/reply pairs were used. Comment/reply sentences of length (no. of words) less than 20, 40 were used in the case of SARC and Twitter dataset respectively. In both cases, the balanced datasets contain equal proportions of sarcastic/non-sarcastic comment/reply pairs, and the imbalanced datasets maintain a 20:80 ratio (approximately) between sarcastic and non-sarcastic comment/reply pairs. Testing was done on 10% of the dataset and the rest was used for training. 10% of the training set was used for validation purposes. Statistics of both the datasets are shown in Table 1.

4.2 Data Preprocessing

The preprocessing of the textual data was done by first lower-casing all the sentences and separating punctuations from the words. We do not remove the stop-words because we believe that sometimes stop-words play a major role in making a sentence sarcastic e.g., *"is it?"* and *"am I?"*. The problem with social media platforms is that, users use a lot of abbreviations, shortened words and slang words like, *"IMO"* for *"in my opinion"*, *lmk"* for *"let me know "*, *"fr"* for *"for"*, etc. These words are challenging to taken care of in the NLP tasks, particularly in the automatic discovery of flexible word usages. So to solve this problem, these words are converted to their corresponding full-forms using abbreviation/slang word dictionaries obtained from urban dictionary[6]. After this, all the sentences were tokenized into a list of words. The proposed model had a fixed input size for both comment and reply, but not all the sentences were of the same length. So all the sentences were padded to the length of the longest sentence (20 in the case of the Reddit dataset and 40 in the case of the Twitter dataset). Word embeddings are used to give semantically-meaningful dense representations to the words. Word-based embeddings are constructed using contextual words whereas character-based embeddings are constructed from character n-grams of the words. Character-based in contrast to the Word-based embeddings solves the problem of out of vocabulary words and performs better in the case of infrequent words by creating word embeddings based only on their spellings. So for generating proper representations for words we have used FastText[7], a character-based word embedding. This would not only give words better representation compared to the word-based model but also incorporate slang/shortened/infrequent words (which commonly appear in social media platforms).

4.3 Training Details

We have used macro-averaged (*F1*) and accuracy (*Acc*) scores as the evaluation metric, as it is standard for the sarcasm detection task. We have also reported Precision (*P*) and Recall (*R*) scores in the case of the Twitter dataset as well as for the Reddit dataset (wherever available). Hyperparameter tuning was used to find optimum values of the hyperparameters. The FastText embeddings used were of size $d = 30$ and were trained for 30 iterations having window size of 3, 5 in the case of SARC, and Twitter dataset respectively. The number of filters in all the convolutional blocks were [64, 64] of height [2, 2]. The learning optimizer used is Adam with an initial learning rate of 0.01. The value of α in all the *Leaky*ReLu layers was set to 0.3. All the models were trained for 20 epochs. L2 regularization set to 10^{-2} is applied to all the feed-forward connections along with early stopping having the patience of 5 to avoid overfitting. The mini-batch size was tuned amongst {100, 500, 1000, 2000, 3000, 4000} and was observed that mini-batch size of 2000, 500 gave the best performance for the SARC and Twitter dataset respectively.

The recent success of transformer-based language models has led to their wide usage in sentiment analysis tasks. They are known for generating high quality high dimensional word representations (768-dimensional for BERT). Their only drawback is that they require high processing power and memory to train. The above-mentioned configuration of the proposed model generates ≈1120K trainable parameters, and increasing either the embedding size or the number of tokens in a sentence led to an exponential increase in the number of trainable parameters. So due to computational resource limitations, we limited our experiments to lower-dimensional word embeddings.

---

[6]https://www.urbandictionary.com/
[7]https://fasttext.cc/

Table 2: (a) Results on the SARC dataset. Models with `blue` backgrounds use only contextual text for detecting sarcasm. (b) Results on the FigLang 2020 workshop Twitter dataset.

(a)

Models	Balance				Imbalanced			
	Acc	F1	P	R	Acc	F1	P	R
CNN-SVM [24] †⋆	68.0	68.0	–	–	69.0	79.0	–	–
AMR [9] ‡	69.5	69.5	74.8	69.7	–	–	–	–
[11] ‡	–	67.8	68.2	67.9	–	–	–	–
CUE-CNN [1] †⋆	70.0	69.0	–	–	73.0	81.0	–	–
MHA-BiLSTM [19] †	–	77.5	72.6	83.0	–	56.8	60.3	53.7
CASCADE [13] ‡⋆	77.0	77.0	–	–	79.0	86.0	–	–
CASCADE (only discourse features) ‡	68.0	66.0	–	–	68.0	78.0	–	–
Bi-ISCA (this paper) ‡	**72.3**	**75.7**	**74.2**	**77.6**	**71.9**	**74.4**	**73.0**	**75.8**
% increase w.r.t CASCADE (only discourse features)	6.3↑	14.7↑	–	–	5.7↑	4.6↓	–	–

‡ Uses only target sentence, ‡ Uses context along with target sentence,
⋆ Uses personality-based features

(b)

Models	P	R	F1
Baseline ($LSTM_{attn}$)	70.0	66.9	68.0
BERT-Large+BiLSTM+SVM [4]	73.4	73.5	73.4
BERT+CNN+LSTM [29]	74.2	74.6	74.1
RoBERTa+LSTM [20]	77.3	77.4	77.2
RoBERT-Large [6]	79.1	79.4	79.0
RoBERT+Multi-Initialization Ensemble [15]	79.2	79.3	79.1
BERT + BiLSTM + NeXtVLAD + Context Ensemble + Data Augmentation [22]	93.2	93.6	93.1
Bi-ISCA (this paper)	**89.4**	**94.8**	**91.7**

5 Results

Bi-ISCA focuses on only using the contextual comment/tweet for detecting sarcasm rather than using any other topical/personality-based features. Using only the contextual information enriches the model's ability to capture syntactical and semantical textual properties responsible for invoking sarcasm in any type of conversation. Table 2a reports performance results on the SARC datasets. For comparison purposes, F1-score (*F1*), Accuracy score (*Acc*), Precision (*P*) and Recall (*R*) were used.

When compared with the existing works, Bi-ISCA was able to outperform all the models that use only conversational context for sarcasm detection (improvement of 11.7% in F1 score when compared to [11]; 8.9% in F1 score and 4% in accuracy when compared to AMR [9]) and was even able to perform better than the models that use personality-based features along with the target sentence for detecting sarcasm (improvement of 11.3% in F1 and 6.3% in accuracy score when compared to CNN-SVM [24]; 9.7% in F1 score and 4.8% in accuracy when compared to CUE-CNN [1]). MHA-BiLSTM [19] had a 2.3% higher F1 score in the balanced dataset but Bi-ISCA was able to show drastic improvement of 31% in the imbalanced dataset, which demonstrated the ability of Bi-ISCA to handle class imbalance.

The current state-of-the-art on the SARC dataset is achieved by CASCADE. Even though CASCADE uses personality-based features and contextual information along with large sentences of average length ≈55-62 (very large compared to our dataset, which gives them the advantage of using a lot more contextual information), Bi-ISCA was able to achieve an F1 score comparable to it (despite using relatively short text). In comparison with CASCADE that only uses discourse-based features, Bi-ISCA performed drastically better with an increase of 14.7% in F1 and 6.3% in accuracy score for the balanced dataset.

Bi-ISCA clearly demonstrated its capabilities to robustly handle an imbalance in the dataset, although it was unable to outperform both the CASCADE models. This slightly poor performance in the imbalanced dataset can be explained by the length of sentences used by CASCADE, which are significantly (≈5 times) greater than the ones on which Bi-ISCA was tested. Longer sentences result in increased contextual information which improves performance especially in the case of imbalance where little extra information can lead to a drastic increase in performance.

Table 2b reports Precision (*P*), Recall (*R*), and F1-score (*F1*) of different models from the leaderboard of FigLang 2020 sarcasm detection shared task using the Twitter dataset. In this case, not only Bi-ISCA was able to outperform the baseline model [12] (improvement of 19%, 28% & 24% in precision, recall, and F1 score respectively), but was also able to perform comparably to the state-of-the-art [22] with a 1.2% increase in recall, which further validates the performance of the proposed model. Even though all the models other than the baseline in Table 2b are a transformer-based model, Bi-ISCA was able to outperform them all.

6 Discussion

The attention scores generated by the attention mechanism makes the proposed model highly interpretable. Table 3 showcases the distribution of the attention scores over four sarcastic (correctly predicted by Bi-ISCA) comment-reply pairs from the SARC dataset. Not only the proposed model

Table 3: Attension weight distribution in reddit comment-reply pairs. Here *CcR* represents "Comment contextualized on Reply" whereas *RcC* represents "Reply contextualized on Comment"; **(R) & (L)** represents forward & backward attention.

1.	*CcR (R):*	dog traumatized by abuse is caressed for the first tim
	CcR (L):	dog traumatized by abuse is caressed for the first tim
	RcC (R):	what amazing idea to force a dog to be pet while cornered when its terrified
	RcC (L):	what amazing idea to force a dog to be pet while cornered when its terrified
2.	*CcR (R):*	no pay gap between men and women at microsoft company say
	CcR (L):	no pay gap between men and women at microsoft company say
	RcC (R):	m also announces that windows never ever crashes and has security holes
	RcC (L):	m also announces that windows never ever crashes and has security holes
3.	*CcR (R):*	its totally okay to treat autonomous ton vehicles with caution
	CcR (L):	its totally okay to treat autonomous ton vehicles with caution
	RcC (R):	when im blind drunk and get behind the wheel its as though its an autonomous vehicle
	RcC (L):	when im blind drunk and get behind the wheel its as though its an autonomous vehicle
4.	*CcR (R):*	what would it change if they reported it during the game
	CcR (L):	what would it change if they reported it during the game
	RcC (R):	it would cause a pause which is super enjoyable for the audience to watch
	RcC (L):	it would cause a pause which is super enjoyable for the audience to watch

was correctly able to detect sarcasm in these pairs of sentences but was also able to correctly identify words responsible for contextual, explicit, or implicit incongruity which invokes sarcasm.

For example in Pair 1, Bi-ISCA correctly identified explicitly incongruous words like *"amazing"* and *"force"* in the reply sentence which were responsible for the sarcastic nature of the reply. Interestingly the word "traumatized" in the parent comment also had a high attention weight value, which shows that the proposed attention mechanism was able to learn the contextual incongruity between the opposite sentiment words like "traumatized" & "amazing" in the comment-reply pair. Pair 2 demonstrates the model's ability to capture words responsible for invoking sarcasm by making sentences implicitly incongruous. Sarcasm due to implicit incongruity is usually the toughest to perceive. Despite this, Bi-ISCA was able to give high attention weights to words like "announces" and "crashes & security holes". Not only this, but the proposed intra-sentence attention mechanism was also able to learn a link between "microsoft" and "m" (slang for microsoft) without having any prior knowledge related to slangs. Pair 3 is also an example of an explicitly and contextually incongruous comment-reply pair, where the model was successfully able to capture opposite sentiment words & phrases like "blind drunk", "cautious" and "behind the wheel" that made the reply sarcastic in nature. Pair 4 is an example of sarcasm due to implicit incongruity between the words, "pause" & "watch", and contextual incongruity simultaneously between "reported" & "enjoyable", both of which were successfully captured by Bi-BISCA.

7 Conclusion

In this paper, we introduced STEPs-RL for learning phonetically sound spoken-word representations using speech and text entanglement. Our approach achieved an accuracy of 89.47% in predicting phonetic sequences when both gender and dialect of the speaker are used in the auxiliary information. We also compared its performance using different configurations and observed that the performance of the proposed model improved by (1) increasing the spoken word latent representation size, and (2) the addition of auxiliary information like gender and dialect. We were not only able to validate the capability of the learned representations to capture the semantical and syntactical relationships between the spoken-words but were also able to illustrate soundness in the phonetic structure of the generated vector space. For future work, we plan to (1) extend the model to use attention mechanisms, (2) improve performance by using transformer-based architecture, and (3) experimenting on larger datasets, and (4) using features other than MFCCs.

References

[1] Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *CoRR*, abs/1607.00976.

[2] David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

[3] Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58, Baltimore, Maryland. Association for Computational Linguistics.

[4] Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. 2020. Context-aware sarcasm detection using BERT. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 83–87, Online. Association for Computational Linguistics.

[5] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

[6] Xiangjue Dong, Changmao Li, and Jinho D. Choi. 2020. Transformer-based context-aware sarcasm detection in conversation threads from social media. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 276–280, Online. Association for Computational Linguistics.

[7] Jodi Eisterhold, Salvatore Attardo, and Diana Boxer. 2006. Reactions to irony in discourse: evidence for the least disruption principle. *Journal of Pragmatics*, 38(8):1239 – 1256. Focus-on Issue: Discourse and Conversation.

[8] Delia Irazú Hernaundefineddez Farías, Viviana Patti, and Paolo Rosso. 2016. Irony detection in twitter: The role of affective content. *ACM Trans. Internet Technol.*, 16(3).

[9] Reza Ghaeini, Xiaoli Z. Fern, and Prasad Tadepalli. 2018. Attentional multi-reading sarcasm detection. *CoRR*, abs/1809.03051.

[10] Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

[11] Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. *CoRR*, abs/1707.06226.

[12] Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. 2020. A report on the 2020 sarcasm detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 1–11, Online. Association for Computational Linguistics.

[13] Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: Contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1837–1848, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[15] Nikhil Jaiswal. 2020. Neural sarcasm detection using conversation context. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 77–82, Online. Association for Computational Linguistics.

[16] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.

[17] Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, and Mark Carman. 2015. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 25–30.

[18] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Linguistic Resource and Evaluation Conference (LREC)*.

[19] A. Kumar, V. T. Narapareddy, V. Aditya Srikanth, A. Malapati, and L. B. M. Neti. 2020. Sarcasm detection using multi-head attention based bidirectional lstm. *IEEE Access*, 8:6388–6397.

[20] Amardeep Kumar and Vivek Anand. 2020. Transformers on sarcasm detection with context. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 88–92, Online. Association for Computational Linguistics.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[22] Hankyol Lee, Youngjae Yu, and Gunhee Kim. 2020. Augmenting data for sarcasm detection with unlabeled conversation context. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 12–17, Online. Association for Computational Linguistics.

[23] Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, Georgia. Association for Computational Linguistics.

[24] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. *CoRR*, abs/1610.08815.

[25] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 97–106, New York, NY, USA. Association for Computing Machinery.

[26] Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268.

[27] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 704–714. ACL.

[28] M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[29] Himani Srivastava, Vaibhav Varshney, Surabhi Kumari, and Saurabh Srivastava. 2020. A novel hierarchical BERT architecture for sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 93–97, Online. Association for Computational Linguistics.

[30] Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020, Melbourne, Australia. Association for Computational Linguistics.

[31] Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *fourth international AAAI conference on weblogs and social media*.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

[33] Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722 – 1743. Language in Mind: A Tribute to Neil Smith on the Occasion of his Retirement.