 doi:10.20944/preprints202101.0302.v1

Article

Interpretable Multi-Head Self-Attention Architecture for Sarcasm Detection in Social Media

Ramya Akula [1,†,‡] **and Ivan Garibay** [2,‡]

[1] University of Central Florida; ramya.akula@knights.ucf.edu
[2] University of Central Florida; igaribay@ucf.edu
* Correspondence: ramya.akula@knights.ucf.edu; igaribay@ucf.edu
† These authors contributed equally to this work.

Abstract: Sarcasm is a linguistic expression often used to communicate the opposite of what is said, usually something that is very unpleasant with an intention to insult or ridicule. Inherent ambiguity in sarcastic expressions, make sarcasm detection very difficult. In this work, we focus on detecting sarcasm in textual conversations from various social networking platforms and online media. To this end, we develop an interpretable deep learning model using multi-head self-attention and gated recurrent units. Multi-head self-attention module aids in identifying crucial sarcastic cue-words from the input, and the recurrent units learn long-range dependencies between these cue-words to better classify the input text. We show the effectiveness of our approach by achieving state-of-the-art results on multiple datasets from social networking platforms and online media. Models trained using our proposed approach are easily interpretable and enable identifying sarcastic cues in the input text which contribute to the final classification score. We visualize the learned attention weights on few sample input texts to showcase the effectiveness and interpretability of our model.

Keywords: Sarcasm Detection; Self-Attention; Interpretability, Social Media Analysis

1. Introduction

Sarcasm is a rhetorical way of expressing dislike or negative emotions using exaggerated language constructs. It is an assortment of mockery and false politeness to intensify hostility without explicitly doing so. In face-to-face conversation, sarcasm can be identified effortlessly using facial expressions, gestures, and tone of the speaker. However, recognizing sarcasm in textual communication is not a trivial task as none of these cues are readily available. With the explosion of internet usage, sarcasm detection in online communications from social networking platforms [1,2], discussion forums [3,4], and e-commerce websites has become crucial for opinion mining, sentiment analysis, and in identifying cyberbullies, online trolls. The topic of sarcasm received great interest from Neuropsychology [5] to Linguistics [6], but developing computational models for automatic detection of sarcasm is still at its nascent phase. Earlier works on sarcasm detection on texts use lexical (content) and pragmatic (context) cues [7] such as interjections, punctuation, and sentimental shifts, that are major indicators of sarcasm [8]. In these works, the features are hand-crafted which cannot generalize in the presence of informal language and figurative slang widely used in online conversations.

With the advent of deep-learning, recent works [9–13], leverage neural networks to learn both lexical and contextual features, eliminating the need for hand-crafted features. In these works, word embeddings are incorporated to train deep convolutional, recurrent, or attention-based neural networks to achieve state-of-the-art results on multiple large scale datasets. While deep learning-based approaches achieve impressive performance, they lack interpretability. In this work, we also focus on the interpretability of the model along with its high performance. The main contributions of our work are:

- Propose a novel, interpretable model for sarcasm detection using self-attention.
- Achieve state-of-the-art results on diverse datasets and exhibit the effectiveness of our model with extensive experimentation and ablation studies.
- Exhibit the interpretability of our model by analyzing the learned attention maps.

 doi:10.20944/preprints202101.0302.v1

This paper is organized as follows: In Sections 2, and 3, we briefly mention the related works and describe our proposed multi-head self-attention architecture. Section 4 includes details on model implementation, experiments, datasets, and evaluation metrics. Performance and attention analysis of our model are described in Sections 5 and 6, followed by the conclusion of this work.

2. Related Work

Sarcasm is studied for many decades in social sciences, yet developing methods to automatically identify sarcasm in texts is a fairly new field of study. The state-of-the-art automated sarcasm detection models can be broadly segregated into content-based and context-based models.

In content-based approaches, lexical and linguistic cues, syntactic patterns are used to train classifiers for sarcasm detection. Carvalho *et al.* [14], González-Ibánez *et al.* [15], use linguistic features such as interjections, emoticons, and quotation marks. Tsur *et al.* [16], Davidov *et al.* [17] use syntactic patterns and lexical cues associated with sarcasm. The use of positive utterance in a negative context is used as a reliable feature to detect sarcasm by Riloff *et al.* [18]. Linguistic features such as implicit and explicit context incongruity, are used by Joshi *et al.* [8]. In these works, only the input text is used to detect sarcasm without any context information.

Context-based approaches increased in popularity in the recent past with the emergence of various online social networking platforms. As texts from these websites are prone to grammatical errors and extensive usage of slang, using context information helps in better identification of sarcasm. Wallace *et al.* [19], Poria *et al.* [20] detect sarcasm using sentiment and emotional information from the input text as contextual information. While, Amir *et al.* [21], Hazarika *et al.* [22] use personality features of the user as context, Rajadesingan *et al.* [23], Zhang *et al.* [24] use historical posts of the user to incorporate sarcastic tendencies. We show that, context information when available helps in improving the performance of the model but is not essential for sarcasm detection.

Existing works by Wallace *et al.* [19], Ptáček *et al.* [25], Wang *et al.* [26], Joshi *et al.* [27], use handcrafted features such as Bag of Words (BoW), Parts of Speech (POS), sentiment/emotions to train their classifiers. Other works by Liu *et al.* [13], Poria *et al.* [20], Amir *et al.* [21], Zhang *et al.* [24], Ghosh and Veale [28], Vaswani *et al.* [29] use deep-learning to learn meaningful features and classify them. The method which uses handcrafted features are easily interpretable but lack in performance. On the other hand, deep learning-based methods achieve high performance but lack interpretability.

In our work, we propose a deep learning-based architecture for sarcasm detection which leverages self-attention to enable the interpretability of the model while achieving state-of-the-art performance on various datasets.

3. Proposed Approach

Our proposed approach consists of five components: Data Pre-processing, Multi-Head Self-Attention, Gated Recurrent Units(GRU), Classification, and Model Interpretability. The architecture of our sarcasm detection model is shown in Figure 1. Data pre-processing involves converting input text to word embeddings, required for training a deep learning model. To this end, we first apply a standard tokenizer which does stop word removal, stemming, lemmatization, etc., to convert a sentence to a sequence of tokens. We employ pre-trained language models, to convert these tokens to word embeddings. These embeddings form the input to our multi-head self-attention module which identifies words in the input text that provide crucial cues for sarcasm. In the next step, the GRU layer aids in learning long-distance relationships among these highlighted words and output a single feature vector encoding the entire sequence. Finally, a fully-connected layer with sigmoid activation is used to get the final classification score.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

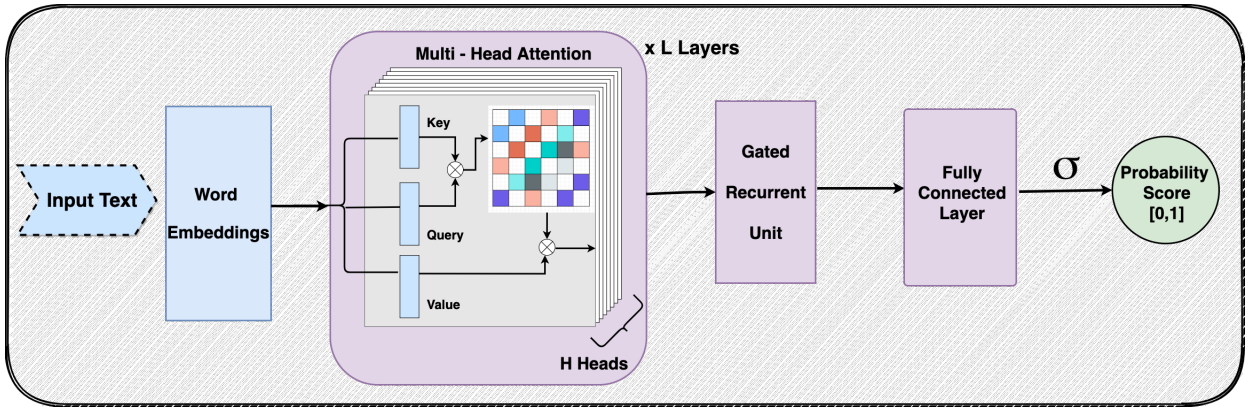

Figure 1. Multi head self-attention architecture for sarcasm detection. Pre-trained word embeddings are extracted for input text and are enhanced by an attention module with L self-attention layers and H heads per layer. Resultant features are passed through a Gated Recurrent Unit and a Feed-forward layer for classification.

3.1. *Data Pre-processing*

Word embeddings range from the clustering of words based on the local context to the embeddings based on a global context that considers the association between a word and every other word in a sentence. Most popular ones that rely on local context are Continuous Bag of Words (CBOW), Skip Grams [30], and Word2Vec [31]. Other predictive models that capture global context are Global Vectors for word representation(GloVe) [32], FastText [33], Embeddings from Language Models (ELMO) [34] and Bidirectional Encoder Representations from Transformers (BERT) [35]. In our work, we employ word embedding which captures global context as we believe it is essential for detecting sarcasm. We show the results of the proposed approach using multiple word embeddings, including, BERT, FastText, and GloVe.

3.2. *Multi-Head Self-Attention*

Given a sentence S, we apply standard tokenizer and use pre-trained models to obtain D dimensional embeddings for individual words in the sentence. These embeddings $S = \{e_1, e_2, ..., e_N\}$, $S \in \mathbb{R}^{N \times D}$ from the input to our model. To detect sarcasm in sentence S, it is crucial to identify specific words that provide essential cues such as sarcastic connotations and negative emotions. The importance of these cue-words is dependent on multiple factors based on different contexts. In our proposed model we leverage multi-head self-attention to identify these cue-words from the input text.

Attention is a mechanism to discover patterns in the input that are crucial for solving the given task. In deep learning, self-attention [29] is an attention mechanism for sequences, which helps in learning the task-specific relationship between different elements of a given sequence to produce a better sequence representation. In the self-attention module, three linear projections: Key (K), Value (V), and Query (Q) of the given input sequence are generated, where $K, Q, V \in \mathbb{R}^{N \times D}$. Attention-map is computed based on the similarity between K, Q, and the output of this module $A \in \mathbb{R}^{N \times D}$ is the scaled dot-product between V and the learned softmax attention (QK^T) as shown in Equation 1.

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V \tag{1}$$

In multi-head self-attention, multiple copies of the self-attention module are used in parallel. Each head captures different relationships between the words in the input text and identify those keywords that aid in classification. In our model, we use a series of multi-head self-attention layers (#L) with multiple heads (#H) in each layer.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

3.3. *Gated Recurrent Units*

Self-attention finds the words in the text which are important in detecting sarcasm. These words can be close to each other or farther apart in the input text. To learn long-distance relationships between these words, we use GRUs. These units are an improvement over standard recurrent neural networks and are designed to dynamically remember and forget the information flow using Reset (r_t) and Update (z_t) gates to solve the vanishing gradient problem.

In our model, we use a single layer of bi-directional GRU to process the sequence A, as these units makes use of the context information from both the directions. Given the input sequence $A \in \mathbb{R}^{N \times D}$, GRU computes hidden states $H = \{h_1, h_2, ..., h_N\}$, $H \in \mathbb{R}^{N \times D}$ for every element in the sequence as follows:

$$\begin{aligned} r_t &= \sigma(W_r A_t + U_r h_{t-1} + b_r) \\ z_t &= \sigma(W_z A_t + U_z h_{t-1} + b_z) \\ \tilde{h}_t &= tanh(W_h A_t + U_h(r_t \odot h_{t-1}) + b_h) \\ h_t &= z_t \odot h_t + (1 - z_t) \odot \tilde{h}_{t-1} \end{aligned} \tag{2}$$

Here $\sigma(.)$ is the element-wise sigmoid function and W, U, b are the trainable weights and biases. r_t, z_t, h_t, $\tilde{h}_t \in \mathbb{R}^d$, where d is the size of the hidden dimension. We consider the final hidden state, h_N, which encodes all the information in the sequence, as an output from this module.

3.4. *Classification*

A single fully-connected feed-forward layer is used with sigmoid activation to compute the final output. Input to this layer is the feature vector h_N from the GRU module and the output is a probability score $y \in [0, 1]$, computed as follows:

$$y = \sigma(W h_N + b), \tag{3}$$

where $W \in \mathbb{R}^{d \times 1}$ are the weights of this layer and b is the bias term. Binary Cross Entropy (BCE) loss between the predicted output y and the ground-truth label $\hat{y}$ is used to train the model.

$$loss(y, \hat{y}) = \hat{y} log(y) + (1 - \hat{y}) log(1 - y) \tag{4}$$

where $\hat{y} \in \{0, 1\}$ is the binary label i.e., 1:Sarcasm and 0:No-sarcasm.

3.5. *Model Interpretability*

Developing models that can explain their predictions is crucial to building trust and faith in deep learning while enabling a wide range of applications with machine intelligence at its backbone. Existing deep learning network architectures such as convolutional and recurrent neural networks are not inherently interpretable and require additional visualization techniques [36,37]. To avoid this, we in this work employ self-attention which is inherently interpretable and allows identifying elements in the input which are crucial for a given task.

4. Experiments

4.1. *Datasets*

4.1.1. **Twitter [18]**

In this dataset, the tweets that contain sarcasm are identified and labeled by the human annotators solely based on the contents of the tweets. These tweets do not depend on prior conversational context. Tweets with no sarcasm or that required prior conversational context are labeled as non-sarcastic.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

4.1.2. **Dialogues [38]**

This Sarcasm Corpus V2 Dialogues dataset is part of Internet Argument Corpus [39] that includes annotated quote-response pairs for sarcasm detection. General sarcasm, hyperbole, rhetorical are the three categories in this dataset. In these quote-response pairs, a quote is a dialogic parent to the response. Therefore, a response post can be mapped to the same quote post or the post earlier in the thread. Here the quoted text is used as a context for sarcasm detection.

4.1.3. **Twitter [9]**

In this dataset, tweets are collected using a Twitter bot named *@onlinesarcasm*. This dataset not only contains tweets and replies to these tweets but also the mood of the user at the time of tweeting. The tweets/re-tweets of the users are the content and the replies to the tweets are the context.

4.1.4. **Reddit [40]**

Self-annotated corpus for sarcasm, SARC 2.0 dataset contains comments from Reddit forums. Sarcastic comments by users are scrapped which are self-annotated by them using \s token to indicate sarcastic intent. In our experiments, we use only the original comment without using any parent or child comments. "Main Balanced" and "Political" variants of the dataset are used in our experiments, the latter consists of comments only from the political subreddit.

4.1.5. **Headlines [41]**

This news headlines dataset is collected from two news websites: Onion and Huffpost. The onion has sarcastic versions of current events whereas Huffpost has real news headlines. Headlines are used as content and the news article is used as context.

Details of these datasets, including the sample counts in train/test splits and the data source, are presented in Table 1.

Source	Train	Test	Total
Twitter, 2013	1,368	588	1,956
Dialogues, 2016	3754	938	4,692
Twitter, 2017	51,189	3,742	54,931
Reddit, 2018	154,702	64,666	219,368
Headlines, 2019	22895	5724	28,619

Table 1: Statistics of datasets used in our experiments. Twitter, 2013 [18], Dialogues, 2016 [38], Twitter, 2017 [9], Reddit, 2018 [40], and Headlines, 2019 [41]. These are sourced from varied online platforms including social networks and discussion forums.

4.2. Implementation Details

We implement our model in PyTorch [42], a deep-learning framework in Python. To tokenize and extract word embeddings for the input text, we use publicly available resources [43]. The embeddings for the words in the input text are passed through a series of multi-head self-attention layers #L , with multiple heads #H in each of the layers. The output from the self-attention layer is passed through a single bi-directional GRU layer with it's hidden dimension $d = 512$. The 512-dimensional output feature vector from the GRU layer is passed through the fully connected layer to get a 1-dimensional output. A sigmoid activation is applied to the final output and BCE loss is used to compute the loss between the ground truth and the predicted probability score. We use Adam optimizer to train our model with approximately 13 million parameters, using a learning rate of 1e-4, batch size of 64, and dropout set 0.2. We use one NVIDIA Pascal Titan-X with 16GB

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

memory for all our experiments. We set $\#H = 8$ and $\#L = 3$ in all our experiments for all the datasets.

4.3. *Evaluation Metrics*

We pose Sarcasm Detection as a classification problem, and use Precision, Recall, F1-Score, and Accuracy as evaluation metrics to test the performance of the trained models. *Precision:* Ratio of the number of correctly predicted sarcastic sentences to the total number of predicted sarcastic sentences. *Recall:* Ratio of correctly predicted sarcastic sentences to the actual number of sarcastic sentences in the ground-truth. *F-score:* Harmonic mean of precision and recall. We use a threshold of 0.5 on the predictions from the model to compute these scores. Apart from these standard metrics we also compute Area Under the ROC Curve (AUC score) which is threshold independent.

Models	**Precision**	**Recall**	**F1**	**AUC**
NBOW	71.2	62.3	64.1	-
Vanilla CNN	71.0	67.1	68.5	-
Vanilla LSTM	67.3	67.2	67.2	-
Attention LSTM	68.7	68.6	68.7	-
Bootstrapping [18]	62.0	44.0	51.0	-
EmotIDM [44]	-	-	75.0	-
Fracking Sarcasm [28]	88.3	87.9	88.1	-
GRNN [24]	66.3	64.7	65.4	-
ELMo-BiLSTM [10]	75.9	75.0	75.9	-
ELMo-BiLSTM FULL [10]	77.8	73.5	75.3	-
ELMo-BiLSTM AUG [10]	68.4	70.8	69.4	-
A2Text-Net [13]	91.7	91.0	90.0	97.0
Our Model	**97.9** (+ 6.2 ↑)	**99.6** (+ 8.6 ↑)	**98.7** (+ 8.7 ↑)	**99.6** (+ 2.6 ↑)

Table 2: Results on Twitter dataset [18].

Models	**Precision**	**Recall**	**F1**	**AUC**
Sarcasm Magnet [9]	73.3	71.7	72.5	-
Sentence-level attention [11]	74.9	75.0	74.9	-
Self Matching Networks [12]	76.3	72.5	74.4	-
A2Text-Net [13]	80.3	80.2	80.1	88.4
Our Model	**80.9** (+ 0.6 ↑)	**81.8** (+ 1.6 ↑)	**81.2** (+ 1.1 ↑)	**88.6** (+ 0.2 ↑)

Table 3: Results on Twitter dataset [9].

5. Results

We present the results of our experiments on multiple publicly available datasets in this section. Results on Twitter datasets are presented in Table 2 and Table 3. In the experiments with Ghosh and Veale [9] dataset, we do not use any additional information about the user or the context tweets. Hence, for a fair comparison, we present the results on this dataset under TTEA (Target Tweet Excluding Addressee) configuration. As evident from these tables, our multi-head self-attention model outperforms previous methods by a considerable margin. In Table 4, we present the results on the Reddit SARC 2.0 dataset which is divided into two subsets: Main and Political. In both the datasets, our proposed approach outperforms previous methods.

Apart from Twitter and Reddit data we also experimented with data from other data sources such as Political Dialogues [38] and News Headlines [41]. In Table 5, we present

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

Models	Main - Balanced		Political	
	Accuracy	F1	Accuracy	F1
Bag-of-words	63.0	64.0	59.0	60.0
CNN	65.0	66.0	62.0	63.0
CNN-SVM [20]	68.0	68.0	70.65	67.0
CUE-CNN [21]	70.0	69.0	69.0	70.0
CASCADE [22]	77.0	77.0	74.0	75.0
SARC 2.0 [40]	75.0	-	76.0	-
ELMo-BiLSTM [10]	72.0	-	78.0	-
ELMo-BiLSTM FULL [10]	76.0	76.0	72.0	72.0
Our Model	**81.0** (+ 4.0 ↑)	**81.0** (+ 4.0 ↑)	**80.0** (+ 2.0 ↑)	**80.0** (+ 5.0 ↑)

Table 4: Results on Reddit dataset SARC 2.0 and SARC 2.0 Political [40].

Models	Precision	Recall	F1	AUC
NBOW	66.0	66.0	66.0	-
Vanilla CNN	68.4	68.1	68.2	-
Vanilla LSTM	68.3	63.9	60.7	-
Attention LSTM	70.0	69.6	69.6	-
GRNN [24]	62.2	61.8	61.2	-
CNN-LSTM-DNN [28]	66.1	66.7	65.7	-
SIARN [45]	72.1	71.8	71.8	-
MIARN [45]	72.9	72.9	72.7	-
ELMo-BiLSTM [10]	74.8	74.7	74.7	-
ELMo-BiLSTM FULL [10]	76.0	76.0	76.0	-
Our Model	**77.4** (+ 1.2 ↑)	**77.2** (+ 1.4 ↑)	**77.2** (+ 1.2 ↑)	**0.834**

Table 5: Results on Sarcasm Corpus V2 Dialogues dataset [38]

Models	Precision	Recall	F1	Accuracy	AUC
Hybrid [41]	-	-	-	89.7	-
A2Text-Net [13]	86.3	86.2	86.2	-	0.937
Our Model	**0.919** (+ 5.6 ↑)	**91.8** (+ 5.6 ↑)	**91.8** (+ 5.6 ↑)	**91.6** (+ 1.9 ↑)	**97.4** (+ 3.7 ↑)

Table 6: Results on New Headlines dataset [41].

results on Sarcasm Corpus V2 Dialogues dataset and in Table 6 we present result on News Headlines dataset. In both the datasets, we see considerable improvements.

5.1. Ablation Study

Sarcasm Corpus V2 Dialogues dataset [38] is used in the following experiments.

5.1.1. **Ablation 1:**

We vary the number of self-attention layers and fix the number of heads per layer (#H = 8). From the results of this experiment presented in Table 7, we observe that as the number of self-attention layers increase (#L = 0, 1, 3, 5) the improvement in the performance of the model due to the additional layers saturate. Also, these results show that the proposed multi-head self-attention model achieves a 2% improvement over the baseline model where only a single GRU layer is used without any self-attention layers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

#L - Layers	Precision	Recall	F1
0 (GRU only)	75.6	75.6	75.6
1 Layer	76.2	76.1	76.1
3 Layers	77.4	77.2	77.2
5 Layers	77.6	77.6	77.6

Table 7: Ablation study with varying number of attention layers #L and fixed Heads **#H = 8** on the Sarcasm Corpus V2 Dialogues dataset [38].

5.1.2. **Ablation 2:**

We vary the number of heads per layer with a fixed number of self-attention layers (#L = 3). Results of this experiments is presented Table 8. We observe that the performance of the model also increases with the increase in the number of heads per self-attention layer.

#H - Heads	Precision	Recall	F1
1 Head	74.9	74.5	74.4
4 Heads	76.9	76.8	76.8
8 Heads	77.4	77.2	77.2

Table 8: Ablation study with varying number of Heads #H and fixed Layers **#L = 3** on the Sarcasm Corpus V2 Dialogues dataset [38].

5.1.3. **Ablation 3:**

To further show the strength of our proposed network architecture, we perform one another ablation, in which we train our model with different word embedding such as Glove-6B, Glove-840B, ELMO, and FastText. Results are presented in Table 9. These results show that the performance of our model is not due to the choice of word embeddings. With #H = 8 and #L = 3, the maximum possible batch size to train the model on 1 GPU with 16GB memory is 64. We set #H = 8 and #L = 3 in all our experiments for all the datasets.

Models	Embeddings	Precision	Recall	F1	AUC
MIARN[45]	-	72.9	72.9	72.7	-
ELMo-BiLSTM FULL [10]	ELMO	76.0	76.0	76.0	-
Our Model	BERT	77.4	77.2	77.2	83.4
	ELMO	76.7	76.7	76.7	80.8
	FastText	75.7	75.7	75.7	81.6
	Glove 6B	76.0	76.0	76.0	82.3
	Glove 840B	77.0	77.0	77.0	82.9

Table 9: Ablation study on various word embeddings on the Sarcasm Corpus V2 Dialogues dataset [38]

6. Model Interpretability

Attention maps from the individual heads of the self-attention layers provide the learned attention weights for each time-step in the input. In our case, each time-step is a word and we visualize the per-word attention weights for sample sentences with and without sarcasm from SARC 2.0 Main dataset. The model we used for this analysis has 5 attention layers with 8 heads per attention. Figures 2 and 3 show attention analysis [46] for sample sentences with and without sarcasm respectively. Each column in these figures corresponds to a single attention layer and attention weights between words in each head are represented using colored edges. The darkness of an edge indicates the strength of the attention weight. CLS and SEP are classifications and separator tokens from BERT. Figures

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

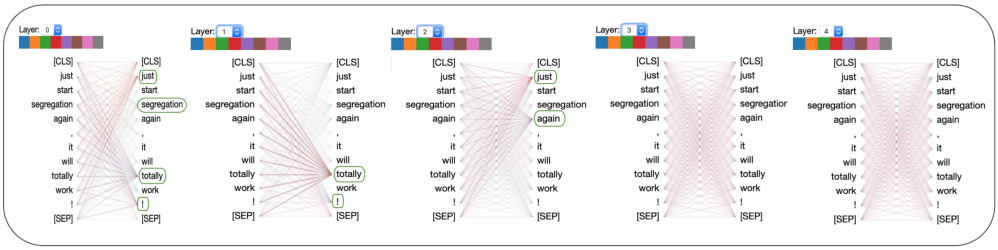

Figure 2. Attention analysis with sample sentence with sarcasm. Words providing cues for sarcasm, highlighted in green, are the words with higher attention weights. The prediction score for this sentence by our model is 0.94.

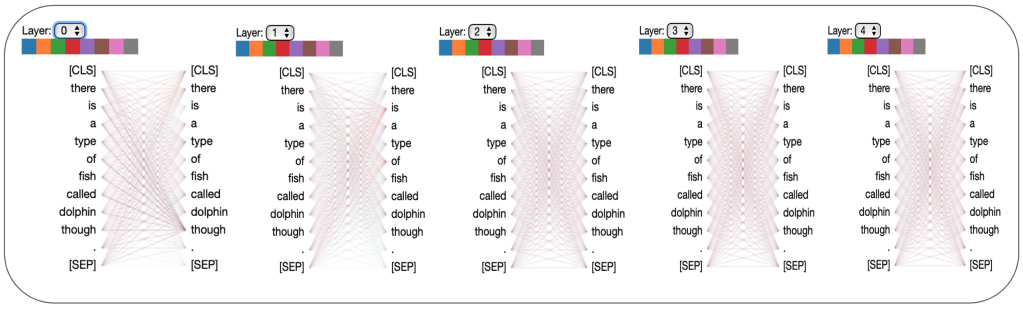

Figure 3. Attention analysis with sample sentence without sarcasm. Due to no presence of cues for sarcasm, every word in a sentence have similar attention weights. The prediction score for this sentence by our model is 0.15.

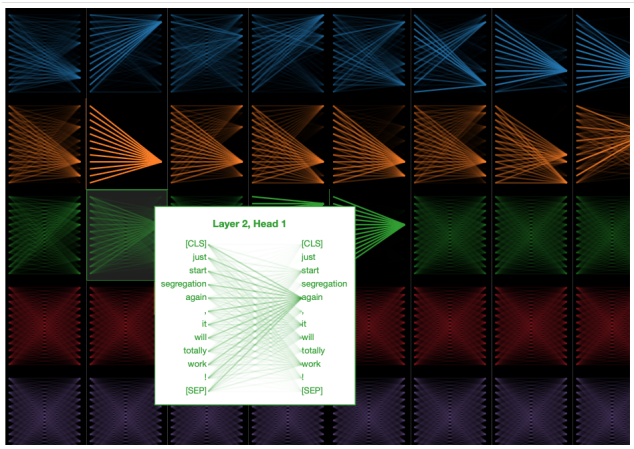

Figure 4. Attention analysis with sample sentence with sarcasm. Rows correspond to the different layers in the model and the columns correspond to the individual heads with a layer. When the input sentence contains sarcasm, we observe multiple heads, across layers attending to cue words in the input.

4 and 5 are yet another visualization which provide a birds-eye view of attention across all the heads and layers in the model. Here rows correspond to 5 attention layers and the columns correspond to 8 heads in each layer. From both the visualizations, we observe that words receiving most attention vary between different heads in each layer and also across layers.

6.1. Attention Analysis

For a sentence with sarcasm, Figure 2 shows that certain words receive more attention than others. For instance, words such as 'just', 'again', 'totally', '!', have darker edges connecting them with every other word in a sentence. These are the words in the sentence

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2021 doi:10.20944/preprints202101.0302.v1

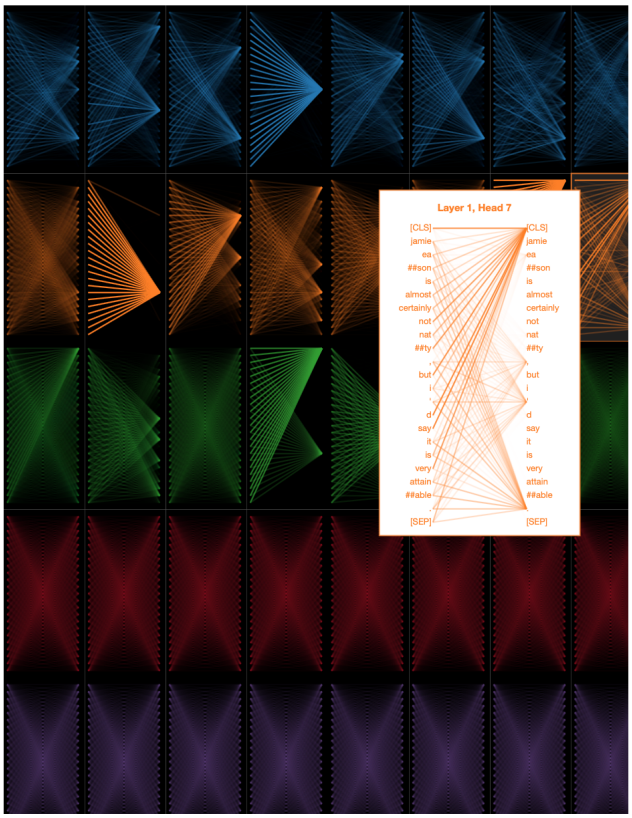

Figure 5. Attention analysis with sample sentence without sarcasm. Rows correspond to the different layers in the model and the columns correspond to the individual heads with a layer. When the input sentence contains no sarcasm, we observe that attention is distributed between multiple words in each head, across layers.

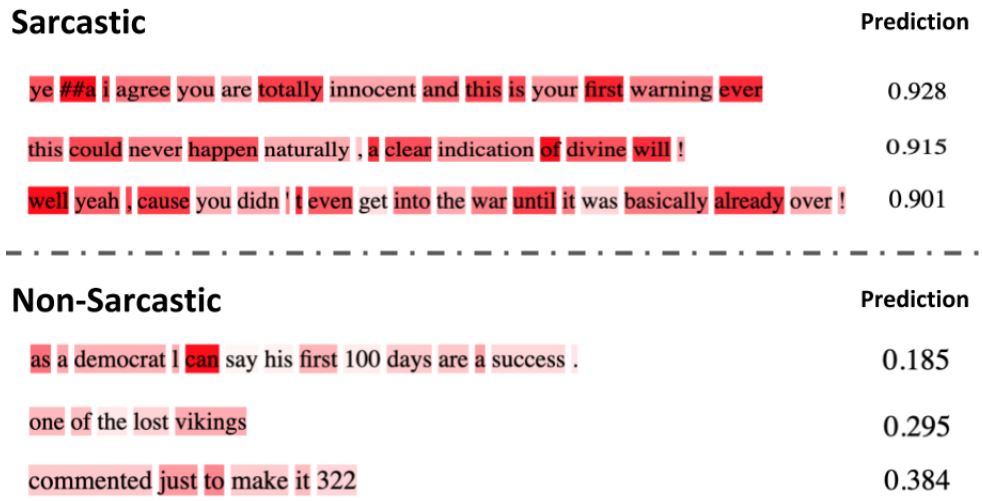

Figure 6. Visualization of the attention on individual words of sample sentences from both Sarcastic and Non-Sarcastic classes are shown in the column to the left. Probability scores predicted by our model are shown in the column to the right. High scores are predicted for sarcastic sentences and low scores for non-sarcastic sentences.

which hint at sarcasm and as expected these receive higher attention than others. Also, note that each cue word is attended by a different head in the first three layers of self-attention. In the final two layers, we observe that the attention is spread out to every word in the sentence indicating redundancy of these layers in the model. A sample sentence shown in

 doi:10.20944/preprints202101.0302.v1

Figure 3 has no sarcasm, thus no word is highlighted by any head in any layer. In Figure 6, we visualize the distribution of attention over the words in a sentence for six sample sentences. Attention weight for a word is computed by first considering the maximum attention it receives across layers and then averaging the weights across multiple-heads in the layer. Finally, the weights for a word are averaged over all the words in the sentence. Stronger the highlight for a word, the higher is the attention weight placed on it by the model while classifying the sentence. Words from the sarcastic sentences with higher weights show that the model is able to detect sarcastic cues from the sentence. For example, the words "totally", "first", "ever" from the first sentence and "even", "until", "already" from the third sentence. These are the words that exhibit sarcasm in the sentences, which the model is able to successfully identify. In all the samples which are classified as non-sarcasm, the weights for the individual words are very low in comparison to cue-words from the sarcastic sentences. The probability of sarcasm predicted by our model for each of the sentences is shown on the right and their respective scores on the left column in Figure 6. Our model is able to predict a high score for sarcastic sentences and low scores for non-sarcastic sentences.

7. Conclusion

In this work, we propose a novel multi-head self-attention based neural network architecture to detect sarcasm in a given sentence. Our proposed approach has 5 components: data pre-processing, multi-head self-attention module, gated recurrent unit module, classification, and model interpretability. Multi-head self-attention is used to highlight the parts of the sentence which provide crucial cues for sarcasm detection. GRUs aid in learning long-distance relationships among these highlighted words in the sentence. The output from this layer is passed through a fully-connected classification layer to get the final classification score. Experiments are conducted on multiple datasets from varied data sources and show significant improvement over the state-of-the-art models by all evaluation metrics. Results from ablation studies and analysis of the trained model are presented to show the importance of different components of our model. We analyze the learned attention weights to interpret our trained model and show that it can indeed identify words in the input text which provide cues for sarcasm.

References

1. Ezaiza, H.; Humayoun, S.R.; AlTarawneh, R.; Ebert, A. Person-vis: Visualizing personal social networks (ego networks). Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2016, pp. 1222–1228.
2. Akula, R.; Garibay, I. VizTract: Visualization of Complex Social Networks for Easy User Perception. *Big Data and Cognitive Computing* **2019**, *3*, 17.
3. Akula, R.; Yousefi, N.; Garibay, I. DeepFork: Supervised Prediction of Information Diffusion in GitHub. *Proceedings of the International Conference on Industrial Engineering and Operations Management* **2019**.
4. Akula, R.; Wieselthier, Z.; Martin, L.; Garibay, I. Forecasting the Success of Television Series using Machine Learning. 2019 SoutheastCon. IEEE, 2019, pp. 1–8.
5. Shamay-Tsoory, S.G.; Tomer, R.; Aharon-Peretz, J. The neuroanatomical basis of understanding sarcasm and its relationship to social cognition. *Neuropsychology* **2005**, p. 288.
6. Skalicky, S.; Crossley, S. Linguistic Features of Sarcasm and Metaphor Production Quality. Proceedings of the Workshop on Figurative Language Processing, 2018, pp. 7–16.
7. Kreuz, R.J.; Caucci, G.M. Lexical influences on the perception of sarcasm. Proceedings of the Workshop on computational approaches to Figurative Language. Association for Computational Linguistics, 2007, pp. 1–4.
8. Joshi, A.; Sharma, V.; Bhattacharyya, P. Harnessing context incongruity for sarcasm detection. Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, 2015, pp. 757–762.
9. Ghosh, A.; Veale, T. Magnets for sarcasm: making sarcasm detection timely, contextual and very personal. Proceedings of the 2017 Conference on EMNLP, 2017, pp. 482–491.
10. Ilic, S.; Marrese-Taylor, E.; Balazs, J.; Matsuo, Y. Deep contextualized word representations for detecting sarcasm and irony. Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2018, pp. 2–7.
11. Ghosh, D.; Fabbri, A.R.; Muresan, S. Sarcasm analysis using conversation context. *Computational Linguistics* **2018**, pp. 755–792.

12. Xiong, T.; Zhang, P.; Zhu, H.; Yang, Y. Sarcasm Detection with Self-matching Networks and Low-rank Bilinear Pooling. The World Wide Web Conference, 2019, pp. 2115–2124.
13. Liu, L.; Priestley, J.L.; Zhou, Y.; Ray, H.E.; Han, M. A2text-net: A novel deep neural network for sarcasm detection. 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI). IEEE, 2019, pp. 118–126.
14. Carvalho, P.; Sarmento, L.; Silva, M.J.; De Oliveira, E. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-. Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, 2009, pp. 53–56.
15. González-Ibánez, R.; Muresan, S.; Wacholder, N. Identifying sarcasm in Twitter: a closer look. Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies: Short Papers-Volume 2, 2011, pp. 581–586.
16. Tsur, O.; Davidov, D.; Rappoport, A. ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. Fourth International AAAI Conference on Weblogs and Social Media, 2010.
17. Davidov, D.; Tsur, O.; Rappoport, A. Semi-supervised recognition of sarcastic sentences in twitter and amazon. Proceedings of the fourteenth conference on computational natural language learning. Association for Computational Linguistics, 2010, pp. 107–116.
18. Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; Huang, R. Sarcasm as contrast between a positive sentiment and negative situation. Proceedings of the 2013 Conference on EMNLP, 2013, pp. 704–714.
19. Wallace, B.C.; Charniak, E.; others. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, 2015, pp. 1035–1044.
20. Poria, S.; Cambria, E.; Hazarika, D.; Vij, P. A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1601–1612.
21. Amir, S.; Wallace, B.C.; Lyu, H.; Carvalho, P.; Silva, M.J. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 167–177.
22. Hazarika, D.; Poria, S.; Gorantla, S.; Cambria, E.; Zimmermann, R.; Mihalcea, R. CASCADE: Contextual Sarcasm Detection in Online Discussion Forums. Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1837–1848.
23. Rajadesingan, A.; Zafarani, R.; Liu, H. Sarcasm detection on twitter: A behavioral modeling approach. Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 2015, pp. 97–106.
24. Zhang, M.; Zhang, Y.; Fu, G. Tweet sarcasm detection using deep neural network. Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2449–2460.
25. Ptáček, T.; Habernal, I.; Hong, J. Sarcasm detection on czech and english twitter. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 213–223.
26. Wang, Z.; Wu, Z.; Wang, R.; Ren, Y. Twitter sarcasm detection exploiting a context-based model. international conference on web information systems engineering. Springer, 2015, pp. 77–91.
27. Joshi, A.; Tripathi, V.; Bhattacharyya, P.; Carman, M. Harnessing sequence labeling for sarcasm detection in dialogue from tv series 'friends'. Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 146–155.
28. Ghosh, A.; Veale, T. Fracking sarcasm using neural network. Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, 2016, pp. 161–169.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in neural information processing systems, 2017, pp. 5998–6008.
30. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* **2013**.
31. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2013, pp. 3111–3119.
32. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. Proceedings of the 2014 conference on EMNLP, 2014, pp. 1532–1543.
33. Joulin, A.; Grave, É.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the ACL, 2017, pp. 427–431.
34. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. Proceedings of NAACL-HLT, 2018, pp. 2227–2237.
35. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of NAACL: Human Language Technologies, 2019, pp. 4171–4186.
36. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.
37. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.
38. Oraby, S.; Harrison, V.; Reed, L.; Hernandez, E.; Riloff, E.; Walker, M. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, pp. 31–41.
39. Walker, M.A.; Tree, J.E.F.; Anand, P.; Abbott, R.; King, J. A Corpus for Research on Deliberation and Debate. LREC. Istanbul, 2012, pp. 812–817.

40. Khodak, M.; Saunshi, N.; Vodrahalli, K. A Large Self-Annotated Corpus for Sarcasm. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.
41. Misra, R.; Arora, P. Sarcasm Detection using Hybrid Neural Network. *arXiv preprint arXiv:1908.07414* **2019**.
42. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; 2019.
43. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Brew, J. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* **2019**.
44. Farías, D.I.H.; Patti, V.; Rosso, P. Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology (TOIT)* **2016**, pp. 1–24.
45. Tay, Y.; Luu, A.T.; Hui, S.C.; Su, J. Reasoning with Sarcasm by Reading In-Between. Proceedings of the 56th Annual Meeting of the ACL, 2018, pp. 1010–1020.
46. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What Does BERT Look at? An Analysis of BERT's Attention. Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2019, pp. 276–286.