Explainable Detection of Sarcasm in Social Media

Ramya Akula
University of Central Florida
USA
ramya.akula@knights.ucf.edu

Ivan Garibay
University of Central Florida
USA
igaribay@ucf.edu

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 makes sarcasm detection very difficult. In this work, we focus on detecting sarcasm in textual conversations, written in English, 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. We show the effectiveness and interpretability of our approach by achieving state-of-the-art results on datasets from social networking platforms, online discussion forum and political dialogues.

1 Introduction
Sarcasm is a rhetorical way of expressing dislike or negative emotions using different language constructs, such as exaggeration or ridicule. It is an assortment of mockery and false politeness to intensify hostility without explicitly doing so. In face-to-face conversation, facial expressions, gestures, and tone of the speaker provide cues that help in identifying sarcasm. 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, discussion forums, and e-commerce websites has become crucial for opinion mining, sentiment analysis, and identifying cyberbullies, online trolls. Thus, developing computational models for automatic detection of sarcasm gathered pace in recent times with multiple studies and collection of new datasets (Ghosh and Veale, 2017; Misra and Arora, 2019; Khodak et al., 2018).

Earlier works on sarcasm detection on texts use lexical (content) and pragmatic (context) cues (Kreuz and Caucci, 2007) such as interjections, punctuation, and sentimental shifts, which are major indicators of sarcasm (Joshi et al., 2015). 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 (Ghosh and Veale, 2017; Ilic et al., 2018; Ghosh et al., 2018; Xiong et al., 2019; Liu et al., 2019), 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. 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: a) Propose an interpretable model for sarcasm detection using self-attention. b) Achieve state-of-the-art results on diverse datasets and exhibit the effectiveness of our model with extensive experimentation and ablation studies. c) Exhibit the interpretability of our model by analyzing the learned attention maps.

2 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. We employ the pre-trained language model, BERT (Devlin et al., 2019), to extract word embeddings. We use these word embeddings which capture global context as we believe context is essential for detecting sarcasm. 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.

Multi-Head Self-Attention Given a sentence S, we apply a standard tokenizer and use pre-trained models to obtain D dimensional embeddings for individual words in the sentence. These embeddings $S = \{e_1, e_2, \ldots, e_n\}$
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.

\[ e_N, S \in \mathbb{R}^{N \times D} \] 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 (Vaswani et al., 2017) 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 \( QK^T \). 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 \((8L)\) with multiple heads \((8H)\) in each layer.

**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.

**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 \( y \in [0,1] \), where \( y \in \{0,1\} \) is the binary label i.e., 1:Sarcasm and 0:No-sarcasm.

**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 (Zhou et al., 2016; Selvaraju et al., 2017). 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.

### 3 Experiments

We implement our model in PyTorch (Paszke et al., 2019), a deep-learning framework in Python. To tokenize and extract word embeddings for the input text, we use publicly available resources (Wolf et al., 2019). Specifically, we use tokenizer and pre-trained weights from the “bert-base-uncased” model to convert words to tokens and then convert tokens to word embeddings. The embeddings for the words in the input text are passed through a series of multi-head self-attention layers \((8L)\) with multiple heads \((8H)\) 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 memory for all our experiments. We set \#H = 8 \#L = 3 \#B = 64 for all our experiments for all the datasets. Details of these datasets, including the sample counts in train/test splits and the data source, are presented in Table 1.

**Evaluation** We pose Sarcasm Detection as a classification problem, and use Precision, Recall, F1-Score,
5 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 the SARC 2.0 Main dataset. The model we used for this analysis has 5 attention layers with 8 heads per attention.

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 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. 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. The 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 can 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 can 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. Our model can predict a high score for sarcastic sentences and low scores for non-sarcasm sentences.

6 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. Expere-
imements are conducted on two datasets from different 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.

Table 2: Results on Twitter dataset (Riloff et al., 2013).

| Models                                  | Precision | Recall | F1    | AUC |
|-----------------------------------------|-----------|--------|-------|-----|
| Fracking Sarcasm (Ghosh and Veale, 2016)| 88.3      | 87.9   | 88.1  | -   |
| GRNN (Zhang et al., 2016)               | 66.3      | 64.7   | 65.4  | -   |
| ELMo-BiLSTM (Ilic et al., 2018)         | 75.9      | 75.0   | 75.9  | -   |
| ELMo-BiLSTM FULL (Ilic et al., 2018)    | 77.8      | 73.5   | 75.3  | -   |
| ELMo-BiLSTM AUG (Ilic et al., 2018)     | 68.4      | 70.8   | 69.4  | -   |
| A2Text-Net (Liu et al., 2019)           | 91.7      | 91.0   | 90.0  | 97.0|
| **Our Model**                           | **97.9**  | **99.6**| **98.7**| **99.6**|

Table 3: Results on Sarcasm Corpus V2 Dialogues dataset (Oraby et al., 2016).

| Models                                  | Main Accuracy | Main F1 | Political Accuracy | Political F1 |
|-----------------------------------------|---------------|---------|-------------------|--------------|
| CASCADE (Hazarika et al., 2018)         | 77.0          | 77.0    | 74.0              | 75.0         |
| SARC 2.0 (Khodak et al., 2018)          | 75.0          | -       | 76.0              | -            |
| ELMo-BiLSTM (Ilic et al., 2018)         | 72.0          | -       | 78.0              | -            |
| ELMo-BiLSTM FULL (Ilic et al., 2018)    | 76.0          | 76.0    | 72.0              | 72.0         |
| **Our Model**                           | **81.0**      | **81.0**| **80.0**          | **80.0**     |

Table 4: Results on Reddit dataset SARC 2.0 and SARC 2.0 Political (Khodak et al., 2018).

| Models                                  | Main Precision | Main Recall | Main F1 | Political Precision | Political Recall | Political F1 |
|-----------------------------------------|---------------|-------------|---------|---------------------|------------------|--------------|
| CASCADE (w/o personality features)      | 68.0          | 66.0        | 68.0    | 70.0                |                  |              |
| **Our Model (w/o personality features)**| **70.0**      | **70.0**    | **71.0**| **72.0**            |                  |              |

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.
Table 5: Ablation study with varying number of attention layers $\#L$ and fixed Heads $\#H = 8$ on the Sarcasm Corpus V2 Dialogues dataset (Oraby et al., 2016).

| $\#L$ - 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 6: Ablation study with varying number of Heads $\#H$ and fixed Layers $\#L = 3$ on the Sarcasm Corpus V2 Dialogues dataset (Oraby et al., 2016).

| $\#H$ - 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  |

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, pages 757–762.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Roger J Kreuz and Gina M Caucci. 2007. Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on computational approaches to Figurative Language, pages 1–4. Association for Computational Linguistics.

Liyuan Liu, Jennifer Lewis Priestley, Yiyun Zhou, Herman E Ray, and Meng Han. 2019. A2text-net: A novel deep neural network for sarcasm detection. In 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI), pages 118–126. IEEE.

Rishabh Misra and Prahal Arora. 2019. Sarcasm detection using hybrid neural network. arXiv preprint arXiv:1908.07414.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 31–41.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andrej Karpathy, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32.

Ellen Riloff, Ashgeul Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruifang Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In Proceedings of the 2013 Conference on EMNLP, pages 704–714.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626.

Yi Tay, Anh Tuan Lau, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. In Proceedings of the 56th Annual Meeting of the ACL, pages 1010–1020.
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. ArXiv.

Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. 2019. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In The World Wide Web Conference, pages 2115–2124.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers, pages 2449–2460.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2921–2929.