
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
 Volume 24, Issues 4, pp. 787–796, DOI 10.12694/scpe.v24i4.2166

EXPLAINING SARCASM OF TWEETS USING ATTENTION MECHANISM

R L KEERTHANA*, AWADHESH KUMAR SINGH*, POONAM SAINI†, AND DIKSHA MALHOTRA†

Abstract. Emotion identification from text can help boost the effectiveness of sentiment analysis models. Sarcasm is one of the more difficult emotions to detect, particularly in textual data. Even though several models for detecting sarcasm have been presented, their performance falls way short of that of other emotion detection models. As a result, few strategies have been introduced in the paper that helped to enhance the performance of sarcasm detection models. To compare performance, the model was tested using the TweetEval benchmark dataset. On the TweetEval benchmark, the technique proposed in this paper has established a new state-of-the-art. Besides the low performance, interpretability of existing sarcasm detection models are lacking compared to other emotion detection models like hate speech and anger. Therefore, an attention-based interpretability technique has been proposed in this paper that interprets the token importance for a certain decision of sarcasm detection model. The results of the interpretability technique aid in our comprehension of the contextual embeddings of the input tokens that the model has paid the greatest attention to while making a particular decision which outperforms existing transformer-based interpretability techniques, particularly in terms of visualisations.

Key words: sarcasm detection, interpretability, task-adaptive pre-training, attention, and heatmap.

1. Introduction. Sentiment analysis is one of the famous domains where many state-of-the-art NLP techniques [17] have been employed in various real-world scenarios such as product recommendation, feedback analysis, social monitoring, and so on. However, the hidden emotions contained within the input data impede the performance [18] of the state-of-the-art sentiment analysis models. Irony is one such emotion, described as the expression of one's meaning through the use of language that generally conveys the opposite, often for hilarious or dramatic effect. Due to the above reasons, deep learning models, especially those optimised for irony detection, are in high demand. Also, pre-trained models [19], specifically trained for a specific area or task, are being widely used for the detection of such sarcasm. While there is an abundance of pre-trained models available for social media data, the intricate nature of sarcastic comments often leads to reduced performance in pre-trained models specifically designed for irony detection.

Ironic expressions convey an opposite meaning, often subtly, especially in text. The irony can be so subtle, especially in the text, that identifying it can be difficult, especially if the model doesn't understand the context. Irony detection is a difficult task for language models since we, as humans, have difficulties understanding sarcasm in context at times. Detecting irony is especially challenging when dealing with unimodal text data, as textual incongruity may or may not imply irony. Therefore, it's crucial not only to develop an effective model for identifying irony but also to grasp the reasoning behind the algorithm's decisions. Model interpretability encompasses methods that aid in deducing the model's outcomes. These interpretations help validate model predictions and understand the token importance, as assessed by our model and the baseline. Consequently, they enhance our comprehension of how the model performs in this challenging task and conceptualising token significance of transformer based models.

Since transformer-based [21] models have been employed to vectorize the input text, the model's attention retains a great deal of context about the input text. The attention scores of the transformer models are related to contextual information; hence, these scores can be employed to analyse the models. Even though there have been disputes that attention cannot be utilised for explanations[20], there are many works that have used attention for interpretability. Explainability of a model is concerned with understanding how the model works to reach a specific conclusion, whereas interpretability is used to assist an observer in understanding the model's decision. Works like [3], [4], [5] have employed model attention for interpretive purposes. The weights

*Department of Computer Engineering, NIT Kurukshetra, Kurukshetra, India.
†Department of Computer Science and Engineering, punjab engineering college (deemed to be college), Chandigarh, India.

generated by the transformers are generally built on attention scores[21]. The attention score at any particular layer indicates the amount of attention received by the input tokens in that layer. As a result, it can be argued that attention can be employed to provide the model interpretability. The attention scores as it is can be used for the sake of interpretability. This paper discusses how raw attention does not provide interpretability for sarcasm detection and models and also proposes a computation with attention scores which aids in better visualisations.

The paper has been broken into the sections listed below. Section 2 discusses related work on sarcasm detection and the interpretability of transformer-based models. Section 3 describes the methods that were adopted to improve the performance of sarcasm detection algorithms. Section 3 also discussed the interpretability technique proposed to visualise the token importance of sarcasm detection models. Section 4 goes over the implementation and results. The final section concluded the paper.

2. Related Work.

2.1. Sarcasm Detection. Information from a variety of sources, including a Twitter dataset, news headline sarcasm, and a Reddit dataset were gathered by [10]for training purposes. This paper tested their dataset on linear regression classifiers utilising a variety of data representations such as bi-grams, BoW, phrase embeddings,and so on. They came to the conclusion that bi-grams and BoW representations outperformed sentence embeddings. Later, [8] used SVM, random forest, and KNN algorithms with k-fold validation to develop a machine learning system for detecting sarcasm. The investigations revealed that Random forests produced the highest F1-score of any of the three techniques on the airline sentiment [18] dataset. And [14]employed BERT_{large} to calculate word embeddings, which were then used to deter-mine if the words were sarcastic or not. The model was evaluated using multipledatasets, and the BERT Classifier stood out in all of them.

The article [9] used Extreme Gradient Boosting to detect sarcasm in an image. The dataset provided the context of the speech, which was also used as a part of the training. The embeddings were acquired using the TF-IDF approach and then processed using the XGBoost ensembling technique. On the test dataset, the model received an F1-score of 92.4.

Attention-based GRUs [12] were also employed to predict sarcasm. To learn the context of the input data, GRUs were used. The contextual embeddings were fed into a simple classification layer, which categorises the text into one of two groups. On the SARC[10] dataset, they obtained an F1-score of 77.2.

Incongruity aware attention network (IWAN) [13] was proposed that combines multimodal data such as textual, visual, and audio information and uses the incongruity between the features to detect sarcasm. To extract characteristics from various data modes, various models were utilised. These characteristics were then scored using their scoring mechanisms. To classify the input, these scores were fed through a softmax classifier. On the MUStARD dataset, the model received an F1-score of 74.5.

Simple classifiers like SVM classifier[11] was also employed to detect sarcasm in a collection of news headlines [17]. They used the TF-IDF concept to obtain word embeddings. Using SVM for data classification, the model earned an F1-score of 94.8 on the test dataset.

2.2. Interpretability. In [12], the GRUs were provided with a layer of attentions, and the weights of these attentions were used for interpretability. The attributions of the input tokens were determined using integrated gradients to help interpret the transformers in general. These attributions indicate the positive, neutral, and negative influencing tokens of the prediction. SHAP and LIME were used in [9] to explain the model predictions. Both methods' output scores were utilised to visualise the importance of each input token in the model's choice.

These methods were not designed especially for the Transformers. As a result, they are unconcerned with attention layers with additional context information. As a result, much research has been conducted in an attempt to interpret the model using attention weights.

BertViz [15] is a visualisation tool that allows us to depict the attention of tokens in a text from several perspectives, such as neuron view, head view, and model view. These views aid in visualising the flow of input token attentions through the model.

Later, the paper [7] suggested aggregating attention weights from previous layers to determine the attention-based token relevance at a specific layer. Under the assumption that attentions can be merged linearly, the

paper [6] proposed two ways for aggregating the attention values gained by each token across the network from the weights in each layer. These methods offer strategies for calculating token importance at higher levels. It is crucial to emphasise that the proposed computations are just for illustrative purposes.

As can be seen, very few interpretability techniques were used for sarcasm detection, and the majority of the work did not use attention scores to interpret sarcasm detection models. For the sarcasm detection model in this paper, an attention-based interpretability technique has been implemented. In addition to proposing the method, previously proposed attention-based interpretable techniques have also been implemented, which did not work for sarcasm detection models. The scope of attention-based interpretable techniques is discussed in this paper.

3. Methodology.

3.1. Workflow. Works in [1] have illustrated the performance improvement of downstream tasks by pre-training the language models. Pre Training techniques can be of two categories based on the kind of data used for pre training. The first type is domain-adaptive pre-training which assists in refining the sentence embeddings and model vocabulary for the domain for which it was trained. Domain specific pre training enhances performance, especially when the target domain language differs greatly from the source domain vocabulary. As the RoBERTa$_{base}$'s pre-training corpus is far too distinct from social media data, it was pre-trained with a large amount of unlabeled data from the social media domain using self-supervised techniques. The RoBERTa$_{base}$ was trained with 58 million tweets using Masked Language Modelling and this pretrained model is popularly called Twitter-RoBERTa [2].

RoBERTa$_{base}$ [23] is an optimised version of the BERT. It was trained on significantly larger datasets than BERT. These datasets were gathered from a variety of domains, including Book-Corpus [24], News [25], Web Text [26], and Stories [27], totaling 160 GB of text. BERT was trained using static Masked Language Modelling and Next Sentence Prediction, whereas RoBERTa$_{base}$ was trained using dynamic Masked Language Modelling only. As a result, RoBERTa is simply the optimised version of the BERT.

The Twitter-RoBERTa [2] model was directly fine tuned on the TweetEVAL-Irony detection dataset, which yielded an F-score of 65.1 on the ironic class. And the top performing model indicated in [2] has an F-score of 70.5 in the ironic class. This model was pre-trained on RoBERTa using the same Twitter data, but from scratch, and then fine-tuned for irony detection. Even if there is an explanation for this performance disparity, that irony observed in social media data differs from irony seen in conventional train text tweets—it can be argued that task-specific pre-training is critical, particularly for challenging tasks like irony detection. Task-adaptive pre-training is the process of pre-training a model with data for a certain task.

The RoBERTa$_{base}$[23] model was initially pre-trained on a domain-specific Twitter dataset [2] using masked language modelling. Subsequently, it underwent further training on the task-specific SARC[10] dataset, specifically designed for sarcasm detection. This continuous pre-training approach played a pivotal role in establishing the model as a state-of-the-art solution for sarcasm detection. The TweetEVAL-Irony detection dataset [2] served both as the basis for fine-tuning the pre-trained model and as the evaluation benchmark. Fig 3.1 depicts the entire process in detail. The observed results were discussed in Section 4. This fine-tuned model's attention has been drawn to interpreting the model's decisions.

3.2. Model Interpretation. Like any deep learning models, Transformer architectures are also black box in nature. Techniques like SHAP, LIME, and integrated gradients [28] were applied to the transformers for interpretability. Even though they provided the explanations, these techniques were not explicitly proposed for transformer architecture, and therefore they do not use the attention weights, which hold a lot of context information. Therefore, in this experiment attention scores were used for interpretability.

In the proposed technique for interpreting transformer-based sarcasm detection models, the attention scores generated by the transformer were leveraged to visualise the importance of individual tokens. It is important to note that BERT-based models consist of multiple encoder layers (denoted as 'n'), each of which processes the input text by vectorizing it. During the vectorization process, the model appends special tokens, such as <CLS> (representing the beginning of a sentence) and <SEP> (representing the end of a sentence), to the input tokens. By examining the attention scores computed by the transformer, the insights into the relative importance and attention assigned to different tokens within the input sequence can be gained. This

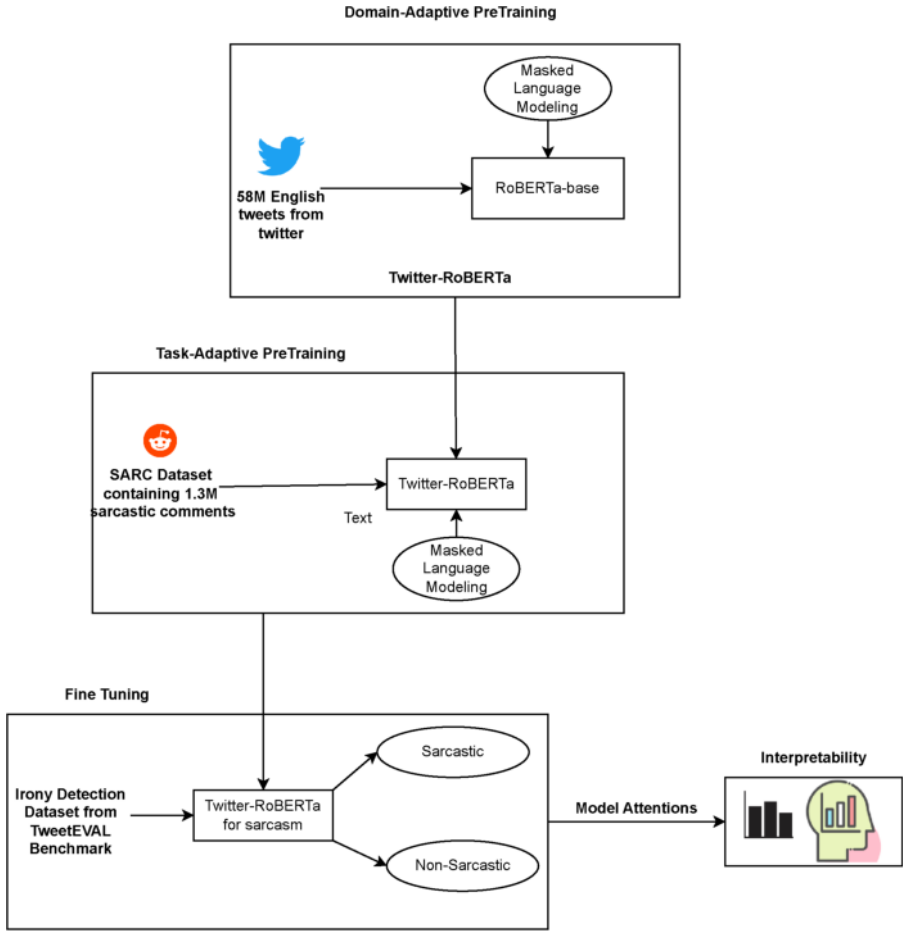

Fig. 3.1: Visual Representation of the whole framework

visualisation technique provides a valuable tool for understanding how the model processes and interprets the textual information, particularly in the context of sarcasm detection. To ensure consistent input representation, every text undergoes padding with a special <CLS> token at the beginning and a <SEP> token at the end. In cases where the input text comprises multiple sentences, a <SEP> token is inserted after the end of each sentence. Following the vectorization process, if an input text consists of t tokens (including the special tokens), the attention scores computed at each layer of the transformer yield a t x t vector. Each t-sized vector captures the degree to which a specific token attends to all other tokens within the sequence.

To gain insights into the interpretability of transformer-based sarcasm detection models, the focus was on understanding the role of the <CLS> token and its attention scores. The <CLS> token's attention scores determine the degree of importance it assigns to all other input tokens. Typically, these attention scores are utilised for classification purposes. However, for the purpose of interpretation, we specifically visualised the attention scores of the <CLS> token in relation to each individual token.

A transformer's encoder layer consists of multiple attention heads[21], each generating a t x t attention vector. For interpretability, the attention scores of the <CLS> token in the final layer was examined, which contains a significant amount of contextual information. To compute the attention scores of the <CLS> token in the last layer, the attention scores of the <CLS> token from each attention head was linearly multiplied in the final layer. This operation can be seen as treating the attention scores as context embeddings, which are subsequently processed for classification purposes.

Table 3.1: Pseudocode for TAPT

Algorithm:
TaskAdaptivePre-training with Domain Adaptive Pre-training
Input:
1.Pre-trained Twitter-RoBERTa model (domain-specific)
2.SARC dataset (task-specific)
3.TweetEval Irony Benchmark dataset (evaluation)
Procedure:
1.Domain-Specific Pretraining:
Initialise the model using the pre-trained Twitter-RoBERTa weights.
2. Task-Specific Pretraining with SARC:
Iterate the model through each data sample of SARC and train the model using Masked Language Modelling.
3. Fine-Tuning and Evaluation on TweetEval Irony Benchmark:
Fine-tune the pre-trained model further and evaluate the model's performance using the TweetEval Irony Benchmark dataset.
Output:
A state-of-the-art model for sarcasm detection.

To illustrate this concept, the transformer architecture must be considered as a computation graph. Within this graph, a linear multiplication of the weights associated with the <CLS> token from all attention heads in the last layer was performed. By applying this approach, the attention scores of the <CLS> token in the final layer are obtained, representing its relevance to the overall context of the input text. By leveraging these attention scores, a deeper understanding of how the model processes is provided and assigns importance to the <CLS> token, which serves as a key element for classification in sarcasm detection tasks.

3.3. Pseudo-code. This subsection provides a pseudocode for the whole framework. This helps the readers to easily understand the whole pipeline. Table 3.1 and 3.2 are the pseudocode for the pre-training method and the attention based interpretation respectively.

4. Results and Discussion.

4.1. Datasets. This experiment employed two publicly available datasets. These data were obtained from the Twitter API and labelled for sarcasm detection. The description is as follows: The Self-Annotated Reddit Corpus (SARC) [10] is a balanced dataset including approximately 90k comments retrieved from the Reddit API, with 50% of the comments being sarcastic and the other 50% non-sarcastic. This dataset is one of the largest for the challenge of sarcasm detection. The authors of the comments contained in the dataset are also responsible for labelling the dataset.

For fine tuning purposes, TweetEVAL-Irony [2] detection dataset has been used which is also a balanced dataset that contains around 4K user-generated tweets that are classified as ironic or non-ironic. The Tweet-EVAL Benchmark dataset, which has been suggested for use in a variety of tweet classification applications, includes this dataset. This benchmark has been used to assess models created for tasks involving Twitter data.

4.2. Data Preprocessing. For both the SARC and TweetEVAL-Irony datasets, the same preprocessing has been employed. The following steps are conducted on both datasets as part of the preprocessing. To eliminate noise from the input text, preprocessing of the data is done. Since all the data samples are consistent after the noise has been removed, the model can be trained more quickly and effectively. The methods used in this research have focused particularly on eliminating noise from the data. All of the words have been made lowercase. The user tags that were not useful for learning from the tweets were excluded. Any links or URLs referenced in the tweets were also removed. The punctuation marks have also been dropped. All of the emojis were converted to text using the Python Emoji Library.

Table 3.2: Pseudocode for attention based model interpretation

Algorithm:
Attention based model interpretation
Input:
Input text for which we want the model's interpretation and attention scores for the input tokens computed across the layers. **Procedure:**
1. $tokens = TokenizeTwitter - RoBERTa(input\_text)$.
2. $attention\_scores = RetrieveAttentionScores(final\_layer)$.
3. $heatmap = InitializeEmptyMatrix(SIZE : (length(tokens) + 2) \times (length(tokens) + 2))$
4.$cls\_attention\_scores = attention\_scores[:, 0, :]$
Slice the attention score matrix to get the attention scores of <CLS> tokens of all the attention heads.
5. For each *attention_head* in $final\_layer$:
   a. cls_scores = cls_attention_scores[attention_head, :]
Extract the attention scores of the <CLS> token from that attention head.
   b. multiplied_scores = cls_scores ∗ attention_scores[attention_head, :, :]
Multiply the attention scores of the <CLS> token by the corresponding attention scores of all other tokens.
   c. heatmap += multiplied_scores
Add the resulting attention scores to the heatmap matrix.
6. heatmap /= number_of_attention_heads
Divide each entry in the heatmap matrix by the number of attention heads to compute the average attention scores.
7. Visualize(heatmap)
Visualize the heatmap matrix, highlighting the token importance and attention distribution.
8. Return heatmap.
Output:
A heatmap displaying the importance of each token in the input text.

4.3. Evaluation Metrics. The F-Score of the irony class has been chosen to evaluate the model, as it is easy to compare our model's performance to the baseline model. The F1 score is a measurement that combines recall and precision. It is calculated by taking the harmonic mean of precision and recall [22].

4.4. Baseline. The twitter-RoBERTa model proposed in [2], which was pre-trained on 58 million tweets using Masked Language Modelling, is considered the baseline. The RoBERTa model was directly pre-trained using task adaptive pre training using SARC is also an other baseline. This model was fine tuned for the irony dataset and is publicly available on the hugging face API.

4.5. Task-Adaptive Pre-training. The hugging face library was used to import the base Twitter-RoBERTa model checkpoint that was used for pre-training. This model was trained using Masked Language Modeling after being fed the SARC dataset. We pre-trained for 25 epochs with a training batch size of 64 by dynamically masking 15% of the input tokens. The final checkpoint model has been finetuned further. The RoBERTa$_{base}$ model was also checkpointed from the hugging face library which was also pre-trained using SARC dataset. This model was also pre-trained with the same parameters as mentioned above. This pre-trained model was considered as the second baseline.

4.6. Fine tuning. The pre-trained models obtained from the previous section were fine-tuned for 100 epochs using early stopping with validation loss and a patience of 10. The training was terminated, and the best model was achieved after 61 epochs for the first model and 76 epochs for the second model. For training and validation, the batch size was set to 12.

Table 4.1: Comparision of the F1-scores of our model with previous Benchmarks

Model	F-Score
Twitter-RoBERTa fine tuned for TweetEVAL-irony dataset(BASELINE1)	65.1
RoBERTa pre-trained for twitter data from scratch and fine tuned for TweetEVAL-irony dataset(State-of-the-art)	70.5
RoBERTa pre-trained from scratch with SARC	71.83
Twitter-RoBERTa pre-trained with SARC(our model)	73.56

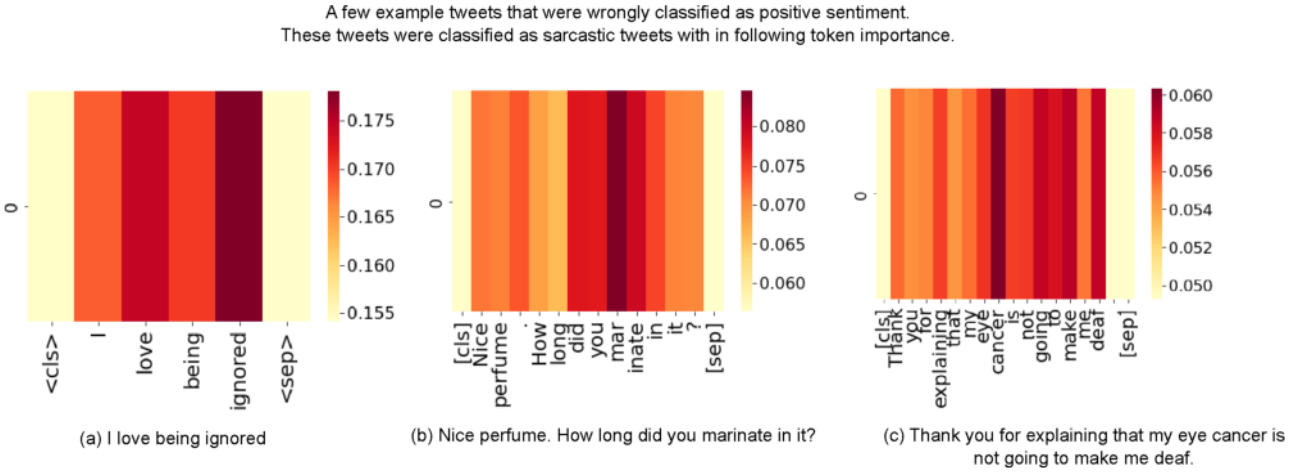

Fig. 4.1: Heat map visualizations of the obtained from the proposed attention based mechanism

4.7. Interpretation. The attention scores of the last layer's <CLS> token (Layer 12) have been utilized to compute the combined attention of tokens in the last layer. All of the layers of the transformer's tokens attend to one another. Contextual embeddings are generally defined as the attention ratings of the <CLS> token to all other tokens. All of the input tokens' relevance is shown by these scores. These attentions so illustrate the significance of each token for model prediction.

4.8. Discussion. In the sarcastic class, our fine-tuned model achieved an F-score of 73.56. In comparison, the baseline Twitter-RoBERTa model scored 65.1, while the state-of-the-art model [2] for the same dataset reached 70.5 (see Table 4.1 for details). When we trained RoBERTa specifically for sarcasm detection using the SARC dataset, it yielded an F-score of 71.83. These results highlight the need for task-adaptive pretraining to enhance performance. Additionally, including domain-specific pretraining[2] alongside task-specific pretraining boosted the model's performance by 8 points, underscoring the importance of continuous pretraining.

A few examples have been chosen where the sentiment was incorrectly identified due to the presence of sarcasm and checked whether or not the tweets were classified as sarcastic. Aside from that, heatmaps have been generated for the same examples using the interpretability method explained in Section 3.2. The attention scores were computed for the last layer and the attention scores of the <CLS> were projected on a heatmap which indicates token importance for each token and helps in interpreting the weight of every token in the model's decision.

From the example (a) in Fig. 4.1, the word *ignored* receives the highest attention score, followed by the word *love*. This interpretation clarifies the prevalence of textual incongruity in sarcastic comments. The sarcasm in example (b) from Fig. 4.1 is fairly subtle. The concealed negative context is properly captured by the model, as evidenced by the heatmap. Similarly, for example (c) from Fig. 4.1, subtle negative context from tokens *cancer, deaf* are attended the most by the <CLS> token. The attentions of the transformer-based models that hold the context can thus be used to interpret the model's predictions.

In addition to visualising token importance with the proposed attention method, a model view of raw

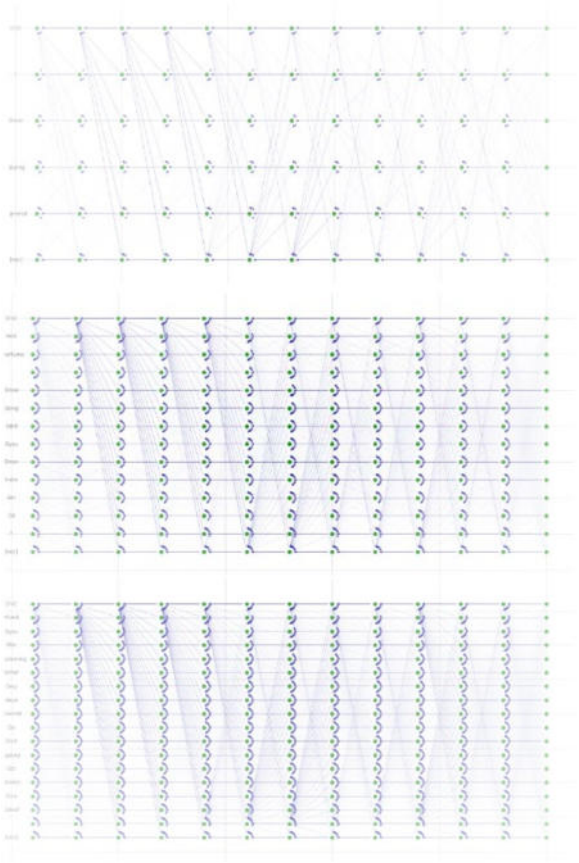

Fig. 4.2: Graph view of raw attention scores for the chosen examples

attention has been generated in Fig 4.2. The raw attention scores across the layers are plotted in a graph manner to understand the flow of attention weights between layers. The graph consists of edges that illustrate the flow of weights which represents the tokens that a particular token attends. In Fig 4.2, it is observed that the model assigns greater importance to the <CLS< and <SEP]< tokens in the initial layers, while the attention weights among the other tokens remain relatively uniform. These findings indicate that existing attention-based explainability methods, as previously proposed, do not effectively capture the intricacies of sarcasm detection.

4.9. Analysis. In the previous subsection, the visualisation of raw attentions were presented in Fig 4.2, which revealed a significant challenge in the deeper layers of the model. In these layers, every token appeared to attend to every other token with almost identical attention scores. This phenomenon resulted in a loss of token identifiability, making it difficult to gauge the true significance of individual tokens solely based on raw attention weights. To address this issue, various studies, including [6] and [7], have proposed techniques to enhance token identifiability. The research in [6] introduced two solutions: attention rollout and attention flow.

These algorithms aim to calculate a token's attention at a specific layer by linearly combining the attention from preceding layers with the attention at that layer. In the attempt to apply the concept of attention rollout, the attention rollouts for the <CLS> token of the last layer were computed and projected onto a heatmap, as depicted in Fig 4.3. Unfortunately, these visualisations yielded no interpretable information. However, when we followed the proposed method of averaging the scores from only the last layer, the resulting interpretations became more meaningful.

Regarding the behaviour where rollouts were not effective, the reason for this outcome remained elusive during our analysis. We encountered challenges in understanding why the rollout visualisation failed to pro-

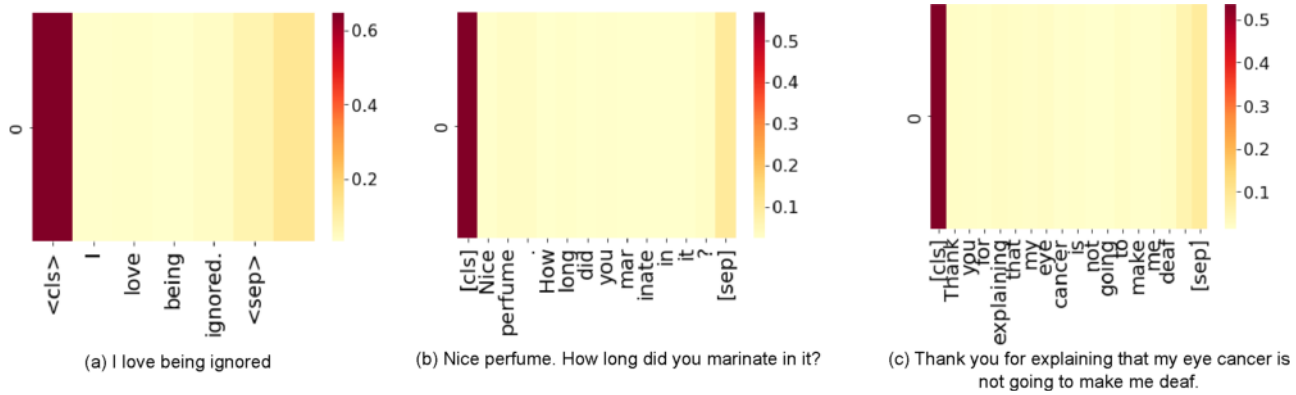

Fig. 4.3: Heat map visualizations obtained from using attention rollouts – version 1

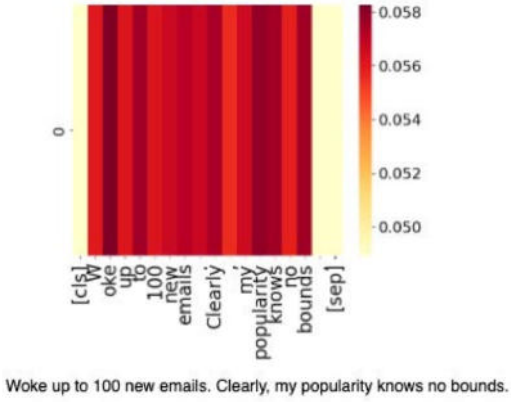

Fig. 4.4: Heat map visualizations obtained from using attention rollouts – version 2

vide clear insights. Further investigation is warranted to uncover the underlying factors contributing to this behaviour and to determine potential solutions for improving the interpretability of token attention rollouts. The reasons behind this behaviour are something to look into in the future.

For instance *"Woke up to 100 new emails. Clearly, my popularity knows no bounds."* as shown in Fig 4.4, the heatmap analysis revealed that the model did not assign higher attention weights to what we consider important keywords. Tokens such as *'bounds'*, *'popularity'* and *'woke up'* were given higher weight while allocating less attention to the word *'no'* which is the essence of textual incongruity in this instance. This observation underscores the need for the model to better capture the incongruity in the sentence, particularly with respect to the word *'no'* in order to yield more accurate interpretations. Although significant attention weight may be allocated to important tokens, it is crucial to prioritise words responsible for incongruity.

5. Conclusion. In this work, we have discussed the performance boost from using task-adaptive pre-training for sarcasm detection models. We also proposed a method to use the attention scores of the input tokens to provide some interpretability. These interpretations aid in understanding the tokens that have had the greatest influence on a certain decision. We believe that not all of the tweets adhere to the traditional definition of sarcasm. Consider the following tweet: *"I just failed my driving test."* There is a reply to the comment that says, *"Very Good! Well done"*.

In light of the situation, the response is sarcastic. However, if we only evaluate the Twitter reply, we cannot conclude that it is sarcastic. As a result, it is critical to provide context in addition to the tweet reply. Besides,

there are other categories of sarcasm such as irony, satire, and so on, as explained in [16]. Despite the fact that the *ISarcasm* dataset has sought to provide training data from several categories of sarcasm, there are relatively few of them. Transformers are attention-based models that have advanced to the forefront of many NLP tasks. However, in order to properly understand their capabilities, they must be trained with a significant amount of data. To conclude this work, we believe that we require a standard dataset for sarcasm detection in social media that comprises a fair amount of data on all varieties of sarcasm as well as other features such as context.

REFERENCES

[1] Gururangan, Suchin, et al., *Don't stop pretraining: Adapt language models to domains and tasks.*, arXiv preprint arXiv:2004.10964 (2020).

[2] F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves, *Tweeteval: Unified benchmark and comparative evaluation for tweet classification,*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 1644–1650.

[3] Wiegreffe, S., & Pinter, Y,*Attention is not not Explanation.*, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)

[4] Vashishth, S., Upadhyay, S., Tomar, G. S., & Faruqui, M., (2019)*Attention interpretability across nlp tasks.*, ArXiv Preprint ArXiv:1909.11218.

[5] Vig, J, (2019). *Visualizing Attention in Transformer-Based Language models.*, ArXiv Preprint ArXiv:1904.02679.

[6] Abnar, Samira, and Willem Zuidema., *Quantifying attention flow in transformers.*, arXiv preprint arXiv:2005.00928 (2020).

[7] Brunner, Gino, et al., *On identifiability in transformers.*, arXiv preprint arXiv:1908.04211 (2019).

[8] Pawar, Neha, and Sukhada Bhingarkar., *Machine learning based sarcasm detection on Twitter data.*, 2020 5th international conference on communication and electronics systems (ICCES). IEEE, 2020.

[9] Godara, Jyoti, et al., *Ensemble classification approach for sarcasm detection.*,Behavioural Neurology 2021 (2021).

[10] Khodak, Mikhail, Nikunj Saunshi, and Kiran Vodrahalli., *A large self-annotated corpus for sarcasm.*, arXiv preprint arXiv:1704.05579 (2017).

[11] Vinoth, D., and P. Prabhavathy., *An intelligent machine learning-based sarcasm detection and classification model on social networks.*, The Journal of Supercomputing 78.8 (2022): 10575-10594.

[12] Akula, Ramya, and Ivan Garibay., *Interpretable multi-head self-attention architecture for sarcasm detection in social media.*, Entropy 23.4 (2021): 394.

[13] Wu, Yang, et al., *Modeling incongruity between modalities for multimodal sarcasm detection.*,IEEE MultiMedia 28.2 (2021): 86-95.

[14] Baruah, Arup, et al. , *Context-aware sarcasm detection using bert.*, Proceedings of the Second Workshop on Figurative Language Processing. 2020.

[15] Vig, Jesse., *A multiscale visualization of attention in the transformer model.*, arXiv preprint arXiv:1906.05714 (2019).

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

[17] Feldman, Ronen., *Techniques and applications for sentiment analysis.*Communications of the ACM 56.4 (2013): 82-89.

[18] Montoyo, Andrés, Patricio Martínez-Barco, and Alexandra Balahur., *Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments.* Decision Support Systems 53.4 (2012): 675-679.

[19] Qiu, Xipeng, et al., *iPre-trained models for natural language processing: A survey.*,Science China Technological Sciences 63.10 (2020): 1872-1897.

[20] Attardo, Salvatore, and Jean-Charles Chabanne. , *Jokes as a text type.*,(1992): 165-176.

[21] Vaswani, Ashish, et al. , *Attention is all you need.*,Advances in neural information processing systems 30 (2017).

[22] Powers, David MW., *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.*,arXiv preprint arXiv:2010.16061 (2020).

[23] Liu, Yinhan, et al., *Roberta: A robustly optimized bert pretraining approach.*,arXiv preprint arXiv:1907.11692 (2019).

[24] Zhu, Yukun, et al., *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.*,Proceedings of the IEEE international conference on computer vision. 2015.

[25] Sebastian Nagel. 2016, *Cc-news. http: //web.archive.org/save/http: //commoncrawl.org/2016/10/newsdataset-available.*

[26] Aaron Gokaslan and Vanya Cohen. 2019. , *Openwebtext corpus. http://web.archive.org/save/http://Skylion007.github.io/ OpenWebTextCorpus.*

[27] Trinh, Trieu H., and Quoc V. Le., *A simple method for commonsense reasoning.*,arXiv preprint arXiv:1806.02847 (2018).

[28] Sundararajan, Mukund, Ankur Taly, and Qiqi Yan., *Axiomatic attribution for deep networks.*,International conference on machine learning. PMLR, 2017.

Edited by: Rajni Mohana
Special issue on: Sentiment Analysis and Affective computing in Multimedia Data on Social Network
Received: Mar 21, 2023
Accepted: Aug 1, 2023