Received: May 4, 2023. Revised: June 14, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Measuring the Quality of Semantic Data Augmentation for Sarcasm Detection

Alif Tri Handoyo[1]* **Aurélien diot[2]** **Hidayaturrahman[1]** **Derwin Suhartono[1]**
Bart Lamiroy[2]

[1]*Computer Science Department, School of Computer Science, Bina Nusantara University,Jakarta 11480, Indonesia*
[2]*Université de Reims Champagne-Ardenne, CReSTIC EA 3804, Reims 51100, France*
* Corresponding author's Email: alif.handoyo@binus.ac.id

Abstract: Sarcasm is a form of figurative speech where the intended meaning of a sentence is different from it literal meaning. Sarcastic expressions tend to confuse automatic NLP approaches in many application domains, making their detection of significant importance. One of the challenges in machine learning approaches to sarcasm detection is the difficulty of acquiring ground-truth annotations. Thus, human-annotated datasets usually contain only a few thousand texts, often being unbalanced. In this paper, we propose two different pipelines of data augmentation to generate more sarcastic data. The first one is SMERT-BERT, a modified SMERTI pipeline that uses RoBERTa as the language model for the text infilling module. The second one is SWORD (semantic text exchange by Word-Attribution), where we modified the masking module in the SMERTI pipeline by utilizing the word-attribution value. These approaches are combined with a SLOR (syntactic log-odds ratio) metric to filter the generated sarcastic data and only select sentences with the best score. Our experiments show that the use of a SLOR filter has a significant positive contribution to the augmentation process. In particular, we achieve the best results when using the SMERT-BERT pipeline and a SLOR filter by improving the F-measure by 4.00% on the iSarcasm dataset, compared to the baseline models.

Keywords: BERT, Data augmentation, Sarcasm detection, SLOR, SMERTI.

1. Introduction

In this paper, we explore the effects of data augmentation in sarcasm detection by generating more training data using a modified SMERTI [1] pipeline with RoBERTa (A Robustly Optimized BERT pretraining approach) for the text infilling module and using Word-Attribution for the mask module. We analyze the results using three different datasets: Ghosh, iSarcasm, and SemEval-18.

Sarcasm is a term that refers to the use of words to ridicule, irritate, or amuse someone. It is widely used on social networks. The metaphorical and creative nature of sarcasm creates considerable difficulty to detect sarcastic sentences when using machine learning approaches [2].

Furthermore, because sarcasm indicates a sentiment [3], its detection in a text is essential to predict the accurate sentiment of the text. Thus, sarcasm detection is a valuable tool with many applications in areas such as safety, health, service, product evaluation, and sales. It is also an essential aspect of creative understanding of language [4] and online opinion mining [5]. Yet even for humans, identification of sarcasm is difficult due to the high contextualization [6].

There have been various works in sarcasm detection in recent years. Goel et al. [7] use various deep learning models such as long short-term memory (LSTM), gated recurrent unit (GRU), and baseline convolutional neural networks (CNN) in sarcasm detection using news headlines and Reddit datasets. To further improve the performance, they proposed ensemble models and found that the weighted average ensemble gave the best performance. Another research by Amer and Siddiqu [8] uses three feature set engineering: context-based on features set, sarcastic based on features, and lexical based on feature, and found that by combining the three feature sets. They achieve the best accuracy when using KNN as the classifier. Another interesting approach was also researched by Wen et al. [9] using sememe knowledge and auxiliary information enhanced approach in Chinese sarcasm detection. Their proposed methods work by first introducing the sememe knowledge to enhance the representation learning of Chinese words at the word level. Then, at the sentence level, they leverage some auxiliary information, such as the news title, to learn the representation of the context and background of sarcasm expression and construct the representation of text expression progressively and dynamically. The evaluation results show that their proposed approach is effective in Chinese sarcasm detection.

Although recent work in sarcasm detection has been improved, there is a problem with the scarcity of sarcastic data in sarcasm detection datasets. Due to the difficulty of acquiring ground-truth annotations in sarcasm datasets, human-annotated datasets usually contain only a few hundred or thousands of sarcastic data items and often are unbalanced with respect to the sarcastic vs. non-sarcastic expressions [10]. One of the possible mitigations against the lack of data and their balance is to artificially enhance the existing data. However, the main challenge in performing data augmentation (especially when generating sarcastic data) is to maintain the sarcastic nature of the text and maintain the quality of the generated text.

In order to achieve this, we measure the quality of the data using the SLOR (syntactic log-odd ratio) metric [11]. SLOR works by measuring the fluency of a sentence. SLOR gives a score to a sentence S based on the log-probability of a given language model, normalized by the length of the sentence and the unigram log-probability. Kann et al. [11] prove in their article that SLOR is a good reference-less metric for natural language generation systems.

In this paper, we use data augmentation with two different pipelines, first by using the pre-trained RoBERTa [12] model to generate sarcastic data and secondly, by utilizing Wordattribution [13] for masking the original text and using pretrained RoBERTa [12] model to generate the sarcastic data.

Our contributions are the following:

1) We have developed a modified SMERTI pipeline [1] to generate more sarcastic data, called SMERT-BERT, by using RoBERTa as pre-trained transformer model for the text infilling module.
2) We created another new pipeline called SWORD (semantic text exchange by Word-Attribution) by using Word-Attribution to take only the most positive and the most negatively charged words within the sentence and use them for the mask module. We then apply RoBERTa for the text infilling module to generate more sarcastic data.
3) We use the SLOR metric to filter the generated sarcastic data, both for SMERT-BERT and semantic text exchange by Word-Attribution (SWORD) then prove that SLOR is a good metric to improve the data augmentation of sarcasm detection on both of those pipelines.

Our contributions considered as state-of-the-art due to how we utilized advanced technique such as RoBERTa, as the text infilling module for the SMERT-BERT pipeline, and also utilizing semantic text exchange by using SWORD pipeline to generate more sarcastic data. Moreover, our study includes an evaluation of the generated data using SLOR metric, which is a novel approach in the field of sarcasm detection.

The remainder of this paper is organized as follows: in section II, we discuss related work, and section III introduces the datasets and the methodology that we used. Section IV details our augmentation algorithms SMERT-BERT and SWORD, and in section V we analyze and discuss the obtained results, before concluding the paper and mentioning directions for future research.

2. Literature review

Data augmentation is often used in computer vision but can also be applied in natural language processing. Its main goal is to compensate for the lack of sufficient training data – often human-annotated – by using techniques to appropriately modify the existing data in a task-consistent way to increase its volume. The main challenges in data augmentation consist of not introducing biases through the augmentation algorithms and avoiding overfitting potential artifacts introduced by it. When applied to NLP (natural language processing) related tasks and data, supplementary constraints consist of generating data that is consistent with the language model under consideration. This may sometimes result in a chicken-and-egg problem when the language model is actually the thing that one wants to model through the data. In our case, we need to be confident that the augmented data actually conveys a sarcastic nature.

Abaskohi et al. [2] defines sarcasm as a term that refers to the use of words to mock, irritate, or amuse someone. Hee et al. [14] and Ilic et al. [15] discovered that simply´ increasing the training sample by scraping more data does not necessarily benefit the classification results of sarcasm detection. This is because the newly scraped data would not sufficiently increase new information that can increase the performance of the models, especially when the data is labeled automatically, thus reducing the beneficial effect of increasing the training data. They conclude that to further improve the performance of the sarcasm detection model, additional manually annotated data may be needed.

Feng et al. [16] tried various NLP data augmentation techniques on Yelp reviews. The first one, "Random Insertion, Deletion, & Swap," consists of adding, then deleting a random word in the sentence, and finally taking two words and swapping them. The second one, "Semantic Text Exchange," consists of using similarity between tokens in a sentence and keywords in a dictionary made by the most occurring words in the dataset, then creating a masking module based on the similarity of the replaced word and the other words in the sentence. The two last techniques studied in [16] is "Synthetic Noise" and "Keyword Replacement." It is indicated in their article that "Synthetic Noise" and "Keyword Replacement" can provide better performance.

Abaskohi et al. [2] also tried data augmentation with generator-based and mutation-based methods in NLP problems. The first one is using the GPT-2 [17] generative model to generate both sarcastic, and non-sarcastic data, and extracting samples from the generated sentences. The second uses synonym replacement, word elimination, and shuffling. They show that those methods do not increase F-measure performance for sarcasm detection. This is because GPT-2 is not pre-trained specifically to generate sarcastic data, thus using either GPT-2 or the new GPT-3, to generate sarcastic data will ultimately still depend on the quality and size of the training data that is used to train the GPT models.

Others augmentation techniques are also already implemented in sarcasm detection. Abdullah et al. [18] use the downsampling and augmentation method to produce a balanced dataset. Shekhawat et al. [19] performed data augmentation using nlpaug library by taking sarcastic tweets given by organizers and using the word-replacement technique to synthesize three additional tweets from each input tweet. According to their research, data augmentation proved to improve the performance of sarcasm detection.

Data augmentation also proved to improve the performance of other text classification tasks. Moreno Monterde et al. [20] propose two models, the first one is a binary multilabel classifier using Bayesian networks, and the second one is using BERT. The dataset that they were using was very unbalanced. Thus, they use balancing techniques in order to achieve good results. They use a synonym augmenter by swapping one random word by its synonym and keeping the label. They also suggest to only applied the technique only once to prevent overfitting. The results show that using a synonym augmenter could improve the performance of sarcasm detection. Although this approach seems to improve the performance, there is a lack of analysis on the performance difference before and after augmentation. Also, augmenting by only replacing one word with the synonym could potentially generate synthetic data that is very similar to the original one, thus risking overfitting the model. In earlier works, Handoyo et al. [21] augmented the data by only changing one word to its synonym. demonstrate the overfitting effects, which cause performance to decline as more augmented data are used.

Stylianou et al. [22] propose a data augmentation method for low-resource and imbalanced datasets by aligning language models to in-domain data prior to generating synthetic examples. They propose alignment to existing generic models in task-specific unlabeled data to boost the performance of text classification tasks. They found that in-domain alignment help creates better synthetic data and improve the performance of text classification. They also found that there is a positive connection between a number of training parameters in language models and the volume of fine-tuning data. Although in-domain alignment proved to boost the performance of text classification tasks, implementing it in sarcasm detection [14, 15] show that it did not help much the performance for detecting sarcasm and irony.

As mentioned before, one of the challenges in performing data augmentation in natural language processing is to ensure that the generated text data provides a decent quality text and maintains the context of the original text. Kann et al. [11] proposes a metric for reference-less fluency for natural language generation output at the sentence level, called syntactic log-odds ratio (SLOR). They show that the SLOR metric correlates much more with human judgment and that they minimize the mean squared error when judging metric performance for sentences with similar quality compared to baseline metrics like ROUGE-L, N-gram-overlap metrics, Negative cross-entropy, perplexity, and BLEU. In our work, we will use this method to produce a new type of masking module by masking only the most positive or the most negative attribution score. We address those attribution scores in the next paragraph.

In the past few years, many works have been done to try to explain the prediction of deep learning models. One of the best approaches is known as the feature attribution method [23-25]. These methods explain a model's prediction by crediting each input feature based on how much it influenced that prediction. Recently, Janizek et al. [13] proposed a

Received: May 4, 2023. Revised: June 14, 2023.

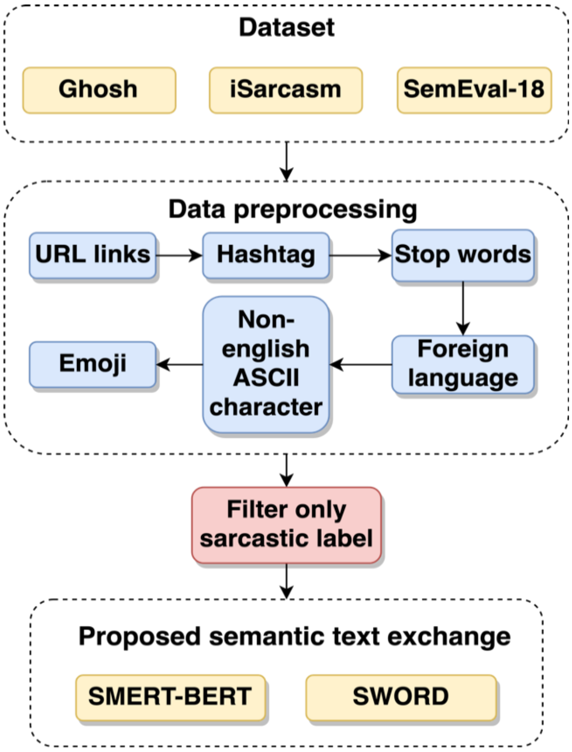

Figure. 1 Workflow of the research

new method of score attribution, by using Integrated Hessians, which is an extension of Integrated Gradients that explains pairwise feature interactions in neural networks. Integrated Hessians methods are not limited to using specific architectures or classes of neural networks. By using a pre-trained transformer architecture, it outperforms basic CNN that was trained from scratch when examining interactions on a given sentence. Our SWORD pipeline will utilise the Integrated Hessians method for the Word-Attribution selection. More details are given in section 5.1.

3. Methodology

Our data augmentation approaches: SMERT-BERT and SWORD are described in subsections IV-A and IV-B. In order to measure the impact of our approaches to sarcasm detection and to compare the results to baseline model, we use the following experimental protocol as illustrated in Fig. 1.

Our experimental setup focuses on generating new sarcastic data using three different datasets: Ghosh [26], Isarcasm [10], and Semeval-18 [14]. We then continue by performing data pre-processing with six different steps, then filter it by only taking the sarcastic data, and perform data augmentation with our two different pipelines SMERT-BERT and SWORD. The training data and the augmented data were then combined, and for each dataset, a validation portion was used to evaluate the model.

Table 1. The proportion of sarcastic and non-sarcastic data in various datasets

Dataset	Non-Sarcastic	Sarcastic	%Sarcasm
Ghosh	22,725	18,478	44.84%
iSarcasm	3,584	766	17.62%
SemEval-18	2,379	2,177	49.12%

1) Dataset

iSarcasm dataset contains tweets that are written by participants of an online survey and is an example of intended sarcasm text as the authors of the tweets themselves have annotated the data. Due to the nature of how this dataset is collected, iSarcasm dataset is one of the most unbalanced datasets that was used in this research. For SemEval-18 dataset, it contains sarcastic tweets that were labeled by third-party annotators and are used for perceived sarcasm detection. Even though the data is labeled manually, SemEval-18 dataset still seems fairly balanced. As for the Ghosh dataset, it was an automatically collected dataset that contain tweets having particular hashtags such as #sarcasm, #not as sarcastic, and others as non-sarcastic. The Gosh dataset were labeled automatically, thus providing a fairly balanced dataset. The distribution of sarcastic and nonsarcastic data in every dataset can be seen in Table 1.

2) Data preprocessing

Pre-processing is an important part of this experiment. It is done to make sure that the text transforms into a more digestible form so that the models can perform well. The steps of pre-processing conducted in this study are by dropping all duplicate tweets across the datasets, and deleting all URL links, hashtags, foreign language characters, stop words removal, non-English ASCII characters, and emoji.

3) Filter only data labeled as sarcastic

Our experiments focus on generating sarcastic labeled text in the datasets. Thus here we filter the data and only use the texts tagged as sarcastic labeled text for performing the data augmentation.

4) Semantic text exchange

Once the data has been cleaned and filtered, we apply our approaches for performing semantic text exchange by maintaining the semantic value of the sarcastic text using two different pipelines: SMERT-BERT and SWORD which we will be defining later.

The amount of generated sarcastic text varies between datasets. This is because different datasets need different amounts of data to be balanced. Details of the amount of generated sarcastic text that was used and the class distribution before and after data augmentation can be seen in Table 3.

Received: May 4, 2023. Revised: June 14, 2023.

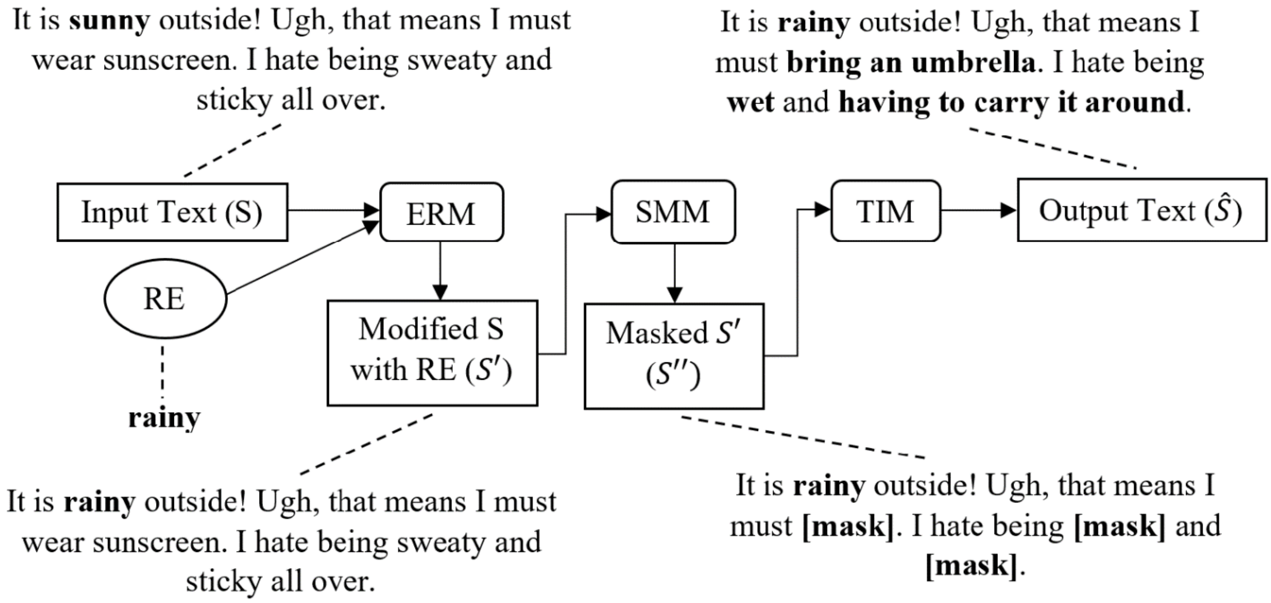
It is **sunny** outside! Ugh, that means I must wear sunscreen. I hate being sweaty and sticky all over.
It is **rainy** outside! Ugh, that means I must **bring an umbrella**. I hate being **wet** and **having to carry it around**.
Input Text (S)
ERM
SMM
TIM
Output Text ($\hat{S}$)
RE
rainy
Modified S with RE (S')
Masked S' (S'')
It is **rainy** outside! Ugh, that means I must wear sunscreen. I hate being sweaty and sticky all over.
It is **rainy** outside! Ugh, that means I must **[mask]**. I hate being **[mask]** and **[mask]**.

Figure. 2 Diagram of SMERTI pipeline

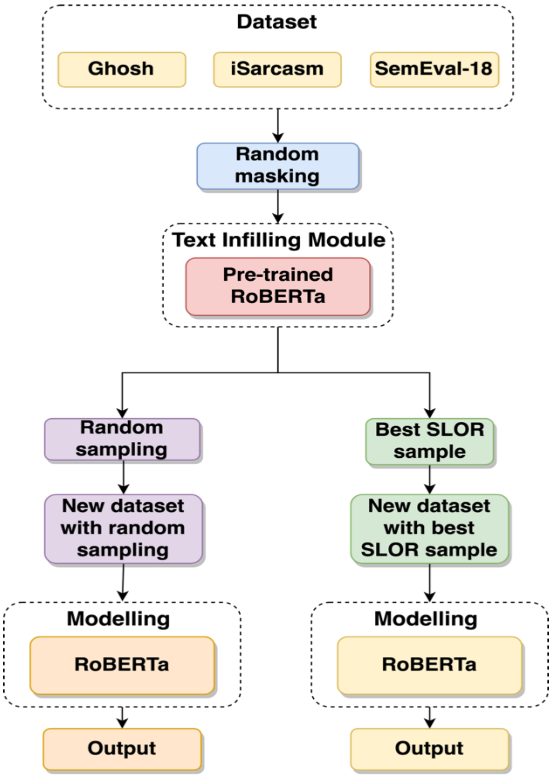

Figure. 3 SMERT-BERT pipeline

4. Data augmentation algorithms

In this experiment, we apply data augmentation only in training data to prevent information leaking to the validation dataset which used to evaluated the models. Among our algorithms, SMERT-BERT takes the sarcastic pre-processed sentences as input, then applies random masking based on the length of the sentence. As for SWORD (semantic text exchange by Word-Attribution), it takes the attribution score of each word from our pre-processed sarcastic data, then masks the words with the best and worst attribution scores. We then make the mask-filling in the text infilling module, which is a pre-trained RoBERTa, adapted to mask-filling tasks. Finally, to make sure our generated sarcastic data still maintain the text quality and maintain the sarcastic nature of the text, we use the SLOR metric to filter the generated sarcastic data. By taking the sample of data augmentation depending on the length of the chosen dataset, we compare the results when using the SLOR metric to filter the generated sarcastic data. To evaluate the quality of the generated data, we use indirect measurement approach by observing the improvement in performance after augmenting the data. The improvement here implies that we successfully preserve the quality of sarcastic text, since the model remains resilient even after merging the original training dataset with the augmented dataset.

Table 2. Initial experiment using different percentages of masking

Metric	Original	20% Masking	40% Masking	60% Masking	80% Masking
F-measure	0.7890	0.7898	0.7779	0.7802	0.7855
Precision	0.8043	0.8078	0.8005	0.8049	0.7924
Recall	0.7742	0.7725	0.7566	0.7569	0.7787
Accuracy	0.8136	0.8150	0.8056	0.8081	0.8087

Table 3. Proportion of sarcastic and non-sarcastic data before and after augmentation for SMERT-BERT and SWORD. pipeline

Dataset	Before Augmentation		After Augmentation		Amount of Aug. Data
	Non-Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic	
Ghosh	22,725	18,478	22,725	21,576	3,098
iSarcasm	3,584	766	3,584	872	106
SemEval-18	2,379	2,177	2,379	2,324	147

4.1 SMERT-BERT

SMERT-BERT is based on a modified SMERTI pipeline [1]. SMERTI has been introduced in data augmentation for sentiment detection. The main goal of this pipeline is to generate data by preserving the fluency and sentiment of the sentence. It combines entity replacement, similarity masking module, and text infilling. Entity replacement module (ERM) will identify which word within the original text that are best replaced with the replacement entity (RE). The similarity masking module (SMM) will then identify the words that have been selected by ERM that are similar to the original text and replace them with a [mask]. Then lastly, text infilling module (TIM) will then fill the [mask] with words that best suit the replacement entity, thus will modify the semantics in the rest of the text. A basic example of how SMERTI works can be seen in Fig. 2.

In our research, we extend SMERTI to generate sarcastic data by using only random masking and adding text infilling using a pre-trained RoBERTa language model. As for the mask-filling module, we are using a pre-trained transformer model, RoBERTa [12], a more robust version of BERT [27], to get new sentences by using context. Details of this pipeline can be seen in Fig. 3. The figure includes two scenarios, one of which has supplementary data selection based on their SLOR score. That specific part will be detailed in subsection IV-D.

As already mentioned, our pipeline is applied to the three reference datasets. For the masking, we use random masking and mask 20% of the original sentence. We determine to use 20% due to an initial experiment that tests the masking using four different configurations, from 20%, 40%, 60%, and 80% of random masking. The results show that using 20% of random masking gives an overall better F-measure when detecting sarcastic sentences when using SMERTBERT pipeline, compared to other percentages. The same outcome can also be seen in precision, recall, and accuracy results. Our initial experiment results, which using SMERTBERT pipeline can be seen in Table 2.

After the random masking process, we then use a pretrained RoBERTa language model for replacing the masked parts with the most similar words, which means the word has the embedding closest to the masked parts while maintaining the context of the sentences. To maintain the context of the sentences, RoBERTa uses contextual embedding, where the vector representation of each word in the vocabulary was obtained after training the whole sentences on the model, thus the pre-trained model also learned the contextual meaning of the word [12]. The newly generated dataset is then used to augment the original data. The proportion of the sarcastic and non-sarcastic data, before and after augmentation can be seen in Table 3.

Table 4. Hyperparameters settings for every experiments scenario

Hyperparameter	value
max_seq_length	40
learning_rate	0.00001
weight_decay	0.01
warmup_ratio	0.2
max_grad_norm	1.0
num_train_epochs	10
train_batch_size	16
fp16	True

Based on Table 3, we can see that for the iSarcasm dataset, we only add 106 sarcastic data elements. For the Ghosh dataset, we add 3,098 sarcastic data items for SemEval-18 dataset we add 147. We add only a small amount of data, and the reason is, if we add too much generated sarcastic data, it will contribute more noise in the model and affect the performance, thus the amount of augmented data that we use is based on our experiments that give the best performance boost for every dataset. We used the exact same amount of augmented data from Table 3 for every experiment scenario. After we generated the new sarcastic data, we concatenated the generated data with the original data and build the model RoBERTa, and BERT have essentially identical architectures, the researchers made a few modest changes to RoBERTa design and training methodology to improve the model’s performance [12]. We use RoBERTa pre-trained model [28] to classify whether the sentence is

Received: May 4, 2023. Revised: June 14, 2023.

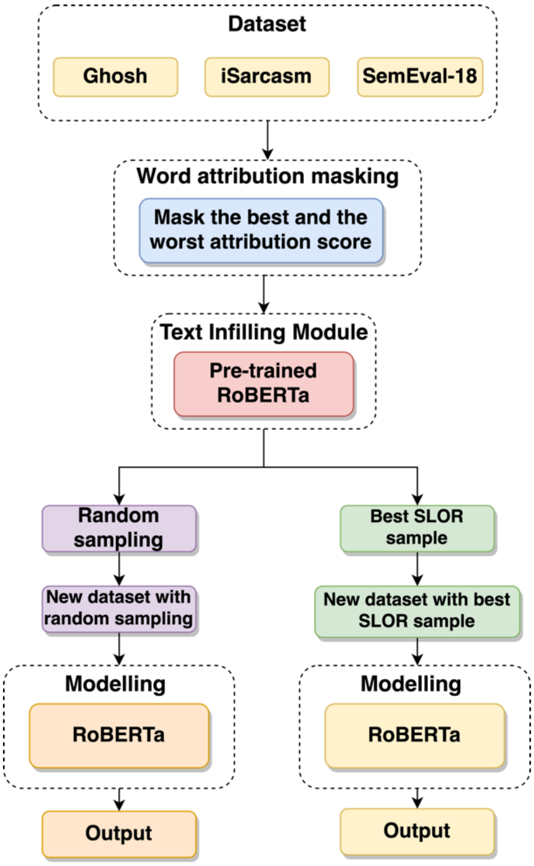

Figure. 4 SWORD Pipeline

sarcastic or non-sarcastic. We use the training dataset to fine-tune the models hyperparameters. In this study, we use a batch size of 16 and an epoch size of 10 for every experiments scenario. Details of the hyperparameters settings on can be seen in Table 4.

The hyperparameters value came from the model default settings. Additionally, we experimenting using the gridsearch method to determine the ideal value for num_train_epochs and train_batch_size.

4.2 SWORD (semantic text exchange by Word-Attribution)

SWORD, uses Word-Attribution for the masking, as for the rest of the pipeline, it is the same as SMERT-BERT pipeline. The details of this pipeline can be seen in Fig. 4.

The SWORD only masks the words with the most positive attribute and most negative attribute scores. We only mask two of those words because of the fact that the most positive attribute word contributes to how the sentence is sarcastic, and the most negative score contributes to how the sentence is less sarcastic. We then use RoBERTa for the text infilling module and generate it with a random sample and retain only the sentence with the best SLOR. The generated sarcastic data that were explained before in Table 3, are then concatenated with the original data and used to build the sarcasm detection model using RoBERTa pre-trained language model.

4.3 Word-attribution

Word-Attribution are a method used for quantifying pairwise feature interactions that can be applicable to any neural network architecture. The pairwise feature interactions here is indicated as word-attribution score. To compute the attributes score, word-attribution uses Integrated Hessians, a paired feature interaction in neural networks explained by an extension of Integrated Gradients [13]. Unlike earlier methods for explaining interactions, Integrated Hessians are not bound to a certain architecture or class of neural network. Integrated Hessians works by applying Integrated Gradients [29], to itself to show how much feature *j* influenced the significance of feature *i*:

$$\Gamma i,j(x) = \phi_j(\phi_i(x)) \tag{1}$$

For $i \neq j$ we can then derive that:

$$\Gamma i,j(x) = (x_i - x_i')(x_j - x_j') \times \int_{\beta=0}^{1}\int_{\alpha=0}^{1} \alpha\beta \frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j} d\,\alpha d\beta \tag{2}$$

In the case of $i = j$, the formula $\Gamma i,j(x)$ has an additional first-order term. We interpret $\Gamma i,j(x)$ as the explanation of the importance of feature *i* in terms of the input value of feature *j*.

4.4 Syntactic log-odds ratio (SLOR)

SLOR is a normalized language model score, it is used as a metric for sentence-level reference-less fluency evaluation of output from natural language generation [11].

SLOR assigns to a sentence *S* a score that consists of its log-probability under a given LM (Language Model), normalized by unigram log-probability and length:

$$SLOR(S) = \frac{1}{|S|}(ln(\mathcal{P}_u\mathcal{P}_M(S) - ln(\mathcal{P}_u(S))) \tag{3}$$

Where $\mathcal{P}_M(S)$ is the probability assigned to the sentence under the LM. The unigram probability $\mathcal{P}_u(S)$ of the sentences is calculated as:

$$\mathcal{P}_u(S) = \prod_{t \in S} \mathcal{P}(t) \tag{4}$$

With $\mathcal{P}(t)$ being the unconditional probability of a token t, i.e., given no context.

SLOR computes the probability of a sentence with a long-short term memory (LSTM) LM. More details on LSTM LMs can be found in [30]. The unigram probabilities for SLOR are estimated using the same corpus.

In our research, we use the SLOR metric to filter the generated augmented data, and only take texts with the best SLOR value. This means that we take only the generated sarcastic sentence with the most similar standard deviation of SLOR when compared to the original sarcastic sentence.

5. Result and discussion

For measuring the model performance when classifying between sarcastic and non-sarcastic sentences, we use Fmeasure, precision, and recall. F-measure uses the harmonic mean of precision and recall to penalize extreme values. It is the proper metric to use in case false negatives and false positives are of the same importance. This also means that F-measure is a good metric for an imbalanced dataset. To fully evaluate the effectiveness of the model, precision and recall results are also provided.

All results are in this section, including baseline, SMERT-BERT, and SWORD with and without selection of data using SLOR. After that, we analyse the results by comparing the results of the baseline method and the proposed method. We also analyse if applying SLOR filtering increases the results when compared to random samples. This section will also present the impact of our data augmentation method for different types of datasets, and then also compare the efficiency of the SLOR metric for sarcasm detection.

5.1 Model performance

The performance comparison between the baseline and the proposed methods can be seen in Table 5. For readability, the performance increase (difference between the baseline and the proposed methods) can be seen in Table 6. The latter can be entirely derived from the former.

Based on Table 5, we can see that data augmentation, whether using SMERT-BERT or SWORD pipeline, gives better performance results when compared to the baseline model. Nevertheless, when we analyse the amount of increased performance from Table 6, we can see that the performance improvement depends on the dataset. Among the three datasets, iSarcasm gives the most significant performance boost when using SMERT-BERT with SLOR filtering, with a +0.04 of performance gain on F-measure. iSarcasm had a characteristic of a small and imbalanced dataset. Although SemEval-18 was also a small dataset, it was still fairly balanced, thus the most significant performance gain was at +0.03 of F-measure. As for the Ghosh dataset, its size was big and was also fairly balanced, thus when we applied data augmentation whether when we are using SMERT-BERT or SWORD pipeline, There are no improvements on the Fmeasure. From here we can assume that data our augmentation approach is effective only when applied to a small and imbalanced dataset.

The performance of the model also depends on the dataset that had been used to create the model. From Table 5 we can see that the best F-measure performance is obtained when using the Ghosh dataset with an F-measure of 0.79. iSarcasm and SemEval-18 were overall quite less performing with the best F-measure of 0.42 and 0.69 respectively.

5.2 Augmentation performance

While the size and the proportion of the class of the Ghosh dataset were the biggest factors of why the Ghosh dataset had a higher F-measure, the way the dataset is collected also contributed to the cause. Ghosh dataset is automatically generated, thus making it easier to create a dataset that was quite big in size and fairly balanced in class proportion. Furthermore, the way the Ghosh dataset labeled the data by detecting the "sarcasm" hashtag, maintain the quality of the label and made it more consistent. Nevertheless, because the label was added automatically, there is a high chance that the labeled sarcastic data was less precise, resulting in a lowerquality model.

As for iSarcasm dataset, the lack of size, and the imbalanced nature of the dataset were the biggest factor in the low F-measure score. However, this happen due to the fact that the iSarcasm dataset was manually collected, making it harder to collect many data, while maintaining the proportion between the sarcastic and non-sarcastic class. SemEval-18 dataset, was also collected manually, thus explaining the nature of the small size. However, it was quite balanced in proportion, resulting in F-measure that was slightly better than iSarcasm dataset. It should be noted while both the iSarcasm and SemEval-18 are manually collected datasets, they both have a different way of labeling. The iSarcasm dataset contains intended sarcasm from the participants of an online survey, while SemEval-18 contains perceived sarcasm that was labeled by third-party annotators.

The different nature of how the data is labeled could also influence the F-measure score results. When applying data augmentation, we consider the amount of generated data to make sure that we do not

Received: May 4, 2023. Revised: June 14, 2023.

Table 5. Experiment results (raw data)

Model	SLOR Filtering		Precision	Recall	F-measure
Baseline	unapplicable	Ghosh	0.79	0.79	0.79
		iSarcasm	0.46	0.33	0.38
		SemEval-18	0.61	0.73	0.66
SMERT-BERT	without SLOR	Ghosh	0.80	0.79	0.79
		iSarcasm	0.47	0.36	0.41
		SemEval-18	0.60	0.76	0.67
	with SLOR	Ghosh	0.78	0.80	0.79
		iSarcasm	0.46	0.39	0.42
		SemEval-18	0.61	0.78	0.69
SWORD	without SLOR	Ghosh	0.80	0.77	0.79
		iSarcasm	0.44	0.38	0.41
		SemEval-18	0.60	0.74	0.66
	with SLOR	Ghosh	0.80	0.79	0.79
		iSarcasm	0.46	0.38	0.41
		SemEval-18	0.60	0.76	0.67

Table 6. Experiment results (increase wrt. Benchmark)

Model	SLOR Filtering		Precision	Recall	F-measure
SMERT-BERT	without SLOR	Ghosh	+0.01		
		iSarcasm	+0.01	+0.03	+0.03
		SemEval-18	-0.01	+0.03	+0.01
	with SLOR	Ghosh	-0.01	+0.01	
		iSarcasm		+0.06	+0.04
		SemEval-18		+0.05	+0.03
SWORD	without SLOR	Ghosh	+0.01	-0.02	
		iSarcasm	-0.02	+0.05	+0.03
		SemEval-18	-0.01	+0.01	
	with SLOR	Ghosh	+0.01		
		iSarcasm		+0.05	+0.03
		SemEval-18	-0.01	+0.03	+0.01

add too much sarcastic data and make our dataset unbalanced due to the generated data. Furthermore, we think that adding too much-augmented data creates bias, and it could reduce the performance of the model.

In this case, we arrive at 3,098 generated data for Ghosh dataset, which balance our dataset. 147 for SemEval-18, because the dataset is already balanced, so we choose to take a few samples, and finally 106 generated data for iSarcasm dataset. The details can be seen in Table 3. Even if iSarcasm dataset is completely unbalanced with the non-sarcastic class, we cannot generate too much data for the sarcastic class, due to the bias effect. In fact, we only had 612 sarcastic sentences in the training dataset, so by adding 106 data, we already added 17,3% of the amount of sarcastic sentences. Generating too much data augmentation could also influence the performance results due to the introduced bias, thus in this experiment, the amount of data we add depends on the size and class proportion of the dataset.

5.3 The influence of SLOR filter

Both SMERT-BERT and SWORD pipeline gives better performance results when compared to the baseline model. We can see from Table 6 on the iSarcasm dataset, the best improvement for SMERT-

Table 7. Comparison table of similar work

Authors	Techniques Used	Discussions
Amirhossein *et al.* [16]	Model: BERT-based Data Augmentation: word removal Dataset: iSarcasm	F1 Score: iSarcasm: 41.4% Limitation: data augmentation only use word removal, thus reducing sentence quality and potentially remove sarcastic nature of sentence
Goyal [31]	Model: RoBERTa Data Augmentation: student-teacher setting Dataset: iSarcasm, Self-Annotated Reddit Corpus (SARC)	F1 Score: iSarcasm : 40.31% iSarcasm + SARC: 45.07% Their research show that by using augmentation method, they can successfully significantly increase the F1 score of the model. However, their technique incorporated new dataset that was used to be combined with the iSarcasm dataset.
Handoyo [21]	Model: BERT, RoBERTa, DistilBERT Data Augmentation: Synonym Replacement	F1 Score: iSarcasm: 40.44% Ghosh: 81.08% Ptacek: 87.41% SemEval-18: 67.46% The research show that using data augmentation in sarcasm detection could increase the F1 performance of the model. However, due to the very basic technique that was used in the research, using more augmented data resulting in the decrease of performance.
Our proposed method	Model: BERT Data Augmentation: SMERT-BERT and SWORD Pipeline Dataset: iSarcasm	F1 Score: iSarcasm: 42% for SMERT-BERT 41% for SWORD Novelty: data augmentation using SMERT-BERT and SWORD pipeline, which utilise semantic text exchange method to augment the sarcastic data while still maintaining the quality of the sentence by incorporating SLOR metric to filter the augmented result.

BERT is when applied with SLOR filtering, with an improvement of +0.04 F-measure. As for the SWORD pipeline, the best improvement is when applied with SLOR filtering, with an improvement of +0.03. In this case, both SMERT-BERT and SWORD pipeline does indeed improve the performance of F-measure when classifying whether a sentence is sarcastic or non-sarcastic. Furthermore, we increase the recall, which means that we reduce the amount of wrong predictions for the sarcastic label.

SLOR filter increases the performance for recall and Fmeasure, which is what we are looking for in our results. This seems to be the case both on SMERT-BERT and SWORD pipelines. However, when we further analyse in Table 6, on the iSarcasm dataset, Using the SLOR filter in the SMERTBERT pipeline seems to improve the performance of Fmeasure by +0.01, and for the SemEval-18 dataset, it increases by +0.02 when compared to SMERT-BERT without SLOR filter. As for the SWORD pipeline, we can see that the only improvement of F-measure when using the SLOR filter is only in SemEval-18 dataset with an improvement of only +0.01.

We can see that choosing the best SLOR values increases results for both pipelines, but it is more effective for our SMERT-BERT pipeline. SLOR filter is not effective on the SWORD pipeline due to how the masking process works. SWORD pipeline only masks the word with the most positive or the most negative score only, thus there is only one word that is masked in the sentence. Less masking made the structure of the generated sentences relatively the same as the original, thus reducing the effectiveness of the SLOR filter.

In any case, our experiments show that SLOR does not reduce the performance of the models. We then conclude that filtering the generated augmented data using the SLOR metric is a nice way to increase performance for sarcasm detection.

5.4 Comparison with prior work.

There are various similar work [2, 14-16, 19, 20, 22] that experimenting with data augmentation to improve the classification task in NLP. To provide an apples-to-apples comparison with our work, we are focusing on comparing our work with other, related research that uses data augmentation to solve the unbalanced data in sarcasm detection problem, specifically when using unbalanced dataset like iSarcasm. A comparison and discussion between our proposed model with other works in detecting sarcasm sentences can be seen in Table 7.

Based on the previous work that we already discussed in section 2, we can conclude that data augmentation in NLP classification problem could significantly boost the performance of model. Although for the case of sarcasm data, building an augmentation approach which can maintain the sarcastic nature of the text proves difficult. Previous work in the sarcasm dataset from Table 7 shows that recent work in augmentation of sarcasm data still lacks on the technique used to implement data augmentation and also on how to evaluate the generated sarcastic data. Thus in this research, we proposed SMERT-BERT and SWORD pipeline, where we use semantic text approach to create the augmented data, and also using SLOR metric to evaluate and filter our generated sarcastic text. Also, our approach only utilize the original dataset, without adding additional external dataset to further augment the data.

6. Conclusion

This article shows that data augmentation applied for sarcasm detection can give better performance, especially for small and imbalanced datasets. Our two proposed augmentation methods SMERT-BERT and SWORD manage to improve the baseline results of small datasets, such as iSarcasm and SemEval-18 by 0.04, 0.03 and 0.03, 0.01 respectively. Nevertheless, the biggest factor that impacts the performance of the model is the quality of the data itself. Bigger and balanced datasets like the Ghosh dataset, which collected and labeled automatically using the "sarcasm" hashtag offer better results in performance, with F-measure of 0.79 in both SMERT-BERT and SWORD pipeline, compared to a small dataset like iSarcasm or SemEval-18 with only 0.42, 0.69 and 0.41, 0.67 in SMERT-BERT and SWORD pipeline rerspectively.

We also found that SLOR metric contributes to better performance in SMERT-BERT methods compared to SWORD method. This happens due to the different amount of masked words in each method. SWORD method masks only one word in sentences, making the structure of the generated sentences have no significant distinctiveness compared to the original one, resulting in the ineffectiveness of the SLOR metric. Further analysis on the impact of sarcasm dataset characteristics like perceived or intended sarcasm could also be investigated to see if there is a significant impact on the model performance.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, A.T.H, A.D, H, D.S, and B.L; software, A.T.H, A.D and H.; validation, formal analysis, investigation, supervision, D.S and B.L; writing—original draft preparation, A.T.H and A.D; writing—review and editing, H., D.S, and B.L.

Acknowledgments

This work is supported by Research and Technology Transfer Office, Bina Nusantara University as a part of Bina Nusantara University's International Research Grant entitled Measuring the

Quality of Semantic Data Augmentation for Sarcasm Detection with contract number: 061/VR.RTT/IV/2022. This work was also supported by "Science et Impact" funding through the French Embassy in Indonesia and by the International Mobility program of the Université de Reims Champagne-Ardenne.

References

[1] S. Y. Feng, A. W. Li, and J. Hoey, "Keep Calm and Switch On! Preserving Sentiment and Fluency in Semantic Text Exchange", In: *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2701–2711, 2019.

[2] A. Abaskohi, A. Rasouli, T. Zeraati, and B. Bahrak, "UTNLP at SemEval-2022 Task 6: A Comparative Analysis of Sarcasm Detection Using Generative-based and Mutation-based Data Augmentation", In: *Proc of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 2022.

[3] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation", In: *Proc of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 704–714, 2013.

[4] T. Veale, F. A. Cardoso and R. P. Y. Pérez, "Systematizing Creativity: A Computational View", *Computational Synthesis and Creative Systems*, pp. 1–19, 2019.

[5] S. Kannangara, "Mining Twitter for Fine-Grained Political Opinion Polarity Classification, Ideology Detection and Sarcasm Detection", In: *Proc of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 751–752, 2018.

[6] M. Walker, J. F. Tree, P. Anand, R. Abbott, and J. King, "A Corpus for Research on Deliberation and Debate", In: *Proc of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, pp. 812–817, 2012.

[7] P. Goel, R. Jain, A. Nayyar, S. Singhal, and M. Srivastava, "Sarcasm detection using deep learning and ensemble learning", *Multimed Tools Appl.*, Vol. 81, No. 30, pp. 43229–43252, 2022.

[8] A. Y. A. Amer and T. Siddiqu, "A novel algorithm for sarcasm detection using supervised machine learning approach", *AIMS Electronics and Electrical Engineering*, Vol. 6, No. 4, pp. 345–369, 2022.

[9] Z. Wen, L. Gui, Q. Wang, M. Guo, X. Yu, J. Du, and R. Xu, "Sememe knowledge and auxiliary information enhanced approach for sarcasm detection", *Inf. Process Manag.*, Vol. 59, No. 3, p. 102883, 2022.

[10] S. Oprea and W. Magdy, "iSarcasm: A Dataset of Intended Sarcasm", In: *Proc of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[11] K. Kann, S. Rothe, and K. Filippova, "Sentence-Level Fluency Evaluation: References Help, But Can Be Spared!", In: *Proc of the 22nd Conference on Computational Natural Language Learning*, pp. 313–323, 2018.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *arXiv preprint arXiv:1907.11692*, p. https://arxiv.org/abs/1907.11692, 2019.

[13] J. D. Janizek, P. Sturmfels, and S. I. Lee, "Explaining Explanations: Axiomatic Feature Interactions for Deep Networks", *arXiv preprint arXiv:2002.04138*, p. https://arxiv.org/abs/2002.04138, 2020.

[14] C. V. Hee, E. Lefever, and V. Hoste, "SemEval-2018 Task 3: Irony Detection in English Tweets", In: *Proc. of the 12th International Workshop on Semantic Evaluation*, pp. 39–50, 2018.

[15]S. Ilić, E. M. Taylor, J. A. Balazs, and Y. Matsuo, "Deep contextualized word representations for detecting sarcasm and irony", In: *Proc of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2018.

[16] S. Y. Feng, V. Gangal, D. Kang, T. Mitamura, and E. Hovy, "GenAug: Data Augmentation for Finetuning Text Generators", In: *Proc. of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2020.

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners", *OpenAI Blog*, Vol. 1, No. 8, 2019.

[18] M. Abdullah, J. Khrais, and S. Swedat, "Transformer-Based Deep Learning for Sarcasm Detection with Imbalanced Dataset: Resampling Techniques with Downsampling and Augmentation", In: *Proc. of 2022 13th International Conference on Information and Communication Systems (ICICS)*, pp. 294–300,

2022.

[19] T. S. Shekhawat, M. Kumar, U. Rathore, A. Joshi, and J. Patro, “IISERB Brains at SemEval 2022 Task 6: A Deep-learning Framework to Identify Intended Sarcasm in English”, In: *Proc. of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 2022.

[20] A. M. Monterde, L. V. Ramos, J. Mata, and V. P. Álvarez, “I2C at SemEval-2022 Task 6: Intended Sarcasm in English using Deep Learning Techniques”, In: *Proc. of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Seattle, United States, pp. 856–861, 2022.

[21] A. T. Handoyo, H. rahman, C. J. Setiadi, and D. Suhartono, “Sarcasm Detection in Twitter - Performance Impact While Using Data Augmentation: Word Embeddings”, *International Journal of Fuzzy Logic and Intelligent Systems*, Vol. 22, No. 4, pp. 401–413, 2022.

[22] D. and T. T. and V. S. and K. I. S. Nikolaos and Chatzakou, “Domain-Aligned Data Augmentation for Low-Resource and Imbalanced Text Classification”, *Advances in Information Retrieval*, pp. 172–187, 2023.

[23] S. M. Lundberg and S. I. Lee, “A Unified Approach to Interpreting Model Predictions”, *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[24] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning Important Features Through Propagating Activation Differences”, In: *Proc. of the 34th International Conference on Machine Learning*, Vol. 70, pp. 3145–3153, 2017.

[25] A. Binder, G. Montavon, S. Bach, K. R. Müller, and W. Samek, “Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers”, *arXiv Preprint arXiv:1604.00825*, p. https://arxiv.org/abs/1604.00825, 2016.

[26] A. Ghosh and T. Veale, “Fracking Sarcasm using Neural Network”, In: *Proc of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 161–169, 2016.

[27] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, *arXiv Preprint arXiv:1810.04805*, p. https://arxiv.org/abs/1810.04805, 2018.

[28] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic Attribution for Deep Networks”, In: *Proc. of the 34th International Conference on Machine Learning*, Vol. 70, pp. 3319–3328, 2017.

[29] M. Sundermeyer, R. Schlüter, and H. Ney, “LSTM Neural Networks for Language Modeling”, In: *Proc. of the Interspeech*, pp. 194–197, 2012.

[30] I. Goyal, P. Bhandia, and S. Dulam, “Finetuning for Sarcasm Detection with a Pruned Dataset”, *arXiv preprint arXiv:2212.12213*, p. https://arxiv.org/abs/2212.12213.