electronics

Article

Comparison of Deep Learning Models for Automatic Detection of Sarcasm Context on the MUStARD Dataset

Alexandru-Costin Băroiu [1] and Ștefan Trăușan-Matu [1,2,*]

[1] Faculty of Automatic Control and Computer Science, Politehnica University of Bucharest, 060042 Bucharest, Romania
[2] Research Institute for Artificial Intelligence "Mihai Draganescu" of the Romanian Academy, 050711 Bucharest, Romania
* Correspondence: stefan.trausan@upb.ro or trausan@gmail.com

Abstract: Sentiment analysis is a major area of natural language processing (NLP) research, and its sub-area of sarcasm detection has received growing interest in the past decade. Many approaches have been proposed, from basic machine learning to multi-modal deep learning solutions, and progress has been made. Context has proven to be instrumental for sarcasm and many techniques that use context to identify sarcasm have emerged. However, no NLP research has focused on sarcasm-context detection as the main topic. Therefore, this paper proposes an approach for the automatic detection of sarcasm context, aiming to develop models that can correctly identify the contexts in which sarcasm may occur or is appropriate. Using an established dataset, MUStARD, multiple models are trained and benchmarked to find the best performer for sarcasm-context detection. This performer is proven to be an attention-based long short-term memory architecture that achieves an F1 score of 60.1. Furthermore, we tested the performance of this model on the SARC dataset and compared it with other results reported in the literature to better assess the effectiveness of this approach. Future directions of study are opened, with the prospect of developing a conversational agent that could identify and even respond to sarcasm.

Keywords: machine learning; natural language processing; sentiment analysis; sarcasm

Citation: Băroiu, A.-C.; Trăușan-Matu, Ș. Comparison of Deep Learning Models for Automatic Detection of Sarcasm Context on the MUStARD Dataset. *Electronics* **2023**, *12*, 666. https://doi.org/10.3390/electronics12030666

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 30 December 2022
Revised: 25 January 2023
Accepted: 27 January 2023
Published: 29 January 2023

1. Introduction

Sarcasm poses a great challenge for natural language processing (NLP) researchers. The presence of sarcasm can completely change the meaning of an utterance and its detection proves to be difficult for most models or systems today, as well as for humans. Sentiment analysis is the area of NLP that most often encounters the difficulties presented by sarcasm, where perceived meaning can be the opposite of intended meaning.

Many approaches have been used to counter sarcasm in sentiment analysis [1–4]. Multimodal approaches that account for different data types, such as text, image, and audio, have been developed [5–7]. Other approaches that model a profile of the speaker and account for his or her previous utterances to better understand the meaning that is conveyed have also proven to be successful [8]. Additionally, approaches that model the profile of the audience and its ability to perceive sarcasm have been researched [9,10]. To build upon these previous findings, this paper prioritizes automatic sarcasm-context detection on an established dataset.

Context is an essential condition for sarcasm. The incongruity between an utterance and the context in which it occurs determines it to be sarcastic. Sarcastic context has also been identified as necessary for an utterance to be considered sarcastic [11–13].

Because sarcasm detection has been a highly researched subject in the past decade, only papers that identify context as an important topic are presented and analyzed below. Ghosh et al. [14] carried out some early work on sarcasm detection using conversation context. They used modeled conversations to determine which parts of a phrase established

sarcasm. They also tried to determine which sentence in a multi-sentence message made it sarcastic. Other approaches attempted to include dialogue context in sarcasm detection. This method was applied by Avvaru et al. [15], using transformer-based models. They used the Twitter and Reddit datasets from the Second Workshop on Figurative Language Processing's sarcasm-detection shared challenge to carry out their research. Context, response, and label fields are all included in these datasets. Avvaru et al. trained different models after preprocessing the data: long short-term memory (LSTM) [16], bidirectional LSTM (BiLSTM), stacked LSTM, convolutional neural network LSTM (CNN–LSTM), bidirectional encoder representations from transformers (BERT) [17], and XLNet [18]. For each dataset, BERT was the top performing model, with BERT-5 for Reddit and BERT-7 for Twitter.

Other datasets, such as the Internet Argument Corpus (IAC), were used in similar studies. Eke et al. [19] proposed three methods to improve sarcasm recognition using IAC and two Twitter datasets. The first suggested model combined a BiLSTM architecture with global vectors for word representation [20] (GloVe) for word embedding and context learning. In the second model, Eke et al. employed a pre-trained representation of BERT. The third model used a context-based feature strategy that combined BERT-generated features with traditional machine learning techniques to increase sarcasm-detection performance. In comparison with the baseline models, the suggested methods exhibited improvement.

To improve sarcasm-detection ability, one method proposed context separators [21]. That solution, presented by Dadu and Pant [21], was part of the FigLang2020 workshop. They used a RoBERTa-large model to detect sarcasm and found that adding separation tokens between the context and the target utterance enhanced the model's performance on the Reddit dataset.

Yao et al. [22] tackled Twitter sarcasm detection by mimicking the brain's cognition of sarcasm. Their paper proposed a multimodal, multi-interactive, and multi-hierarchical model that was trained on Twitter, image, image-caption, and text-in-image data. They applied this approach by drawing similarities to the brain's perception of sarcasm, which requires multiple modalities. In addition to sarcasm detection, their model displayed good generalization performance in multimodal emotion recognition and sentiment analysis. Alathur et al. [23] studied the implications of metaphors on electronic participation in healthcare, in the context of the COVID-19 pandemic. They found that a lack of automated tools and region specificity limited the use of metaphors for awareness assessment.

Castro et al. [24] authored the main publication upon which the present paper is based. They proposed MUStARD, a novel multimodal dataset for sarcasm recognition based on a famous comedy series, with the goal of bringing the sarcasm-detection problem closer to real-life situations. The collection consisted of tagged audiovisual utterances, each with its own context. The goal was to incorporate sarcastic indications into the analysis, such as drawn-out syllables, shifts in tone, or straight expressions. When contextual signals are unavailable, humans employ these paralinguistic cues, such as facial expressions and vocal prosody, to communicate and interpret sarcasm. However, research has demonstrated that when there are adequate contextual clues, such paralinguistic cues are not required. Therefore, based on the knowledge in the field and the previous literature, the research question we asked is: *Which of the Deep Learning architectures considered in this paper best performs for Automatic Sarcasm Context Detection on the MUStARD dataset?*

2. Materials and Methods

2.1. Methodology

The goal of the research presented herein was to identify the situations in which sarcasm is used. Therefore, this work was conducted only on the text transcripts of the MUStARD dataset, which will be presented and analyzed in further detail in the next section. For now, we note that it contains 690 observations and that it is balanced, with two classes (true and false). Additionally, to benchmark our best performing model, we used the SARC dataset [25], which is a larger and more popular dataset than MUStARD. SARC was selected because it is one of the most popular datasets for sarcasm detection,

it is publicly available, and most of the data is still available, a quality that most popular sarcasm-detection datasets do not have, as tweets or comments have been deleted, as highlighted in our systematic literature review of the automatic sarcasm-detection task [26].

The text was preprocessed: it was lower-cased, stop words were eliminated, special characters were removed, and the words were stemmed in this phase. All these steps were performed to increase the classifiers' performance.

Each model was tested using stratified 5-fold cross-validation. Due to the small number of cases in the dataset, cross-validation was used. The folds were stratified to ensure that each class was represented evenly, which improved the overall findings and ensured their validity. The text data were then n-gram vectorized in a bag-of-words manner and fed into an SVM classifier to train the baseline model. This model was the baseline for future tests.

TensorFlow and Keras were then used to build and train deep learning models. Multilayer perceptron (MLP), deep neural networks (DNNs), convolutional neural networks (CNNs), long short-term memory (LSTM), bidirectional LSTM (BiLSTM), CNN–LSTM, and attention–LSTM were the architectures that were employed. Transformers, mainly bidirectional encoder representations from transformers (BERT), were employed in addition to the other DNN models. To attain the greatest performance, parameter hyper-tuning and network construction were undertaken after the initial results. Final data were obtained, examined, and discussed once no substantial gains were detected. Future work was laid out, with the major focus being on improving and building upon the achieved results.

2.2. Data

MUStARD is the dataset that was employed in this study, as previously indicated. There were 690 total observations in the dataset. Only the textual data in MUStARD were used in this study. Below is an example from this dataset:

- "utterance": "It's just a privilege to watch your mind at work."
- "speaker": "SHELDON"
- "context": ["I never would have identified the fingerprints of string theory in the aftermath of the Big Bang." "My apologies. What's your plan?"]
- "context_speakers": ["LEONARD", "SHELDON"]
- "show": "BBT"
- "sarcasm": true

with:

- Utterance (String): the target utterance that is analyzed;
- Speaker (String): the speaker of the target utterance;
- Context (List): the sentence exchange prior to the target utterance;
- Context_speakers (List): the speakers of the context sentences;
- Show (String): the show from which the instance is taken;
- Sarcsam (Bool): the label of the target utterance.

Only the context and sarcasm fields from the MUStARD textual data were used. The context "speakers" field was left out of the study, since the goal was to create a system that could detect sarcasm only from knowledge that was obtained in a conversation, rather than modeling speaker profiles and inferring from them. Table 1 shows various information to consider while looking into the context field.

There were 3205 unique words in the context field. The average exchange had 38.69 words, with a standard deviation of 22.45 words, and half of the total exchanges had 35 words or fewer. The minimum number of words in an exchange was 1; in contrast, the maximum of words in an exchange was 123. The average exchange had 4.39 sentences, with a standard deviation of 2.52 sentences. Half of the total exchanges had 4 or fewer sentences and 75% had 6 or fewer. The minimum number of sentences was 1 and the maximum number of sentences was 13. It must be noted that there was no case in which the context was missing, further proving the responsorial nature of sarcasm: it must occur

as a response to a situation or a chain of events. The distribution of context by both words and sentences can be observed in Figure 1.

Table 1. General statistics of the context field.

Statistics	Context
Words	
Mean	38.69
Standard Deviation	22.45
Min	1
25%	21
50%	35
75%	54.75
Max	123
Unique words	3205
Sentences	
Mean	4.39
Standard Deviation	2.52
Min	1
25%	3
50%	4
75%	6
Max	13

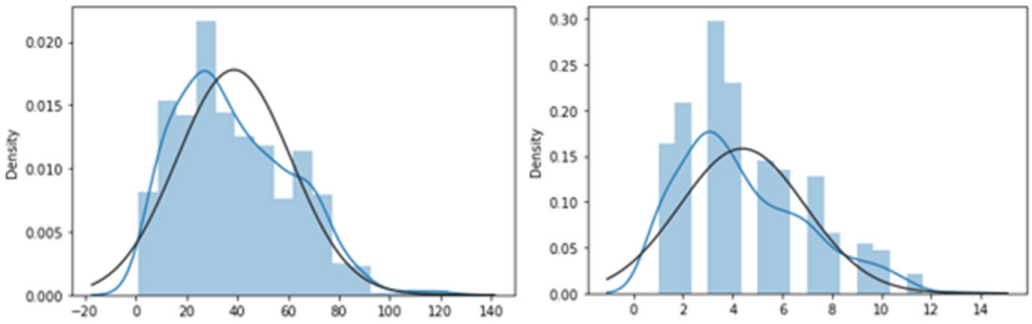

Figure 1. Words' (**left**) and sentences' (**right**) distribution of MUStARD context. Data distribution (blue line) vs Normal distribution (black line).

Most cases, as seen in Table 1, were under 55 words, with the third quartile being 54.75 words. With a skew value of 0.56, the distribution had positive skewness, or skewness to the right. With a kurtosis score of only −0.21, it matched a normal distribution more closely, possibly indicating a weak platykurtic tendency.

Most cases in the sentence distribution were under six sentences, with the third quartile being at this point. The distribution had positive skewness, or skewness to the right, with a skew score of 0.71, which was greater than the skew score for words. The deviation for kurtosis was considerably smaller than that for words, with a score of −0.17. A small platykurtic tendency was seen here, although it was minor. The past statistics will be examined, categorized by the sarcastic label, in Table 2.

There were several distinctions between the two classifications. Sarcastic context had more words on average (41.68) than non-sarcastic context (36.69). However, the standard deviation for non-sarcastic cases was higher, at 23.87 versus 20.53. The lowest, first, and second quartiles were higher for one class than for the other (2 to 1, 27 to 15 and 39 to 31, respectively). Non-sarcastic utterances, on the other hand, gained momentum on the higher spectrum, with the third quartile about identical for the two classes (55 for positive and 54 for negative) and the maximum length in the negative class (123 to 118 for positive).

Table 2. General statistics of the context field, grouped by sarcasm.

Statistics	**Context**	
Label	True	False
	Words	
Mean	41.68	36.69
Standard Deviation	20.53	23.87
Min	2	1
25%	27	15
50%	39	31
75%	55	54
Max	118	123
	Sentences	
Mean	4.8	3.98
Standard Deviation	2.3	2.67
Min	1	1
25%	3	2
50%	4	3
75%	6	6
Max	11	13

In terms of language, the snapshot confirmed what the data revealed: on the lower end, a greater mean and count for the positive class, and on the higher end, a balancing and even an exceeding of the negative class. The negative class's standard deviation was similarly larger. Figure 2 presents the boxplots of the word and sentence counts, split by the sarcasm label, to gain a better understanding of the data.

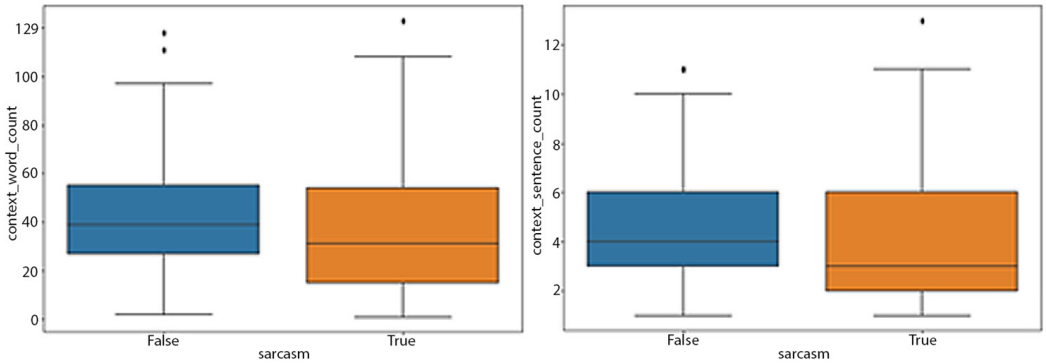

Figure 2. Words (**left**) and sentence (**right**) boxplot, grouped by sarcasm. Dots represent outlier values.

Outlier values were not a cause for worry, as shown by the boxplots. For the sentences count, there were just two outliers, one for each class, and three for the word count, one for the negative class, and two for the positive class. Next, Figure 3 highlights the distribution of word and sentence counts, divided by the sarcasm label.

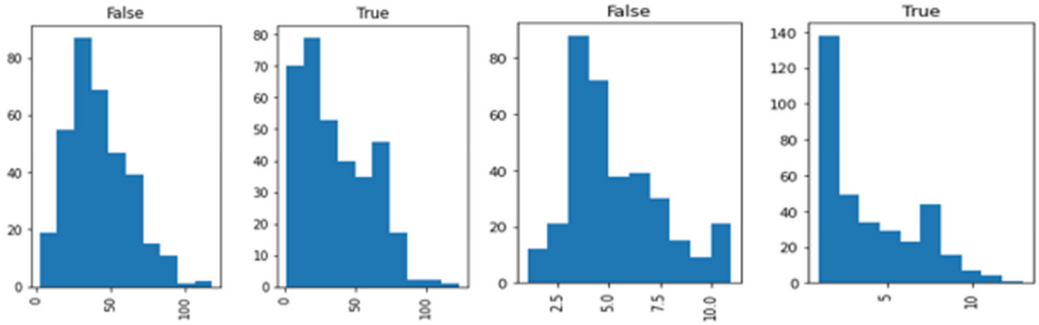

Figure 3. Word (**left**) and Sentences (**right**) distribution, grouped by sarcasm.

It is worth noting that the positive class dominated the bottom end of the word distribution, with shorter examples. This might mean that sarcasm was more likely to occur in a fast response to a single speech or short conversation, and that the chances of sarcasm decreased as the encounter lengthened. The greater values of the negative class near the top end, when the positive class appeared to completely die out, seem to support this argument.

The sentence distribution also revealed that sarcasm appears to be adversely linked with context length. While the negative class was more consistent, the positive class was heavily skewed to the right, with most cases consisting of only two phrases. The Pearson correlation coefficient was evaluated for both word and sentence lengths to see if there was a link between sarcasm and context length.

The negative link between sarcasm and context length is shown by the coefficient indications (-). The likelihood of sarcasm increased as the duration decreased. However, the correlation's strength (0.13, 0.16) indicates that it was poor, but additional research on different datasets might provide fresh findings for sarcasm detection in connection with context or utterance length. The number of words and sentences were positively connected (0.8). With more words consistently translating into more phrases, this was to be expected.

2.3. Preprocessing

The text data must be preprocessed before being used to train and test the classification models, as indicated in Section 2.1. As a result, several actions were required. The text data were handled as follows: all letters were lowered to lower case, all special characters (-, ;, etc.) were deleted, all stop words (a, is, the, etc.) were removed, and the words were stemmed (walking -> walk). Finally, the text was vectorized for processing by the classification algorithms. The resulting vocabulary was reduced from 3205 to 2592 distinct words after preprocessing.

Various tokenization techniques were used. CountVectorizer from the sklearn package was used for the baseline model. This tokenizer turned a collection of text documents into a sparse representation of token counts—a matrix of token counts. Word unigrams were used to tokenize the text.

Word embeddings were utilized in deep learning models. The embeddings were trained alongside the model for the initial batch of models. The GloVe pre-trained word embeddings were employed in the second batch. Specific tokenizers were used for transformer-based models. All sequences were padded to the same length of 100 before training.

2.4. Experiments

The models or architectures used to categorize the data are emphasized in this section. A support vector classifier (SVC) with a Gaussian radial basis function (RBF) kernel function was first utilized. The rest of the parameters were left as default.

The basic deep learning models are next discussed. The binary cross entropy loss function and the Adam optimizer were used to train all of the models for 300 epochs. An MLP with an input layer, two dense hidden layers, and a dense classification layer with a sigmoid activation function was the initial network trained. The neural networks used this model as a starting point. The second model was a basic LSTM network with an input layer, an embedding layer, an LSTM layer, a dense hidden layer, and a dense layer with a sigmoid activation function. The third model was a CNN, which consisted of an input layer, an embedding layer, a convolutional layer, a global max pooling layer, a dense hidden layer, and a dense layer with a sigmoid activation function, which provided the output.

More complex networks were developed using these simple structures. First, a BiLSTM network was created on top of the previous LSTM network. An input layer, an embedding layer, a bidirectional LSTM layer, a hidden dense layer, a dropout layer, and a final dense layer with a sigmoid activation function made up the network. An input layer, an embedding layer, a convolutional layer, a max pooling layer that fed into a bidirectional LSTM layer, a dense hidden layer, and a final layer with a sigmoid activation function

were all integrated into a CNN–LSTM network. An input layer, an embedding layer, four convolution–max pooling–Dropout layers, a dense hidden layer, a dropout layer, and a dense layer with a sigmoid activation function comprised the CNN-large model.

In addition, an attention-based architecture was used. Between the LSTM layer and the dense hidden layer, this network added a basic attention layer, similar to the one previously proposed by Bahdanau et al. [27]. The final layer was identical to the prior two. The GloVe-100D vectors, which were pre-trained word embeddings, were also implemented. The BiLSTM network, the CNN-large network, and the CNN–LSTM network were all trained with them. The transformer architecture was BERT-large uncased.

3. Results

In this section, the trained models' findings are presented and assessed. The baseline benchmark is set first, then the performance of the various models are emphasized under various training situations. Precision, recall, and F1-score were the measures used to evaluate performance. The results of Castro et al. [24] were compared to the base models. Because Castro el al. employed the target utterance for training, which was lacking from the tests done in this study, the comparison was not entirely accurate. The case that was compared was the speaker-independent scenario, in which the authors employed both text and context information to train their model. In the literature, the MUStARD dataset is most used for multimodal sarcasm detection. As such, we were unable to identify other results reported strictly on the textual data from MUStARD. Therefore, we used as a baseline only the results reported by the creators of the dataset, together with the previously mentioned factors.

From Table 3, it can be observed that the best performing model was the Attn–LSTM model, which had the highest recall score of 62.3 and the highest F1 score of 60.1. The SVM model achieved the highest precision score of 61. In addition to the achieved results, other discoveries of interest are discussed.

Table 3. Model's results. Bold highlights the best result for the specific metric (column).

Model	Precision	Recall	F1
Castro et al.	57.9	54.5	54.1
SVC	**61.0**	45.5	48.7
MLP	53.7	56.2	54.6
LSTM	52.5	41.7	45.7
CNN	53.6	45.2	48.7
BiLSTM	51.4	50.1	50.4
CNN-Large	51.6	53.3	52.1
CNN–LSTM	52.0	46.7	49.1
Attn–LSTM	58.4	**62.3**	**60.1**
BiLSTM–GloVe	52.3	49.3	50.6
CNN-Large–GloVe	39.8	40.3	39.9
CNN–LSTM–GloVe	54.3	54.2	53.9
BERT	54.3	54.2	53.9

First, it can be observed that the GloVe embeddings improved the performance of the BiLSTM and CNN–LSTM models, while decreasing the performance of the CNN-large model, which achieved the lowest F1 score of only 39.9. It seems that the embeddings had beneficial effects for LSTM-based architectures and detrimental effects for CNN-based architectures.

Second, the mediocre performance of BERT must also be noted. Transformers have achieved state-of-the-art standing in a plethora of tasks and have displayed superior performance compared with that of other architectures; however, in our experiments, BERT did not achieve the expected results This underperformance could be attributed to the small size of the MUStARD dataset and the transformer's inability to learn from such a reduced sample size.

Next, we benchmarked the performance of the Attn–LSTM model on the SARC dataset, compared with other results reported in the literature. The SARC dataset is a large corpus of 1.3 million sarcastic statements collected from Reddit, along with non-sarcastic statements and the comments they respond to. The performance of the Attn–LSTM model, along with the results reported by Vitman et al. [28], are presented in Table 4.

Table 4. SARC benchmarking. Bold highlights the best result for the specific metric (column).

Dataset	Model	Precision	Recall	F1-Score
SARC–movies	Our model	48.7	**83.2**	61.5
	Vitman et al.	**72.1**	74.4	**73.2**
SARC–technology	Our model	60.7	79.7	68.9
	Vitman et al.	**76.7**	**83.8**	**80.1**

Vitman et al. reported state-of-the-art results on the SARC dataset. It can be seen that their model outperformed our approach on both the movies and technology subsets, with a 73.2 F1 to 61.5 F1 score for movies and an 80.1 to 68.9 F1 score for technology. However, these approaches tackled different tasks: Vitman et al. performed standard sarcasm detection, while our approach performed automatic sarcasm-context detection. Therefore, the difference in performance was not necessarily a downside for our approach; this comparison is better explored in the following section.

4. Discussion

Several major points emerge from these findings. First, LSTM and CNN architectures showed poor performance, which might imply that there were too few instances in the data for these networks to learn and infer excellent outcomes. Furthermore, their low performance might imply that the length of texts was too long for these models to extract the key elements that signal the existence of sarcasm. The strong performance of the Attn–LSTM systems support this claim. The network can scan bigger chunks of text and discover sarcastic triggers in contexts, which are frequently dispersed. The attention layer can prioritize important terms associated with sarcasm, increasing the likelihood of extracting the essential elements needed to accurately characterize the context.

Pre-trained embeddings are also a topic of discussion. While the LSTM-large architecture benefited the most from its implementation, the CNN networks showed a decrease in performance. No clear explanation stems from our research, but future studies can explore this decrease in performance on this task.

BERT's mediocre performance must be addressed. These architectures have dominated the scene in recent years, consistently producing cutting-edge outcomes. One cause might be the small amount of data, especially since transformers are more effective for bigger volumes of data. A debate might be held over the relationship between data and models, with newer solutions depending more on large quantities and constraining AI development's accessibility. This topic, however, is not within the focus of this study and may be addressed in the future.

Finally, the benchmarking on the SARC dataset highlights interesting findings. To better understand the results, the differences between the approaches must be presented. Vitman et al. developed a complex model that used a pre-trained transformer and a CNN to capture context features. Additionally, they used pre-trained transformers on sentiment analysis and emotion-detection tasks. The reported results outperformed state-of-the-art results on the SARC dataset; accordingly, their model was selected for benchmarking with our approach. However, a major difference was the tasks of the papers. Vitman et al. achieved state-of-the-art results for detecting if a target utterance was sarcastic by accounting for the context in which that utterance occurred, while our approach sought to determine if a context was appropriate for sarcasm, did not account for the target utterance at all, and only analyzes the context. Therefore, while the performance of our model was lower than that of the one reported by Vitman et al., it was achieved only on context data,

as it tackled a different task. However, it is interesting to assess the performance of the two approaches by comparison, as such a comparison displays the increased difficulty of sarcasm-context identification when compared with traditional sarcasm detection.

5. Conclusions

After training and comparing the performance of different deep learning models on the MUStARD dataset and then benchmarking the performance of the best performing model with state-of-the-art results on the SARC dataset, the answer to the research question posed at the beginning of this article was achieved, beginning with the observation that the Attn–LSTM model is the best solution among the ones tested for the automatic context-detection task on the MUStARD dataset. The model was best able to identify the salient features of sarcasm context from a small dataset. The increased performance could be attributed to the attention layer that was applied on the LSTM architecture. Furthermore, by comparing the results of our model with the state-of-the-art result achieved on the SARC dataset, we were able to better contextualize the difficulty of the automatic sarcasm-context detection task. It was more difficult to identify whether context was appropriate for sarcasm than to identify whether an utterance was sarcastic, because the salient features of sarcasm itself could have been missing. We encourage future research to pursue the task of automatic sarcasm-context detection, as it could lead to a better understanding of natural language and it could prove to be paramount in developing human-like conversational agents.

There are research avenues that might be further pursued, based on the findings of this study. Further research on the MUStARD dataset, particularly the text data, is one potential possibility. Although the dataset has grown in popularity for multimodal techniques, its usefulness for text-only challenges should not be overlooked. Future research might concentrate on text-only data and build better performing algorithms that can detect sarcasm from single-target utterances, target utterances plus their context, or, as in this study, merely the context to assess if it is suited for sarcasm. The dataset's shortcomings, however, persist, with one major issue being its size.

Second, methods for detecting sarcasm in longer texts can be investigated. Models fail to perform effectively on longer texts, as this study shows, especially for a subtle and sophisticated task such as sarcasm detection. Models fail to recognize the important elements that hint at sarcasm, since they are typically subtle and spread throughout the text. Large CNN architectures and attention-based LSTM architectures are two good places to start. Transformers might potentially be a good option, but they require larger datasets.

The construction of a conversational AI that can detect sarcasm and reply appropriately is the third direction in which research could go. The concept is that by incorporating a performing solution into a conversational model, the agent will be able to identify sarcasm. As a result, it will be able to respond to an utterance's intended meaning and even deploy sarcasm, bringing machine communication skills closer to human-level performance.

Detecting sarcastic context, a novel approach, has proven to be challenging. According to the results, it may be more difficult than the typical sarcasm-detection task. Without the target sarcastic comment and longer texts that might not have an obvious relationship to sarcasm, the models failed to understand and provide excellent outcomes. Several over-achievers, on the other hand, were discovered, setting the groundwork for further inquiry.

Author Contributions: Conceptualization, A.-C.B. and Ș.T.-M.; methodology, A.-C.B.; software, A.-C.B.; validation A.-C.B. and Ș.T.-M.; formal analysis, A.-C.B.; investigation, A.-C.B.; resources, A.-C.B. and Ș.T.-M.; data curation, A.-C.B.; writing—original draft preparation, A.-C.B.; writing—review and editing, A.-C.B. and Ș.T.-M.; visualization, A.-C.B.; supervision, Ș.T.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The authors confirm that all relevant data were included in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abercrombie, G.; Hovy, D. Putting Sarcasm Detection into Context: The Effects of Class Imbalance and Manual Labelling on Supervised Machine Classification of Twitter Conversations. In Proceedings of the ACL 2016 Student Research Workshop, Berlin, Germany, 13–15 June 2016.
2. Ghosh, A.; Veale, T. Fracking Sarcasm Using Neural Network. In Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, San Diego, CA, USA, 16 June 2016.
3. Bouazizi, M.; Otsuki, T. A Pattern-Based Approach for Sarcasm Detection on Twitter. *IEEE Access* **2016**, *4*, 5477–5488. [CrossRef]
4. González-Ibánez, R.; Muresan, S.; Wacholder, N. Identifying sarcasm in Twitter: A closer look. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; short papers; Volume 2.
5. Cai, Y.; Cai, H.; Wan, X. Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
6. Pan, H.; Lin, Z.; Qi, Y.; Fu, P.; Wang, W. Modeling Intra and Inter-modality Incongruity for Multi-Modal Sarcasm Detection. In Proceedings of the EMNLP 2020, Online, 16–20 November 2020.
7. Schifanella, R.; de Juan, P.; Tetreault, J.; Cao, L. Detecting Sarcasm in Multimodal Social Platforms. In Proceedings of the 2016 ACM on Multimedia Conference, ACM, Amsterdam, The Netherlands, 15–19 October 2016.
8. Baruah, A.; Das, K.; Barbhuiya, F.; Dey, K. Context-aware sarcasm detection using BERT. In Proceedings of the 2nd Workshop on Figurative Language Processing, Seattle, WA, USA, 9 July 2020.
9. Bamman, D.; Smith, N. Contextualized sarcasm detection on twitter. In Proceedings of the 9th International AAAI Conference on Web and Social Media, Oxford, UK, 26–29 May 2015.
10. Marwick, A.; Boyd, D. I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media Soc.* **2011**, *13*, 114–133. [CrossRef]
11. Colston, H. Contrast and assimilation in verbal irony. *J. Pragmat.* **2002**, *34*, 111–142. [CrossRef]
12. Ivanko, S.; Pexman, P. Context incongruity and irony processing. *Discourse Process* **2003**, *35*, 241–279. [CrossRef]
13. Ackerman, B.P. Contextual integration and utterance interpretation: The ability of children and adults to interpret sarcastic utterances. *Child Dev.* **1982**, *53*, 1075–1083. [CrossRef]
14. Ghosh, D.; Fabbri, A.; Muresan, S. The Role of Conversation Context for Sarcasm Detection in Online Interactions. In Proceedings of the SIGDIAL 2017 Conference, Saarbrucken, Germany, 15–17 August 2017.
15. Avvaru, A.; Vobilisetty, S.; Mamidi, R. Detecting sarcasm in conversation context using Transformer based model. In Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA, 9 July 2020.
16. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
17. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 3–5 June 2019.
18. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Adv. Neural Inf. Process. Syst.* **2020**, *32*, 5753–5763.
19. Eke, C.I.; Norman, A.A.; Shuib, A.L. Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model. *IEEE Access* **2021**, *9*, 48501–48518. [CrossRef]
20. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the EMNLP 2014, Doha, Qatar, 29 October 2014; pp. 1532–1543.
21. Dadu, T.; Pant, K. Sarcasm detection using context separators in online discourse. In Proceedings of the 2nd Workshop on Figurative Language Processing, Seattle, WA, USA, 9 July 2020.
22. Yao, F.; Sun, X.; Yu, H.; Zhang, W.; Liang, W.; Fu, K. Mimicking the Brain's Cognition of Sarcasm from Multidisciplines for Twitter Sarcasm Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 228–242. [CrossRef] [PubMed]
23. Alathur, S.; Chetty, N.; Pai, R.; Kumar, V.; Dhelim, S. Hate and False Metaphors: Implications to Emerging E-Participation Environment. *Future Internet* **2022**, *14*, 314. [CrossRef]
24. Castro, S.; Hazarika, D.; Pérez-Rosas, V.; Zimmermann, R.; Mihalcea, R.; Poria, S. Towards Multimodal Sarcasm Detection (An Obviously Perfect Paper). In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
25. Khodak, M.; Saunshi, N.; Vodrahalli, K. A Large Self-Annotated Corpus for Sarcasm. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
26. Baroiu, A.; Trausan-Matu, S. Automatic Sarcasm Detection: Systematic Literature Review. *Information* **2022**, *13*, 399. [CrossRef]

27. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
28. Vitman, O.; Kostiuk, Y.; Sidorov, G.; Gelbuks, A. Sarcasm Detection Framework Using Context, Emotion and Sentiment Features. *arXiv* **2022**, arXiv:2211.13014.