**Sarcasm Detection using LSTM**

Aahna Sethi1, Akanksha Sharma1, Ananya Singh Gautam1, Sakshi1, Nisheeth Joshi1,2, Pragya Katyayan1,2[aahnasethi18@gmail.com](mailto:aahnasethi18@gmail.com), [aakankshash1311@gmail.com](mailto:aakankshash1311@gmail.com), [78ananyasingh@gmail.com](mailto:78ananyasingh@gmail.com), [pandeysakshi9203@gmail.com](mailto:pandeysakshi9203@gmail.com) , [nisheeth.joshi@rediffmail.com](mailto:nisheeth.joshi@rediffmail.com), [pragya.katyayan21@gmail.com](mailto:pragya.katyayan21@gmail.com)

1Department of Computer Science, Banasthali Vidyapith, Rajasthan.

2Speech and Language Processing Lab, Centre for Artificial Intelligence, Banasthali Vidyapith, Rajasthan.

**A****bstract:** Sarcasm is subtle and context-dependent, it poses a significant barrier to natural language processing. This work explores the efficacy of using contextual embeddings to identify sarcasm with the goal of revealing the nuanced semantics and temporal dynamics present in sarcastic statements. Through the integration of contextual embeddings with LSTM networks, we are able to leverage sequential modeling and contextual richness to effectively interpret sarcasm in a variety of linguistic contexts. We investigate the cooperative relationship between contextual embeddings and LSTM networks in great detail using a range of empirical analyses and experiments. Our work reveals the intricate mechanisms behind contextual understanding and sequential reasoning in textual data analysis, in addition to achieving state-of-the-art performance in sarcasm detection across multiple datasets. Our results benefit the larger field of natural language processing as well as the field of sarcasm detection. We shed important light on the complex interaction between contextual embeddings and LSTM networks and how these methods can be used to enhance textual data interpretation. Our study demonstrates the potential of contextual embeddings to improve sarcasm identification, with implications for a broad range of sentiment analysis and natural language understanding applications.

**Keywords:** Contextual embeddings, LSTM.

**1 INTRODUCTION**

Sarcasm is the sharp use of language, frequently in a lighthearted manner, to make fun of someone or something. Even while sarcasm isn't always ironic, it can use ambivalence. Satire is more noticeable when spoken, and it mostly depends on the context. It can be identified by the inflection with which it is delivered or, in an ironic way, by how out of proportion the comment is to the scenario. It is a prevalent issue in casual conversations, internet communication, and social media, necessitating robust computer methods for detecting it. Different types of sarcasm, including conversational, deadpan, self-deprecating, ironic, and exaggerated, require unique approaches. Accurate sarcasm identification is crucial for understanding online user interactions, sentiment analysis, and opinion mining. It improves chatbot performance, promotes better human-computer interaction, and enhances customer service efficiency by providing contextually relevant responses to sarcastic remarks.

Figurative language, such as irony and sarcasm, can convey different meanings. Irony is used for humor or to highlight the difference between expectations and reality, while sarcasm conveys mockery or contempt through a caustic tone and deceptive words. Sarcastic irony is a sharper, more direct form of verbal contradiction, aiming to ridicule individuals or circumstances.

Contextual embeddings and recurrent neural networks—like Long Short-Term Memory (LSTM) networks—have emerged recently, opening up new approaches to the problem of sarcasm detection. Compared to typical static word embeddings, contextual embeddings provide a more complex representation of language by encoding word meanings dependent on their surrounding context. Because LSTM networks can represent sequential dependencies in data, they are a good fit for simulating the temporal dynamics found in sardonic expressions.The goal of this study is to determine how well LSTM networks and contextual embeddings function together to identify sarcasm. Our goal is to enhance the precision and resilience of sarcasm detection systems by merging the sequential modeling capability of LSTMs with the contextual richness of embeddings. LSTM networks are especially good at capturing the temporal relationships found in sarcastic utterances because of their capacity to retain and use information across extended sequences.Our method aims to improve the state-of-the-art in sarcasm recognition by capturing the minute language clues and contextual details that distinguish sarcastic remarks. We include a thorough analysis of our methods in this work, together with the architectural setups, training regimens, and experimental outcomes. We showcase the efficacy of our methodology in sarcasm identification in a range of circumstances by showing how it outperforms existing methods across multiple datasets and language conditions. In addition, we examine the underlying processes of sequential reasoning and contextual comprehension in sarcasm detection, offering insights into the intricate nature of sarcastic language and how NLP systems interpret it.

All things considered, this study adds to the expanding corpus of research in NLP and sarcasm detection by presenting a fresh method that makes use of the advantages that contextual embeddings and LSTM networks have to offer. Our findings have implications for a broad range of applications where effective interpretation and decision-making depend on a comprehension of minor nuances in language.

**2 Literature Review**

Riloff et al. (2013) created a sarcasm recognizer using positive sentiment in negative situations from tweets, improving detection. However, it was limited in scope. They recommended identifying stereotypical negative activities and gauging negativity intensity for better detection [1]. Maynard et al. (2013) proposed an approach blending text and multimedia analysis for social media. Combining rule-based text analysis with multimedia techniques, it tackled challenges like noisy text and complex interactions. The research emphasized structured preservation for archivists, aiding in selecting relevant content. It offered insights into sentiment and sarcasm detection, with customizable algorithms for social media nuances. The modular design allows for future advancements in understanding social media dynamics [2].

Ptacek et al. (2014) investigated sarcasm detection in English and Czech tweets, favoring a language-independent approach (F-measure 0.947) and stressing the significance of rich morphology, particularly in Czech. Their research emphasized document-level sarcasm detection via supervised machine learning, extensively evaluating classifiers, features, and preprocessing techniques for both languages [3]. Liu et al. (2014) explored sarcasm in English and Chinese, introducing the Multi-strategy Ensemble Language Approach (MSELA) to tackle class imbalance. Their explicit sarcasm feature set captured unique characteristics in both languages, aiming to automate sarcasm detection and enhance models for diverse textual contexts [4].

Rajadesingan et al. (2015) introduced SCUBA for sarcasm detection on Twitter, blending user behavior analysis with linguistic cues. SCUBA utilized past tweets and behavioral traits to discern sarcasm in unlabeled tweets, showing high efficiency with limited data. Considering psychological and behavioral aspects, it's suitable for real-time applications [5]. Bamman and Smith (2015) emphasized sarcasm's contextual nature, integrating non-linguistic context data from Twitter to enhance detection accuracy. Their model highlighted the author-audience relationship, crucial for sarcasm comprehension, with #sarcasm serving as a communication aid rather than a direct indicator, facilitating audience interpretation [6].

Zhang et al. (2016) explored sarcasm detection in tweets with a neural network approach, using bi-directional GRNN and pooling neural networks. Their model surpassed traditional ones by eliminating manual feature engineering, relying on distributed embeddings and recurrent networks for semantic features [7]. Bouazizi and Ohtsuki (2016) devised a pattern-based method for Twitter sarcasm detection, leveraging sentiment analysis and part-of-speech tags. Their approach focused on pattern-based features, disregarding temporal context, to improve sentiment analysis accuracy [8]. Amir et al. (2016) introduced CUE-CNN, a deep neural network for sarcasm detection, emphasizing speaker context and lexical cues without complex feature engineering. Using convolutional layers, they represented lexical cues and obtained user embeddings, enabling a comprehensive analysis of sarcastic comments on social media [9].

Khodak et al. (2017) introduced the Self-Annotated Reddit Corpus (SARC), containing 1.3 million self-labelled sarcastic statements annotated with "/s". This dataset includes user profiles, topics, and conversation context, providing rich data for sarcasm research. Despite challenges like noise from the "/s" marker, SARC's credibility was validated through meticulous comparisons. Human evaluators outperformed baseline methods, setting the stage for advanced sarcasm detection algorithms [10].

Kolchinski and Potts (2018) investigated the link between author traits and sarcasm recognition in text, employing Bayesian modeling and dense embeddings in a bidirectional RNN to capture author-specific nuances. Using the SARC dataset, they highlighted the effectiveness of these approaches, with Bayesian modeling suited for smaller forums and dense embeddings for larger datasets, emphasizing the importance of considering author-specific traits in language tasks and suggesting further exploration of computational tools for enhanced contextual analysis [11]. Hazarika et al. (2018) addressed challenges in automated sarcasm recognition, presenting CASCADE, a model integrating context-driven discourse modeling with user embeddings, surpassing existing models by capturing both content and contextual cues, thereby highlighting the significance of context and user behavior in sarcasm detection and establishing CASCADE as a premier model in this domain [12]. Ahuja et al. (2018) synthesized research on sarcasm detection in text, emphasizing the significance of user traits, recommending Gradient Boosting as a top classifier, and advocating for ensemble methods, expanded feature sets, and broader platform applicability for future studies [13]. Agarwal and An (2018) proposed Affective Word Embeddings for Sarcasm (AWES), advocating for emotion-infused word representations to enhance sarcasm detection, with sentiment-aware embeddings effective for short texts like tweets and emotion-aware embeddings for longer texts such as product reviews and forum posts, thereby advancing data-driven strategies in this domain [14].

Subramanian et al. (2019) highlighted the importance of emojis in enhancing communication and expressing emotions, advocating for integrating text and emoji signals for improved sarcasm detection. They stressed understanding users' underlying messages and intentions [15]. Liu et al. (2019) introduced A2Text-Net, a deep neural network for sarcasm recognition, outperforming conventional techniques by incorporating auxiliary variables like punctuation and part of speech. Their framework offers flexibility for various deep learning models, benefiting sentiment analysis studies [16]. Castro et al. (2019) introduced the Multimodal Sarcasm Detection Dataset (MUStARD) and advocated for incorporating multimodal cues to enhance sarcasm classification. They emphasized modeling multi-party discussions and suggested exploring speaker localization and advanced neural approaches [17].

Ren et al. (2020) introduced the Sentiment Semantics Enhanced Multi-level Memory Network, highlighting sentiment semantics' importance in sarcasm detection often overlooked in deep learning models. Their approach uses a multi-level memory network to capture sentiment semantics, with one level focusing on sentiment and another on the contrast between sentiment and situation semantics in each sentence. They addressed the lack of local information by incorporating an improved CNN [18]. Potamias et al. (2020) devised a transformer-based approach for detecting irony and sarcasm, tackling figurative language prevalent in social media. Their neural network architecture integrates pre-trained transformer-based models with a recurrent CNN, minimizing data preprocessing and handling complex language nuances without extensive feature engineering. This represents a significant advancement in understanding figurative language in short texts [19].

Yaghoobian et al. (2021) focused on automatic sarcasm detection in written text, discussing methods including Hashtag-Driven Learning, Semi-Supervised Pattern Finding, and Utilizing Extra Information, alongside rule-based approaches. Their study covered single-sentence texts, longer texts with sarcastic sentences, and conversation transcripts, employing deep learning techniques to improve detection accuracy. Razali et al. (2021) integrated deep learning features with contextual features for sarcasm detection in tweets. They combined features from a CNN with handcrafted ones, finding Logistic Regression to be the most effective classifier. Their study utilized five feature sets, enhancing detection accuracy even with less prevalent features like temporality and dislike [20]. Lou et al. (2021) introduced the Affective Dependency Graph Convolutional Network (ADGCN) framework for sarcasm detection, capturing emotional and structural information with affective and dependency graphs. Bidirectional LSTMs and Graph Convolutional Networks were employed to understand sentence context and emotional connections between words [21]. Handoyo et al. (2021) utilized RoBERTa augmented with GloVe embeddings for sarcasm detection on Twitter, achieving better balance and effectiveness across various datasets. Their method improved recognition of non-sarcastic text and demonstrated effectiveness in different sarcasm detection tasks [22]. Eke et al. (2021) proposed a hybrid approach combining deep learning and traditional machine learning for sarcasm detection in text. They experimented with Bi-LSTM with GloVe, BERT, and a Feature Fusion Model, achieving promising results across multiple benchmark datasets [23]. Bedi et al. (2021) introduced MaSaC for Hindi-English code-mixed dialogues and MSH-COMICS for sarcasm detection, utilizing context and language identification [24]. AL-An et al. (2021) while enhanced detection with LSTM and Auto-Encoder techniques, uncovering hidden patterns and relationships in documents for improved accuracy [25].

Baroiu and Matu (2022) reviewed the interdisciplinary evolution of sarcasm detection, addressing challenges in sentiment analysis, particularly in handling figurative language, aiming to provide insights for researchers in selecting suitable approaches within sentiment analysis and NLP [26].

Ren et al. (2023) introduced a knowledge-augmented neural network model for sarcasm detection, utilizing contextual information from an external source, achieving an F1 score of 82.79% on the Semeval-2018 Task 3 dataset, emphasizing the importance of context in detection [27].

**3 Methodology**

**3.1 Corpus Creation:**

Corpus creation is a major step in sarcasm detection because it is necessary to select the proper data for training. There are three types of datasets available on the internet viz short text, long text and other datasets according to Joshi et al. (2017). Short texts are like tweets and threads, which have length limitations set by the platforms. Long texts are like discussion forums and posts and other datasets are like text from novels or tv shows.

We will be using the "News Headlines" dataset, which has an abundance of sarcastic and non-sarcastic headlines. It will be helpful in training and testing the model. It is a dataset of 26,710 lines containing 11,725 sarcastic and 14,985 non-sarcastic news headlines respectively. The model will be trained on the 80% of the dataset and tested on the remaining 20% of the same.

For Example:

**Sarcastic sentence:** *I love working on Sundays.*

**Non-sarcastic sentence:** *I love Fridays*.

Examples from Dataset:

1. HEADLINE: mom starting to fear son's web series closest thing she will have to grandchild

SARCASM: True

2. HEADLINE: j.k. rowling wishes snape happy birthday in the most magical way

SARCASM: False

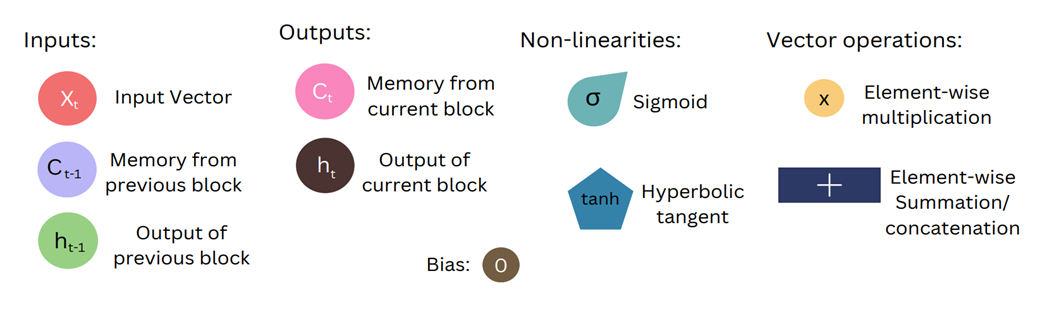
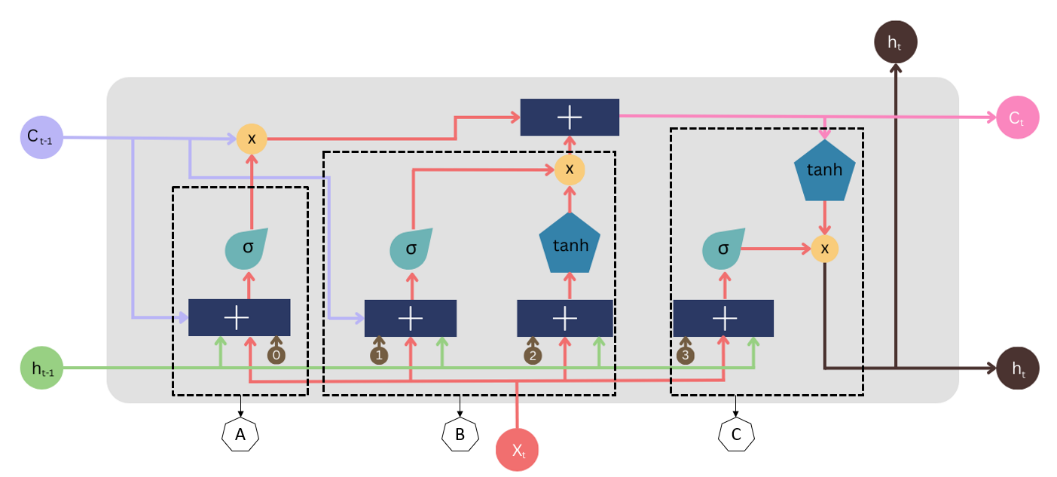
**3.2 LSTM Model:**

Long Short -Term Memory Network is a powerful in AI and deep learning, was created by Hochreiter and Schmidhuber, a type of RNN architecture that addresses the problems like vanishing or exploding gradient problem and allows learning of long-term dependencies in sequential data, which makes them appropriate for tasks such as speech recognition, language translation, and time series forecasting.

The working of LSTM can be understood as, while watching a movie or reading a book sometimes we feel a sense of similarity that we have seen or read in some previous scenes or past chapters. So, similarly we can use this feature for sarcasm detection, if we feed a sentence that is sarcastic while training the model, that information will be remembered for the future use and to predict the nature of sentences while testing the model.

Given below, Fig.1 explains the internal working and the major components of an LSTM cell.

Our LSTM model is divided into three gates that are forget gate, input gate and output gate. The forget gate decides whether the information received from the previous cell is to be remembered or is inapt. In the second step the cell tries to learn new information for input to this cell. And at last, the updated information from the current cell state.

******Fig.1:** LSTM Cell and its components

LSTM consists of 4 layers that interact with each other to produce output of the current cell along with cell state. And these 2 things are passed onto the next hidden layer unlike RNN. It consists of 3 logistic sigmoid gates and tanh layer.  
  
Gates are basically used to limit the information that is passed through the cell. There are 3 types of gates present in LSTM:

### **A) Forget Gate:**

### Information that is not required anymore by the cell state is removed with a forget gate. Xtand ht-1 are fed to the gate and multiplied with weighted matrices followed by addition of bias. And the resultant is passed through an activation function that gives binary output.

### **B) Input Gate:**

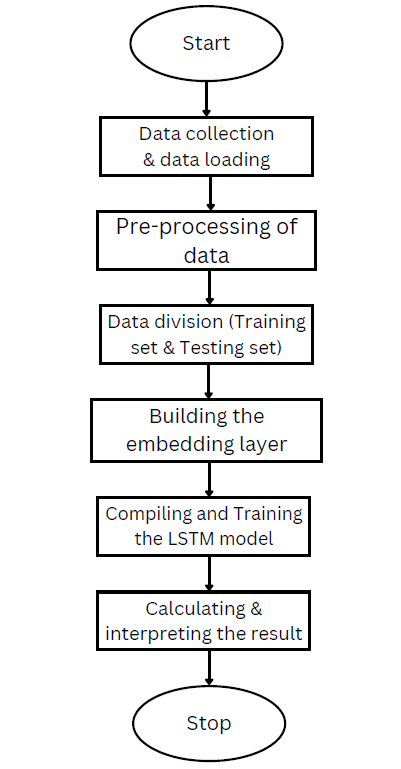
### Now, the addition of useful information is done by the input gate. First Information is regulated using a sigmoid function similar to forget gate, then vector is created using the tanh function that gives output from -1 to 1 values of vector. And regulated values are multiplied to obtain useful information.

**C) Output Gate:**

The task of obtaining useful information from the current cell state to be presented as output is done by output gate. Firstly, tanh function on a cell creates a vector and after that information is regulated using sigmoid function and filtered by values to be remembered, then the value of vector and regulated vector are multiplied to obtain output of current cell and input for another block.

Cell state vector is updated by forgetting information (from forget gate) and addition of information (from input gate).

**3.3 Workflow:** To achieve our objectives, we utilized the methodology represented in Fig.2.

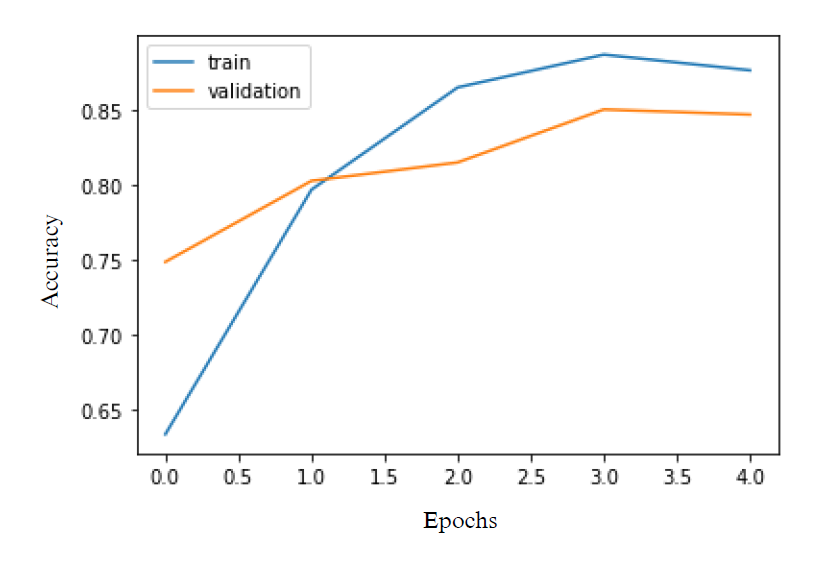


**Fig.2:** LSTM Cell and its components

We start with data collection and loading of all the necessary data. For implementing our model, we install and use TensorFlow. TensorFlow is a rich system for managing all aspects of a machine learning system. Then, we clean the dataset by removing duplicates, handling missing values, standardizing formats and correcting errors. We also use tokenizer to break down a sequence of text into smaller, meaningful units called tokens. Then the punctuations and stop words are removed. The preprocessing of data is then done to improve the quality, usability, and efficiency of the data for the intended task or analysis. Data is then divided into appropriate subsets for effective analysis for training and testing sets. The core of our approach involved embedding the LSTM model using keras function, in which we check if the word is present or not in our vocabulary and then their vector representation is taken in consideration, followed by compilation and training phases for calculating the accuracy of our model. Finally, our result is interpreted. This comprehensive methodology ensured a systematic and thorough approach towards achieving our objectives.

**4 Results and Discussion**

The results of our study demonstrates that our model is 81.95% accurate. Here, we have used news headlines as a dataset. While working with LSTM, it is important to make all the inputs in a fixed size. To achieve this objective, we pad the review sentences. As we know LSTM is better in terms of capturing the memory of sequential information better than simple RNNs. We then compiled and trained our model by fixing the hyperparameters like batch size to 32, epochs to 25 and verbose to 2. Finally, we displayed the model accuracy on test data.

**4.1 Quantitative analysis:**

**Fig.3:** Graph representing the performance of the model

The graph in Fig.3 illustrates the accuracy of our model, with the x-axis denoting number of iterations and the y-axis representing accuracy. The blue line signifies the training phase, while the red line represents testing. Notably, both training and testing accuracies progressively increase over epochs, indicating effective model learning. Efficiency for the training dataset increased steeply over the 1st epoch, while there was a gradual increase for the testing phase. The minimal disparity between training and testing accuracies suggests the absence of overfitting, reflecting the robustness of our approach.

Here, Table 1 shows the comparisons of past works done in this domain, along with their accuracies.

|  |  |  |  |
| --- | --- | --- | --- |
| **Research Papers** | **Authors** | **Models** | **Accuracy** |
| Sarcasm Detection using Contextual Word Embedding with Gaussian Model for Irony Type Identification | Krishnan et. al. (2022) | ELMO | 70% |
| Affective Dependency Graph for Sarcasm Detection | Lou et. al. (2021) | ADCGN | 72.25% |
| A contextual word embedding for Arabic sarcasm detection with random forests | Elagbry et. al. (2021) | BERT Model | 73% |
| CASCADE: Contextual Sarcasm Detection in Online Discussion Forums | Hazarika et. al. (2018) | CUE-CNN | 77% |
| Tweet Sarcasm Detection Using Deep Neural Network | Zhang et. al. (2016) | Deep Neural Network | 78.55% |
| **Sarcasm Detection using LSTM** | **Sethi et. al. (2024)** | **LSTM** | **81.95%** |

**Table 1:** Past Machine Learning models along with their accuracies

Upon comparing our work with other models, it becomes evident that our model has

outperformed the above-mentioned ones.

**4.2 Qualitative analysis:**

**Drawbacks:** Our model at times struggles with detecting sarcasm, leading to comparatively loweraccuracy rates than other models.

For Example:

* Statement: mom starting to fear son&#39;s web series closest thing she will have to grandchild

This is a sarcastic sentence from our dataset but our model detects it as a non-sarcastic sentence.

* Statement: can’t make a simple cup of coffee without everyone freaking out

This is also a sarcastic sentence from our dataset but our model detects it as a non-sarcastic sentence.

**Efficiency:** While our model may have failed in detecting a few sarcastic sentences, it successfullyidentified the majority of sentences with accuracy.

For Example:

* Statement: whale regrets eating 290,000 plastic poker chips that fell off container ship

This is a sarcastic sentence from our dataset and our model also detects it as a non-sarcastic sentence.

* Statement: right to own handheld device that shoots deadly metal pellets at high-speed worth all of this.

This is a sarcastic sentence from our dataset and our model also detects it as a non-sarcastic sentence.

**5 Conclusion and Future Scope**

This paper shows our performed experiment done using LSTM to detect whether a sentence is sarcastic or not. We have mentioned the past works done in this domain and realized that the research has evolved from positive emotions and negative situations to the use of deep neural network models. We further mentioned our goal, emphasizing the desired response of our LSTM model when presented with a sarcastic sentence. We then discussed about our methodology which included corpus creation, the working of our LSTM model and the workflow was then explained. After compiling and training our model we finally calculated the model’s accuracy and displayed it with the help of graph.

In our future endeavors, we plan to explore by employing various models, such as GRU,

BiLSTM, RoBERTa and many more, to compare their accuracies and address the challenges we faced while working with LSTM. Our objective is to achieve our goal by leveraging diverse methodologies and optimizing our approach for success. We commit to keep working on this experiment to refine and advance the model for even greater effectiveness and impact.

**References**

1. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., &amp; Huang, R. (2013, October). Sarcasm as contrast between a positive sentiment and negative situation. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 704-714).
2. Maynard, D., Dupplaw, D., &amp; Hare, J. (2013). Multimodal sentiment analysis of social media. Misra, R., &amp; Arora, P. (2019). Sarcasm detection using hybrid neural networks. arXiv preprint arXiv:1908.07414.
3. Ptáček, T., Habernal, I., &amp; Hong, J. (2014, August). Sarcasm detection on Czech and English twitter. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers (pp. 213-223).
4. Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., &amp; Lei, K. (2014). Sarcasm detection in social media based on imbalanced classification. In Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15 (pp. 459-471). Springer International Publishing.
5. Rajadesingan, A., Zafarani, R., &amp; Liu, H. (2015, February). Sarcasm detection on twitter: A behavioural modelling approach. In Proceedings of the eighth ACM international conference on web search and data mining (pp. 97-106).
6. Bamman, D., &amp; Smith, N. (2015). Contextualized sarcasm detection on twitter. In proceedings of the international AAAI conference on web and social media (Vol. 9, No. 1, pp. 574-577).
7. Zhang, M., Zhang, Y., &amp; Fu, G. (2016, December). Tweet sarcasm detection using deep neural networks. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers (pp. 2449-2460).
8. Bouazizi, M., &amp; Ohtsuki, T. O. (2016). A pattern-based approach for sarcasm detection on twitter. IEEE Access, 4, 5477-5488.
9. Amir, S., Wallace, B. C., Lyu, H., &amp; Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976.
10. Khodak, M., Saunshi, N., &amp; Vodrahalli, K. (2017). A large self-annotated corpus for sarcasm. arXiv preprint arXiv:1704.05579.
11. Kolchinski, Y. A., &amp; Potts, C. (2018). Representing social media users for sarcasm detection. arXiv preprint arXiv:1808.08470.
12. Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., &amp; Mihalcea, R. (2018). Cascade: Contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805.06413.
13. Ahuja, R., Bansal, S., Prakash, S., Venkataraman, K., &amp; Banga, A. (2018). Comparative study of different sarcasm detection algorithms based on behavioural approach. Procedia computer science, 143, 411-418.
14. Agrawal, A., &amp; An, A. (2018, June). Affective representations for sarcasm detection. In The 41st International ACM SIGIR Conference on Research &amp; Development in Information Retrieval (pp. 1029-1032).
15. Subramanian, J., Sridharan, V., Shu, K., &amp; Liu, H. (2019). Exploiting emojis for sarcasm detection. In Social, Cultural, and Behavioral Modeling: 12th International Conference, SBP-BRiMS 2019, Washington, DC, USA, July 9–12, 2019, Proceedings 12 (pp. 70-80). Springer International Publishing.
16. Liu, L., Priestley, J. L., Zhou, Y., Ray, H. E., &amp; Han, M. (2019, December). A2text-net: A novel deep neural network for sarcasm detection. In 2019 IEEE first international conference on cognitive machine intelligence (CogMI) (pp. 118-126). IEEE.
17. Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., &amp; Poria, S. (2019). Towards multimodal sarcasm detection (an \_obviously\_ perfect paper). arXiv preprint arXiv:1906.01815.
18. Ren, L., Xu, B., Lin, H., Liu, X., &amp; Yang, L. (2020). Sarcasm detection with sentiment semantics enhanced a multi-level memory network. Neurocomputing, 401, 320-326.
19. Potamias, R. A., Siolas, G., &amp; Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications, 32, 17309-17320.
20. Razali, M. S., Halin, A. A., Ye, L., Doraisamy, S., &amp; Norowi, N. M. (2021). Sarcasm detection using deep learning with contextual features. IEEE Access, 9, 68609-68618.
21. Lou, C., Liang, B., Gui, L., He, Y., Dang, Y., &amp; Xu, R. (2021, July). Affective dependency graph for sarcasm detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1844-1849).
22. Handoyo, A. T., &amp; Suhartono, D. (2021). Sarcasm detection in Twitter--performance impact while using data augmentation: Word embeddings. arXiv preprint arXiv:2108.09924.
23. Eke, C. I., Norman, A. A., &amp; Shuib, L. (2021). Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model. IEEE Access, 9, 48501-48518.
24. Bedi, M., Kumar, S., Akhtar, M. S., &amp; Chakraborty, T. (2021). Multi-modal sarcasm detection and humor classification in code-mixed conversations. IEEE Transactions on Affective Computing.
25. AL-Ani, M. M., Omar, N., &amp; Nafea, A. A. (2021). A Hybrid Method of Long Short- Term Memory and Auto-Encoder Architectures for Sarcasm Detection. J. Comput. Sci, 17(11), 1093-1098.
26. Băroiu, A. C., &amp; Trăușan-Matu, Ș. (2022). Automatic sarcasm detection: Systematic literature review. Information, 13(8), 399.
27. Ren, Y., Wang, Z., Peng, Q., &amp; Ji, D. (2023). A knowledge-augmented neural network model for sarcasm detection. Information Processing &amp; Management, 60(6), 103521.