JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Sarcasm Detection Using NLP

***Rupali Sharma[1], Shubham Mohite[2], Abhijeet Vadarge, Vaibhav Jaiswal, Mandar Mhatre***

[1]*Department Of Computer Engineering, Pillai College of Engineering (Autonomous), Mumbai University, Mumbai, India.*

[2]*Department Of Computer Engineering, Student, Pillai College of Engineering (Autonomous), Mumbai University, Mumbai, India.*

***Abstract :*** Sarcasm is a form of communication that involves saying something but meaning the opposite. It is a type of irony that is often used to convey a message or express a sentiment in a humorous or critical way. Sarcasm is often used to make a point, express frustration, or to convey humour. It can also be used to mock or ridicule someone or something. Relationships. Spotting sarcasm in language is a big challenge for computers. It's important because it helps them understand our feelings better. To do this, they need to grasp the context and emotion behind our words.. This proposed model will use three components such as C-Net. BERT with Glove embeddings , as an ensemble model, will take contextual information of a sentence in a sequential manner along with the sentiment of the comment, its length, and source to classify it as sarcastic or non-sarcastic. .This model show-cases competitive performance in the Sarcasm.

## I.INTRODUCTION

Sarcasm detection is the task of identifying and interpreting sarcastic language in text or speech. Sarcasm is a form of communication that relies on irony, exaggeration, or understatement to convey a message that is different from or opposite to the literal meaning of the words used. Sarcasm can be challenging to detect because it often requires an understanding of the context, the speaker's tone, and the cultural or social norms surrounding the communication. Sarcasm detection is an important area of research in natural language processing (NLP) because it has many applications, including sentiment analysis, opinion mining, and social media analysis. It also has implications for human communication, such as detecting and interpreting sarcasm in customer service interactions or political debates. There are various approaches to sarcasm detection, including rule-based methods, machine learning techniques, and deep learning models. These approaches use features such as lexical, syntactic, and semantic cues to identify sarcastic language. However, sarcasm detection remains a challenging task in NLP, and there is ongoing research to improve the accuracy and efficiency of sarcasm detection algorithms

## II. LITERATURE SURVEY

### II.I Sarcasm Detection using Context Separators in Online Discourse [ 1 ]

This work presents the importance of context while detecting sarcasm from responses sampled from Twitter and Reddit. Our proposed architecture using three different inputs performs competitively for both datasets showing that the addition of contextual information to target response improves the performance of the fine-tuned contextual word embeddings for detecting sarcasm. We further show that the addition of a separation token between context and target response also performs competitively, markedly showing an improvement of 5.13% in the F1-score of Reddit dataset. Future works include exploring different contextual cues, including user-specific attribute information and extending this hypothesis to other figurative speeches like irony detection and humour detection.

**II.II Sarcasm Detection Using an Ensemble Approach [ 2 ]**

We described an ensemble approach for sarcasm detection in Reddit and Twitter comments. TheModel consists of an ad boost classifier with the decision tree algorithm as base estimator and learns the sarcasm probabilities predicted by four different component models: an LSTM model that uses word, emoji and hashtag representations; a model that uses CNNs and LSTM networks to learn word embeddings, and dense networks to learn punctuation, casing, sentiment and stop word features; an MLP based on Infer sent embeddings; and an SVM approach that captures the stylometric and emotional characteristics of sarcastic content. All component models (except SVM) use conversational context to make predictions, namely the two turns preceding the response.

**II.III "Did you really mean what you said?" : Sarcasm Detection in**

**Hindi-English Code-Mixed Data using Bilingual Word Embeddings [3]**

Social media, in recent years, has become a medium widely used by people for expression of thoughts and opinions, further leading to the realization of tasks like emotion analysis and opinion mining. Sarcastic content in these texts make it even more challenging to figure out the overall sentiment of the text, thus needing proper processing and analysis.

**II.IV Transformers on Sarcasm Detection with Context [4]**

To conclude, we showed an effective method for sarcasm detection and how much context is necessary for it. We didn't use any dataset (reddit and twitter) specific pre-processing or hyperparameter tuning in order to evaluate effectiveness of models across various types of data. In future, we would like to experiment with supplementing external data or merging different types of data on this task.

**II.V C-Net: Contextual Network for Sarcasm Detection [5]**

In this paper, we compared the performances of various approaches for the Sarcasm Detection task. We experimented with traditional machine learning based approaches, and the latest state of the art transformers architectures. For Reference[6]

The results obtained show that our proposed model, C-Net, has the potential to effectively use the conversation context of an utterance to capture the sarcastic nature of a conversation. The variable number of context sentences for each response sentence makes it difficult to capture the long rangedependency. Hence, as future work, approaches that can effectively deal with variable context set size can be investigated.

## III. OBJECTIVES

1. Improve sentiment analysis :- Sarcasm can often flip the sentiment of a statement, and accurately detecting sarcasm can help opinions that may not be expressed directly.

2. Enhance opinion mining:- Sarcasm can be used to express disagreement or criticism in a subtle way, and detecting sarcasm can help identify opinions that may not be expressed directly.
3. Understand social media conversations:- Sarcasm is prevalent in social media conversations, and detecting sarcasm can help understand the underlying meaning of posts and comments.

4. Improve customer service interactions:- Detecting sarcasm in customer service interactions can help identify frustrated customers who may need more attention or support

### III.I SCOPE

The scope of sarcasm detection is quite broad, as it can be applied to various types of text and speech, including social media posts, online reviews, customer support interactions, news articles, and more. Some specific areas where sarcasm detection is used include: Sentiment analysis: Sarcasm can flip the sentiment of a statement, and detecting sarcasm can help improve the overall accuracy of sentiment analysis. Opinion mining: Sarcasm can be used to express disagreement or criticism in a subtle way, and detecting sarcasm can help identify opinions that may not be expressed directly. Social media analysis: Sarcasm is prevalent in social media conversations, and detecting sarcasm can help understand the underlying meaning of posts and comments. Customer service interactions: Detecting sarcasm in customer service interactions can help identify frustrated customers who may need more attention or support. Political discourse analysis: Sarcasm is often used in political discourse to express criticism or irony, and

detecting sarcasm can help understand the true intent behind political statements. Overall, the scope of sarcasm detection is vast and can be applied to various areas where understanding the nuances of human

## IV PROPOSED SYSTEM

### IV.I Existing System Architecture

Sarcasm detection using AI is a challenging task as it involves understanding the context and the nuances of language. While there are various approaches to sarcasm detection using AI, most of them involve a combination of machine learning techniques such as natural language processing (NLP), sentiment analysis, and deep learning algorithms. One existing system architecture for sarcasm detection using AI is based on a combination of NLP and deep learning algorithms. The system involves several components, including text pre-processing, feature extraction, and classification. The text pre-processing component involves cleaning the text data and transforming it into a format that can be fed into the model. The feature extraction component involves extracting relevant features from the text data, such as lexical and syntactic features. The classification component involves training a deep learning algorithm, such as a recurrent neural network (RNN) or a convolutional neural network (CNN), As shown in fig (3.1)to classify the text data as either sarcastic or non-sarcastic. The system may also use a sentiment analysis component to help distinguish between sarcastic and non-sarcastic text. Another approach to sarcasm detection using AI is based on a rule-based system. In this approach, the system uses a set of rules or heuristics to identify sarcastic text. The rules may be based on features such as the use of irony, hyperbole, or rhetorical questions. Overall, sarcasm detection using AI is an active area of research, and there are many different approaches and architectures that have been propose

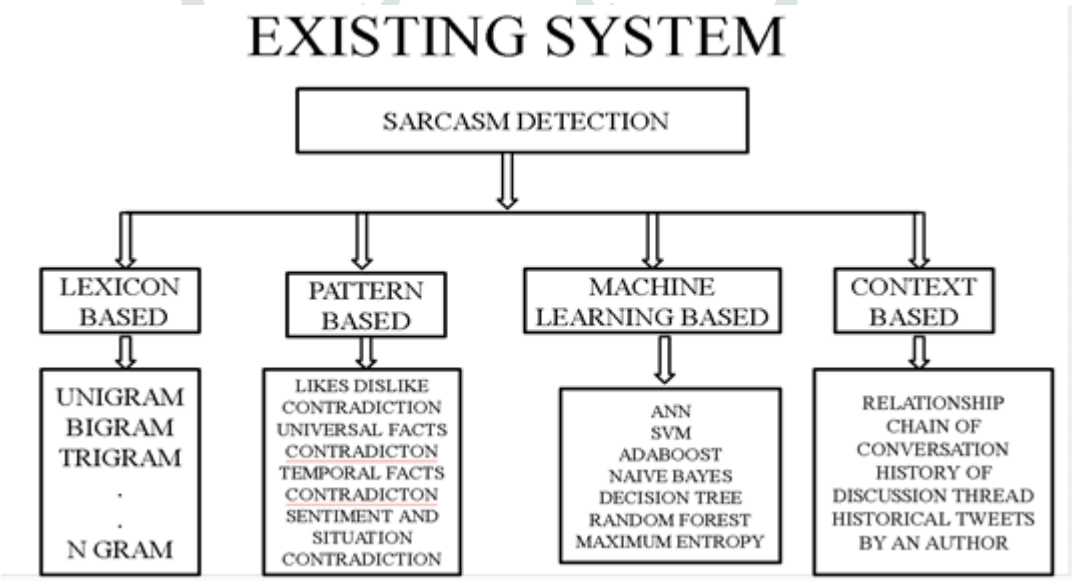



**Fig. I.V.I Existing system architecture used for Content based Systems**

### IV.II Problem Statement

Despite the widespread use of sarcasm in human communication, detecting sarcasm remains a challenging task for machines. The inability to detect **sarcasm** can lead to misinterpretation of messages, causing issues in areas such as sentiment analysis, opinion mining, and social media analysis. Therefore, the objective of this project is to develop an AI-based system that can accurately detect sarcasm in textual data, to improve the accuracy of **sentiment** analysis **and** related applications.

### IV.III Methodology

To develop a methodology for a project on sarcasm detection using AI, you need to define the problem, gather data, pre-process it, perform feature engineering, select an appropriate AI model, train it, test it, fine-

tune it, evaluate its performance using appropriate metrics, deploy it, and regularly check its performance and retrain the model using new data to improve its accuracy over time. You should ensure that the data is diverse and represents the context and tone of various sources. Feature engineering involves selecting or creating relevant features from the text data to help the AI model learn and identify sarcasm. Some popular models for text classification include Naive Bayes, Decision Trees, and Neural Networks. Regularly checking the model's performance and retraining the model using new data to improve its accuracy over time is crucial.

## V. Algorithms / Tools / Techniques

**[1] Natural Language Processing (NLP) libraries** libraries are essential tools for processing text data in natural language processing tasks such as sarcasm detection. They offer a range of functionalities to preprocess, tokenize, and parse text data. Here are some examples:

**Pre-processing:** Pre-processing refers to the tasks that prepare text data for analysis. This includes tasks such as removing punctuation, converting text to lowercase, and removing stop words (words that are commonly used in language but carry little meaning, such as "the" or "and"). This is the initial step in any NLP project, where the raw text data is cleaned and formatted to make it suitable for further analysis. Text pre-processing may involve severalsteps such as removing punctuation marks, converting all text to lowercase, removing stop words (common words such as "the" and "and"), handling contractions (expanding words like "can't" to "cannot"), and correcting spelling mistakes. The goal of text pre-processing is to simplify the text data and make it easier to analyse.

**Tokenization:** Tokenization refers to the process of splitting text data into individual words or tokens. This is an essential step for many NLP tasks, including sarcasm detection. Tokenization is the process of breaking down the text into individual units, or tokens, which can be analysed separately. In its simplest form, tokenization involves splitting the text by whitespace or punctuation marks. However, more advanced tokenization techniques may be used, such as using pre-trained models that can identify multi-word expressions and named entities. Tokenization is an important step in NLP because it allows us to work with the text data at a more granular level and apply further analysis techniques to each individual token.

**Part-of-speech tagging:** Part-of-speech tagging is the process of assigning a part of speech (such as noun, verb, or adjective) to each word in a text. This can be useful for sarcasm detection because it allows the model to understand the role each word plays in the sentence. Part-of-speech (POS) tagging involves assigning each token in the text data a part of speech label, such as noun, verb, adjective, etc. POS tagging is typically performed using pre-trained models that have been trained on large amounts of text data. By assigning a part of speech label to each token, we can better understand the syntactic structure of the text data and apply further analysis techniques that are specific to each part of speech

**Dependency parsing:** Dependency parsing is the process of identifying the relationships between words in a sentence. This can be useful for sarcasm detection because it allows the model to understand the syntax and structure of the sentence. Dependency parsing involves identifying the grammatical relationships between words in a sentence. A dependency parser analyses the text data to identify the subject, verb, object, and other grammatical components of each sentence, and then constructs a tree diagram to show how these components are related to each other. Dependency parsing is an important technique in NLP because it allows us to better understand the structure of sentences and apply more advanced analysis techniques that are specific to the relationships between words. **[2] Deep learning:** Deep learning is a subset of machine learning that uses neural networks with multiple layers to learn complex patterns in data. Here are some examples of deep learning architectures that can be used for sarcasm detection.

**Convolutional Neural Networks (CNNs):** CNNs are commonly used for image classification tasks, but they can also be used for text classification tasks such as sarcasm detection. A CNN works by passing a filter over the text data to extract features at different levels of granularity, such as individual words or phrases.

**Recurrent Neural Networks (RNNs):** RNNs are designed to process sequences of data, such as text or speech. They work by passing information from one-time step to the next, allowing them to capture contextual information. This can be useful for sarcasm detection because it allows the model to understand the relationship between different words in a sentence.

**Long Short-Term Memory (LSTM) networks:** LSTMs are a type of RNN that are designed to remember information over longer periods of time. This can be useful for sarcasm detection because it allows the model to understand the context of a sentence and make use of information .

**V.II Proposed System Architecture**

- A proper proposed system for our project on sarcasm detection using NLP in AI could involve a combination of deep learning techniques, such as recurrent neural networks (RNNs) and attention mechanisms, to capture the complex relationships between words and phrases in the text data.
- The system would start by pre-processing the text data to remove stop words, punctuation, and other irrelevant information. Next, the system would use a pre-trained word embedding model, such as Word2Vec or Glove, to convert each word in the text data into a vector representation that captures its semantic meaning.
- After that, the system would use an RNN to analyse the sequence of words in the text and capture the context of each word. The output of the RNN would then be fed into an attention mechanism that would highlight the most important parts of the text, helping the model to identify the sarcastic elements. Finally, the system would use a SoftMax layer to classify the text as either sarcastic or not sarcastic.

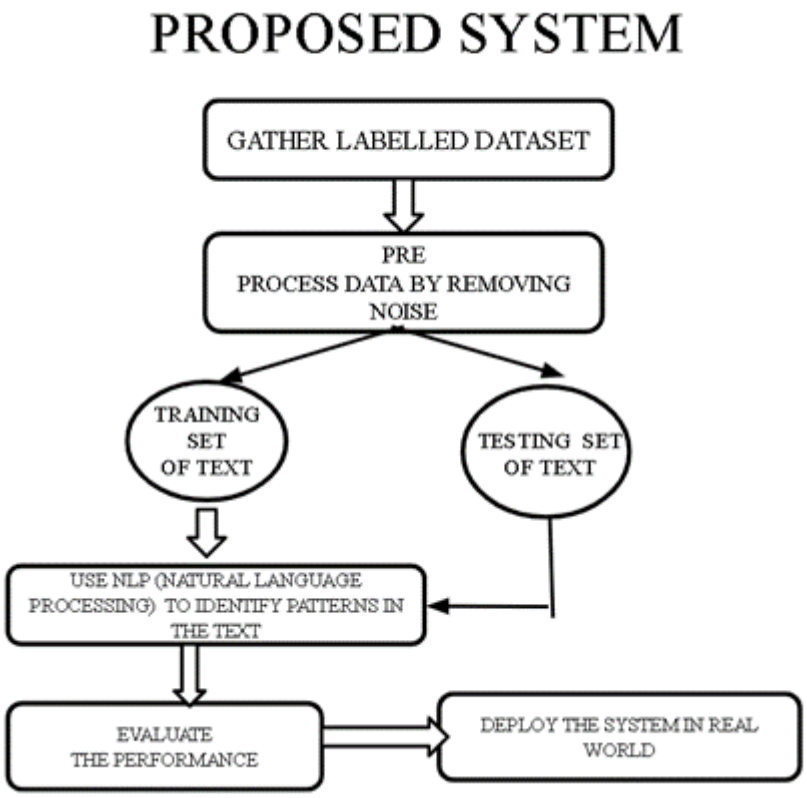



**Fig. II.II Classification of domain techniques**

- **GATHER LABELLED DATASET**

The first step in building a sarcasm detection system is to gather a labelled dataset. This means that we need a large number of text examples that are labelled as either sarcastic or non-sarcastic. We can obtain these examples by manually annotating a dataset or by using existing annotated datasets.

- **PREPROCESSING THE DATA BY REMOVING NOISE**

After collecting the dataset, we need to pre-process the data by removing noise, such as special characters, punctuation marks, and stop words. (As shown in fig 3.2) This step will ensure that the data is clean and ready for analysis.

- **USE NATURAL LANGUAGE PROCESSING TO IDENTIFY PATTERNS IN THE TEXT**

Once the data has been pre-processed, we can use NLP techniques to identify patterns in the text that are indicative of sarcasm. We can use techniques like sentiment analysis, part-of-speech tagging, and named entity recognition to extract features from the text that are relevant to sarcasm.

❖ **USE NATURAL LANGUAGE PROCESSING TO IDENTIFY PATTERNS IN THE TEXT**

Once the data has been pre-processed, we can use NLP techniques to identify patterns in the text that are indicative of sarcasm. We can use techniques like sentiment analysis, part-of-speech tagging, and named entity recognition to extract features from the text that are relevant to sarcasm.

❖ **USE NATURAL LANGUAGE PROCESSING TO IDENTIFY PATTERNS IN THE TEXT**
Once the data has been pre-processed, we can use NLP techniques to identify patterns in the text that are indicative of sarcasm. We can use techniques like sentiment analysis, part-of-speech tagging, and named entity recognition to extract features from the text that are relevant to sarcasm.

❖ **DEPLOY THE SYSTEM IN REAL WORLD:**

Once we are satisfied with the performance of the sarcasm detection system, we can deploy it in the real world. This could involve integrating the system into social media platforms, (As shown in fig 3.2) chatbots, or other applications where sarcasm detection could be useful. It is important to continuously monitor and update the system to ensure that it remains accurate and effective over time.

## APPLICATIONS AND RESULTS

### VI.I APPLICATIONS

There are various applications of this domain system. The application is listed here.

1. Social Media Monitoring: Social media platforms like Twitter, Facebook, and Instagram are filled with sarcastic comments, which can often be misunderstood by brands and companies. By using AI to detect sarcasm, businesses can gain a better understanding of how their customers perceive their products and services, and make more informed decisions based on that information. Sentiment Analysis: Sarcasm can often be used to express a sentiment opposite to the literal meaning of the words used. In sentiment analysis, detecting sarcasm can help accurately identify the sentiment of a piece of text.

2. Social Media Monitoring: Social media platforms such as Twitter and Facebook are often filled with sarcastic remarks. Sarcasm detection can help companies monitor social media conversations about their products and services, and understand how people feel about them.

3.Sarcasm detection can help chatbots identify when a customer is being sarcastic or annoyed and provide appropriate responses.

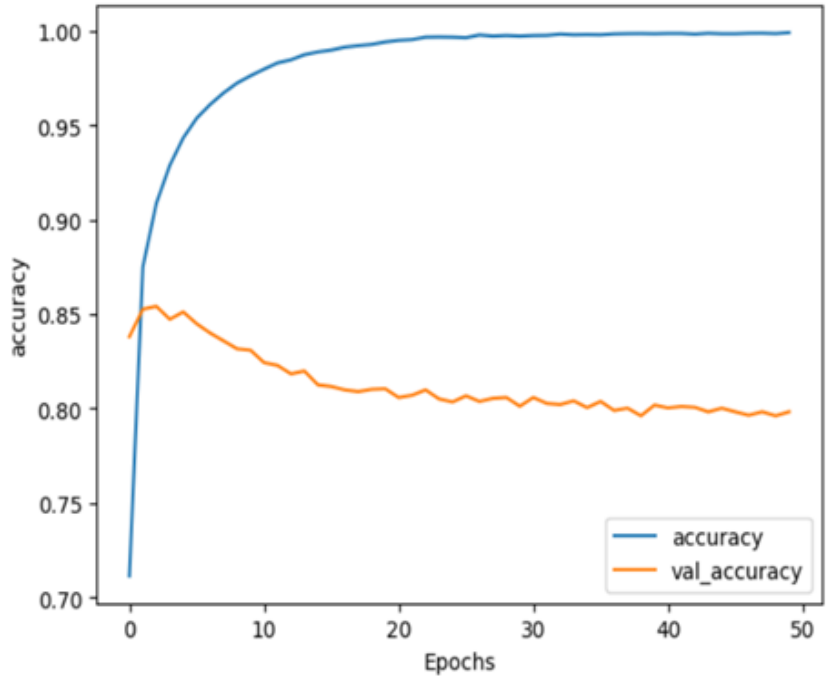
4.Political Analysis: Sarcasm is often used in political discourse to criticize opponents. Sarcasm detection can help political analysts understand the true meaning of political statements.

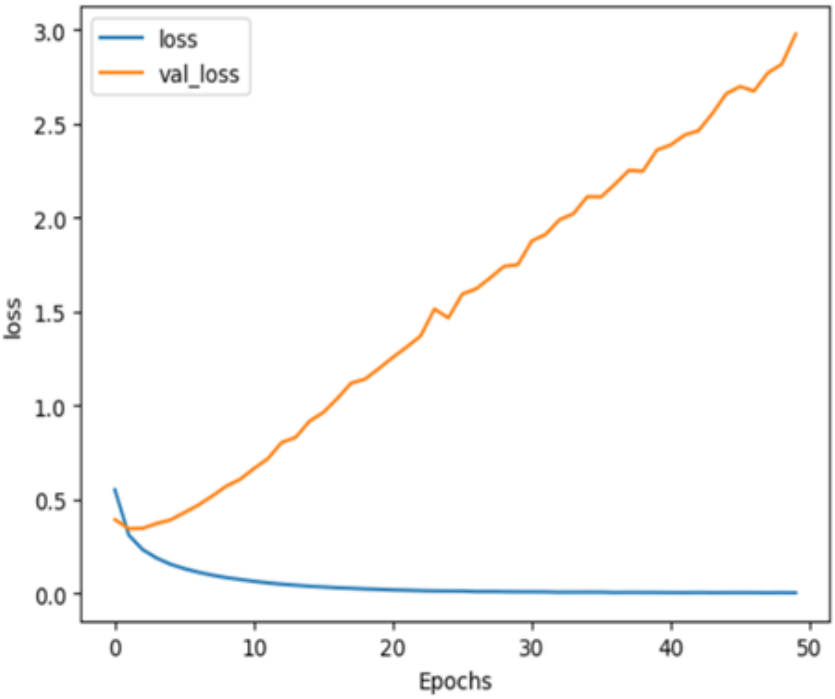
5.Humor Generation: Sarcasm is often used in humor, and sarcasm detection can help generate more accurate and appropriate jokes.

6.Market Research and Brand Perception: Companies can use sarcasm detection to analyse market research data and consumer feedback. It can provide insights into brand perception and help in making informed business decisions

**V.II RESULTS**

**ACCURACY GRAPH**



**FOR NOT SARCASTIC**

```
sentence = ["I had a great time at the party last night"]
sequences = tokenizer.texts_to_sequences(sentence)
padded = pad_sequences(sequences, maxlen=max_length, padding=padding_type, truncating=trunc_type)
print(model.predict(padded))
```

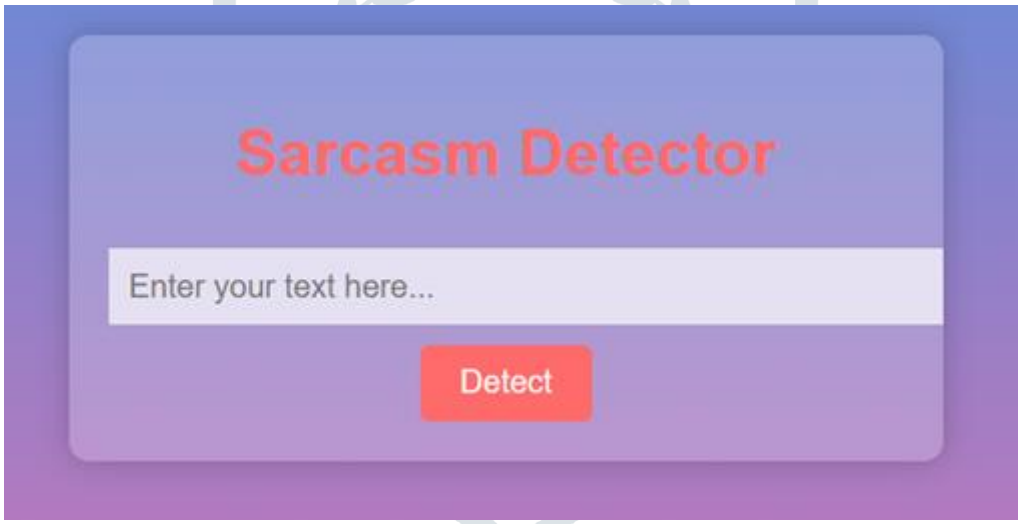
```
1/1 [==============================] - 0s 22ms/step
[[2.5876796e-08]]
```

**FOR SARCASTIC**

```
sentence = ["Oh, great. Another Monday to look forward to"]
sequences = tokenizer.texts_to_sequences(sentence)
padded = pad_sequences(sequences, maxlen=max_length, padding=padding_type, truncating=trunc_type)
print(model.predict(padded))
```

```
1/1 [==============================] - 0s 20ms/step
[[0.9999951]]
```

**WEBPAGE:**




**V.III Summary**

A project on sarcasm detection using AI involves the use of natural language processing (NLP) techniques to pre-process and analyse text data in order to identify features that are indicative of sarcasm. The goal of this project is to train an AI model that can accurately detect sarcasm in text data, which can be used in a variety of applications such as sentiment analysis, social media monitoring, and customer service chatbots.

The project typically involves several steps, including data collection, text pre-processing, feature extraction, model training, and model evaluation. Data collection involves gathering a large dataset of text data that contains examples of sarcastic language, which can be obtained from a variety of sources such as social media, news articles, and online forums. Text pre-processing involves cleaning and formatting the text data to make it suitable for further analysis, which may involve techniques such as removing stop words, correcting spelling mistakes, and tokenization.
Feature extraction involves identifying linguistic features in the text data that are indicative of sarcasm, which may include the use of irony, hyperbole, and other types of figurative language. This may involve using NLP techniques such as part-of-speech tagging and dependency parsing to identify specific linguistic patterns in the text data. These features are then used to train an AI model, which can be done using a variety of machine learning algorithms such as logistic regression, decision trees, or neural networks.

Once the AI model has been trained, it is evaluated on a test dataset to determine its accuracy in detecting sarcasm. Model evaluation typically involves measuring metrics such as precision, recall, and F1 score, which provide a quantitative measure of

the model's performance. The goal of the project is to develop an AI model that can accurately detect sarcasm in text data with a high degree of accuracy, which can be used in a variety of applications to improve communication and understanding between humans and machines.

**REFERENCES**


- Razali, Md Saifullah, et al. "Sarcasm Relation to Time: Sarcasm Detection with Temporal Features and Deep Learning." (2022).
- Razali, Md Saifullah, et al. "Sarcasm detection using deep learning with contextual features." IEEE Access 9 (2021): 68609-68618.
- Jena, Amit Kumar, Aman Sinha, and Rohit Agarwal. "C-net: Contextual network for sarcasm detection." Proceedings of the second workshop on figurative language processing. 2020.
- Kumar, Akashi, Shubham Dikshit, and Victor Hugo C. Albuquerque. "Explainable artificial intelligence for sarcasm detection in dialogues." Wireless Communications and Mobile Computing 2021 (2021): 1-13
- Razali, Md Saifullah, et al. "Sarcasm Relation to Time: Sarcasm Detection with Temporal Features and Deep Learning." (2022).
- Ajayan, A. K., Mohanan, K., Anugraha, S., Premjith, B., & Kp, S. (2022, July). Amrita_CEN at SemEval-2022 Task 6: A Machine Learning Approach for Detecting Intended Sarcasm using Oversampling. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022) (pp. 834-839).
- Kumar, Akshi, Shubham Dikshit, and Victor Hugo `C. Albuquerque. "Explainable artificial intelligence for sarcasm detection in dialogues." Wireless Communications and Mobile Computing 2021 (2021): 1-13.
- Razali, Md Saifullah, Alfian Abdul Halin, Lei Ye, Shyamala Doraisamy, and Noris Mohd Norowi. "Sarcasm detection using deep learning with contextual features." IEEE Access 9 (2021): 68609-68618.
- Ashwitha, A., G. Shruthi, H. R. Shruthi, Makarand Upadhyaya, Abhra Pratip Ray, and T. C. Manjunath. "Sarcasm detection in natural language processing." Materials Today: Proceedings 37 (2021): 3324-3331.
- Vinoth, D., & Prabhavathy, P. (2022). An intelligent machine learning-based sarcasm detection and classification model on social networks. The Journal of Supercomputing, 78(8), 10575-10594.
- Godara, Jyoti, Rajni Aron, and Mohammad Shabaz. "Sentiment analysis and sarcasm detection from social network to train health-care professionals." World Journal of Engineering 19.1 (2022): 124-133.
- Yuan, Y., & Lam, W. (2022, February). Sentiment Analysis of Fashion Related Posts in Social Media. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (pp. 1310-1318).
- Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), 5731-5780.
- Băroiu, Alexandru-Costin, and Ștefan Trăușan-Matu. "Automatic Sarcasm Detection: Systematic Literature Review." Information 13.8 (2022): 399.
- Mahdaouy, A. E., Mekki, A. E., Essefar, K., Skiredj, A., & Berrada, I. (2022). Cs-um6p at semeval-2022 task 6: Transformer-based models for intended sarcasm detection in english and arabic. arXiv preprint arXiv:2206.08415.
- Zhang, Y., Ma, D., Tiwari, P., Zhang, C., Masud, M., Shorfuzzaman, M., & Song, D. (2022). Stance Level Sarcasm Detection with BERT and Stance-Centered Graph Attention Networks. ACM Transactions on Internet