Natural Learning Processing: DATS 6312

Individual Report

Sarcasm Detection Project

Instructor: Amir Jafari

Presented By Group 6:

Tanmay Kshirsagar

Date: 11th Dec 2023

Table of Contents

1. Introduction

The challenge of detecting sarcasm in textual content is a fascinating and complex problem in the realm of Natural Language Processing (NLP).

- Contextual Understanding and Misinterpretation: Sarcasm poses a significant challenge in news headlines due to its nuanced and context-dependent nature. The subtle interplay of language in sarcastic headlines can lead to misinterpretations, potentially causing readers to misunderstand the intended message. Detecting sarcasm becomes crucial to ensure accurate comprehension and prevent the spread of misinformation or unintended messages in news reporting.
- Maintaining Credibility and Trustworthiness: Sarcasm, when undetected, can undermine the credibility and trustworthiness of news sources. Readers may perceive sarcastic statements as factual, leading to misinformation and potentially influencing public opinion. The need for sarcasm detection in news headlines stems from a broader concern for maintaining the integrity of journalistic content, ensuring that headlines accurately reflect the intended tone and meaning.
- Mitigating Negative Social Impact: Sarcasm in news headlines, if not identified, can contribute to the dissemination of biased or inflammatory information. Incorrectly interpreting sarcastic content may lead to unwarranted emotional reactions, social polarization, or even the propagation of false narratives. Detecting sarcasm in news headlines is, therefore, essential for mitigating negative social impact and fostering a more informed and discerning readership.

Our project aims to tackle this issue by focusing on sarcasm detection in news headlines. Sarcasm detection is crucial because it helps in understanding the underlying sentiment and tone in written language, which can significantly differ from the literal meaning of the words used. In addition to sarcasm detection, our project explores the application of text summarization techniques to distill key information from sarcastic news articles. This comprehensive analysis involves utilizing various NLP techniques and models to distinguish between sarcastic and non-sarcastic news headlines, providing a holistic approach to understanding and processing sarcastic textual content.

2. Dataset Description

We selected the "News Headlines Dataset For Sarcasm Detection" available on Kaggle. This dataset is particularly suited for our purpose due to its structure and content:

- Volume and Source: It comprises 55,328 news headlines with corresponding articles. This dataset was carefully compiled from two distinct news websites to minimize the noise and ambiguity often found in similar datasets from social media platforms like Twitter.
- Composition and Reliability: The sarcastic headlines are sourced from TheOnion, a website known for its satirical portrayal of current events. These headlines are mainly from its "News in Brief" and "News in Photos" categories. On the other hand, the non-sarcastic headlines are collected from HuffPost, providing a balance with real-world, serious news content.
- Attributes: Each entry in the dataset is characterized by three attributes:
  - is_sarcastic: A binary indicator, where 1 denotes a sarcastic headline and 0 denotes a non-sarcastic headline.
  - headline: The text of the news headline.
  - article_link: A URL linking to the original news article.
- Data Enrichment: To add depth to our analysis, we have extended the dataset by scraping the main content of the articles using the BeautifulSoup Python package. This allows us to compare the headlines with their corresponding articles, providing a more nuanced understanding of sarcasm in news media. This enhanced dataset is hosted in the cloud, facilitating efficient access and analysis.
- Web Scraping: We also scraped the URLs available in our dataset where is_saracastic has value 1. The actual news text was saved and used for Text Summarization.

3. Description of Individual Work

3.1. Proposed System

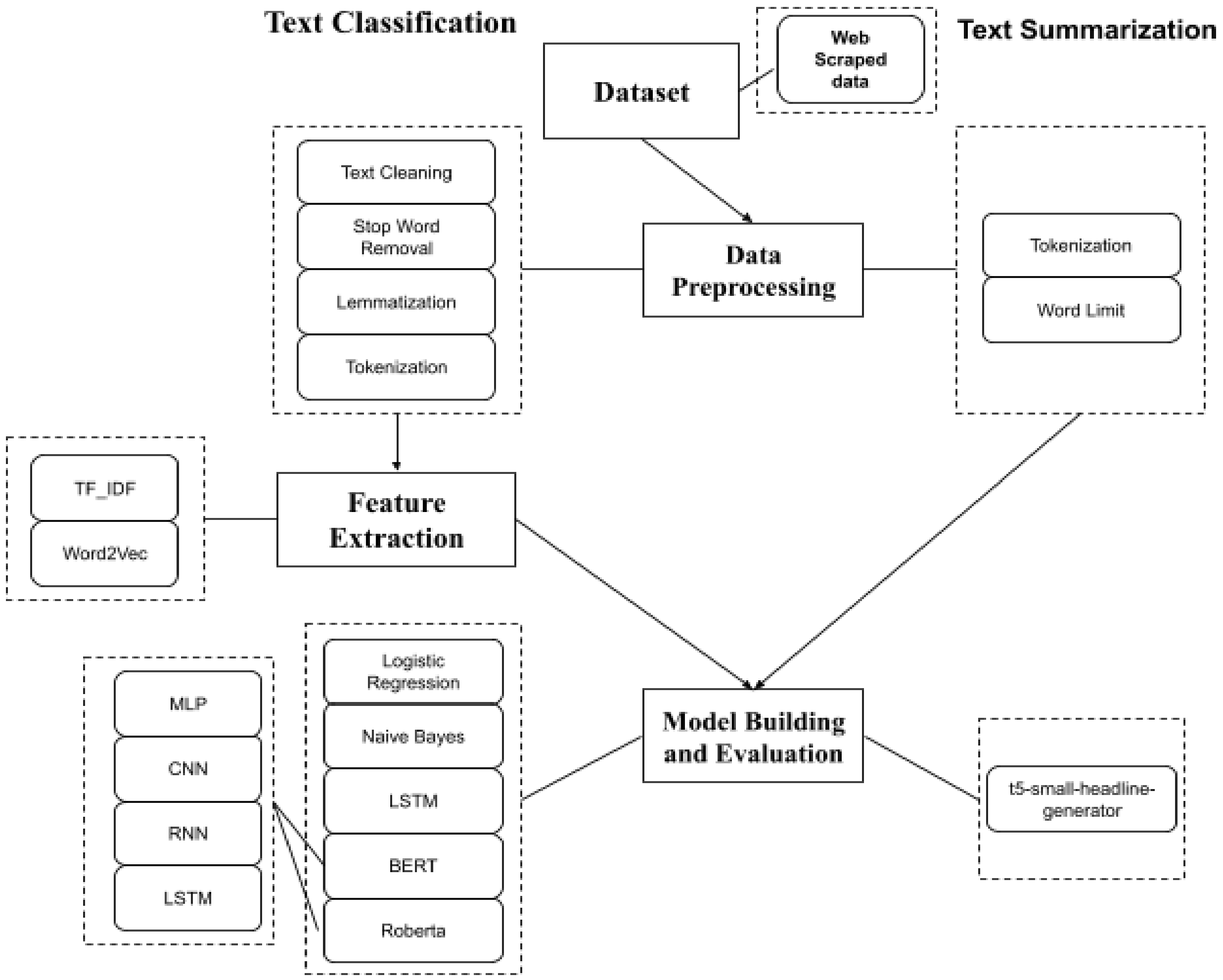


Figure 1 : Proposed System

Our Natural Language Processing project aims to develop algorithms capable of classifying and summarizing large volumes of text data efficiently. We employ a robust pipeline that begins with scraping for available news text from the URLs available in the dataset and then meticulous data preprocessing to ensure data quality, followed by sophisticated feature extraction methods to capture the essence of the text.

- Data Preprocessing: The initial phase involves cleansing the dataset by removing stop words, applying lemmatization to reduce words to their root form, and tokenizing the text, which breaks it down into individual units for analysis.

- Feature Extraction: We leverage both traditional and advanced methods to extract features from the text. Techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) and Word2Vec are used to transform text into a format that can be easily interpreted by our models. These features serve as the input for the subsequent modeling phase.
- Model Building and Evaluation: Our approach explores a variety of models to determine the most effective for our objectives. For text classification, we experiment with models ranging from Logistic Regression and Naive Bayes to more sophisticated neural network architectures like LSTM (Long Short-Term Memory) and transformer-based models such as BERT and Roberta. For text summarization, we leverage the t5-small-headline-generator, which is designed to condense information into concise headlines.

Throughout the project, we continuously refine our models and evaluate their performance to ensure high accuracy and reliability in classifying and summarizing text, which could be pivotal for applications in content analysis, sentiment detection, and information retrieval.

3.2. Text Summarization

To ensure effective training, the dataset undergoes preprocessing first by using tokenization of the t5-small model. Imposing a word limit on input text for text summarization models is a pragmatic approach to address computational constraints and enhance the coherence of generated summaries. Thus, articles are filtered based on a word limit of 30 words to manage the complexity of the summarization task.

3.3. T5

T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task. T5-small is a variant of the T5 model, specifically representing a smaller and computationally lighter version. Despite its reduced size, T5-small retains the core architecture and capabilities of T5, making it suitable for applications with limited computational resources like this.

3.4. Evaluation metrics

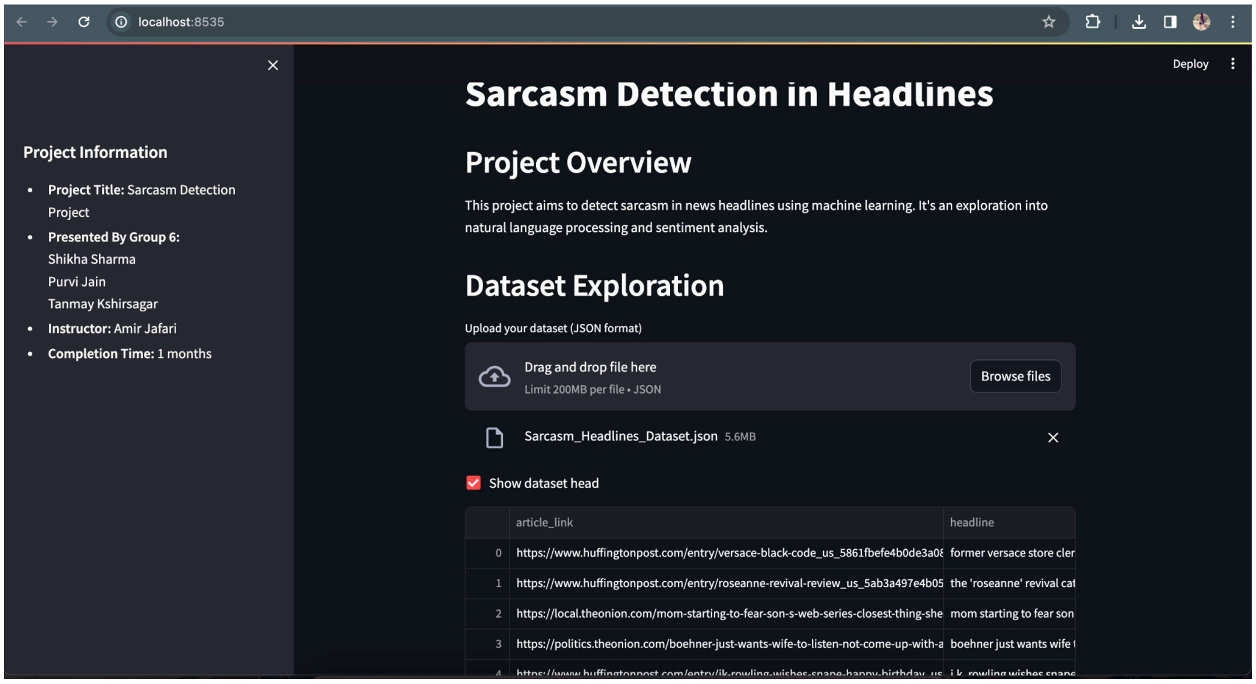
For model evaluation, various metrics were strategically employed to assess the performance and characteristics of the trained summarization model. Train accuracy and test accuracy provided insights into the model's ability to correctly predict headlines during both the training and evaluation phases, respectively. The number of epochs and batch size were crucial parameters monitored during training, offering a glimpse into the model's convergence and computational efficiency. Train loss, a measure of the model's error during training, served as an indicator of its learning progress over time. Learning rate, a key hyperparameter, was optimized to regulate the magnitude of updates during training. The choice of the optimizer, in this case, AdamW or Adam, influenced the optimization process. These metrics collectively formed a comprehensive evaluation framework, enabling a thorough understanding of the model's training dynamics, generalization performance, and parameter tuning effectiveness. Additionally, for text summarization, the ROUGE metric is employed to assess the effectiveness of the baseline on the validation dataset. ROUGE scores provide insights into the quality of the generated summaries concerning the ground truth. ROUGE-1 measures unigram overlap, ROUGE-2 evaluates bigram overlap, ROUGE-L assesses the longest common subsequence, and ROUGE-Lsum considers unigrams, bigrams, and longest common subsequence in automatic text summarization evaluation.

3.5. SHAP

SHAP (SHapley Additive exPlanations) is a model-agnostic framework for interpreting the output of machine learning models. It assigns fair contributions to individual input features, allowing for a nuanced understanding of a model's decision process. By providing feature-level insights, SHAP values help quantify the impact of each feature on model predictions, facilitating enhanced interpretability. This method is applicable across various model architectures, making it versatile for both simple and complex models. Importantly, SHAP values aid in addressing the "black-box" nature of certain models, offering transparency and explanations for specific predictions. The framework is widely used to gain insights into feature importance and enhance trust in machine learning models.

3.6. Streamlit

In our project, we developed a Streamlit application focused on detecting sarcasm in news headlines, leveraging machine learning and natural language processing. The application encompasses several integral features: This introductory section outlines the application's objective to analyze and identify sarcasm in news headlines. Users have the flexibility to upload datasets in JSON format. The application includes real-time text summarization and sarcasm prediction, allowing users to input text for immediate analysis and results, showcasing the application's practical application in processing and understanding natural language data.



Project Information

- **Project Title:** Sarcasm Detection Project
- **Presented By Group 6:** Shikha Sharma Purvi Jain Tanmay Kshirsagar
- **Instructor:** Amir Jafari
- **Completion Time:** 1 months

Sarcasm Detection in Headlines

Project Overview

This project aims to detect sarcasm in news headlines using machine learning. It's an exploration into natural language processing and sentiment analysis.

Dataset Exploration

Upload your dataset (JSON format)

Drag and drop file here
Limit 200MB per file • JSON
Browse files

Sarcasm_Headlines_Dataset.json 5.6MB

[✓] Show dataset head

	article_link	headline
0	https://www.huffingtonpost.com/entry/versace-black-code_us_5861fbefe4b0de3a08	former versace store cler
1	https://www.huffingtonpost.com/entry/roseanne-revival-review_us_5ab3a497e4b05	the 'roseanne' revival cat
2	https://local.theonion.com/mom-starting-to-fear-son-s-web-series-closest-thing-she	mom starting to fear son
3	https://politics.theonion.com/boehner-just-wants-wife-to-listen-not-come-up-with-a	boehner just wants wife
4	https://www.huffingtonpost.com/entry/jk-rowling-wishes-snape-happy-birthday_us	j.k. rowling wishes snape

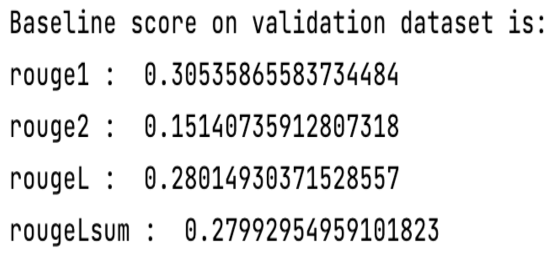
Figure 2: Streamlit app demo

4. Results

The dataset is split into training, validation, and test sets in 80%, 10%, and 10% splits respectively. Subsequently, the aforementioned models were trained on the training data and evaluated on validation data.

4.1. Text Summarization

Before delving into the model training, a baseline evaluation is performed using a lead-1 summarization approach. The dataset is tokenized using the T5 (Text-to-Text Transfer Transformer) model tokenizer, a state-of-the-art model for sequence-to-sequence tasks. The model used for headline generation is a fine-tuned version of t5-small. Specifically, it is based on the JulesBelveze/t5-small-headline-generator model, which is pre-trained for headline generation using the tldr_news dataset.
The T5 model is fine-tuned on the sarcastic news articles using the AdamW optimizer with a specified learning rate. The training process involves multiple epochs, and a linear learning rate schedule is employed to gradually adjust the learning rate during training. The performance of the trained model is evaluated on the validation set using the ROUGE metric. ROUGE scores are tracked over epochs to observe the model's convergence and identify potential areas for improvement.



```
Baseline score on validation dataset is:
rouge1 :  0.30535865583734484
rouge2 :  0.15140735912807318
rougeL :  0.28014930371528557
rougeLsum :  0.27992954959101823
```

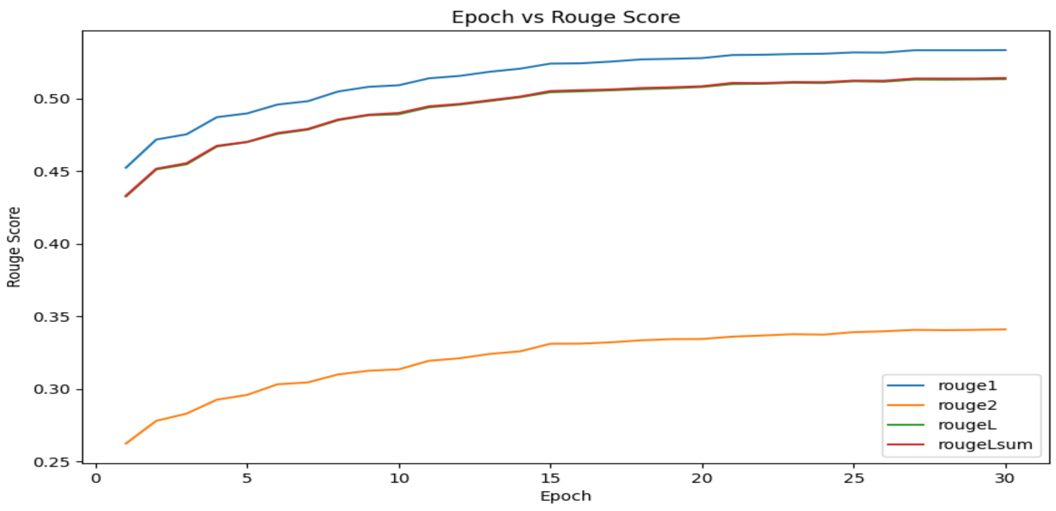
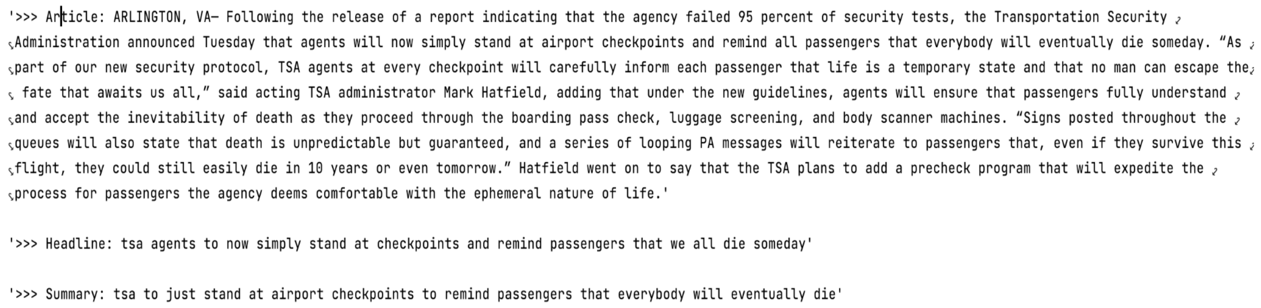


Figure 3: Epoch vs Rouge Score for Summarization model

Here we can see that the ROUGE scores improve as the model is trained over the epochs. The ROUGE scores around 0.50 are considered to be good. Thus, we can say that our text summarization model is appropriately trained. Here we can see that the summary created by the model is pretty close to what the actual headline is.



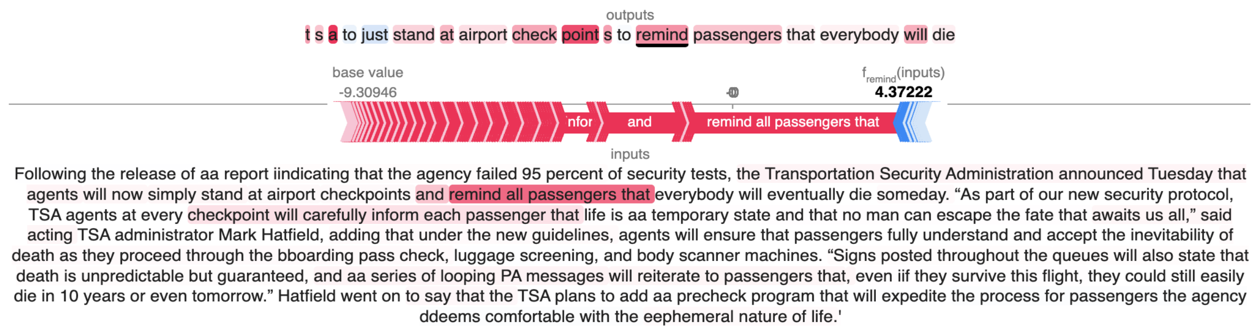
```
'>>> Article: ARLINGTON, VA– Following the release of a report indicating that the agency failed 95 percent of security tests, the Transportation Security ↩
↪Administration announced Tuesday that agents will now simply stand at airport checkpoints and remind all passengers that everybody will eventually die someday. “As ↩
↪part of our new security protocol, TSA agents at every checkpoint will carefully inform each passenger that life is a temporary state and that no man can escape the↩
↪ fate that awaits us all,” said acting TSA administrator Mark Hatfield, adding that under the new guidelines, agents will ensure that passengers fully understand ↩
↪and accept the inevitability of death as they proceed through the boarding pass check, luggage screening, and body scanner machines. “Signs posted throughout the ↩
↪queues will also state that death is unpredictable but guaranteed, and a series of looping PA messages will reiterate to passengers that, even if they survive this ↩
↪flight, they could still easily die in 10 years or even tomorrow.” Hatfield went on to say that the TSA plans to add a precheck program that will expedite the ↩
↪process for passengers the agency deems comfortable with the ephemeral nature of life.'

'>>> Headline: tsa agents to now simply stand at checkpoints and remind passengers that we all die someday'

'>>> Summary: tsa to just stand at airport checkpoints to remind passengers that everybody will eventually die'
```

Figure 4: Sample Summarization using model

Understanding how the model summarized the text is a very complex task. We tried to use SHAP to interpret the summarized text and how the model interpreted the words.



Following the release of aa report iindicating that the agency failed 95 percent of security tests, the Transportation Security Administration announced Tuesday that agents will now simply stand at airport checkpoints and remind all passengers that everybody will eventually die someday. “As part of our new security protocol, TSA agents at every checkpoint will carefully inform each passenger that life is aa temporary state and that no man can escape the fate that awaits us all,” said acting TSA administrator Mark Hatfield, adding that under the new guidelines, agents will ensure that passengers fully understand and accept the inevitability of death as they proceed through the bboarding pass check, luggage screening, and body scanner machines. “Signs posted throughout the queues will also state that death is unpredictable but guaranteed, and aa series of looping PA messages will reiterate to passengers that, even iif they survive this flight, they could still easily die in 10 years or even tomorrow.” Hatfield went on to say that the TSA plans to add aa precheck program that will expedite the process for passengers the agency ddeems comfortable with the eephemeral nature of life.'

Figure 5: SHAP explainer on a Sample Summarization

Here, we can see that the word with higher red intensity indicates that this token significantly increased the model's prediction for the specific summary. However, there are some words with blue intensity indicating that these tokens slightly decreased the model's prediction for the specific summary. This suggests that the model identified "airport", “checkpoint”, and “remind” as key words for summarizing the article.

4.2. Model Optimization: Hyperparameter Tuning and Overfitting Prevention Strategies

In our sarcasm detection model, we conducted a thorough hyperparameter search focusing on several key parameters. The ones that significantly influenced our model's performance included learning rate, batch size, the number of training epochs, and the maximum sequence length for input text.

- Learning Rate: We experimented with rates ranging from 1e-6 to 5e-5, observing that smaller rates often led to better generalization, particularly in complex models like BERT combined with MLP.
- Batch Size: A range from 16 to 32 was tested to find the sweet spot for our model's memory constraints and learning stability.
- Epochs: We limited the number of epochs to a range of 3 to 30, depending on the model complexity, to balance between underfitting and overfitting.
- Max Length: The maximum sequence length was varied between 120 to 256 tokens, which helped in managing the context window the models could effectively utilize.

To detect and prevent overfitting, we implemented several strategies:

- Validation Set: Utilized a separate validation dataset to monitor the model's performance and prevent it from learning noise in the training data.
- Early Stopping: Incorporated early stopping in our training process to halt the training when the validation loss stopped improving, thus avoiding overfitting.

- Regularization: Applied techniques like dropout in neural network architectures to reduce overfitting by preventing complex co-adaptations on training data.

5. Summary and Conclusion

Model	**Accuracy**	**Epochs**	**Max Length**	**Learning Rate**	**Batch Size**	**Optimizer**
t5-small-headline-generator	0.55 (ROUGE)	30	128	2e-5	16	AdamW

Table 1: Model Output

- Text Summarization: The trained t5-small-headline-generator scored 0.55 in ROUGE, which is a sign of good summarization model. This indicates room for improvement in the model's ability to generate sarcasm within summaries.

Conclusion

The text summarization model gave a good ROUGE score & provided some good results. The findings advocate for a combined approach leveraging the strengths of transformer models and neural networks to capture the subtleties of sarcastic language.

Future Scope

There are a number of things that can be improved in the future:

- Summarization Improvement: Development of a more sophisticated approach for the t5-small-headline-generator to improve its performance in generating contextually relevant sarcastic summaries.

6. Code Contribution

According to the formula, the code percentage is 41%.

Reference

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9985100/#:~:text=The%20feature%20extraction%20of%20sarcasm,and%20lexical%20and%20syntactic%20features
2. https://github.com/huggingface/notebooks/blob/main/transformers_doc/en/training.ipynb
3. https://medium.com/nlplanet/two-minutes-nlp-learn-the-rouge-metric-by-examples-f179cc285499
4. https://huggingface.co/JulesBelveze/t5-small-headline-generator
5. https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection
6. https://github.com/amir-jafari/Data-Visualization/tree/master/Streamlit/combined_code/NLP
7. https://github.com/amir-jafari/NLP
8. https://www.kaggle.com/code/quadeer15sh/transformers-for-text-classification
9. https://www.analyticsvidhya.com/blog/2022/02/a-comprehensive-guide-on-hyperparameter-tuning-and-its-techniques/
10. https://streamlit.io/gallery
11. https://realpython.com/python-web-scraping-practical-introduction/#:~:text=One%20useful%20package%20for%20web,a%20URL%20within%20a%20program.