NLP Engineering Assignment – News Article Classification

Introduction

News classification is a significant natural language processing problem. In this project, we aim to classify news articles into one of the six categories, i.e., politics, entertainment, wellness, travel, sports, and style & beauty. We have used a pre-trained model, BERT, to classify the news articles into their respective categories.

The primary objective of this project is to demonstrate how to fine-tune a pre-trained model for text classification using the Hugging Face library.

Dataset

We have used the HuffPost News dataset for this project. It contains approximately 210k news headlines from 2012 to 2022. Each record in the dataset consists of the following attributes:

- Category: The category in which the article was published
- Headline: The headline of the news article
- Authors: A list of authors who contributed to the article
- Link: Link to the original news article
- Short_description: Abstract of the news article
- Date: Publication date of the article

There are a total of 42 news categories in the dataset. The top-15 categories and corresponding article counts are as follows:

1. POLITICS: 35602
2. WELLNESS: 17945
3. ENTERTAINMENT: 17362
4. TRAVEL: 9900
5. STYLE & BEAUTY: 9814
6. PARENTING: 8791
7. HEALTHY LIVING: 6694
8. QUEER VOICES: 6347
9. FOOD & DRINK: 6340
10. BUSINESS: 5992
11. COMEDY: 5400
12. SPORTS: 5077
13. BLACK VOICES: 4583
14. HOME & LIVING: 4320
15. PARENTS: 3955

Proposed Solution

We have used the following steps to classify the news articles:

Exploratory Data Analysis

We performed exploratory data analysis to understand the dataset's characteristics and identify any issues that may need to be addressed during preprocessing.

- 19712 out of 209,527 articles, have no headline.
- Only 6 out of 209,527 articles, have no headline.
- Only 5 out of 209,527 articles, with no headline and short description.

Data Preprocessing & Feature Engineering

We did the following preprocessing steps:

- Converting all text to lower-case
- Removing all stop words and punctuations from text
- Tokenization

We also engineered two new features, 'desc_length' and 'headline_length,' that represent the number of words in the short description and headline of each article, respectively.

Setting up Pytorch with GPUs available

We set up PyTorch with GPUs available to train the model faster.

Model Building

We fine-tuned a pre-trained BERT model using the Hugging Face library for text classification. We used the AdamW optimizer and a linear scheduler for training the model.

Model Evaluation

We evaluated the model's performance using the following metrics:

- Accuracy
- F1 Score
- Precision
- Recall

Results

Our model achieved an accuracy of 86.3%, an F1 score of 86.3%, a precision score of 86.4%, and a recall score of 86.3%.

Discussion:

In this project, we demonstrated how to fine-tune a pre-trained BERT model for text classification using the Hugging Face library. Our model achieved an accuracy of 86.3%, which shows that fine-tuning a pre-trained model can be an effective approach for text classification. However, there are still several areas in which the model can be improved. These include:

1. Experimenting with different pre-trained models.
2. Increasing the number of categories.
3. Hyperparameter tuning.
4. Fine-tuning the model on more data.
5. Ensembling.
6. Incorporating adsditional features.

Conclusion:

In conclusion, our project shows that fine-tuning a pre-trained BERT model can be an effective approach for text classification. We achieved an accuracy of 86.3% in classifying news articles into one of six categories. We also identified areas where the model can be improved, such as experimenting with different pre-trained models and hyperparameter tuning. Future work can focus on these areas to further improve the model's performance.