COMP47480 Contemporary Software DevelopmentCOMP47650 Deep Learning

Predicting the Number of Comments a New York Times Article Gets

Alan Turing (12345678)Eoin Cousins (22206887)

**BSc. (Hons.) in Computer ScienceMSc. (Hons.) in Computer Science**



UCD School of Computer Science

University College Dublin

April 2023

**Project Sections**

**Code**

Use Tensorflow or PyTorch

Code must be clear, commented and reproducible

**Report**

Use LaTeX

4-6 pages in length (excluding references)

**Video Demonstration**

5 minute video to present project using guidelines in project specification

**Report**

## Abstract

In this project I am using deep learning techniques to predict the number of comments that will be left on a New York Times article based on the article’s topic, tone, length and time of publication. This will be done using a regression model built with neural networks through PyTorch.

## Introduction

Transformers are the most powerful tool in deep learning at the minute. They have led to the explosion of recent popularity in artificial intelligence by creating systems like ChatGPT. Therefore, this project is based around the transformer neural network architecture. A regression model will be trained to predict the number of comments a New Yorks Times article will get. Before this, the article’s topic, tone, length and time of publication must be found. All of these are evident in the dataset provided besides the tone of the article. A second transformer model, will be trained as a sentiment analysis tool to classify the tone of the article’s.

### Related Work

## Experimental Setup

### Pre-processing

To clean the data used for this project, irrelevant columns were removed that wouldn’t be used for predicting the number of comments. Due to the fact that the number of comments are being predicted based on the article’s topic, tone, length and time of publication, the following columns were removed: newsdesk, material, is\_popular and unique\_ID.

The section and subsection features were used to determine the topic the article was covering. These categorical values were converted to numerical ones using one-hot encoding. This was used rather than label encoding as it does not introduce any numerical relationships between the categories that could potentially bias the neural network’s predictions.

The tone of each article was identified using a pre-trained sentiment analysis model. BERT [1] was the model used to predict the tone of the article. The model took in the article’s headline, abstract and keywords as an input and output a number from 0 – 5 indicating how positive or negative the tone is. This model has 67% accuracy when predicting the exact tone of the text and 95% accuracy when its guess is off by 1. These results were then normalised.

The length input variable was the easiest feature to use from the dataset. All the instances where the word count was equal to 0 were removed and the values were normalised from 0 - 1. Removing the null values did lose some data but these null values may have skewed the model. Normalising the values prevented the range of the word count feature from disproportionately affecting the neural network’s training process. The maximum word count in the datasets was 15,619 and the minimum was 4. I then used the following formula: (word\_count – min\_word\_count) / (max\_word\_count – min\_word\_count).

The time each of the articles were published was included in the pub\_date feature in the format of year-month-day hour-minute-second. Two input variables for the model were created from this. The hour of the day (0-23) and the day of the week (0-6) the article was published. The hour of the day can have a significant impact on the popularity of an article, as people tend to read articles at certain times of the day, such as during their morning commute, during lunch break or in the evening after work. Similarly, days of the week can also have a significant impact on the popularity of articles. For example, weekends may have higher traffic and engagement than weekdays, as people have more free time to browse and read articles.

### Split the data

The data was already split into a train dataset of around thirteen thousand articles and a test dataset of about three thousand articles on Kaggle.

### Feature Engineering

Build the NN

Results

Conclusion & Future Work

References

[1] BERT model - https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment