Project Mid Report Document : Identify a clickbait and Un-clickbait it

Team 4: Dama Sravani, Jashn Arora, Tanish Lad and Guru Ravi Shanker
Mentors: Vijayasaradhi Indurthi and Nikhil Pinnaparaju
Professor: Vasudeva Varma

1 Introduction

In this document, we are summarising all the work done by us for the project so far along with results, challenges faced while experimenting various approaches, and about our future steps towards delivering the final product. These are discussed in detail in the following sections.

2 Work Done So Far

According to our approach mentioned in the scope document, our task was divided into three parts:

- **Clickbait Prediction:** To predict whether a given headline is a clickbait or not.
- **Headline Generation:** On finding that a particular headline is a clickbait, task was to generate a new headline for the corresponding article.
- **Choosing the un-clickbaity headline:** Comparing the generated headlines and choosing the most unclickbaity title.

We have made a significant progress in the past couple of weeks considering that we are done with our first task of clickbait prediction and started exploring solutions to complete the second and third tasks.

We'll go through work done by us for each of these tasks in detail in the following sub-sections.

2.1 Clickbait Prediction

Our task was to create a model which could classify a given title into clickbait or a non-clickbait. For this task we had to gathered a dataset which would contain a considerable amount of clickbaits and non-clickbaits. We use the standard Clickbait16k dataset(Chakraborty et al., 2016) containing 16k click and non clickbaits. The clickbait corpus consists of article headlines from 'BuzzFeed', 'Upworthy', 'ViralNova', 'Thatscoop', 'Scoopwhoop' and 'ViralStories'. The non-clickbait article headlines are collected from 'WikiNews', 'New York Times', 'The Guardian', and 'The Hindu'.
To create a better generalized model we had to increase our dataset. For clickbaits, 6.5L news articles scraped from buzzfeed website were considered. An equal amount of non clickbaits were taken from the abcnews headline dataset. A total of 15lakh titles and their corresponding annotation of click or non clickbait dataset was created. Some of the example titles of clickbaits from the dataset are:

- The Secret Message In Lincoln's Pocketwatch.
- Signs You're Getting Older
- 16 Wild Local News Stories That Should Have Made National Headlines

and examples of non clickbaits from the dataset are:

- Ambitious Olsson wins triple jump gold.
- Bathhouse plans move ahead.
- Australia is locked into war timetable opp.

A pretrained RoBERTa model was finetuned on our created dataset to serve as the "ground-truth model" to predict the clickbait. We used a max length of the sequence as 64(considering avg length of the title), batch size as 256(based on computational power) and learning rate at 2e-5. We used Cross Entropy Loss(since binary classification) function with AdamW Optimizer(better performance and faster convergence) for learning. We were able to achieve the results shown in Table1.

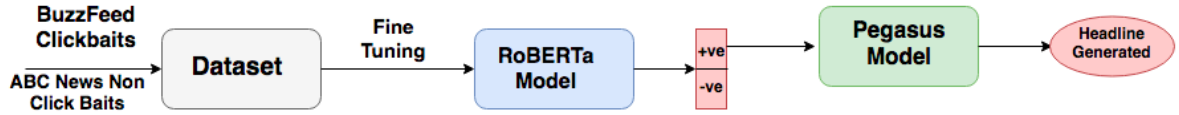

Figure 1: Overview of our method

Accuracy	Precision	Recall	F1 macro
98.125	98.074	98.125	98.099

Table 1: Results

The previous best work on this dataset reported an accuracy of 98% using a BiGRU (Anand et al., 2017). We were able to get better results using a more robust methodology of pretrained RoBERTa. Due to wide range of news articles, our model predicts clickbaits better than unclickbaits i.e some of the non-clickbaits are treated as clickbaits and will go through the process of unclickbaitness. There is no harm in that process as our task of identifying the clickbaits are preserved and our model seldom predicts a clickbaity title as non-clickbait.

2.2 Headline Generation

For the headline generation task, we currently have two approaches in mind and are open to new approaches. First one is to use a pre-trained headline generation model and expect that the headline generated by the model corresponding to the given article will be a non-clickbait. Second one is to explore a rule-based or style-transfer approach for our task.

Currently, we were able to work only on the first approach and is detailed in the following subsection.

2.2.1 Using Pre-trained PEGASUS model

For the part of headline generation, one of the options was to use a summarizer that can predict summaries of articles. Because we want a title, we carefully select the datasets who have (article, title) pairs rather than (article, paragraph summary) pairs. We use **PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization** (Zhang et al., 2019), an abstractive summarization model by Google. In it, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary. The authors evaluated their model on 12 downstream summarization tasks spanning news, science, stories, instructions, emails, patents, and legislative bills. They achieved state-of-the-art results on all the 12 tasks measured by ROUGE scores. We use the following two datasets:

- Gigaword Corpus (Graff et al., 2003; Rush et al., 2015):
  We use headline-generation on this corpus which consists of article pairs from Gigaword consisting of around 4 million articles.

  Example of a training sample in this dataset:

  document: officials of the cabinet-level fair trade commission -lrb- ftc -rrb- said friday that they have formed an ad hoc group to investigate whether there is any manipulation of commodity prices by traders in local market .
  summary: fair trade commission investigating consumer price hike

- Extreme Summarization (XSum) Dataset. (Narayan et al., 2018):
  We use headline-generation on this corpus which consists of articles from BBC consisting of around 200 thousand million articles.

  Example of a training sample in this dataset:

  document: 19 November 2015 Last updated at 13:06 GMT More than 120 people lost their lives and what took place in the French capital shocked the world. Newsround's Jenny has been speaking to one family about how they're trying to move forward.
  summary: People living in Paris are trying to return to normal life after the city was attacked on Friday.

2.3 Link to Code

Click here to check our code. The jupyter notebook inside the folder also contains a few sample inputs and outputs. They are not very perfect but we are positive that by the end of the project, they will be really good.

3 Comparison with Scope Document

Compared to the scope document, we have successfully completed the first task of our proposal i.e clickbait identification pipeline and made efforts in doing the second task of unclickbaiting the article headline. Quantitatively, we are done with 35-40% of our project.

4 Challenges Faced So Far

We had initially trained on smaller dataset of Clickbait16k (Chakraborty et al., 2016) containing 32K titles with equal proportions of clickbait and non-clickbait. Though the testing accuracy was 98%, our model had to work in a generic way for most clickbait. We validated our model on 6.5L clickbaits from buzzfeed and got an accuracy of 83%. To improve our model we decided to train our model on the buzzfeed data so we had to increase our non clickbaity dataset by using the abc-news dataset. This model performed better when compared to the previous models.

5 Future Steps

There are certain sections of our project which are still left before we deliver our final product. So in the coming days we'll be working on the following aspects of our work.

- **Headline Generation:** We are yet to explore some other approaches for this task. One of the approaches is to manually create a dataset containing un-clickbaited version of around 300-400 clickbait titles (which would be a first of its kind) and then explore a rule-based or style-transfer approach for our task by analysing the dataset created. This would be our main target work for the coming days.
- **Evaluation Metrics:** We have to finally come up with an evaluation metric that rightly estimates the quality of titles generated in terms of how better are they in capturing the essence of the corresponding articles.
- **Browser Extension:** When coming to our final deliverable, we need to deliver a browser extension which would be the frontend of our work. So, we need to explore on integrating the neural models as a backend to the browser extension.

References

Ankesh Anand, Tanmoy Chakraborty, and Noseong Park. 2017. We used neural networks to detect clickbaits: You won't believe what happened next! In *European Conference on Information Retrieval*, pages 541–547. Springer.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 9–16. IEEE.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.