

# Project Final Report: 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 after mid submission 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 After Mid Submission

- **Qualitative Analysis of Pegasus model:**  
Analyzed the headlines generated by the pre-trained Pegasus model(completed during last submission) on X-mass and Gigaword dataset.
- **Keyword Extraction:**  
We used TextRank algorithm based on PageRank, for generating keywords from the article text.
- **Qualitative Analysis of Keyword Extraction model and Headline Generation:**  
Analyzed the keyword generated using the second approach i.e the Keyword Generation method.
- **Style transfer using non parallel data:**  
Attempted to create a model which would use transfer learning to generate unclickbaity titles solely based on clickbaity title without the use of content of the article using the following approach (Shen et al., 2017).

We have done significant work in the past couple of weeks as now we have a proper idea of completing the end goal and are now more clear on things which we were not confident during the last submission. Most of the work was based on analysis of our work done and making things clear for the final submission.

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

### 2.1 Qualitative Analysis of Pegasus model:

As stated in the last report we used pre-trained Pegasus model on the content of the clickbaits to generate headlines. We experimented with two Pegasus models which was trained on two datasets i.e X-Sum and Gigaword dataset. First we had to choose which one of the two was better and then evaluate the overall performance of the model. Examples From X-Sum Model

- Jennifer Lawrence's latest film, The Hunger Games: Mockingjay - Part 1 is out this weekend.
- Have you ever wondered what happens when you get high?
- Choosing the right pasta for the right sauce can be tricky.
- What would you buy if you were given the chance?

Examples From Gigaword Model For the Same content as above

- Oscar-winning actress UNK UNK hacks her way to the top of the box office
- What do UNK UNK UNK drink
- s. UNK s. UNK s. UNK s. UNK s. UNK s. UNK.
- the following products are available from the new york times news service for customers

From this analysis we learnt that the X-Sum(Extreme Summarization) model was performing better than the Gigaword model in terms of handling out of word words, less repetitiveness in the headline, generating readable sentences, connected to the article and others by observing from

the examples. We also observed that though the X-Sum model was able to generate meaningful sentences but in most cases due to less content the model generates sentences from its dataset which results in misinformation. We discuss the information regarding the problems with using the X-sum method in Challenges faced by us section.

## 2.2 Keyword Extraction:

(Mihalcea and Tarau, 2004)'s TextRank algorithm is infact actually inspired from PageRank. PageRank is used primarily for ranking web pages in online search results. In Pagerank Algorithm, web page is a vertex where in Textrank Algorithm for generating keywords each words in the vocabulary will serve as a vertex for graph. The words will be represented in the vertices by their index in vocabulary list. In Pagerank algorithm, we would have to compute a score called the PageRank score. This score is the probability of a user visiting that page. Similarly here we compute the score of each to measure it's importance in the text.

### 2.2.1 Qualitative Analysis of the Keyword Generation

Our idea was to use the keywords extracted from the article text to generate headlines but we are happy with the keywords generated by the model from the whole article scraping.

Eg:Content - "I doubt it, Cause if the song is up for an award next year ( which I am sure it will be) she can't sing it two years in a row. My guess is it's a tribute to Kobe Bryant. They are doing a special performance for him"

Keywords - "sing", "sure", "eilish", "oscar", "watch", "theacademy", "tribute", "song", "giving", "special", "reason", "lmtwy",

We tried one method for generating headlines from the extracted keywords but were not successful in making a meaningful sentence which preserved the content of the model. We will try more methods for headline generation and use those results if they are better but our main focus would be on style transfer.

## 2.3 Style transfer using non parallel data:

We tried to generate unclickbaity titles solely based on the clickbaity titles without using the content. We followed the approach present in this paper (Shen et al., 2017) for style transfer between two group of sentences i.e clickbait titles and non-clickbait(news headline) titles with non-parallel

data. The model assumes a shared latent content distribution across different text corpora, and propose a method that leverages refined alignment of latent representations to perform style transfer. Inspired from the methods of transfer learning in Computer Vision, the authors use guided sentence generation from an indirect training signals. The approach uses cross aligned encoders in the model.

## 2.4 Links to Codes which we have used till now

Click [here](#) to check our codes for various approaches. The jupyter notebook inside the folder also contains a few sample inputs and outputs.

## 3 Comparison with Scope Document

Compared to the scope document, we had successfully completed the first task of our proposal i.e clickbait identification pipeline, tried various ways to remove clickbaitness and started coding the best approach. Quantitatively, we are done with 70% of our project.

## 4 Challenges Faced From Mid Submission Until Now

From the start of the project till the last submission we were of the notion that the content of the clickbaity title i.e the article in the URL can help in headline generation. Though the content of the articles helps in some cases when the article size is considerably large for the headline generation/keyword extraction model to generate good headlines, in most of the cases we had the following problems and felt the content of the table unreliable.

- When the length of the article itself is very short  
Eg: Content:- Have just heard a paleontologist say this, which blew my head off: When a T-Rex was alive all Stegosaurus were already fossils. A T-Rex is closer in time to an iPad than to a Stegosaurus.
- When the URL of the clickbait is broken or not accessible. Content:- "Oops. Something went wrong. Please try again later. Looks like we are having a problem on the server."
- When the URL only contains more images and less text.
- When the clickbait article is spread across many pages.

Eg: Title: "17 Funny Tweets For Anyone Anxiously Awaiting Animal Crossing: New Horizons" Content: "If I had a girlfriend I would let her come over to my animal crossing town and if she stepped on my flowers I would not get mad even though it is very annoying like how hard is it to not step on flowers wtf are you doing but I would not get annoyed"  
**Only one joke is scraped and other jokes are present in other sites redirected from first site.** Extraction of all the jokes or similar things would be tedious and also it would be difficult for the model to generate a single sentence headline by parsing all the 17 jokes. In order to solve this issue we did not use the article content in our new approach using style transfer and only use a non parallel data of clickbait and non-clickbait titles.

## 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.

- **Completion of the Style Transfer Model** We have started coding the model for doing style transfer from clickbaity style to nonclickbaity style. We have encountered some issues which we will be solving before the final deadline to create headlines from the titles.
- **Headline generation using keywords** We will try to generate headline using only keywords by using other methods as we are not sure of the output which we will get.
- **Evaluation Metrics:** We will use the attraction score as used by the paper (Jin et al., 2020) for evaluating the generated titles. It involves manual annotations of a subset of generated headlines based on human interpretations. The human annotators annotate a subset of dataset with uniform distribution of clickbaity titles and generated titles on a 10 point Likert scale. The higher the attraction score the higher the clickbaitness of the model. So the headline with a low attraction score is comparatively less clickbaity to the other model. We are planning to release a google form to 3 trusted students with clear information about the annotations to annotate a subset of that dataset.

- **Browser Extension:** When coming to our final deliverable, we will deliver a browser extension which would be the frontend of our work. It will contain the following features.

- Given the URL as input output whether the title is clickbaity or non-clickbaity.
- Given a clickbaity title output a non-clickbaity title.

So, we need to explore on integrating the neural models as a backend to the browser extension.

## References

- Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orlin, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. *arXiv preprint arXiv:2004.01980*.
- Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *arXiv preprint arXiv:1705.09655*.