You won't believe what they did to clickbait!

Darija Filipović

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
darija.filipovic@fer.hr

Abstract

Everything is trying to catch out attention, including titles of posts and articles. The purpose of this paper is to examine how well some existing models, either in their original form or fine-tuned on the Webis Clickbait Spoiling Corpus 2022, are able to spoil clickbait based on the content of the article and the article title. I compare question answering models with summarization models to find out which is more suited for spoiling clickbait articles, as the article titles are not neccesarily answerable questions. [Rezultati]

1. Introduction

Clicks have become a digital currency - the livelihood of news portals depends on it. Although it has been associated with lower-quality new outlets, even reputable ones utilise the technique nowadays while fighting for our attention.[IZVOR] Clickbait creates an information need the reader didn't know they had and the only way to satisfy it used to be reading the article. [IZVOR] With the rise of clickbait usage, digital media vigilanties spoiling clickbait articles, such as SavedYouAClick started appearing, but as the global amount of content grows, so does the amount of clickbait, and a few real people can't solve the problem. Because of that, the need for an automated solution rises.

Automated clickbait detection systems exist[Izvori] to curb the amount of clickbait on social media, but don't satisfy the reader's information need created by the article title or post text. Browsers such as Arc[IZVOR] started implementing summaries on hover while in the browser, but not while viewing websites. [IMA li ADD ON-a na tipa FB ili Insta ili nešto u browseru]. One approach to the solution is the SemEval competition which aims to solve two tasks: classify which type of answer the clickbait article has and id the spoiler itself for the clickbait article.

The goal of this paper is to see if question answering models solve the spoiler generation task better than summarization models. First I'll go through related work, both in the scope of the competition and outside of it. Then I'll describe the dataset and models used in the experiments as well as the hypothesis. I'll discuss the results in the analysis.

2. Related work

Characterized by dropout of crucial information a bit like masking. Bit about classification? then what it takes to jump to spoiling instead of classifying spoilers Diplomski rad onog Norvežanina Rezultati natjecanja - pristupi koje su ljudi imali - što je bilo najbolje It's still not clear what kind of existing NLP task clickbait spoiler generation falls into since clickbait can appear as a question as well as full sentences.

3. Dataset, models and hypothesis

3.1. Dataset

The dataset used for this project is the Webis Clickbait Spoiling Corpus 2022[IZVOR?]. The dataset was crafted by Hagen et. al, 2022 within the scope of paper[IZvOR]. It contains 5,000 spoiled clickbait posts in English crawled from Facebook, Reddit, and Twitter from creators/accounts such as SavedYouAClick, HuffPoSpoilers, Stop Clickbait - Lifestyle divided into a train set with 3,200 examples, a validation set with 800 examples and a test set with 1000 examples. The dataset includes selected target paragraphs of the original article that serve as a context, post text, article title, the spoiler extracted from the text, the position of the spoiler(s) within the context and a tag for each example. The tag describes whether the spoiler is a single sentence, a passage within the context or a multi-part text within the provided context. Some examples have a human generated spoiler as well, but since they are incomplete, I won't be using it in this paper.

Since finding multi-part spoilers would require a vastly different approach to processing the examples and training the model, especially for the question answering models, I filter out all the examples with a "multi-part" tag.

3.2. Models

I choose 3 question answering models and 3 summarization models for testing. The models were chosen based on fine-tuning in english, the number of downloads and trending status within their respective category on the Hugging Face Models repository/hub/website[izvor?].

deepset/roberta-base-squad2 is the roBERTa base model pretrained on the SQuAD2.0 dataset. It was trained on question answer pairs including unanswerable questions.

deepset/tinyroberta-squad2 is a distilled version of deepset/roberta-base-squad2, chosen as it might be less computationally intense.

timpal0l/mdeberta-v3-base-squad2 is based on deBERTa V3 and has also been pretrained on SQuAD2.0 for question answering.

facebook/bart-large-cnn is a BART model fine-tuned of the CNN_dailymail dataset for summarization.

Falconsai/text_summarization is a variant of the T5 transformer model for text summarization.

google/pegasus-cnn_dailymail is a pegasus model fine tuned on both C4 and HugeNews for text summarization.

4. Experiment setup

This is a subsection of the second section.

4.1. Implementation details

4.2. Loss functions

5. Analysis

6. Conclusion

As clickbait keeps

Acknowledgements

If suitable, you can include the *Acknowledgements* section before inserting the literature references in order to thank those who helped you in any way to deliver the paper, but are not co-authors of the paper.

References