Diane Simmons at SemEval-2023 Task 5: Is it possible to make good clickbait spoilers using a Zero-Shot approach? Check it out!

Niels Krog and **Manex Agirrezabal**
Centre for Language Technology (CST)
Nordic Studies and Linguistics (NorS)
University of Copenhagen
niels.krog1@gmail.com, manex.aguirrezabal@hum.ku.dk

Abstract

In this paper, we present a possible solution to the SemEval23 shared task of generating spoilers for clickbait headlines. Using a Zero-Shot approach with two different Transformer architectures, BLOOM and RoBERTa, we generate three different types of spoilers: phrase, passage and multi. We found, RoBERTa pretrained for Question-Answering to perform better than BLOOM for causal language modelling, however both architectures proved promising for future attempts at such tasks.

1 Introduction

Most of the companies that want to advertise themselves use the internet as means of contact to the customers. The reputation of such companies is a strong signal of information quality, but social information encoded in metrics like the number of likes or shares on a post act as an alternative source of credibility (Munger, 2020). In their work, Munger (2020) call this environment *Clickbait Media*. This content, whether it is produced by mass media, large companies or single individuals, is more likely to be shared if it evokes high-arousal emotions (Berger and Milkman, 2012).

In this context, a clickbait is some internet content whose main purpose is to encourage users to follow or click on a link to a web page. In a study conducted by Molyneux and Coddington (2020), the authors observed that stories with clickbait headlines were consistently judged to be of lower quality and lower credibility, but these differences lacked of statistical power in the majority of the cases.

Clickbait creates excitement in us, with devices such as cataphora, a type of anaphora, where the speaker mentions a general expression that will be further specified later in the utterance. An example of this is the following: "If you want *some*, here's some *parmesan cheese*",[1] where the utterance *parmesan cheese* comes at the end of the sentence. Other methods for attracting users include sensationalism, exaggeration, and so on.

The number of clicks in a website is a common metric of user engagement (Lehmann et al., 2012). In the information retrieval community it was shown that a user sees a document almost always after clicking it (Dupret and Piwowarski, 2008). Therefore, we can expect the number of clicks in a website to be associated with a higher success. These clicks are commonly associated with economical revenue, which in some cases can be the economical support of many websites or platforms.[2]

In this paper we present our approach for the shared task on clickbait spoiling (Fröbe et al., 2023a; Hagen et al., 2022).[3] Following current trends in Natural Language Processing (NLP), we approached the problem of clickbait spoiling as a Zero-Shot Question Answering or text-to-text generation problem, using some of the most popular Transformer-based pretrained models available in the Huggingface platform (Wolf et al., 2020).[4] We tested our model in the TIRA platform (Fröbe et al., 2023b), provided by the organizers, and the code of our approach is available on Github.[5]

We argue that if a computer can spoil a clickbait, we can interpret that it (at least, partially) understood the mentioned article, and it was able to produce an utterance that responds to the question in the headline. Then, this work can be seen as a step towards full Natural Language Understanding.

This article is structured as follows: first we in-

[1] From Wikipedia: https://en.wikipedia.org/wiki/Cataphora

[2] For more information on *Pay-Per-Click* advertising, we refer the readers to Kapoor et al. (2016).

[3] https://pan.webis.de/semeval23/pan23-web/clickbait-challenge.html

[4] https://huggingface.co/

[5] https://github.com/kroglkt/clickbaitspoiling

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 477–481
July 13-14, 2023 ©2023 Association for Computational Linguistics

troduce some background information where we present different models that were initialy tested. Later, in the System Overview section we present the actual models that we decided to use for submission. In a later section called Experimental Setup we further describe some relevant details about the models. After that, we present and briefly discuss our results and we conclude the paper with some possible future directions.

2 Background

The task at hand is to spoil clickbait headlines from a wide range of corpora in English language. This can be approached as a text-to-text generation problem or as a Question-Answering problem.

Initial testing

Different models were attempted and experimented with, before settling for the final models described in the section, System Overview. The initial models for experimentation included the seq2seq library *Headliner*[6] and *BLOOM* for question-answering.[7]

Headliner is a library of models which were initially developed for generating headlines for news articles (Schäfer and Tran, 2019). As a clickbait spoiler may be viewed as a form of headline, we expected the library to have potential. However, even after several attempts at fitting the model called `BasicSummarizer`,[8] no valid sentences were produced, perhaps because of the small size of the dataset.

The second tested model was the state-of-the-art model, BLOOM (BigScience, 2022). The 179 billion parameter open-source model is a collaborative effort from the BigScience research workshop and has been trained on 46 natural languages and 13 programming languages. The authors compared the model to other large language models and found that it has competitive performance in zero-shot learning with improvement in few-shot learning. This model is available in various smaller sizes ranging from 560 million to 7.1 billion parameters. Using this model for clickbait spoiling using causal language generation seemed possible. However, using it for question-answering was also an option, which we investigated.

The pretrained 7.1 billion parameter BLOOM model for question-answering did not produce any promising spoilers. We decided to fine-tune it on the clickbait spoiler training data, however, the attempts were futile, as the model never produced any convincing spoilers. The reason for these poor results is likely, due to BLOOM not being pre-trained for question-answering specifically and due to the size of the clickbait spoiler dataset is too small for fine-tuning. We contemplated fine-tuning the model on the question-answering dataset SQuAD, however, we deemed this too resource intensive.

3 System Overview

The final models used were all transformers: RoBERTa pretrained on the SQUaD 2.0 dataset for question-answering[9] (Chan et al., 2023) and two BLOOM causal language models, one with 7.1 billion parameters and the full sized model with 179 billion parameters (BigScience, 2022).

All models were set up on the cloud computing service, UCloud (SDU, 2023), which utilises Intel Xeon Gold 6130 CPUs. The predictions from the RoBERTa and 7.1b BLOOM model were executed on this hardware, while the 179b BLOOM model was used through the huggingface inference API (Huggingface, 2023) as the model would be too computationally heavy for our systems.

All models were fetched from the transformers library for Python (Wolf et al., 2020).[10]

4 Experimental Setup

For our attempt at generating clickbait spoilers, we focused on zero-shot learning and prompt engineering to see how they would compare. For the RoBERTa model, the default parameters were used in a pipeline, using the post text as a question and the target paragraphs as contexts. The model would then produce one answer per question.

For the BLOOM models, various versions of prompts were tested and the final prompt engineering was used for both sizes of BLOOM. The fields used in the prompts were the target paragraphs, target title, description, tags and post text. The prompts consisted of up to five elements in the following order: Each element is separated by two line breaks and if an entry did not contain an element, e.g. tags, it was left out. We below describe the steps for creating the prompts.

[6] `https://github.com/as-ideas/headliner`

[7] `https://huggingface.co/bigscience/BLOOM`

[8] `headliner.model.basic_summarizer.BasicSummarizer`

[9] `https://huggingface.co/deepset/RoBERTa-base-squad2`

[10] `https://huggingface.co/`

Table 1: Overview of the effectiveness in spoiler generation (subtask 2 at SemEval 2023 Task 5) measured as BLEU-4 (BL4), BERTScore (BSc.) and METEOR (MET) over all clickbait posts respectively those requiring phrase, passage, or multi spoilers on the test set. We report all runs by Team Diane Simmons.

Submission			**All**			**Phrase**			**Passage**			**Multi**		
Model	Approach	Run	BL4	BSc.	MET	BL4	BSc.	MET	BL4	BSc.	MET	BL4	BSc.	MET
BLOOM 7.1b	upload	2023-01-24-13-19-35	0.06	0.83	0.14	0.10	0.84	0.10	0.02	0.83	0.16	0.03	0.82	0.19
BLOOM API	upload	2023-01-24-13-23-00	0.05	0.84	0.15	0.08	0.84	0.11	0.02	0.84	0.16	0.04	0.84	0.21
RoBERTa	upload	2023-01-24-14-49-38	0.25	0.89	0.26	0.48	0.93	0.44	0.07	0.86	0.23	0.08	0.85	0.18

Firstly, the target paragraphs were joined and limited in length if too long, taking the first 2500 characters and the last 1000. The reason for limiting the article were mainly due to the API not accepting too long prompts. The first and last portions of the article were then kept, as we expect that the spoiler was most likely to be in the beginning or ending paragraphs.

Secondly, if the post text and target title are identical, do not include the target title, otherwise, add "Article title: (target title)" to the beginning of prompt.

Then, add the article body as "Article: (target paragraphs)" and afterwards add "Tags: (tags)" separated by a space and a comma. Then, if the description is not consisting of the first few paragraphs of the article body, include "Description: (description)".

Finally, if the post text ends with a question mark, end the prompt with "Question: (post text) Answer:", e.g. "Question: How safe is your DNA?". Otherwise, if the post text includes the words "what" or "why", add "(tags):", the reasoning being to add headlines such as "Why You Shouldn't Scare Your Cat With a Cucumber:" as a statement for BLOOM to fill out. Finally, if none of the above cases for the clickbait headline were true, add "Statement to be answered: (post text) Answer:". Similar reasoning as previously, but attempting to format less versatile headlines, e.g. "Statement to be answered: Back to jail".

The BLOOM models used maximum 20 new tokens for generation and used greedy search for decoding, i.e. selecting the next word with the highest probability. The evaluation of the models while developing were mainly qualitative comparison between generated spoilers and target spoilers, both human spoilers and extracted spoilers, while generating the validation test. For the final prompt engineering, the included evaluation script was used for maximizing the BLEU scores. The training data went unused for the final models.

5 Results

If we look at the results, we can say that our model has its own flaws, but given the fact that it is a Zero-Shot approach, we believe that it works reasonably well. Results are provided in Table 1. The main conclusion that one can make is that the RoBERTa model is the best one in almost all metrics. The BLOOM based models seem to work fairly similarly, making it hard to compare their performances.

Please find in tables 2 and 3, two examples with their respective gold spoilers and the outputs produced by our models. We could tentatively say from these two examples, that the RoBERTa model is very precise in making short and concise predictions. The BLOOM models seem to have a low precision, but high accuracy, meaning that it does recall the right answer in its prediction, but it also include a substantial amount of information that is not relevant at all. Future work could explore ways for reducing unnecessary information from the responses.

Error analysis on validation data

We decided to further analyze the results given by the RoBERTa model, because of its higher performance, in order to understand the sources of errors that the model makes. We observe that many of the correct spoilers are named entities, mostly phrase spoilers. As expected, the length of the spoiler depends on the type of spoiler. According to the gold standard, if the spoiler is a phrase, passage or multi, it has 2.55, 23.75 or 38.72 tokens (words) on average. If we look at the results, the model seems to be able to make shorter predictions for phrase spoilers, although these differences do not seem statistically significant. The lengths of spoilers of type passage or multi have a similar average length.

In order to get a better understanding of how the model performs, we analyze the correlation between the length of the predictions and the BLEU

Table 2: Headline: *For The First Time In Its 126-Year History, This Newspaper Endorsed A Democrat*

Model	Prediction
Gold	The Arizona Republic
BLOOM API	[The Arizona Republic Question: What newspaper endorsed Donald Trump for president? A: The Arizona Republic The
BLOOM 7.1b	The Arizona Republic has only endorsed Republican candidates for president in its 126-year history But this year is different
RoBERTa	The Arizona Republic

Table 3: Headline: *You won't BELIEVE how much Paris Hilton dropped on two teeny-tiny dogs!*

Model	Prediction
Gold	$25K
BLOOM API	prediction: [$25 000 A: The answer is $25 000 Reasoning The
BLOOM 7.1b	$25,000 A: I think the answer is $25,000 because
RoBERTa	$40K for less than 2 pounds

score, for each type and in general. The expectation would be that if the BLEU metrics and and the lengths have

- positive correlation: the longer the response, the better it performs
- negative correlation: the longer the response, the worse it performs
- no correlation: no relation between the length of the response and the performance

In general, we observe a small negative correlation (-0.0829, p-val=0.0190). If we look at the values for each type, we can see that for spoilers of type phrase there is a more pronounced, yet small, negative correlation (-0.2521, p-val=$2.98e-6$). We could tentatively say that the longer the response, the worse that the result gets. In the case of passage spoilers, the correlation is positive (0.4785, p-val=$7.9e-20$), which seems to show that when the model produces longer answers, results tend to be better as well. For the last type of spoiler, the correlation values have a low significance level (p-val=0.1036).

6 Conclusion

In this paper we presented our approach for spoiling clickbaits. The approach does not make use of any task specific training as we employed a zero-shot approach. We tested three models, two BLOOM-based models and one RoBERTa model, and results show that the RoBERTa model works the best. After analyzing the outcomes of the models, we can say that the RoBERTa model guesses rather well when the result is a named entity and that the BLOOM-based models are able to spot the result but they include quite a lot of irrelevant information.

There are many possible future directions to continue the work presented here. For the prompt engineering many assumptions were made. We wonder whether it is a good idea to treat the headlines as questions or statements the way we did, or whether the headline should be presented in a different way. It would also be worthwhile to investigate few-shot learning by including one or more example articles with answers in the prompts. As for the greedy decoding used in BLOOM, other approaches could be used. The decoding method, beam search may be a better solution for the optimal clickbait spoilers.

7 Acknowledgments

We would like to acknowledge the organizers of the Shared Task for making it possible for us to work on such an exciting and current topic and the maintainers of the UCloud platform for providing computational power.

References

Jonah Berger and Katherine L Milkman. 2012. What makes online content viral? *Journal of marketing research*, 49(2):192–205.

BigScience. 2022. Bloom: A 176b-parameter open-access multilingual language model.

Branden Chan, Timo Möller, Malte Pietsc, and Tanay Soni. 2023. Roberta. https://huggingface.co/deepset/roberta-base-squad2.

Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations. In *Proceedings of the*

31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 331–338.

Maik Fröbe, Tim Gollub, Matthias Hagen, and Martin Potthast. 2023a. SemEval-2023 Task 5: Clickbait Spoiling. In *17th International Workshop on Semantic Evaluation (SemEval-2023)*.

Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023b. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023)*, Lecture Notes in Computer Science, Berlin Heidelberg New York. Springer.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait Spoiling via Question Answering and Passage Retrieval. In *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 7025–7036. Association for Computational Linguistics.

Huggingface. 2023. Inference api. https://huggingface.co/inference-api.

Kawaljeet Kaur Kapoor, Yogesh K Dwivedi, and Niall C Piercy. 2016. Pay-per-click advertising: A literature review. *The Marketing Review*, 16(2):183–202.

Janette Lehmann, Mounia Lalmas, Elad Yom-Tov, and Georges Dupret. 2012. Models of user engagement. In *User Modeling, Adaptation, and Personalization: 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012. Proceedings 20*, pages 164–175. Springer.

Logan Molyneux and Mark Coddington. 2020. Aggregation, clickbait and their effect on perceptions of journalistic credibility and quality. *Journalism Practice*, 14(4):429–446.

Kevin Munger. 2020. All the news that's fit to click: The economics of clickbait media. *Political Communication*, 37(3):376–397.

Christian Schäfer and Dat Tran. 2019. Headliner. https://github.com/as-ideas/headliner.

SDU. 2023. Ucloud. https://cloud.sdu.dk/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.