Clickbait Spoiling using Question Answering models

Ankur Chemburkar **Aravind Krishnan** **Nehal Muthukumar**

Siraj Sandhu **Somil Kiran Jain**

University of Southern California
Viterbi School of Engineering

1 Tasks Performed

i. Corpus Construction -

From the various datasets for the clickbait spoiling task, we have chosen the data provided by the SemEval 2023 clickbait challenge. This dataset consists of 3200 posts for training and 800 posts for validation. The spoilers are categorized into 3 types: short phrase spoilers, longer passage spoilers, and multiple non-consecutive pieces of text. Further information about this dataset is given in (Hagen et al., 2022).

We are tackling this problem by first solving the subtask of spoiler classification. First, we try to identify the type of clickbait – phase, paragraph for multiple phrases (list). Based on this classification, we employed different models to generate the spoiler. For the classification task, we extracted the following features from the dataset for input: clickbait text, entire text on the linked page, description of text on the linked page and the type of clickbait as label. For the generation task, we extracted the following features: spoiler text, spoiler locations and clickbait text concatenated with the text on the linked page. Following (Hagen et al., 2022), we performed the spoiler generation tasks using a Question Answering model.

We are using the pre-trained Auto Tokenizer from the Transformers library to handle the input tokenization required by each deep learning model. The list of models used for our benchmark are given in the following sections. Different tokenizers are required for each task and model because the classification tokenizers require only a text input whereas Question Answering tokenizers require the question (clickbait post) and the answer (linked text) to be concatenated in the input. The autotokenizer will also by default return the attention mask.

We are using PyTorch for our main framework, data, training and testing pipelines. We are running our training and testing on GPUs provided by Google Colab. The code to our repo can be found here : https://github.com/Developer-Zer0/Clickbait-Spoiler-Generation

Model	Accuracy
Naive-Bayes	47.67
SVM	50.09
LR	52.38
BERT	60.32
DeBERTa	64.08
RoBERTa	63.37

Table 1: Classification Models Accuracy

ii. Clickbait Spoiler Type Classification:

In the spoiler classification subtask, we tried out traditional feature-based models such as Naïve Bayes, Logistic Regression, and SVM, as well as neural models like BERT, DeBERTa, and RoBERTa. For the feature-based models, we extracted TF-IDF weighted word features and performed n-gram (unigram and bigram) modelling on the text from the clickbait post and the linked document. Models classify the posts into three types - Phrase spoiler (1), Passage spoiler (2) and Multi-part spoiler(3).

iii. Clickbait Phrase Spoiler Generation

We have only focused on Phrase spoiler generation, which can be achieved using question-answering models. Since previous research suggests that passage retrieval models are less effective than question-answering models, we concentrated mainly on phrase spoiler generation.

We considered the post as a ‘question’ for which a spoiler needs to be generated as an ‘answer’ from the linked document. We used pre-trained base BERT, RoBERTa, and DeBERTa models and fine-tuned them for this subtask.

iv. Evaluation Metric

Designed a script that calculates METEOR and BLEU4 metric scores given truths and predic-

Model	Phrase
BERT	44.46
DeBERTa	57.91
RoBERTa	57.11

Table 2: METEOR Scores of spoilers

tions.

2 Challenges and Resolution Plan

Multipart Approach for Clickbait Spoiling: One of the significant challenges we encountered in our project was developing a multipart approach for clickbait spoiling. This is because there is no existing work that we can use as a reference. Multipart spoilers involve more than one non-consecutive phrase or passage of the linked document, making it challenging to develop a systematic approach. Therefore, we plan to tackle this challenge by approaching it as a text summarization problem. We aim to experiment with extractive and abstractive approaches and compare the models to develop an effective multipart approach for clickbait spoiling.

Generating Feature Vectors for ClickBait Classification: Another challenge we faced was generating proper feature vectors for tf-idf. Our corpus is small, making it difficult to generate accurate feature vectors. To mitigate this challenge, we plan to use some global datasets to get better feature vectors. By doing so, we hope to improve the accuracy of our classification model and achieve better results.

Reproducing the Question Answering SOTA Results: We were not able to reproduce the results achieved in the paper (Hagen et al., 2022) for the question-answering model BERT, RoBERTa, and DeBERTa. This is an essential task as it allows us to establish a baseline and further improve upon it. However, we are making progress and are close to achieving our goal. One of the challenges we face is that the paper does not provide exact hyperparameters. To mitigate this challenge, we plan to explore different hyperparameters and evaluate the performance of the models to achieve better results.

Despite the challenges we have encountered, we believe that we do not need to adjust our project goals. We are on track to completing the project with our expected objectives.

3 Individual Contributions

Ankur Chemburkar:

Involved in the planning of our method for tackling clickbait spoiler generation. Implemented the main code framework, data, training and testing pipelines. Compared between various available datasets and performed the necessary feature extraction for spoiler classification and spoiler generation. Also contributed in the hyperparameter tuning in the model training phase. Performing literature survey for tackling the unsolved multipart spoiling problem using methods such as summarization and Passage Retrieval.

Aravind Krishnan:

Implementation of statistical classification models using TF-IDF features to effectively handle spoiler generation task. Built and trained DeBERTa-lg architecture to perform QA. Involved in hyperparameter finetuning for the QA models.

Nehal Muthukumar:

Worked on building and training the BERT model to establish the baseline for QA models. Understood the working of transformers and QA type NLP tasks. Analyzed various techniques and finetuned hyperparameters to develop the most effective model possible, since the exact hyperparameters for the SOTA Hagen et al. (2022) were not available. In addition, conducted thorough research on summarization methods(abstractive (Etemad et al., 2021) and extractive (Liu, 2019)). This also involved exploring the architecture of BART and Pegasus. This research was necessary to tackle the multipart spoiling problem, which is not a part of our reference paper.

Siraj Sandhu:

Implementation of a standalone evaluation script for integration into training pipeline. Complicated by the convoluted nature of the baseline evaluation, which did not allow ease of use among team members. Performing literature survey for tackling the unsolved multipart spoiling problem using methods such as summarization and Passage Retrieval.

Somil Kiran Jain:

Contributed to the development of the RoBERTa model for spoiler generation to achieve the SOTA result. Additionally, implemented classifiers for the types of spoilers and conducted research on effective methods for classifying multipart spoilers.

References

Abdul Ghafoor Etemad, Ali Imam Abidi, and Megha Chhabra. 2021. A review on abstractive text summarization using deep learning. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–6.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait spoiling via question answering and passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7025–7036, Dublin, Ireland. Association for Computational Linguistics.

Yang Liu. 2019. Fine-tune bert for extractive summarization.