Stephen Colbert at SemEval-2023 Task 5: Using Markup for Classifying Clickbait

Sabrina Spreitzer
University of Regensburg
Sabrina.Spreitzer@student.ur.de

Hoai Nam Tran
University of Regensburg
Hoai-Nam.Tran@student.ur.de

Abstract

For SemEval-2023 Task 5, we have submitted three DeBERTaV3$_{\text{LARGE}}$ models to tackle the first subtask, classifying spoiler types (passage, phrase, multi) of clickbait web articles. The choice of basic parameters like sequence length with BERT$_{\text{BASE}}$ uncased and further approaches were then tested with DeBERTaV3$_{\text{BASE}}$ only moving the most promising ones to DeBERTaV3$_{\text{LARGE}}$. Our research showed that information-placement on webpages is often optimized regarding e.g. adplacement. Those informations are usually described within the webpages markup which is why we conducted an approach that takes this into account. Overall we could not manage to beat the baseline, which we lead down to three reasons: First we only crawled markup for Huffington Post articles, extracting only <p>- and <a>-tags which will not cover enough aspects of a webpages design. Second Huffington Post articles are overrepresented in the given dataset, which, third, shows an imbalance towards the spoiler tags. We highly suggest re-annotating the given dataset to use markup-optimized models like MarkupLM or TIE and to clear it from embedded articles like "Yahoo" or archives like "archive.is" or "web.archive" to avoid noise. Also, the imbalance should be tackled by adding articles from sources other than Huffington Post, considering that also multi-tagged entries should be balanced towards passage- and phrase-tagged ones.

1 Introduction

The shared task "Clickbait Challenge at SemEval 2023 - Clickbait Spoiling" on PAN was about classifying and predicting spoilers from English (gossip) articles (Fröbe et al., 2023a) provided via TIRA (Fröbe et al., 2023b). It poses several technical challenges, especially handling text data with a high degree of diversity in genre, writing style, and structure. The task aims to close the curiosity gap clickbait posts cause within their readers by providing an informative summary. Solving this problem would benefit the use of social media since click baits are an annoying phenomenon aiming to manipulate one into visiting a page which may affect the user's credibility and quality perception in a negative way (Hagen et al., 2022). The task consists of two subtasks:

1. classifying spoilers into three categories: "passage," "phrase," or "multi"
2. predicting spoilers for an article based on those categories

Our system focuses on solving the classification task with a fine-tuned DeBERTaV3$_{\text{LARGE}}$ model trained on the given Webis Clickbait 2022 Corpus, which we enriched with information from markup crawled from the original websites. We have chosen this approach since we discovered that (gossip) articles, in most cases, as webpages, are subject to some professional style guide aiming to make visitors stay on the page, which is essential, especially when it comes to advertising revenues (Nielsen and Pernice, 2009). The results fell short of expectations since we only tested with markup from Huffington Post articles and only from the training dataset. Nevertheless, out of 31 submitting teams for task 1, our team, for most measures, got among the top 10 teams and the top 3 with the highest F1 score in classifying passage spoilers.

2 Background

With the Webis Clickbait 2022 Corpus, the dataset contains 4000 clickbait articles, split into training and validation sets. For solving the classification task, we focused on the following:

- the articles cleared from the advertisement or other web-page-related-noise, e.g., markup (column "targetParagraphs")
- article titles (column "targetTitle")

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1844–1848
July 13-14, 2023 ©2023 Association for Computational Linguistics

- post text (the text that was initially posted on social media when providing a link to the article) (column "postText")
- the article URLs (column "targetUrl")
- spoiler classification (column "tags")

We utilized the state-of-the-art transformer-based architectures BERT (Devlin et al., 2019) and DeBERTaV3 (He et al., 2021), which, by adding a disentangled attention mechanism and an enhanced mask decoder, significantly improves performance on various tasks compared to BERT (He et al., 2021):

- $\text{BERT}_{\text{BASE}}$ uncased for initial parameter evaluation
- $\text{DeBERTaV3}_{\text{BASE}}$ for fast evaluation, especially on our balancing approaches
- $\text{DeBERTaV3}_{\text{LARGE}}$ for the most promising approaches evaluated from the base model and for our markup-based approach

Our experiments show that taking markup into account regarding webpage analysis is quite common in visually rich document understanding. However, since the given dataset does not meet the requirements to utilize models like MarkupLM (Li et al., 2022) or TIE (Zhao et al., 2022), we had to come up with our approach. Due to the small size of the training dataset, containing only 3200 rows, we also used the validation dataset for fine-tuning, which consists of 800 rows, to improve accuracy on every run.

3 System Overview

We initially started with fine-tuning $\text{BERT}_{\text{BASE}}$ uncased to gain fast insights into how parameters influence its accuracy.

Table 1: System setup components for both model sizes

Model size	small	large
Processor	Intel® i7-4790	Intel® Xeon® Gold 6230
RAM	16 GB	128 GB
GPU	NVIDIA® GeForce® GTX 1060	NVIDIA® RTX A6000
VRAM	6 GB	48 GB

Because of the different hardware setups available to us (see Table 1), we used $\text{BERT}_{\text{BASE}}$ and $\text{DeBERTaV3}_{\text{BASE}}$ on the smaller setup for pre-testing purposes and $\text{DeBERTaV3}_{\text{LARGE}}$ to check for any improvements over the pre-tests on the most promising ones. Please mind that the values reported are the results achieved during training.

Runs with $\text{BERT}_{\text{BASE}}$ uncased

First, we tested different sequence lengths on the article text (column "targetParagraphs").

Table 2: $\text{BERT}_{\text{BASE}}$ Uncased results on “targetParagraphs” columns for different sequence lengths

sequence length	acc.	balanced acc.	mcc	macro F1
postText	0.593	0.568	0.340	0.574
targetTitle	0.539	0.521	0.270	0.520

With best performing on a sequence length of 512 (see Table 2), we moved on to combine the “targetParagraphs” column with other columns.

Table 3: $\text{BERT}_{\text{BASE}}$ Uncased results combining “targetParagraphs” column with columns “postText” and “targetTitle”

column	acc.	balanced acc.	mcc	macro F1
postText	0.593	0.568	0.340	0.574
targetTitle	0.539	0.521	0.270	0.520

After conducting a thorough evaluation of the different combinations, we found that combining the "targetParagraphs" and "postText" columns gave us the best results (see Table 3).

Moving from $\text{BERT}_{\text{BASE}}$ to DeBERTaV3

The high level of performance achieved by our first approach demonstrated the effectiveness of fine-tuning BERT and encouraged us to continue exploring the potential of deep learning models for natural language processing. So we re-evaluated our findings with DeBERTaV3.

Both models achieved better results than our pre-evaluated ones (see Table 4).

After we noticed that articles from Huffington Post are by far the most significant cluster with over 774 (19.35%) articles compared to the next highest domain (see Table 5), we also checked the occurrences of tags in the dataset without Huffington Post articles as well as in the Huffington Post articles cluster (see Table 6).

Table 4: Results for DeBERTaV3$_{\text{BASE}}$ and DeBERTaV3$_{\text{LARGE}}$ models on sequence length of 512 and combining columns "postText" and "targetParagraphs"

model	acc.	balanced acc.	mcc	macro F1
base	0.643	0.630	0.426	0.645
large	0.740	0.718	0.585	0.730

Table 5: Domain occurrences listing nones, embedded (yahoo, archives (archive.is, web.archive, etc.)), and domains occurring >= 30

domain	occurrence	occ. in %
none	483	12.08
archives	689	17.23
yahoo	37	0.93
huffington	774	19.35
fraghero	39	0.98
iflscience	35	0.88
business insider	34	0.85
buzzfeed	30	0.75
washington	30	0.75

We figured out two attempts at balancing the dataset: proportional balancing and tag-oriented balancing, which will be described in the following.

Proportional balancing

For the balanced approach, the final portion was gathered by brute-forcing: running the model on the data with a sampled portion on half of all Huffington Post articles and further reducing this amount (see Table 7).

Our model performed best on a sixteenth with a balanced accuracy of 0.661 and a macro F1 score of 0.648, also better than our initial DeBERTaV3$_{\text{BASE}}$ run (Table 4). Therefore we tried this approach also with DeBERTaV3$_{\text{LARGE}}$ (Table 8). Unlike the improvements this adjustment made to our base model, the large one performed worse.

Tag-oriented balancing

We recognized that in the cleared dataset (no Huffington Post articles), while passage- and phrase-tagged articles make up 80.5%, only 19.5% are multi-tagged ones, with passage and phrase not diverging that much (6.9% compared to 23.1% between passage and multi and 17.2% between phrase and multi) (see Table 8). Due to this observation, we ran our DeBERTaV3$_{\text{BASE}}$ model on a combination of the cleared dataset and the dataset containing only Huffington Post articles limited to multi-tagged articles. For the record, we also ran on

Table 6: Tag occurrences in the dataset (without Huffington Post articles) and in Huffington Post only cluster

dataset	cleared	huffington post
total	3053	947
multi	596	106
multi in %	19.5	11.19
passage	1334	262
passage in %	43.6	27.67
phrase	1123	579
phrase in %	36.7	61.14

Table 7: Results for DeBERTaV3$_{\text{BASE}}$ runs on brute-forced amount of Huffington Post samples added to the dataset

sample size	acc.	balanced acc.	mcc	macro F1
1/2	0.654	0.644	0.446	0.658
1/4	0.646	0.634	0.434	0.648
1/8	0.637	0.649	0.437	0.648
1/16	0.655	0.661	0.454	0.664
1/24	0.659	0.652	0.457	0.662

adding the phrase- and passage-limited Huffington Post dataset (see Table 9). This apporach was not further tested with DeBERTaV3$_{\text{LARGE}}$ due to the lower results compared to the achieved balanced accuracy with $\frac{1}{16}$ of Huffington Post.

3.1 Dataset enrichment with markup

While checking the domains and URLs, we found that 483 entities had no URL (see Table 5). Also, 689 were embedded articles from websites like "Yahoo" or archived versions of the original articles (which influences the markup structure making it noisier). Furthermore, while Huffington Post was overrepresented, it was the domain with the best reachable articles. So for fast evaluation of our markup hypothesis, we focused on crawling the source code for Huffington Post articles in the training and validation dataset. To make it utilizable within machine learning, we cleared the HTML code to take items in <p>- and <a>-tags only into account. This approach was directly tested with DeBERTaV3$_{\text{LARGE}}$ (see Table 10).

4 Experimental Setup

We evaluated three transformer-based models (for specifications, see Table 11), combining train and validation datasets. Utilizing BERT$_{\text{BASE}}$ to find the best working specifications, DeBERTaV3$_{\text{BASE}}$ to fast evaluate our balancing approaches (see Sys-

Table 8: Results for $\frac{1}{16}$ of Huffington Post on DeBERTaV3$_{\text{LARGE}}$ run

acc.	balanced acc.	mcc	macro F1
0.696	0.677	0.517	0.687

Table 9: Results for DeBERTaV3$_{\text{BASE}}$ runs on dataset combined with samples from Huffington Post cluster based on given tag

tag	acc.	balanced acc.	mcc	macro F1
multi	0.675	0.647	0.488	0.668
phrase	0.650	0.648	0.443	0.655
passage	0.650	0.652	0.443	0.662

tem Overview and DeBERTaV3$_{\text{LARGE}}$ as final submission model on the most promising runs from DeBERTaV3$_{\text{BASE}}$ and our markup approach. We obtained the models from the Hugging Faces transformers library for PyTorch.

For our markup-based approach, we crawled the original articles using BeautifulSoup and cleared the results with a simple RegEx pattern (see section Dataset enrichment with markup for a more detailed description)

We submitted three DeBERTaV3$_{\text{LARGE}}$ models to TIRA, fine-tuned on the validation set, the validation set containing only $\frac{1}{16}$ of all Huffington Post articles and the validation set enriched with markup for Huffington Post articles.

5 Results

As shown in Table 12, we could not surpass the baseline accuracy of 0.74 with any of our approaches achieving the highest accuracy with the combined approach. Comparing more specifically based on the three tags, we achieved a higher precision on multi-tagged articles with our full-combined dataset model and the $\frac{1}{16}$ HuffPost one. With 0.54 precision, the markup-enriched model does worse on a multi-tag prediction but achieves the highest precision on phrase-tagged ones. Finally, we see the highest recall of 0.79 for the full-combined dataset and the $\frac{1}{16}$ HuffPost run.

Compared to other submissions, we achieved the highest recall in detecting multi-part spoilers and got among the top 3 with the highest F1 score in classifying passage spoilers.

Table 10: DeBERTaV3$_{\text{LARGE}}$ run on markup enriched dataset

acc.	balanced acc.	mcc	macro F1
0.693	0.691	0.527	0.679

Table 11: Specifications for best performing transformer-based evaluation

model	learning rate	epochs	sequence length
BERT$_{\text{BASE}}$	2e-5	4	512
DeBERTaV3$_{\text{BASE}}$	2e-5	4	512
DeBERTaV3$_{\text{LARGE}}$	6e-6	5	512

6 Discussion and Future Work

Since we had many mixed results, we recommend further balancing the dataset, especially to compensate for the overrepresentation of Huffington Post articles and the underrepresentation of multi-tagged articles. Since our markup-based approach resulted in the lowest accuracy, we assume this is because we only considered articles from Huffington Post. Also we only managed to focus on <p>- and <a>-tag extraction which won't cover all relevant aspects within webpage design. Since with MarkupLM and TIE, there are high-performing models to use for domains like webpages to solve prediction and classification tasks, we strongly recommend taking this into account for annotating clickbait articles in the future, always saving the pages' entire markup within the dataset and avoiding embedded sources like "Yahoo" or archives like "archive.is" since they build markup around the embedded article and might leave out content like an advertisement which could hint to the position of the spoiler in the text.

7 Conclusion

We submitted several approaches, utilizing DeBERTaV3$_{\text{LARGE}}$ to solve the classification task of SemEval-2023's Task 5. Although we could not quite reach the baseline's accuracy we could point out weaknesses within the given dataset caused by the overrepresentation of Huffington Post articles and the underrepresentation of multi-tagged entries. Also, it misses providing the original markup, which we tried to crawl retroactively after we figured out that design is one of the critical parts of information placement on the website. This approach was not very successful, which might be because

Table 12: Overview of the effectiveness in spoiler type prediction (subtask 1 at SemEval 2023 Task 5) measured as balanced accuracy over all three spoiler types and precision (Pr.), recall (Rec.), and F1 score (F1) for phrase, passage, and multi spoilers on the test set. We report all runs by Team stephen-colbert.

Submission	**Accuracy**	**Phrase**			**Passage**			**Multi**		
		Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1
Baseline	0.74	0.76	0.75	0.76	0.73	0.76	0.74	0.74	0.70	0.72
DeBERTaV3$_{\text{LARGE}}$ (full combined)	0.70	0.75	0.74	0.74	0.71	**0.79**	**0.75**	0.76	0.57	0.65
DeBERTaV3$_{\text{LARGE}}$ (with $\frac{1}{16}$ HuffPost)	0.68	0.75	0.69	0.72	0.66	**0.79**	0.72	**0.77**	0.57	0.65
DeBERTaV3$_{\text{LARGE}}$ (HuffPost Markup)	0.67	**0.77**	0.60	0.67	0.67	0.76	0.71	0.54	0.67	0.60

we only crawled the markup for Huffington Post articles since they were best reachable at this time, only extracting <p>- and <a>-tags and therefore only covering a small aspect of markup design possibilities. After finding models like MarkupLM and TIE, which promise to perform better on markup-based presentations like webpages, we recommend re-annotating the dataset to use those.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maik Fröbe, Tim Gollub, Benno Stein, 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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *arXiv preprint arXiv:2111.09543*.

Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. 2022. MarkupLM: Pre-training of text and markup language for visually rich document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6078–6087, Dublin, Ireland. Association for Computational Linguistics.

Jakob Nielsen and Kara Pernice. 2009. Eyetracking web usability. USA. New Riders Publishing.

Zihan Zhao, Lu Chen, Ruisheng Cao, Hongshen Xu, Xingyu Chen, and Kai Yu. 2022. TIE: Topological information enhanced structural reading comprehension on web pages. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1808–1821, Seattle, United States. Association for Computational Linguistics.