ClickBait Spoiling using Transformer and Classical Ensemble

1. Abstract

The primary motivation behind this study is to develop a machine-learning model capable of classifying click-bait spoilers using parameters such as the post-text length, source of the post etc. This can serve as a tool for detecting spoilers in a particular social media post, and the feeling of being played can be avoided.

In this study, various classification models (both classical and transformer-based models) have been trained on the webis clickbait spoiling (2022) dataset. Moreover, we have devised ensemble methods, first a transformer model and then a classical model applied to the logits obtained from the transformer to get the results.

Additionally, we have also devised models to generate spoilers with the help of the pre-trained T5 model. This is done in order to directly show the spoiler to the person and save their time, this would also help the person stay clear of the feeling of getting played.

2. Introduction

The Goal of SemEval-2023 Task 5 is to spoil click-bait. Posts all around the internet use click-baits to arouse user curiosity to increase the number of clicks. While seeming harmless, this method preys on user's curiosity to show them advertisement by purposefully teasing and leaving out key information from an article to advertise a web page's content. The aim of this task is to classify click-baits into three classes, and provide a short text that satisfies this curiosity of the user, as in "spoil" the click-bait.

We felt motivated to do this task because of how click-baits right now completely overrun the internet, and it feels as if each and every link we encounter on the internet wants to gobble up our attention, but does not provide us any information of value. Most of the click-baits are on Twitter, Facebook, News websites which can be quite misleading for the readers.

Figure 3 shows an example of a short text that we need to generate to spoil the click-bait. In this paper, we attempt both task 1 and task 2. We have shown a comparison between classical models and different transformer models. For task 1, we found an ensemble between transformer and classical models to be the best with the limited resources we had. For task 2, we tried zero-shot passage retrieval for training the T5-base and also the ensemble between the T5-base.

3. Literature Survey

[3] nancy-hicks-gribble at SemEval-2023 Task 5: Classifying and generating clickbait spoilers with Roberta

The research paper aimed to classify and generate spoilers with the help of classical machine learning and transformer-based approaches. For the classification task, the author proposed the Shallow Learned Approach and the Transformer Approach.

For the Shallow Learned Approach, classical machine learning models, like Random Forest, Logistic Regression, Multinomial Naive Bayes, etc, were applied to a dataset that was pre-processed via standard techniques like lemmatization stemming, etc, to extract relevant features, which were then concatenated with the target paragraph and fed to the models via TF-IDF vectoriser. These classical models were used to establish baselines for further experimentation.

Figure 1. spoiler

For transformer-based approaches, RoBERTa-based models, without any pre-training and the ones that were pre-trained on the HuffPost Dataset, were utilised. Different combinations of the concatenation of the various fields of the input instances were fed into the transformer-based models, and the final classification was done. Overall it was observed that the Roberta model pre-trained on the HuffPost dataset performed better, but the performance jump observed was minimal.

For the Spoiler Generation task, RoBERTa models were again utilised, and the task was modelled as a question-answering cum retrieval task. The post-title of the click-bait post was fed as a question to the model, which retrieved the top-k most feasible spans from the target paragraph as answers, and out of the ones that were extracted one or more were concatenated together to generate the spoiler. Overall it was observed that the best performance was obtained in terms of the BLEU score in case of the phrase spoilers.

[2]Clickbait Spoiling via Question Answering and Passage Retrieval

The research paper models the spoiler generation as a Question Answering task which takes the post text as the question and target link paragraph as the context. The question is answered from the context. The models used were RoBERTa, ELECTRA, ALBERT, DeBERTa.

The paper also models the task as a Passage Retrieval task. The spoiler is generated by retrieving a passage from the context. The paper explores the use of various retrieval models and algorithms. The authors experimented with MonoBERT, MonoT5 and BM25.

The phrase generation was done by training the QA models on

SQUAD dataset and using the context to answer the questions. Passage generation used the retriever models to retrieve the best passage from the linked text. The results showed that modelling the task as a question-answering task gave better spoilers. **[1]Clickbait Spoiling via Simple Seq2seq Generation and Ensembling**

The research paper attempts only task-2. It tries to model the spoiler generation task as a Question answering task. They use various versions of T5 to do this task. They create an ensemble model that uses n T5 models trained using different seeds. Their conjuncture being that these models will learn a bit differently from each other. They use these to generate n spoilers for a single click-bait. They used edit distance as the metric to select the best spoiler. They calculated edit distance between all n predictions, and choose the one with least cumulative distance from all the others. Their model's results for this task were surprising both for the authors of the paper, and that of the task. Their common belief was that retriever type models would be better, but their model was able to achieve best score the task.

[4] Chick Adams at SemEval-2023 Task 5: Using RoBERTa and DeBERTa to extract post and document-based features for Clickbait Spoiling

The main aim of this research paper is to do the spoiler classification taks using an ensemble of Roberta and Dberta models. The first step involved in the methodology followed by this paper focused on removing all the social media hashtags and emojis. The cleanup step was then followed by the splitting step in which the document based features were separated from the text based features. The text based features involved a concatenation of post and targetTitle. The document based features were summarized using sentence transformers to a certain token length not beyond 512 . This resulted in the creation of two seperate datasets one, consisting of all the document based features and the other consisting of all the spoiler based features. These were then input into the RoBERTa and DeBERTA based models and the final output was taken as average or max of the outputs of the two models.

4. Data Analysis

4.1. Size and shape of the dataset

The dataset for this project has been obtained from the *webis clickbait spoiling corpus 2022*. The original training dataset consists of 3200 posts, the validation set contains 800 posts and the testing dataset contains 1000 posts, a total of **5000** posts (rows) and **14** columns. The posts were gathered from various social media platforms, such as Facebook, Reddit, and Twitter.

4.2. Exploratory Data Analysis

We have used a pie chart (Figure 2) to display the distribution of the tags. The distribution shows a clear imbalance with the *phrase* and *passage* spoilers, being almost 2 times more than that of *multi*. To tackle this imbalance in the dataset, we decided to increase the entries for multi-label spoilers. We did this by doubling the number of multi-label entries. After training our model on this dataset, we noticed an increase in the balanced accuracy (0.03) and a decrease in the F1 score (0.07). Therefore, we decided to continue with the imbalanced dataset.

The number of *targetMedia* entries that were 0 or None in

Feature	Description
uuid	Unique identifier of the entry
postId	Identifier of the post
postText	Text of the post that is to be spoiled
postPlatform	Platform of the post
targetParagraphs	Main content of the linked webpage
targetTitle	Title of the linked webpage
targetDescription	Description of the linked webpage
targetKeywords	Keywords involved in the linked webpage
targetMedia	Media associated with the linked webpage
taregtUrl	URL of the linked webpage
provenance	It provides temporal, spacial and background information of the post
spoiler	The human extracted spoiler for the post from the linked webpage. Not available in the testing dataset.
spoilerPositions	The position of the human extracted spoiler for the clickbait. Not available in the testing dataset.
tags	These are the labels that are to be classified in task 1, they are not available in the task1 however they can be used in the task 2.

Table 1. Dataset columns

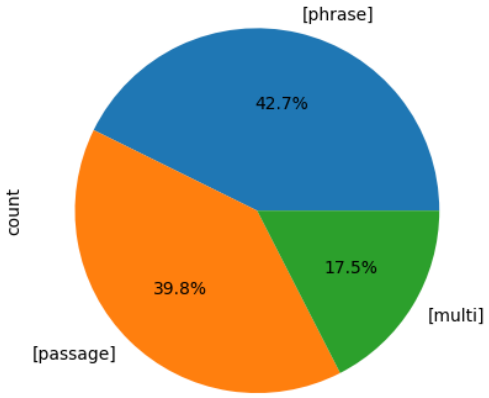

Figure 2. Pie Chart

Tags	Post Text	Paragraph Number	Paragraph Length
multi	62.99	22.94	4404.15
passage	61.16	13.14	2993.54
phrase	62.22	11.59	2485.60

Table 2. Mean values

length, was also almost half of the total entries for the multi-label (0.39), however it was near to 10% for both passage (0.08) and phrase (0.14) labels

4.3. Pair Plots and Correlation Heat Map

The correlation heatmap shown in **Figure-3** is the correlation between the different data features, including the target attribute. The values and colour of each cell indicate the degree

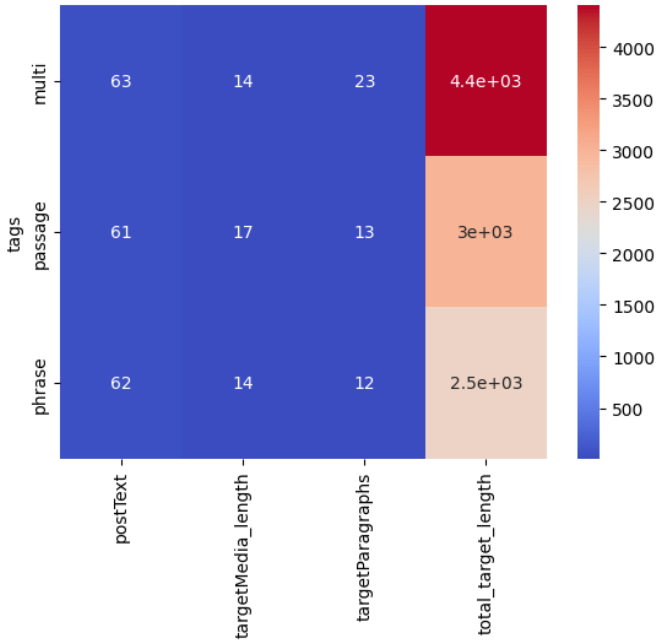

Figure 3. Correlation Heatmap

of correlation.

5. Task-1: Spoiler Type Classification

5.1. Task Description

In the Spoiler Type Classification task, the input provided consists of a click-bait post along with other metadata and provenance information about the post and other information like a description of the linked target article and keywords in the target article, along with the spoiler generated for the click-bait post. The goal is to classify the type of the spoiler into one of the three categories, viz: **Phrase**, **Passage**, **Multi** denoting the kind of spoiler needed to spoil the post.

1. Phrase spoilers consisting of a single word or phrase from the linked document
2. Passage spoilers consisting of a few sentences
3. Multi spoilers consisting of multiple non-consecutive passages or phrases.

The metrics used to judge the model performance on these sub-tasks are: **balanced accuracy** and **macro F1-Score**

5.2. Models and Methodology

Our methodology consisted of modelling the problem as a sequence classification task, which involved three main steps:-

The first step involved training classical machine learning models on the training data and using them to establish baselines. The second step involved fine-tuning transformer-based models like RoBERTa and DeBERTa on the dataset for the Sequence classification task. The third step involved creating ensemble-based machine-learning models using a combination of the above-trained classical/fine-tuned transformer-based models.

We report Macro F1-Score and Balanced Accuracy as evaluation metric for each of our models.

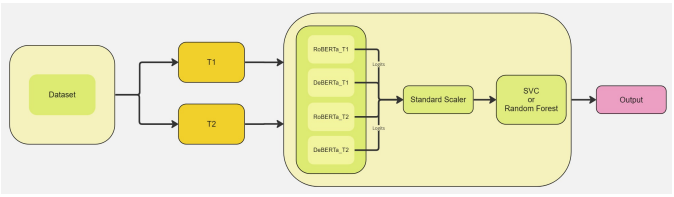

Figure 4. Model Architecture

5.2.1 Pre-Processing for Classical Models

For applying the classical machine learning models to the input data, our pre-processing step consisted of concatenating all the fields for a given input instance from the dataset post using a [SEP] token between each field.

The concatenated input was passed through a TF-IDF vectoriser, and the output was fed to the machine learning models.

5.2.2 Classical Machine Learning Models

For each of the classical machine learning models, a pipeline was setup consisting of the TF-IDF vectoriser followed by the model which allowed for simultaneous tuning of the hyperparameters of both the vectoriser and the model via K-Fold Cross Validation using GridSearchCV. The classical models used were Multinomial Naive Bayes, Logistic Regression, Support Vector Machines, XGBoost and Random Forest . For XGBoost, TF-IDF was omitted from the pipeline.

The details about the metrics obtained are summarised in Table 5.2.2. The hyperparameters are provided at the end in Table 7.

Model	**Balanced Accuracy**	**Macro F1-Score**
Multinomial Naive Bayes	0.526	0.577
Logistic Regression	0.555	0.519
Support Vector Machines	0.523	0.534
XGBoost	0.560	0.555
Random Forest	0.593	0.556

Table 3. Balanced accuracy and macro F1-score obtained using classical models

5.2.3 Pre-Processing for Transformer Based Models

Taking inspiration from the ensemble of the RoBerta and DBerta as mentioned in [4] we propose our own way of creating an ensemble of the Roberta and DBerta based models as discussed below:-

For the transformer-based models RoBERTa-base and DeBERTa-base were fine-tuned, and for each of the models, two types of pre-processing techniques were followed:-

Technique-1: The first technique involved concatenating the **postText**, **targetTitle** and **targetKeywords** using a space between them for each input instance. To this string, the **targetDescription** was appended, separating it from the rest using **[SEP]** token.

Input-Format for T1

"postText targetTitle targetKeywords [SEP] targetDescription"

Technique-2: The second technique involved concatenating the **targetParagraphs** field with the **postText** using the **[SEP]** token.

Input-Format for T2
"targetParagraphs [SEP] postText"

5.2.4 Transformer Based Models

Passing the input in the format specified above, resulted in four sets of models being created that have been labelled as Roberta_T1, DeBerta_T1, Roberta_T2, DeBerta_T2 where T1 and T2 refer to the technique in which the input data has been fed into the model during the fine-tuning process as described above. The final results obtained are summarised in Table 5.2.4

5.2.5 Ensemble Based Models

After fine-tuning the transformer-based models, a set of four logits was generated for each instance of the dataset after passing the input to each of the four models in the format described above.

These four logits were then concatenated and passed once through the Random Forest classifier and the second time through the SVM Classifier (Figure 4). Grid search was applied for hyper-parameter tuning of the classical models via GridSearch CV. The results obtained for each ensemble method are summarised in Table 5.2.5.

5.3. Intuition behind the choice of models :

- An interesting observation we made was that our classical machine-learning models trained via simple concatenation of the fields followed by tf-idf vectoriser outperformed the classical machine-learning models implemented in [3] even without any explicit feature extraction.
- Our choice of the RoBERTa base model over the standard BERT model for our sequence classification task can be attributed to its dynamic masking ability, allowing the model to focus on different parts of the input text. Since the input to our models involves concatenating various fields of the input instances from the dataset using the techniques specified above, we did not want our model to focus on one part of the input text, which often happens in the case of static masking as in the case of BERT model.
- Our choice of DeBerta model over standard BERT model for our task can be justified by the disentangled attention mechanism followed by DeBERta. In case of standard BERT models, the positional embeddings and the word-embeddings are added together, thus leading to loss of information making it impossible to give more-attention to the position of the word or the embedding of a word. In case of DeBerta, however due to the dientagled attention mechanism, the model is able to decide whether to give more attention to the word-embedding or the positional embedding. This is particularly useful in our case, since our dataset includes a target paragraph along with the information about the span of the spoiler, so having dynamic priority for positional encoding and word embedding is useful.
- The intuition behind using an ensemble based-approach can be attributed to the fact that, a direct concatenation of all the input fields would exceed the max-sentence length for RoBerta and DeBerta based models which is 512 tokens, and our main goal was to capture as much information as possible is retained. Thus we decided to train two different models of RoBerta and DeBerta model using two techniques T1 and T2 as mentioned above so each technique having a different set of fields in ensemble, and then concatenating the inputs together so that the information loss is minimized.

5.4. Observations and Analysis

- It can be seen that the classical machine learning models performed decently well, giving both macro f1-score and balanced accuracy in the 50%-60% range.
- An interesting observation we made was that our classical machine-learning models trained via simple concatenation of the fields followed by tf-idf vectoriser out-performed the classical machine-learning models implemented in [3] even without any explicit feature extraction.
- For the transformer based models, we observed an increase in the performance on both the evaluation metrics because of the presence of attention mechanisms.
- Ensemble based models, resulted in a further improvement of the performance due to the averaging of results of multipled models and reduced information loss.

6. Task-2: Spoiler Generation

6.1. Task Description

In spoiler Generation task, we had to generate a short text to satisfy the curiosity introduced by a clickbait post. The data for this consisted of the same data as the Task-1, just that we could access the tags as well consisting of phrase, passage and multi, the type classification used in task-1. For every tag we had to generate a spoiler of it type.

6.2. Models and Methodology

For the task of spoiler generation, we decided to use T5-base and fine tune it for question answering task. We treat the postText as the question and provide the context through targetParagraphs and targetDescriptions. We have reported Blue-4, Rouge-4, BertScore, and Meteor for all the models that we have trained.

6.2.1 QA-T5

For Fine tuning T5-base model on Question answering, we took a simple approach of possing the clickbait itself as the question, and the spoiler as the answer that we wish to generate. We provide the textParagraphs as the context for the question. This naive approach may not seem to be very interesting, but it was able to generate phrase type spoilers pretty well. While it struggled with learning to generate phrase and multi spoilers

Model	Technique	Balanced Accuracy	Macro F1 Score
Roberta	Technique-1	0.671	0.692
Roberta	Technique-2	0.605	0.616
Deberta	Technique-1	0.66	0.68
Deberta	Technique-2	0.61	0.63

Table 4. Description of Transformer Fine-tuning

Model	Balanced Accuracy	Macro F1 Score
Random Forest Ensemble	0.696	0.706
SVM Ensemble	0.698	0.706

Table 5. Description of the Ensemble Models

when trained on the whole dataset, it was able to generate these which much better when trained specifically on these tags only. We also tried to use t5-base-finetuned-question-answering similarly. Table 6 has the results to all these models

6.2.2 Retriever-QA

We used a retriever based approach for providing only the relevant context to the QA model T5-base. The T5-base was further fine-tuned to answer the questions. For retriver pipeline langchain was used which creates an embedding database and a model retrives the most similar context to the click-bait text. The embeddings were created with Instructor-XL and the FAISS langchain library was used to retrive similar text. This approach showed good performance for phrase but a poor performance for passage and multi.
The poor performance may be due to the fact that the passage which could spoil the click-bait was not retrived as we hadn't tuned the retriever specifically for our task.

6.2.3 T5-Ensemble

We used Hiroto Kurita's[1] to create an ensemble model which predicts the spolier using n models and then selects the best one using edit-distance. This model did not work good for us, as we were only able to train 3 models and the edit-distance was not able to select the best result. We also ran into an error while that might non trivial. For larger spoilers, edit distance might not be the best metric to select the best spoiler as it takes $O(n^2)$time to compute for a singular spoiler. Doing this ensemble for larger spoilers can be very resource intesive.

We also tried to use the create and Ensemble by using the tag specific models. This model performed better than all other models that we tried and gave the best results. Table 6.2.3 contains the results of a single T5, and ensemble of tag-wise T5 models.

6.2.4 Observations and Analysis

1. The results indicate that the performance of different models varied across different types of spoilers. While some models performed well for generating phrase spoilers, they struggled with passage and multi spoilers. This suggests that the complexity and diversity of spoiler types pose a challenge for the models.

2. Fine-tuning the T5-base model for question answering seemed to be a promising approach. However, its performance still showed limitations, especially for generating spoilers of certain types. This could imply that the task of generating spoilers requires more nuanced understanding and fine-tuning than simple question answering.

3. The retriever-QA approach, aimed at providing relevant context to the QA model, faced challenges in generating spoilers for passage and multi types as it heavily relies on the quality and relevance of the retrieved context.

4. The ensemble models, while conceptually sound, faced challenges in implementation due to the computational complexity of the algorithm and on the other hand increased the complexity of the models by many folds only to provide diminishing returns.

7. Conclusion and Future Tasks

- From the above analysis, we can conclude that, the generation task is a much more complicated task as compared to the generation task as is evident from the evaluation metrics and the results obtained. The reason for this can be attributed to the fact that generation requires complex understanding of the language and the nature of the post by the model.

- The process of classification of a clickbait type post, helps in the generation process, since then by the type of the clickbait generated, the model knows what is the length of the spoiler which is to be generated and hence classification and generation go hand in hand.

- Generating spoilers by simple retrieval of the data, did not prove to be benificial ,simply because, it needed explicit fine-tuning of the model on the training dataset, that was not possible due to the lack of resources.

We could have produced by results by training retriever to get much better context for our QA models which would allow models to learn and predict better. This could also be paired with an LLM to generate spoilers that are not directly retrieved from the context but are still able to satisfy the the curiosity of the user.

Further, we realized that the spoilers that generated by our models usually contained the gold spoilers or were spoiling the

	BLEU-4			Rouge-1			BertScore			Meteor		
	Passage	Phrase	Multi	Passage	Phrase	Multi	Passage	Phrase	Multi	Passage	Phrase	Multi
T5-base	0.13	0.21	0.22	0.31	0.65	0.40	0.87	0.92	0.88	0.25	0.56	0.34
T5-phrase	0.00	0.43	0.00	0.15	0.66	0.22	0.84	0.92	0.85	0.10	0.54	0.14
T5-pass-age	0.21	0.04	0.08	0.36	0.26	0.22	0.88	0.85	0.84	0.32	0.33	0.18
T5-multi	0.10	0.07	0.28	0.23	0.40	0.41	0.83	0.86	0.88	0.20	0.41	0.34
T5-Qa	0.11	0.09	0.28	0.25	0.52	0.36	0.86	0.90	0.87	0.21	0.45	0.30
Dist-Ens	0.12	0.20	0.22	0.31	0.64	0.40	0.87	0.92	0.88	0.25	0.55	0.32

Table 6. Scores

	B4	R1	BSc	MTR
T5	0.29	0.49	**0.90**	0.43
T5-ensemble	**0.31**	**0.51**	0.89	**0.45**

Table 7. Result of singular T5 and ensemble T5 on the the whole data

click-bait pretty well, but still were not resulting in good metrics. We would love to implement a metric that takes these things into account like presence of subarrays in two sentences, and how they might affect the identification of good spoiler.

The goal of people using click-baits is to increase the number of clicks that their post/website gets. On one hand, this task could reduce the number of click-through rates and revenue of websites that rely on click-bait tactics. However, there is a counter-ethical argument that clickbait is a form of deception and detecting them would help the people save their time, and avoid getting tricked by clicking on content that doesn't match the headline that it was assigned because of the click-bait.

Model	**Hyperparameters**
Multinomial Naive Bayes	tf-idf vectoriser: Ngram-range: (1,2) Stop-words: None Model:Alpha () : 0.5
Logistic Regression	tf-idf vectoriser: Ngram-range: (1,2) Stop-words: None Model: Regulaization Constant (C): 2.5
Support Vector Machines	tf-idf vectoriser: Ngram-range: (2,2) Stop-words: None Model:Regulaization Constant (C) : 1.8 Kernel: Linear
Xg-Boost	Model: learning_rate:0.1 n_estimators=1000 max_depth=7
Random Forests	tf-idf vectoriser: Ngram-range: (1,1) Stop-words: None Model:Optimal estimatmors: 200 Optimal Depth: 10 Optimal Min Leaf: 1 Optimal Max Features: sqrt Optimal Bootstrap: False

Table 8. Hyperparameters

References

[1] Tugay Bilgis, Nimet Beyza Bozdag, and Steven Bethard. Gallagher at SemEval-2023 task 5: Tackling clickbait with Seq2Seq models. In Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori, editors, *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1650–1655, Toronto, Canada, July 2023. Association for Computational Linguistics.

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

[3] Jüri Keller, Nicolas Rehbach, and Ibrahim Zafar. nancy-hicks-gribble at SemEval-2023 task 5: Classifying and generating clickbait spoilers with RoBERTa. In Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori, editors, *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1712–1717, Toronto, Canada, July 2023. Association for Computational Linguistics.

[4] Ronghao Pan, José Antonio García-Díaz, Franciso García-Sánchez, and Rafael Valencia-García. Chick adams at SemEval-2023 task 5: Using RoBERTa and DeBERTa to extract post and document-based features for clickbait spoiling. In Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori, editors, *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 624–628, Toronto, Canada, July 2023. Association for Computational Linguistics.