Sam Miller at SemEval-2023 Task 5: Classification and Type-Specific Spoiler Extraction using XLNET and other Transformer Models

Pia Bernards
TH Köln
pia.stoermer@smail.th-koeln.de

Tobias Esser
TH Köln
tobias_vincent.esser@smail.th-koeln.de

Patrick Thomasius
TH Köln
patrick_christian.thomasius@smail.th-koeln.de

Abstract

This paper proposes an approach to classify and generate spoilers for clickbait articles and posts. For the spoiler classification, XLNET was trained to fine-tune a model. With an accuracy of 0.66, 2 out of 3 spoilers are predicted accurately. The spoiler generation approach involves preprocessing the clickbait text and post-processing the output to fit the spoiler type. The results were evaluated on a test dataset of 1000 posts, with the best result for spoiler generation achieved by fine-tuning a RoBERTa Large model with a small learning rate and sample size, reaching a BLEU score of 0.311. The paper provides an overview of the models and techniques used and discusses the experimental setup.

1 Introduction

In advertising and social media, there are common patterns and tactics that are employed to capture and capitalize the attention of consumers. One of those is the use of so-called "Clickbait", which create curiosity in the users that motivate them to click on the articles (Hagen et al., 2022a), (Loewenstein, 1994). The proposed solution, therefore, is to generate an approach that automates the generation of a spoiler, that spoils the information needed to close the users curiosity gap. The user can then make a more informed decision whether they want to proceed. The clickbait posts and articles are mostly in English language. The task is split into two parts, in the first task, the type of the required spoiler is determined, whether it's a shorter phrase, a longer passage or a multipart spoiler. In the second part, the actual spoilers will be generated or extracted.

Given task 2, our approach was to fine-tune a question answering model and use post-processing the respective spoiler type, so that the format and length of the answer fits the requested spoiler type. For multipart spoilers, recursive spoiler generation was used to generate multiple answers. Furthermore, the possibility of doing preprocessing in the form of query rewriting was explored.

We tried different pretrained models, and larger models with more pretraining (RoBERTa Large Squad-2) outperformed smaller models significantly. Our result for task 2 was comparable to the other groups, but worse than the Baseline (0.311 BLEU on our approach compared to 0.38 on the transformer baseline (Hagen et al., 2022b)).

Models were added to the Docker Images (Merkel, 2014) to ensure replicability.

1.1 Data

The training data consists of 3200 posts, the validation dataset contains 800 posts and the test dataset, which was private during development, contains another 1000 posts. Most spoilers are extracted, with a small minority of abstractive spoilers in train and evaluation datasets. Multipart spoilers are supposed to consist of 5 items that can stem from nonconsecutive points in the post. Phrase spoilers should be shorter than 5 words, spoilers longer than 5 words are classified as passage spoilers.

Out of the corpus of training and validation data, the vast majority of posts could be classified as written in the English language (3909 out of 4000). The remaining posts were distributed over numerous languages.

2 Background

The tasks specified input and output formats of the Software. As input, a JSON File with several features for the respective tasks was given. All paragraphs were separated in the input files. The output file must also be a JSON file, while the format depends on the task. For task 1, it must contain the UUID and the predicted spoiler type. For task 2, it must contain the UUID and the spoiler as a string. The train dataset was used to refine our approaches, and the validation dataset was used to test our approaches.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1217–1224
July 13-14, 2023 ©2023 Association for Computational Linguistics

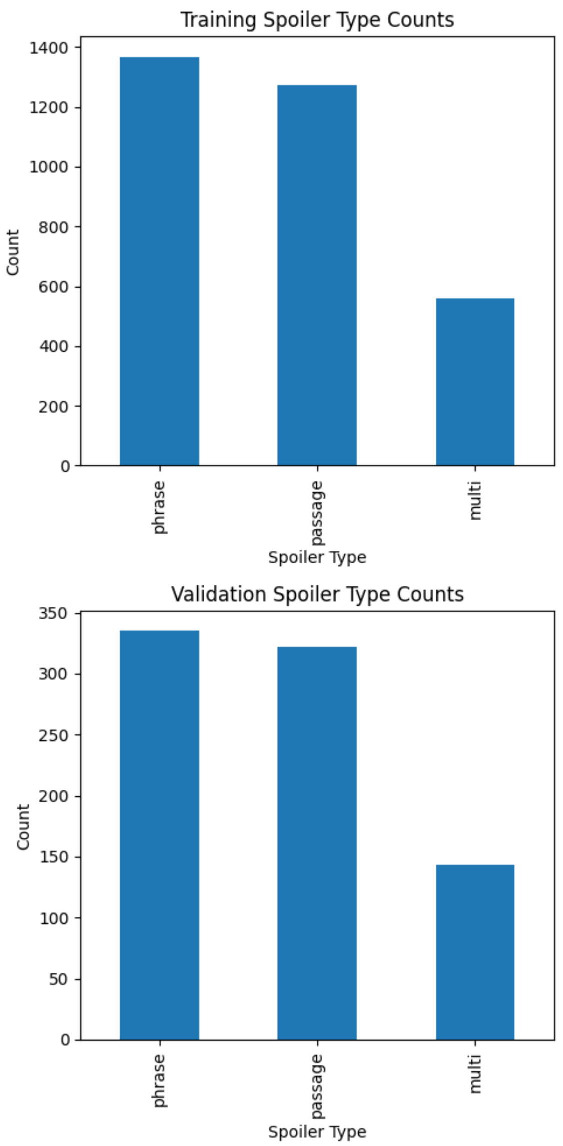

Figure 1: Distribution of different Spoiler Types per Dataset.

The evaluation was generally performed via Docker images and Tira [1]. The BLEU score (Papineni et al., 2002) was calculated on the Tira service(Fröbe et al., 2023a). For task 1 and 2, the best performing images can be accessed via the link given in the appendix.

3 System Overview

3.1 Task 1

The spoiler classification part is done with a fine-tuned XLNET model (Yang et al., 2019). XLNET works as a generalized autoregressive model, where generalized means that it does not only train on close permutations like left-right or right-left, but all possible permutations. This is called Permutation Language Modeling (PLM). Another addition to XLNET was the model to be autoregressive. Bert does not use the predicted tokens to predict new tokens in a multipart spoiler, whereas XLNET does use the newly predicted token in their prediction of new tokens (Yang et al., 2019).

The XLNET model is one of the largest Language models, as it is trained with 32.89 Billion tokens (Yang et al., 2019). This means that the model works for a variety of use cases. Fine-tuning the model leads to additional optimization for a specific use case.

3.2 Task 2

We decided against the use of XLNET for task 2 because implementing and training the model would have been very time consuming, and team members had previous experience with BERT models. DistilBERT and RoBERTa are two popular models for natural language processing, such as question answering. Both are based on the Transformer architecture, which has proven to perform well on many natural language tasks (Wolf et al., 2019).

Both models are pre-trained models, which means that they were trained on large amounts of text data to learn underlying patterns and structures of natural language. This pre-training process allows the models to be fine-tuned on smaller datasets, such as the one we used for our task. This enables them to achieve high performance.

DistilBERT is well suited for tasks with limited computational resources, since it is a smaller and faster version of the original BERT model. It uses the same architecture as BERT, but has fewer layers and fewer parameters, which reduces the size and computation required (Sanh et al., 2019).

RoBERTa is a variant of the BERT model. It is trained with a larger data set and a longer training period. It uses a similar architecture as BERT, but introduces some changes to the pre-training process, such as the removal of the next sentence prediction task and the use of dynamic masking instead of static masking. It is these changes that allow RoBERTa to achieve top-notch performance on a wide variety of natural language tasks (Liu et al., 2019).

The process of the spoiler generation performed for task two is shown in Figure 2. Before the actual

[1] https://www.tira.io/

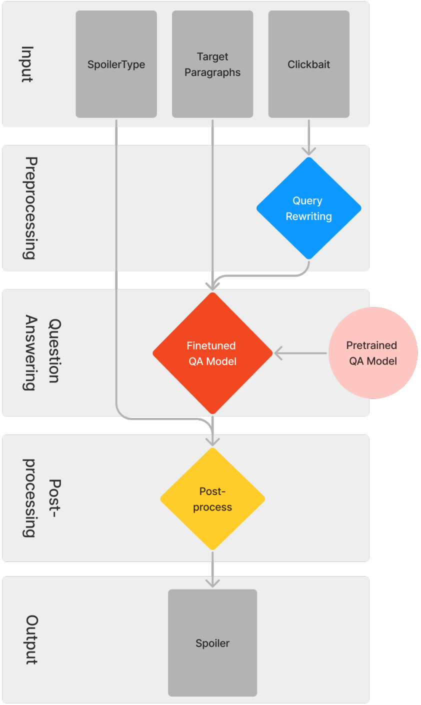

Figure 2: Process of Spoiler Generation (Task 2).

question answering, a preprocessing in the form of query rewriting was performed. In our question answering task, we used DistilBERT and RoBERTa to fine-tune the pre-trained models on our particular train dataset. Based on the information in the passages, the fine-tuned models were then used to predict the answer to new questions. To generate more accurate spoilers, the spoiler type was taken into consideration and a type-specific post-processing was performed.

The models were fine-tuned using google colab and paperspace gradient. In the beginning, for the DistilBERT models 3, query preprocessing was not included in the fine-tuning process. It was, however, added later for the larger models.
DistilBERT was trained with a moderate learning rate (3e-5). In the later steps, bigger models were trained. Performances for Roberta trained with and without the preprocessing can be seen in table 1. In the final model, RoBERTa large was trained using a small batch size and a small learning rate (1e-5). We decided against the use of the post-processing to calculate the loss during the fine-tuning process, since post-processing mainly presents a "correction" on answers that are inadequate for the specific spoiler type in the later generation process. If predicted spoilers are too short, the post-processing will retrieve the entire sentence. For DistilBERT, we also attempted to solve the task once with the squad-2 pretrained model without fine-tuning and once with the same model but with the additional fine-tuning applied. For RoBERTa, pretrained models used were RoBERTa and RoBERTa large, both pretrained on squad-2 and fine-tuned on the training data. Afterward, the differences in model performance were compared to the data in the literature (Adoma et al., 2020) (Khan, 2019).

According to hardware limitations and the models used, some hyperparameters were changed for the fine-tuning of different models. It therefore cannot be ruled out that the differences in model performance can be partially attributed to different hyperparameters.

4 Experimental Setup

For Task 1 and 2, the given train and validation datasets were used to train and test the models. We fine-tuned models for both tasks on the training dataset provided by the Sem-Eval organizers and then tested it locally on subsets of the data, before submitting it with the use of a docker (Merkel, 2014) image to be run against the validation dataset on the Tira system (Fröbe et al., 2023b).

Task 1 Since Task 1 is a simple text classification problem, the model is fine-tuned by preprocessing the input data in a way XLNET understands it, which includes special tokenizing, adding separators. XLNET does not allow for non-numerical labels, so the labels got encoded into either 0, 1 or 2. The total calculation time per training iteration comes down to a little over 14 hours. Although additional information could have been used, the only text to predict the label was the "postText".

Task 2 For task 2, RoBERTa (Liu et al., 2019) and DistilBERT (Sanh et al., 2019) were used with a Question Answering head. These models were fine-tuned using different hyperparameters. To prepare the clickbaits for the question-answering model and to optimize the output of the spoilers, a number of pre- and post-processing techniques were applied. For the preprocessing, the clickbait was first converted to lowercase and tokenized using spaCy (Honnibal and Montani,

Model	Fine-tuned	Post-processing	Pre-processing	BLEU Score
DistilBERT raw				0.111
DistilBERT with sentence removal	X			0.196
DistilBERT without sentence deletion	X	X		0.202
RoBERTa	X	X		0.2417
RoBERTa	X	X	X	0.2715
RoBERTa Large (small learning rate)	X	X	X	0.31196

Table 1: BLEU Scores of different models for Task 2 using the Test Set

2017) [2]. If the clickbait contained an interrogative word (such as "how", "what", "when", etc.), it was moved to the beginning of the clickbait. Additionally, to handle interrogative words like "how much" or "how many", a special condition was added. A transformer question-answer pipeline (Wolf et al., 2019) [3]from was used to obtain spoilers for all three spoiler types. For phrase spoilers, a post-processing function was implemented to limit the returned answer to a maximum of 5 words/tokens. Two post-processing approaches were implemented for sentence spoilers. The first procedure removed the entire sentence if the predicted response contained less than 5 words/tokens, and repeated the process until a suitable response is obtained. The second method was to return the whole sentence that contained the predicted answer if the predicted answer contained at least 5 words/tokens. In the case of multi-part spoilers, a spoiler was predicted, then the sentence that contained the spoiler was removed, and the pipeline was run again. This process was repeated until a total of 5 spoilers had been obtained.

5 Results

5.1 Task 1 – Spoiler Classification

The results for the spoiler classification task are pretty decent in comparison to both baselines. The model was deployed and can be accessed via docker [4].

With a balanced accuracy of 0.66, the run comes close to the transformer baseline (Hagen et al., 2022b) and, thinking of the model only using the "postText"-content, which means, there could be a lot more of possibilities to optimize the model. The most content and the most information, what type of spoiler there is, is found in the body of the article. This not only includes the spoiler, but also a lot of information and data about the spoiler and spoiler type. So by including the article body, there is a high chance that this turns out to be even better.

5.2 Task 2- Clickbait Spoiling

The scores and results in the second tasks highly depended on which model we used and how it was trained. An overview is displayed in Table 1 Assumptions from the literature about the difference in model performance did hold up. DistilBERT performed significantly worse than RoBERTa, which in turn performed noticeably better than RoBERTa Large.

According to other sources, a small performance decrease is to be expected for DistilBERT, while RoBERTa and RoBERTa Large have been observed to lead to significant increases in performance (Adoma et al., 2020) (Khan, 2019).

When used on the final post-processing approach, DistilBERT was able to achieve a BLEU Score of 0.202 on the test dataset with fine-tuning. Without fine-tuning, it only reached a BLEU Score of 0.11. Larger models performed a lot better. RoBERTa reached a BLEU score of 0.242. When the preprocessing steps (query re-writing) were included as well, the results increased further to 0.272.

The best model that was produced by our approach was RoBERTa Large, that was fine-tuned with a small learning rate and sample size and achieved a BLEU Score of 0.312. It is also deployed as a docker image that can be accessed via docker.[5]

These results are comparable to the results of other groups, but are worse than the transformer baseline presented by the challenge organizers, which is 0.38 (Hagen et al., 2022b).

Using a multi-language version of this model might increase results slightly further, since some posts are written in other languages, although the amount

[2] v3.5.0 - https://spacy.io/

[3] v4.25.1 - https://huggingface.co/docs/transformers/index

[4] registry.webis.de/code-research/tira/tira-user-pan23-sam-miller/task1:22

[5] registry.webis.de/code-research/tira/tira- user-pan23-sam-miller/generation001:35

of posts in different languages is small enough to consider it noise.

6 Conclusion

Task 1 shows how strong Large Language Models have become. Even by fine-tuning them by a small degree, they prove to be very strong in predicting and classifying text spoilers. The results from the second task also show that larger models such as RoBERTa outperform smaller models like DistilBERT. Additionally, using preprocessing techniques like query rewriting and post-processing based on spoiler type improved the quality of the generated spoilers. For future work, the approach could be extended to handle additional spoiler types or incorporate other preprocessing techniques by using more advanced text normalization techniques. Additionally, exploring the use of multi-language models could improve the performance on non-English posts.

References

Acheampong Francisca Adoma, Nunoo-Mensah Henry, and Wenyu Chen. 2020. Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 117–121. IEEE.

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. 2022a. 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.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022b. semeval23-clickbait-spoiling-call-for-participation-for-teaching-frame.pdf. https://pan.webis.de/semeval23/pan23-figures/semeval23-clickbait-spoiling-call-for-participation-for-teaching-frame.pdf. (Accessed on 02/28/2023).

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Suleiman Khan. 2019. Bert, roberta, distilbert, xlnet — which one to use? | by suleiman khan, ph.d. | towards data science. https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82b (Accessed on 02/28/2023).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological Bulletin*, 116:75–98.

Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

A Appendix

https://github.com/pstoermer/Clickbait_Challenge_SamMiller
registry.webis.de/code-research/tira/tira-user-pan23-sam-miller/generation001:35
registry.webis.de/code-research/tira/tira-user-pan23-sam-miller/task1:22

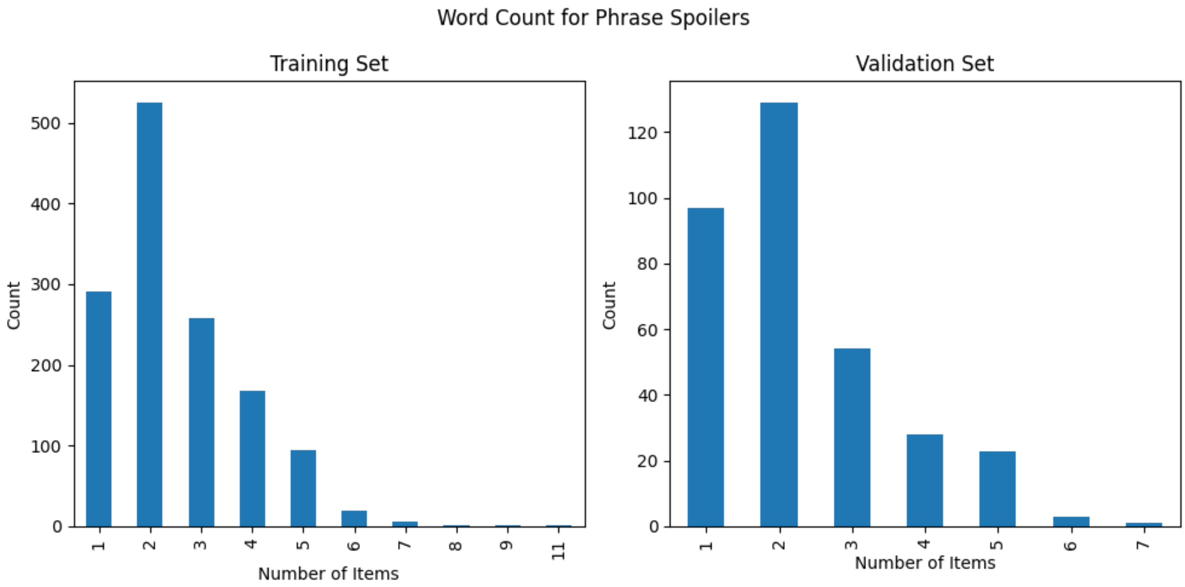

Figure 3: Word Count of Phrase Spoilers per Dataset

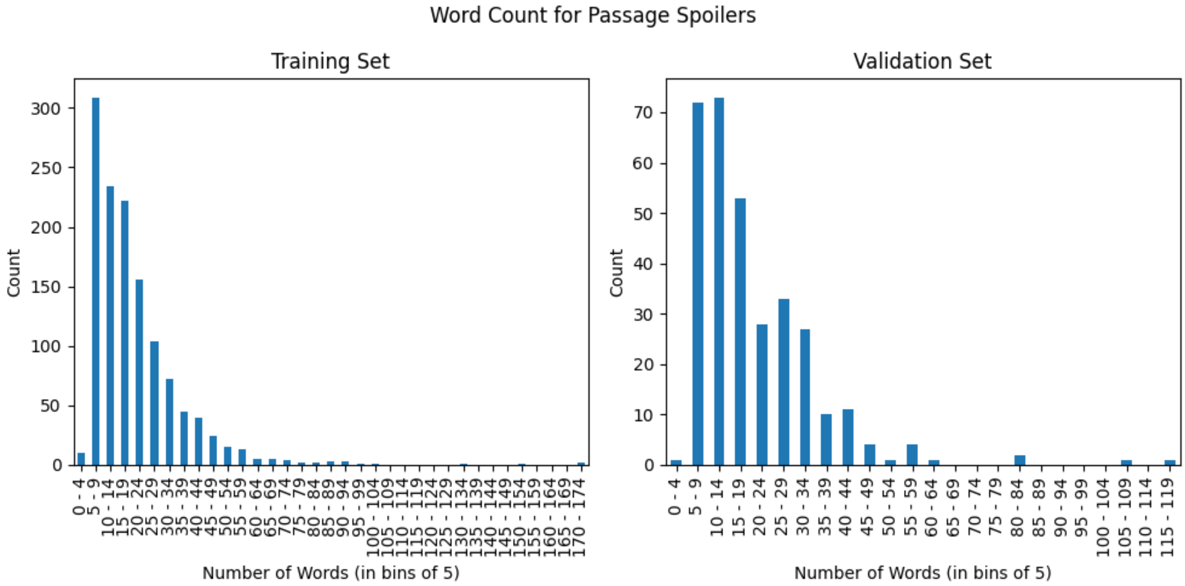

Figure 4: Word Count of Passage Spoilers per Dataset

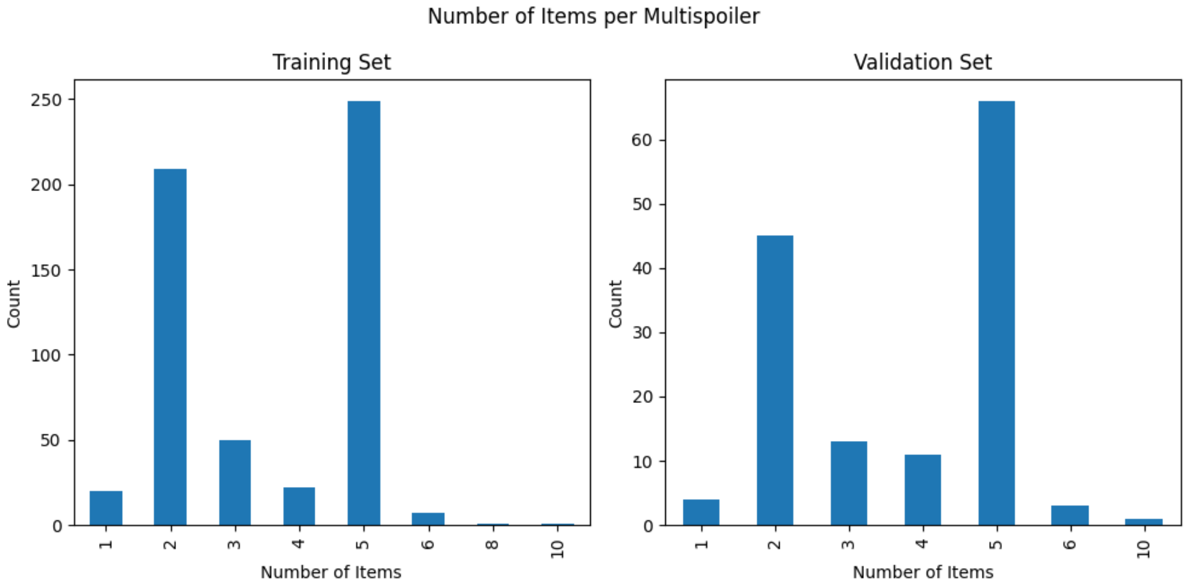

Figure 5: Item Count of Multi Spoilers per Dataset

Implementation	
Research	All
Task 1	Tobias Esser
Task 2 (Pre- & Postprocessing)	Pia Bernards
Task 2 (Model Fine-tuning)	Patrick Thomasius
Submission (Docker/Tira)	Patrick Thomasius
Report	
Abstract	Pia Bernards
Introduction	Patrick Thomasius
Background	Patrick Thomasius
System Overview	
Task 1	Tobias Esser
Task 2	Pia Bernards / Patrick Thomasius
Experimental Setup	
Task 1	Tobias Esser
Task 2	Pia Bernards / Patrick Thomasius
Results	
Task 1	Tobias Esser
Task 2	Patrick Thomasius
Conclusion	Pia Bernards

Table 2: The distribution of work within the team regarding the preliminary work and the implementation, as well as the writing of the report.