ML Mob at SemEval-2023 Task 5: "Breaking News: Our Semi-Supervised and Multi-Task Learning Approach Spoils Clickbait"

Hannah Sterz* **Leonard Bongard*** **Tobias Werner***
Clifton A. Poth **Martin B. Hentschel**
Technical University of Darmstadt
{hannah.sterz, leonard.bongard, tobias.werner}@stud.tu-darmstadt.de

Abstract

Online articles using striking headlines that promise intriguing information are often used to attract readers. Most of the time, the information provided in the text is disappointing to the reader after the headline promised exciting news. As part of the SemEval-2023 challenge, we propose a system to generate a spoiler for these headlines. The spoiler provides the information promised by the headline and eliminates the need to read the full article. We consider Multi-Task Learning and generating more data using a distillation approach in our system. With this, we achieve an F1 score up to *51.48%* on extracting the spoiler from the articles.

1 Introduction

The modern web is flooded with clickbait headlines and articles - text snippets that try to lure the reader into clicking a link (Potthast et al., 2016) - so readers often fall prey to misleading information.

Clickbait spoiling is a pressing issue that refers to the phenomenon of clickbait headlines that are used to gain attention and clicks to an article. The answer to the question or statement raised in the headline is typically a small part of the article. Clickbait spoiling, the process of getting the answer (the spoiler) to the raised problem in the article, provides this small part of the article which enables readers to get the relevant information without having to go through the entire article.

This work is part of the SemEval 2023 challenge where our team chose to focus on the topic of clickbait spoiling (Fröbe et al., 2023a), as it is a widespread problem that affects millions of internet users every day.

The first of two tasks is about spoiler type classification, where an input text is given and the model classifies the spoiler as either "phrase", "passage" or "multi". A phrase spoiler denotes an article, whose headline can be answered with a single word or short phrase. For "passage" spoilers the answer is a few sentences long. Answers with multiple spans are classified as the "multi" spoiler type.

The second task focuses on spoiler generation. A clickbait post and a linked document are used as input to generate a spoiler (Hagen et al., 2022a). Within our work, we adopted the proposed baselines by Hagen et al. (2022a). Specifically, we applied Multi-Task Learning and extended the training dataset by incorporating additional data.

2 Background

This section introduces relevant background for the distillation approach and multi-task learning.

2.1 Clickbait

Clickbait posts attempt to attract users with a controversial or striking title (Zannettou et al., 2022). The motivation behind this is to gain traffic on the site or to scam users. There are two tasks related to protecting users from clickbait: clickbait detection and clickbait spoiling. Fröbe et al. (2023a), as the presenters of the SemEval-2023 Task 5, define clickbait spoiling as providing the information promised by the headline of an article. Previous approaches to clickbait spoiling used question answering (QA) and passage retrieval (Hagen et al., 2022b). The clickbait title promises some intriguing information, hence it acts as the question. A model addressing this task needs to find the answer to the title from the article body. There are several specialized pipelines to address QA problems, depending on whether the answer is extracted from text or generated, the domain of the questions, and the text type (Baumgärtner et al., 2022). A simple approach to extractive QA is to use transformer models with a question-answering head (Devlin et al., 2019). This can be enhanced by using ensembles (Li et al., 2021) or passage retrieval methods to find relevant passages. Especially for open-domain questions this can be used in combination with a

* equal contribution

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1818–1823

retrieval system (Yamada et al., 2021).

2.2 Distillation

Distillation is a common approach in deep learning where a model, the student, is trained by a teacher model (Hinton et al., 2015). This transfer learning setup allows adding unlabeled data into the training, by letting the teacher create samples for the student to train on. This semi-supervised data collection method is suitable for tasks like ours with only a limited amount of labeled datasets. (Wong and Gales, 2016)

In addition to the data given, another corpus from the "Clickbait Resolving Challenge" (Hättasch and Binnig, 2022) is related to clickbait spoiling. However, this corpus does not contain usable labels. To address this challenge, we employed the student-teacher approach to leverage the unlabeled dataset. The dataset in question consists of 2635 samples for English clickbait articles with the corresponding title and was gathered from social media accounts trying to resolve clickbait manually (Hättasch and Binnig, 2022). Our task differs from the clickbait resolving task in that, while the challenge requires generative models to produce the resolution, our task involves extracting the resolution from the given article.

2.3 Multi Task Learning

To train a model for a task we often fine-tune the model on the dataset to optimize a specific metric. But for some tasks, there might be information that the model does not take advantage of (Ruder, 2017). Multi Task Learning (MTL) fine-tunes the model on multiple tasks and optimizes several metrics simultaneously. This can enable the model to use the information required for one task to solve another task. With MTL, the tasks share parameters and have task-specific parameters. Some possibilities to implement this include using task-specific heads (Ruder, 2017), or using parameter-efficient training methods in combination with hyper network (Karimi Mahabadi et al., 2021). We propose a multitask setting to train on both tasks, the classification and spoiler retrieval to make use of the overlap between the two tasks and help the model learn the difference between the spoiler types.

3 System Overview

We propose a semi-supervised setup, allowing us to use other clickbait datasets that are not annotated with a spoiler. This enables us to use more data for training. To combine the training of both tasks we explore a multi-task setup to train both sub-tasks jointly. The system is evaluated using the Tira platform (Fröbe et al., 2023b).

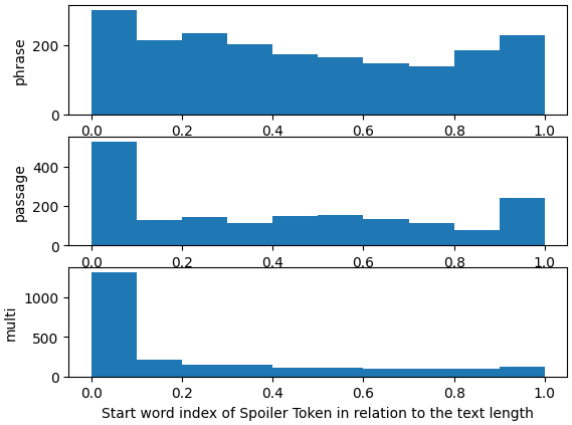

Figure 1: Spoiler positions with respect to the article length for each label

3.1 Analysis of given datasets

The dataset provided for this SemEval 2023 challenge includes a collection of articles and their spoiler, along with their type and position. To gain a better understanding of the data, we conduct an analysis of the spoilers' position. The dataset consists of three labels: "phrase", "passage", and "multi", which categorize spoilers as small sentence spoilers, longer spoilers, and multiple spoilers across the article, respectively. Figure 1 depicts the spoiler position in relation to the article length. Spoilers labeled as "multi" are counted individually. Our analysis reveals that a large part of spoilers occurs in the first tenth of the article, with a slight increase observed in the last tenth. Especially for passage and multi-spoiler types, spoilers tend to be in the beginning while phrase spoilers are more equally distributed. The remaining data is relatively evenly distributed.

3.2 Semi-Supervised Learning

We want to assess the efficacy of augmenting unsupervised data to the training and compare its performance against the baseline. Focusing on clickbait spoiling, we gather more labeled data by predicting clickbait resolutions from the "Clickbait Resolving Challenge" dataset (Hättasch and Binnig, 2022) using a distillation approach for transfer learning. Incorporating additional data contains the risk of overlapping samples within the data. This

presents a potential issue as including identical samples from the validation or test sets in the training set would distort the performance results. Hence, we perform text similarity analysis to identify and remove duplicates.

Given two samples *A* and *B*, tokenize both titles. If at least 70% of the words in the title overlap, remove *B*. This results in the identification of 477 duplicates between the two datasets. The combined datasets comprise 5244 entries, representing an increase of 1863 entries compared to the original dataset.

3.3 Multi Task Learning

To incorporate both tasks in the training, we fully fine-tune the model on both tasks jointly. The two tasks of identifying what answer type is necessary to spoil the clickbait and then providing the information to spoil it are closely related. We assume that learning what type of spoiler is required is beneficial for clickbait spoiling and the other way around. Therefore, we propose a multitask setup that addresses both tasks jointly. Given an input article with the clickbait heading q and the article body $body$, the model is required to classify the spoiler type and provide the spoiler jointly. We approach the two tasks as follows:

- **Classification:** Given $(q, body)$ the model classifies the article as passage, paragraph, or multi by providing a score on how likely the given sample is to be of that class.
- **Extractive QA:** The spoiling of the clickbait can be modeled as extractive QA. Hence, the model needs to identify the text spans that provide the answer to the clickbait. The title of the article acts as the question.

We use the RoBERTa model(Liu et al., 2019) with two heads, a classification head, and a question-answering head. For each training sample, both heads compute their prediction. For both tasks, we use the cross entropy loss. The combined loss is the sum of the losses for the individual tasks.

$$L_{multi} = CLL(x_{class}, l_{class}) + CLL(x_{qa}, l_{qa})$$

Hence, each training sample leads to parameter updates taking into account both tasks. The model architecture is illustrated in figure 2. During inference, we can get the results for the individual tasks by only adding the corresponding head to the model.

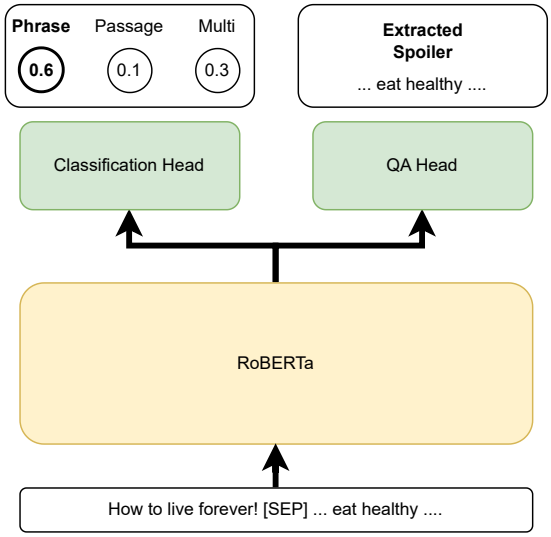

Figure 2: Multi Task Setup

4 Experimental Setup

In the following, we describe the experimental setup for both of our system variations.

4.1 Semi-Supervised Learning

During the experiments, we fine-tuned several pre-trained models to achieve the most optimized performance. These models include "bert-base-uncased", "roberta-base", "deepset/roberta-base-squad2", and "deepset/deberta-v3-base-squad2"

To achieve the most optimal set of hyperparameters for each pre-trained language model, a random search approach was employed, resulting in each model being trained 15 times with different hyperparameters. To prevent over-fitting, we used early stopping with patience set to three. The hyperparameters that we tuned include the learning rate, warmup ratio, maximum sequence length, and gradient accumulation steps. The ranges of values for each hyperparameter are summarized in Table 1.

Hyperparameter	Value Range
Learning Rate	$1e^{-4}$ - $5e^{-4}$
Warmup Ratio	0.02 - 0.1
Maximum Sequence Length	256, 384, 512
Gradient Accumulation Steps	1 - 8

Table 1: Ranges of values for each hyperparameter used in our random search optimization.

We used the F1 score on the evaluation set as our primary metric for selecting the best-performing model. We assume it to be a more appropriate

evaluation metric than "exact match" for our task, as it provides a more balanced evaluation over all spoiler types. We are using a distillation approach with two models. The teacher model is trained on the original data split called the *Webis* corpus. The student model is trained upon the *Webis* corpus and on an additional corpus, referred to as the *CRC* corpus, which comprises data labeled by the teacher model. Only the teacher model with the highest performance was selected to serve as the basis for training the student model. The selection of the student models architecture was predicated on the evaluation of the teacher model's performance.

4.2 Multi Task Learning

We use adapter-transformers (Pfeiffer et al., 2020) for implementing the multitask setup since it allows flexible loading and managing of multiple heads. Results from the semi-supervised experiments suggest that the model benefits from pertaining on squad. Hence, we are using the "deepset/roberta-base-squad2" pre-trained weights during our experiments. We train the model for 5 epochs with a learning rate of $2e{-}5, 1e{-}4$ and a batch size of 32. We use the data as proposed in (Hagen et al., 2022b). To compare the different setups, we report the balanced accuracy for the first task and the F1 score and exact matches for the second task.

5 Results

Table 2 shows the performance of various finetuned classifiers on the task, measured in terms of F_1 score. As a baseline we use a naive bayes (Hagen et al., 2022b) and a fully fine-tuned RoBERTa model for both tasks and. The table 2 presents the results of models trained on the Webis corpus. Additionally, the table shows the performance of the student models trained on both the Webis corpus and the CRC corpus, denoted as "Webis + CRC". The "deepset/deberta-v3-large-squad2" model trained on Webis + CRC achieves the highest F_1 score of *51.48%*, outperforming all other models.

Looking at Table 2, we can see that for the "deepset/deberta-v3-base-squad2" model, fine-tuned on the Webis corpus, the F_1 score is *49.69%*. However, when this same model is additionally trained on the CRC corpus (denoted as "Webis + CRC"), its performance improves to *50.7%*. This represents an improvement of *1.01%* in F_1 score, indicating that the additional training on the CRC corpus has a positive impact on the model's performance. These results demonstrate the effectiveness of our distillation approach and the importance of additional training data in improving the model's performance.

Table 3 summarizes the performance of our best-performing model on a new set of evaluation metrics measuring more precisely how far the results differ from the ground truth. The reported results are for the test set. We evaluated the performance of our Question Answering model in extracting spoilers from a given text using three different approaches: extracting spoilers from individual phrases, passages, and multiple spoilers in a given text, given the spoiler title. Our model achieved a BLEU score of *62.02*, a BERT score of *94.83*, and a Meteor score of *55.32* in extracting spoilers from individual phrases. In comparison, the model achieved a BLEU score of *19.45*, a BERT score of *88.77*, and a Meteor score of *41.13* in extracting spoilers from passages, and a BLEU score of *11.39*, a BERT score of *88.65*, and a Meteor score of *38.46* in extracting spoilers from multiple spoilers. These results indicate that our model performs best in extracting spoilers from individual phrases. However, the performance drops significantly regarding passage and multi spoilers.

The MTL setup leads to an accuracy of *67.39%* in the development for the spoiler classification task (Table 2) and an F1 score of *44.60%* for the spoiler extraction. The answer type classification seems to be working rather well in the MTL setup, but the spoiler extraction on the other hand does not seem to benefit from this setup. Interestingly, the MTL benefits from larger learning rates. Probably, the model needs to adapt to solving both tasks jointly first and needs larger updates for this.

Table 4 shows the results of the test set for the spoiler type classification task. The precision, recall, and F1 score for the different spoiler types suggest that the classifier tends to classify the answer type as "phrase" more often than the other two classes. In the training dataset, the share of phrase and passage spoilers are similar. Only the "multi" spoilers have notably fewer training samples.

For both tasks, our models perform best for the samples with the phrase spoiler type. One possible explanation for this is the usage of models that were already fine-tuned on the squad dataset. The squad

Data	Model	Accuracy	F_1	Exact Match
Webis	Naive Bayes	56.15	-	-
Webis	roberta-base	73.08	-	-
Webis	roberta-base	-	44.44	28.37
Webis	bert-base-uncased		34.63	21.62
Webis	deepset/roberta-base-squad2	-	48.53	33.38
Webis	deepset/deberta-v3-base-squad2	-	49.69	34.0
Webis + CRC	deepset/deberta-v3-base-squad2	-	50.70	35.25
Webis + CRC	deepset/deberta-v3-large-squad2	-	**51.48**	**36.13**
Webis	MTL + deepset/roberta-base-squad2[lr=2e−5]	66.41	27.05	15.00
Webis	MTL + deepset/roberta-base-squad2[lr=1e−4]	67.39	44.60	29.38

Table 2: Comparison of the baseline, provided by (Hagen et al., 2022b), and our different approaches on the Webis development dataset with their respective accuracy, F_1 scores, and exact match scores.

Model	BLEU	BERT	Meteor
Phrase-Spoiler	62.02	94.83	55.32
Passage-Spoiler	19.45	88.77	41.13
Multi-Spoiler	11.39	88.65	38.46
All-Spoilers	36.06	91.31	43.13

Table 3: Evaluation results in percent for task 2 of our fine-tuned "deepset/deberta-v3-large-squad2" model on the test set.

dataset contains questions with short answers that are most similar to the phrase spoilers. Hence, the pretraining might benefit the phrase spoilers the most. Additionally, the distribution varies between the spoiler types. As noted in section 3.1 the distribution of the phrase and multi spoiler types is skewed towards the beginning of the article. This might make the transfer harder.

6 Conclusion

In this work, we present a distillation approach improving the performance of a fine-tuned deep learning model on a clickbait spoiling task. We extended the given dataset with additional automatically labeled examples through a semi-supervised learning approach from a different dataset, which led to an improvement in the model's performance.

Additionally, we explored a Multi-Task learning setup, where we tried to improve the performance on both tasks further. As our results indicate a lack in performance for longer spoiler extractions sequences, we suggest for future work to further investigate finding ways to extract these sequences.

	Precision	Recall	F1
Phrase-Spoiler	66.28	80.85	72.84
Passage-Spoiler	73.85	63.77	68.44
Multi-Spoiler	69.85	54.60	61.29

Table 4: Results of the MTL setup with learning rate set to 2e−5 for task 1 on the test set.

Acknowledgments

We thank Myra Zmarsly and Gianluca Zimmer for their support on this project.

References

Tim Baumgärtner, Kexin Wang, Rachneet Sachdeva, Gregor Geigle, Max Eichler, Clifton Poth, Hannah Sterz, Haritz Puerto, Leonardo F. R. Ribeiro, Jonas Pfeiffer, Nils Reimers, Gözde Şahin, and Iryna Gurevych. 2022. UKP-SQUARE: An online platform for question answering research. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 9–22, Dublin, Ireland. Association for Computational Linguistics.

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. 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. 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.

Benjamin Hättasch and Carsten Binnig. 2022. Know better–a clickbait resolving challenge. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 515–523.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

Shilun Li, Renee Li, and Veronica Peng. 2021. Ensemble ALBERT on squad 2.0. *CoRR*, abs/2110.09665.

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.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait detection. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 810–817. Springer.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *ArXiv*, abs/1706.05098.

Jeremy Heng Meng Wong and Mark JF Gales. 2016. Sequence student-teacher training of deep neural networks. ISCA.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online. Association for Computational Linguistics.

Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. 2022. The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans.