nancy-hicks-gribble at SemEval-2023 Task 5: Classifying and generating clickbait spoilers with RoBERTa

Jüri Keller
TH Köln
Claudiusstr. 1
50678 Köln
jueri.keller@smail.th-koeln.de

Nicolas Rehbach
TH Köln
Claudiusstr. 1
50678 Köln
nicolas_alexander.rehbach@smail.th-koeln.de

Ibrahim Zafar
TH Köln
Claudiusstr. 1
50678 Köln
ibrahim.zafar@smail.th-koeln.de

Abstract

Clickbait spoiling and spoiler type classification in the setting of the SemEval2023 shared task five was used to explore transformer based text classification in comparison to conventional, shallow learned classifying models. Additionally, an initial model for spoiler creation was explored. The task was to classify or create spoilers for clickbait social media posts.

The classification task was addressed by comparing different classifiers trained on hand crafted features to pre-trained and fine-tuned RoBERTa transformer models. The spoiler generation task was formulated as a question answering task, using the clickbait posts as questions and the articles as foundation to retrieve the answer from.

The results show that even of the shelve transformer models outperform shallow learned models in the classification task. The spoiler generation task is more complex and needs an advanced system.

1 Introduction

Clickbait describes the presentation of content on the internet in such a manner, that it encourages the viewer to click on a link[1]. To increase the likelihood of a click, lurid headlines or misleading descriptions are oftentimes used. Clickbait spoiling oppositely tries to counteract by precisely summarising the most important information of a given article or document. Task five of SemEval2023's Discourse and Argumentation section revolves around creating Clickbait spoilers based on clickbait posts on social media, organized and evaluated through (Fröbe et al., 2023b,a). Further information about clickbait spoiling and the underlying data of the challenge is presented in (Hagen et al., 2022).

The Clickbait Challenge consisted of two tasks. The first is spoiler type classification and the second is spoiler creation. Based on a social media post and the given article the correct spoiler type shall be classified for the first task. For task two, the correct spoiler shall be generated based on the information provided such as the advertised article. Three systems based on the RoBERTa (Liu et al., 2019) transformer model were submitted for the first task: A multiclass classifier, a system based on three binary classifiers and a multiclass classifier that was further pre-trained. Additionally, these systems were compared to different shallow learned classifiers beforehand. As for task 2, a system based on an off the shelve transformer model and spoiler type dependent post processing is proposed for the spoiler creation.

The submitted systems are made publicly available on GitHub[2].

2 Background

The dataset for the shared task is provided through the organizers website and contains 3200 rows for training and 800 for validation. 1000 additional spoilers were held back as testing data for the submission. Out of 14 existing columns, the most relevant are the text of the post (`postText`), the title of the advertised article (`targetTitle`), the articles text (`targetParagraphs`), the spoiler itself (`spoiler`) and the type of the spoiler (`tag`). In general, three spoiler types exist in the dataset. *Phrase*

[1]Oxford Learner's Dictionaries

[2]https://github.com/jueri/nancy-hicks-gribble-at-SemEval-2023-Task-5

spoilers are a single or a few words, while *passage* spoilers consist of a whole sentence. *Multi* spoilers have several forms, such as relevant parts of an article's phrases, enumerations or single words. After analyzing the training dataset, one can observe that the classes are not evenly distributed. 1367 phrase, 1274 passage and 559 multi spoilers exist. Additionally, the publisher of the articles could be traced back. The articles mainly originate from the HuffPost[3], with more than 700 articles, followed by archive.is[4], an internet archive with articles from various sources, with around 500 articles. The spoilers are generally part of the article and oftentimes formulated as a question. Therefore it seems sensible to use an extractive question answering method for generating spoilers in task two.

3 System Overview

The systems developed for both sub-tasks can be categorized into three main branches: Systems based on shallow learned models and transformer based models for the first task and a transformer based system for question answering for the second task.

3.1 Task 1: Spoiler Type Classification

The shallow learned models were used as a local baseline only. Since the transformer models outclassed them, they were not submitted to the online evaluation.

3.1.1 Shallow Learned Approaches

The key idea behind the shallow learned approaches was to compare their performance to the current state-of-the-art transformer methods. Besides the general performance, the ease of implementation and computational intensiveness should be explored along the way.

For this approach, the dataset was preprocessed using typical Natural Language Processing (NLP) techniques such as tokenization, stopword removal, and lemmatization before applying further processing to derive meaningful features for the different spoiler types. The text of the post and the paragraphs of the target article were combined before extracting these features, such as the overall text length in comparison to the average text length, enumerations, named entities and occurring question marks. The main idea of these features was to discover possible patterns in the data. Multi-part spoilers, for example, might be longer than the average post and contain enumerations or named entities such as cities, company names or names of celebrities. Phrase spoilers oppositely could be shorter on average but contain a question mark in the post. These patterns could be observed in several clickbait posts. Examples are posts such as *"Hot Sauce Taste Test: Find out which we named number 1"*[5], a multi-part spoiler where a hot sauce test was conducted using a numbered list and consisting of 699 words. In contrast, *"Google paid HOW MUCH in overseas taxes!?"*[6] is a phrase spoiler using named entities and a question mark in the post text consisting of only 374 words.

Subsequently, the text was vectorized to a bag of words consisting of the post text and the target article paragraphs in conjunction with the extracted features. Additionally, the words were weighted using Term Frequency Inverse Document Frequency (TF-IDF) before applying classification algorithms. The classifiers were then trained on the pre-processed train data and evaluated on the provided validation dataset.

Afterwards, five classification algorithms performed the spoiler type classification. These included Random Forest, Logistic Regression, Naïve Bayes, XGBoost and K-Nearest Neighbours. These algorithms were chosen as they seem to be a variety of standard algorithms and are well supported, which makes them straightforward to implement. They cover a variety of different methods, such as tree-based methods, distance-based classification and Bayesian probability.

3.1.2 Transformer Approaches

The systems using transformer models are all based on the RoBERTa model (Liu et al., 2019) and are fine-tuned as single and multi-label models. An additional new dataset was used for pre-training, and the input data source was varied between combinations of the post text, article title and the article paragraphs.

The pre-trained RoBERTa transformer model was chosen as the foundational model mainly because it was initially trained on a much bigger dataset, compared to the original BERT model (Devlin et al., 2019), comprising a large part of news articles, perfectly in line with the prevalent domain of the spoilers. Further, superior results to the BERT

[3] https://www.huffpost.com/
[4] https://archive.is/
[5] postId: 4oeqo8
[6] postId: 384820915285921792

model were reported in various tasks (Liu et al., 2019). Before fine-tuning the model for the classification task, the model was pre-trained on news headlines from the HuffPost dataset (Rajpurkar et al., 2018). News headlines were chosen as they match the domain of the task and are short and concise, much similar to clickbait social media posts. The dataset was originally designed for news categorization tasks and contained approximately 200k news articles from 2012 to 2018 (Rajpurkar et al., 2018). The articles are from the HuffPost post, where majority of the posts also originate from.

Further, the original and the pre-trained model is fine-tuned for the classification task. This is done as a single-label multi-class classification and a single-label binary classification. While the multi-class models directly yield the class of the post, three binary models for each of the three tags, *multi*, *passage* and *phrase*, are combined into one system to predict the final class. This one-vs-rest approach was employed based on the observation that the binary classification models on their own achieve greater results than a multi-class approach.

Regarding the input text source, all models were trained on the post text as well as further adding the target article title and paragraphs. In total, through all combinations, 24 models were trained for comparison.

3.1.3 Task 2: Spoiler Creation

To perform spoiler creation, a transformer model was chosen. While linear approaches were a consideration, the given data seemed to be too complex to produce meaningful results using these models. After analyzing the data, the model roberta-base-squad2 [7] was chosen, as it is one of the most popular ones for question answering. This model was trained on the Stanford Question Answering Dataset 2 (SQuAD) by (Rajpurkar et al., 2018) and covers a wide range of topics suitable for diverse datasets such as the one of the shared task. For the purpose of spoiler creation, the task was formulated as a question-answering task. Many of the posts were or could be formulated as a question, and the needed spoiler can be seen as the answer to this question. For example, the post *"What happens if your new AirPods get lost or stolen, will Apple do anything?"*[8] is a literal question and the desired spoiler, *"Apple says that if AirPods are lost or stolen, you'll have to buy new ones, just like any other Apple product."*, is a direct answer to this question. As the spoilers can literally be found in the target article almost every time, an extractive method was feasible for this task. This means that most of the time, given the target article, the span needed to be identified rather than abstractively created.

As the spoiler types differ strongly, the type information (tag) was used to determine what spoiler should be generated and a dedicated post-processing performed. Instead of using the type information, a model from the first task could be used instead. The question-answering model returns the top k spans as answers to the input question like post. For *phrase* spoilers, the top one answer was used directly. *Passage* spoilers were created by using the complete sentence from the article, the top one predicted answer span occurred in. *Multi* spoilers could not directly be created by using the top five predicted answers as they were too similar. Instead, five predictions were made in a row, while the sentence that holds the predicted span was removed for the next prediction.

4 Experimental Setup

A series of experiments were conducted to evaluate the spoiler classification and creation systems. The final systems and configurations were determined for the final submission during these experiments. The weighted F1 score was used as the main metric for the spoiler classification problem because the three classes are distributed unevenly across the datasets. Results for the spoiler creation system are reported through the BLEU score (Papineni et al., 2002).

4.1 Task 1: Spoilertype Classification

4.1.1 Shallow Learned Approaches

Shallow learned approaches have been applied to different combinations of TF-IDF n-grams and count vectorized unigrams. For text pre-processing, standard NLP libraries were used, such as NLTK (Bird et al., 2009) to perform the tokenization and lemmatization and SpaCy (Honnibal et al., 2020) to perform named entity recognition based on the `en_core_web_lg` model. The vectorization, TF-IDF and classifier models were all supported by Scikit-learn (Pedregosa et al., 2011). Pre-processing and model training was achieved by using Scikit-learn's pipeline functions.

[7] `https://huggingface.co/deepset/roberta-base-squad2`

[8] postId: 521bee

While not submitting these approaches to the challenge due to being outperformed by the transformer approaches, the best performing algorithm was XGBoost. Generally, the tree-based methods slightly outperformed the other models. While higher TF-IDF weighted n-grams increased the classification for the XGBoost model, it resulted in similar or worse results for the other models. In-depth hyperparameter tuning was not done for the models due to time constraints. For the MultinomialNB and Logistic Regression, the default hyperparameters were used. K-Nearest Neighbors uses a k of 20. For the Random Forest 100 estimators and a tree depth of 50 were used. Also for XGBoost the default hyperparameters were used with the objective `multi:softmax` and three classes, to enable the classification of each spoiler type. The F1 score for each model using named pre-processing, a count vectorizer of unigrams and TF-IDF of one to three with an occurrence in at least 10 documents can be observed in Table 1.

Table 1: An overview of the applied algorithms and their performance for spoiler type classification. The best result is highlighted in bold.

Model	F1
Random Forest	0.483
Logistic Regression	0.467
Multinomial Naïve Bayes	0.471
XGBoost	**0.543**
K-Nearest Neighbors	0.399

4.1.2 Transformer Approaches

The transformer systems are implemented through PyTorch (Paszke et al., 2019) and the Transformers library (Wolf et al., 2020) from Hugging Face, where the HuffPost dataset was also taken from. Each model is trained for five epochs with a starting learning rate of 2e-5 and a weight decay of 0.1. The models are saved after each epoch, and the instance with the lowest loss is taken as the final model of a training run.

Table 2 shows the results for the systems using the RoBERTa model. The model using all available texts as input, the post text, article title and paragraphs, achieves the best validation result with an F1 score of 0.741. The results for the binary classifiers are distributed evenly across all input text configurations. Phrases are best detected based on the post text alone (0.74 F1), the passages are best detected based on the post text and the title of the article (0.77 F1), and for the multi-posts, all fields are used as input (0.9 F1). Generally, the results for one post type are close to each other, with a few percent differences. Multi spoilers can be detected with the highest F1 score of 0.902.

Table 2: Overview of the effectiveness in spoiler type prediction measured as F1 score for all systems configurations and inputs based on the RoBERTa model. The best results for each class are highlighted in bold.

Model	Approach	Input	F1
RoBERTa	Multi	post, title, paragraphs	**0.902**
RoBERTa	Phrase	post, title, paragraphs	0.741
RoBERTa	Passage	post, title, paragraphs	0.773
RoBERTa	Multi	post, title	0.880
RoBERTa	Phrase	post, title	0.734
RoBERTa	Passage	post, title	**0.774**
RoBERTa	Multi	post	0.888
RoBERTa	Phrase	post	**0.774**
RoBERTa	Passage	post	0.736
RoBERTa	Multiclass	post, title, paragraphs	**0.741**
RoBERTa	Multiclass	post, title	0.710
RoBERTa	Multiclass	post	0.724

Results achieved through pre-training the model on the HuffPost dataset improved the general performance little. These results are reported in Table 3. The multiclass model improves slightly to an F1 score of 0.786. For the other models, the best results are achieved on different input texts compared to the RoBERTa models, but besides the model for phrase detection (0.889 F1) only minor improvements could be achieved if any.

Based on these results, three submissions were made:

1. INJ-TASK1_MULTYCLASS based on the RoBERTa model finetuned for multiclass classifications

2. INJ-TASK1_OAO based on the three best RoBERTa models for binary classification

3. INJ-TASK1_NEWS based on the RoBERTa model pre-trained on the HuffPost dataset and finetuned for binary classifications

The different inputs for the binary classifiers were considered during prediction as the models in the submitted system were only provided with the inputs they performed best on during the experiments.

4.2 Task 2: Spoiler Creation

As mentioned in Section 3, the spoiler creation is based on the roberta-base-squad2 transformer model. As the model was not fine-tuned but used

Table 3: Overview of the effectiveness in spoiler type prediction measured as F1 score for all systems configurations and inputs based on the RoBERTa model that was pre-trained on the HuffPost dataset. The best results for each class are highlighted in bold.

Model	Approach	Input	F1
RoBERTa News	Multi	post, title, paragraphs	0.735
RoBERTa News	Phrase	post, title, paragraphs	**0.889**
RoBERTa News	Passage	post, title, paragraphs	0.731
RoBERTa News	Multi	post, title	0.886
RoBERTa News	Phrase	post, title	0.886
RoBERTa News	Passage	post, title	0.767
RoBERTa News	Multi	post	**0.889**
RoBERTa News	Phrase	post	0.773
RoBERTa News	Passage	post	**0.778**
RoBERTa News	Multiclass	post, title, paragraphs	**0.786**
RoBERTa News	Multiclass	post, title	0.709
RoBERTa News	Multiclass	post	0.723

in a zero-shot fashion, no hyperparameter tuning was done. Additions were made by increasing the maximal input length to 500 subword tokens, to account for possible long posts, and enabling truncation.

5 Results

5.1 Task 1: Spoiler Classification

Three systems were submitted for the first task, with the best performing being INJ-TASK1-MULTYCLASS with a balanced accuracy of 0.7. INJ-TASK1-NEWS and INJ-TASK1-OAO scored a balanced accuracy of 0.66 as seen in Table 4. While the balanced accuracy gives an insight into the overall performance of each model, valuable information can be gained by analyzing the spoiler type specific metrics and values. Interestingly, the best performing model does not outperform the other models in predicting phrase or multi spoilers in terms of accuracy. Especially INJ-TASK1-OAO achieved a precision of 0.99 for the multi spoilers and 0.8 for phrase spoilers, being the highest precision in the shared task. However, when analyzing the other metrics, one can observe that the best performing model has a higher F1 score for each spoiler type. This suggests that INJ-TASK1-NEWS and INJ-TASK1-OAO are rather conservative when classifying spoilers as multi, leading to a high amount of true positives but also an increased amount of false negatives in comparison to INJ-TASK1-MULTYCLASS. It is worth noting that the recall and F1 scores are generally the highest for the passage spoilers, indicating that these spoiler types could be classified with high confidence.

5.2 Task 2: Spoiler Creation

For task two, an average BLEU score of 0.27, BERTScore of 0.88, and METEOR score of 0.26 was achieved. Based on all metrics, but especially the BLEU score, one can observe a performance decrease when it comes to passage and multi spoilers as seen in Table 5. Since the exact spoiler should be extracted from the article, the BLEU score will be the metric mainly focused on. Phrase spoilers achieved a BLEU score of 0.48. Passage and multiple spoilers only achieved a BLEU score of 0.14 and 0.05 respectively, indicating that almost no spoilers were correctly extracted for both types. This can likely be traced back to the increased complexity of extracting the exact sentences or enumerations. Additionally, the created spoilers were not further post-processed to provide a filtered string. It is therefore possible that special characters such as "\n" are part of the created spoilers resulting in a decreased BLEU score.

6 Conclusion

Through the use of different approaches, several ideas could be implemented and partly submitted to the SemEval2023 challenge. Three fine-tuned RoBERTa models were submitted for spoiler classification, with the best performing model showing good predictions for each spoiler type but not exceeding at a specific one. Additionally, one of the submitted models has the highest precision in predicting phrase and multi spoilers in the shared task. However, the best performing model did not classify certain spoiler types perfectly but rather classified a large fraction of each correctly.

For the spoiler generation, one system could be submitted based on a RoBERTa model. While the results are not satisfactory, first attempts towards type specific spoiler generation could be made. The results suggest that transformer models are currently able to extract and spoil short information out of a text.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of

Table 4: 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 nancy-hicks-gribble.

Submission		**Accuracy**	**Phrase**			**Passage**			**Multi**		
Team	Approach		Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1
nancy-hicks-gribble	INJ-TASK1-MULTYCLASS	0.70	0.78	0.66	0.71	0.67	0.84	0.75	0.75	0.59	0.66
nancy-hicks-gribble	INJ-TASK1-NEWS	0.66	0.78	0.61	0.69	0.62	0.89	0.73	0.94	0.46	0.62
nancy-hicks-gribble	INJ-TASK1-OAO	0.66	0.80	0.61	0.70	0.61	0.91	0.73	0.99	0.45	0.62

Table 5: Overview of the effectiveness in spoiler generation (subtask 2 at SemEval 2023 Task 5) measured as BLEU-4 (BL4), BERTScore (BSc.) and METEOR (MET) over all clickbait posts respectively those requiring phrase, passage, or multi spoilers on the test set. We report all runs by Team nancy-hicks-gribble.

Submission		**All**			**Phrase**			**Passage**			**Multi**		
Team	Approach	BL4	BSc.	MET	BL4	BSc.	MET	BL4	BSc.	MET	BL4	BSc.	MET
nancy-hicks-gribble	short-screw	0.27	0.88	0.26	0.48	0.93	0.44	0.14	0.84	0.25	0.05	0.85	0.23

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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. 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.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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. *CoRR*, abs/1907.11692.

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 of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

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. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.