Clickbait Spoiling: A Combination of Question Answering and Summarization

Gabriel Chen and **Krishna Jayakumar**
University of Waterloo

Abstract

Clickbait has been an issue to internet users for years. It arouses curiosity in readers to click on a link by providing only partial information on interesting topics and only provides low information in the articles. This creates profit for certain websites but wastes readers' time, therefore, to tackle this issue the Clickbait Challenge was created as SemEval 2023. We adopted the classification, question-answering, and summarization natural language processing models and assemble them into a pipeline (Figure 1) to tackle this challenge and discovered it outperforms the question-answering model alone for clickbait spoiling. In this paper we mainly explore DeBERTa for classification, RoBERTa-base for question-answering and t5-base for summarization.

[1]

1 Introduction

Clickbait is a text or link designed to grab a user's attention and get them to click on the link which is quite common on the internet to either drive site traffic or generate advertisement revenue. It can be deceptive and frustrating for the user to click on a highly catchy title or link, thinking they are going to be presented with content that is of value to the individual only to be presented with something that has no value or spreads misleading information. As part of the Clickbait Challenge at SemEval 2023 (Fröbe, Maik and Gollub, Tim and Stein, Benno and Hagen, Mattias and Potthast, Martin, 2023), we aim to spoil clickbaits, which means we will providing brief texts to users that quench their desire of clickbait posts. The challenge consists of two tasks. Task 1 is spoiler type classification where given a post, the type of spoiler needed by the post is to be classified. The type of spoiler, also known as the tag of the post has three classes which are "passage", "phrase" and multi. Task 2 is the spoiler generation where the actual spoiler has to be generated for the post.

There were 2 primary methods tested in task 1 which both involved using pretrained models for classification. One was using a single DeBERTa (He et al., 2021) for multi-class classification and the second method (Thirumala and Ferracane, 2023) was using two DeBERTa models to give binary classifications and choose the spoiler type from those.

In task 2, clickbait spoiling, we first tried the question-answering model alone for predicting all types of spoilers content. Nevertheless, by analyzing the performance of each type, we decided to use a hybrid pipeline, as shown in Figure 1. We adopt the result of the classification model in task 1, clickbait type classification, and merged the output of type "phrase" and "passage" into "non-multi". Next, we feed the content classified as "non-multi" to the question-answering model and feed the one classified as "multi" to the summarization model. In this way, the summarization model is helping the question-answering model on the part, "multi" which it performed poorly, and outperforms the method with the question-answering model alone.

2 Related Work

The clickbait started to be spotted everywhere ever since the rise of the social media era, especially Facebook. The operation of the social media platform encourages users to post all types of content, which means clickbait could be in some of those posts. In the beginning, we saw (Rubin et al., 2015) and (Chakraborty et al., 2016) be the pioneers of making the first detector. There are also some works endeavored on clickbait generation: (Shu et al., 2018) and (Xu et al., 2019). Recently, a new clickbait dataset (Hagen et al., 2022) is created and hosts a challenge (Fröbe, Maik and Gollub, Tim

[1] GitHub repository link: https://github.com/gabrielchen65/clickbait_spoiler Or click here

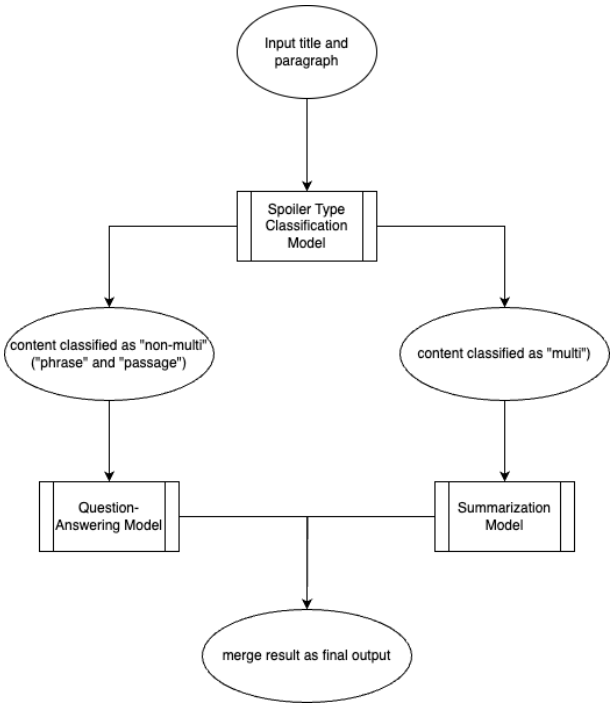

Figure 1: System pipeline.

and Stein, Benno and Hagen, Mattias and Potthast, Martin, 2023). There was a work (Thirumala and Ferracane, 2023) that tried to use LLM (large language model) API (Brown et al., 2020) for generating clickbait spoilers, though it yield a poor results due to the difficulty of fine-tuning. Furthermore, this work (Krog and Agirrezabal, 2023) tried to do the clickbait challenge with the zero-shot approach.

2.1 Spoiler Classification

Classification for NLP tasks have been explored and there are many state-of-the-art models giving high performance (Gasparetto et al., 2022) with models such as BERT and related models giving good accuracy on sequence classification. Within the context of the clickbait challenge there has been success with using DeBERTa(Fröbe et al., 2023) as it improves on BERT(Devlin et al., 2019) and RoBERTa(Liu et al., 2019) by using disentangled representations and an enhanced mask encoder(He et al., 2021)

2.2 Question Answering

The question-answering task is the most relevant task the clickbait spoiling. The question, context, and answer in the question-answering task are corresponding to the title, article, and spoiler in the clickbait spoiling task. There is a remarkable work of dataset SQuAD (Rajpurkar et al., 2016) collecting 100K+ questions from 500+ Wikipedia articles. We adopt the evaluation metric from them and use the pretrained model on it. They created the following work SQuAD v2 (Rajpurkar et al., 2018) which also included un-answered questions. However, we did not use it because we expect the spoiler would always be in the paragraph. The transformer models is popular in this task, the models we use in this work are RoBERTa-base (Liu et al., 2019) and DistilBERT (Sanh et al., 2019).

2.3 Passage Retrieval

Since passage retrieval has similar nature to the question answering but to the extent of relaxing the restriction of only extracting the short answer from the context. In these works (Guo et al., 2020; Lin et al., 2021; Nguyen et al., 2016), we saw some success in passage retrieval. Additionally, we saw (Hagen et al., 2022) using the passage retrieval model to do the clickbait task. However, it is outperformed by the question-answering model.

2.4 Transfer Learning

Transfer learning is a method in machine learning whereby a model is trained for a specific target dataset using an existing model that was pre-trained on a generic dataset. This method is mainly used when the quantity of data in the target domain is considerably low, which can make training a model difficult without the model overfitting. Transfer learning also helps in reducing the training time of the model.

3 Approach

3.1 The Clickbait Type classification

For clickbait type classification for task 1, a variety of approaches were tried. Initially, we trained a few models from scratch. First was a Multinomial Naive Bayes model for multi-class classification using unigram, bigram and a mixed model consisting of uni+bigrams. Next, we tried a using a simple neural network with an embedding layer and a single linear layer. These models did not produce the results we wanted and had poor accuracy. We noticed that the models were overfitting to our training data but not generalizing well and hence had poor validation accuracies which meant that training more complicated models from scratch would not be the best way forward. This was because of the lack of training data with only 3200

samples. Another issue was the class imbalance in the training set provided.

We decided to tackle this by using transfer learning in which we took pretrained BERT model and modified the final layer to support 3-class classification. This worked as intended giving us a much better score than before. To improve on this further we decided to use DeBERTa as it generally performs better than RoBERTa. We named this first model Multiclass-DeBERTa. Our next model was a combination of two DeBERTa models used for sequence classification inspired by(Thirumala and Ferracane, 2023) which are binary classifiers which we call model-1 and model-2. Model-1 is a classifier which classifier between "multi" and the other classes("non-multi"), while model-2 classifies between "passage" and "phrase". To perform the final classification, we first use model-1 to decide between "multi" and "non-multi". If model-1 classifies as "multi", the output is given as "multi". Otherwise we use model-2 to classify between "phrase" and "passage".

3.2 The clickbait spoiler generation

For the clickbait spoiling, after observing the provided data, including the clickbait title, the clickbait paragraph, and the spoiler, we decided to tackle this challenge in the question answering manner. The question-answering task comes with a question, a context, and an answer, and the corresponding parts are the title, paragraph, and spoiler in the clickbait spoiler generation task, which are similar. We can find the same conclusion from (Hagen et al., 2022). According to the exeperiments from (Hagen et al., 2022), we used the following models: DistilBERT (Sanh et al., 2019) and RoBERTa-base (Liu et al., 2019) from Huggingface model hub (hug), and chose the one pretrained on either SQuAD (Rajpurkar et al., 2016) or SQuAD v2 (Rajpurkar et al., 2018) since they are the most commonly used question-answering task dataset and we are expecting the best fine-tuning result from it.

We fine-tuned the models with all of the data. Naturally, the one with more parameters yielded a better result (see Table 1).

Furthermore, we want to improve the performance by taking the features of the data into account. There are three types of spoilers and they all have different average lengths, which indicates they might need different solutions individually. Therefore, we split the data according to their types

	RoBERTa-base	distilbert
meteor	0.3202	0.2774
squad: exact match	19.0	14.25
suqad: f1_score	37.86	31.58

Table 1: The performance of different size model.

and discovered our model performed the best on spoiler type "phrase", which is the shortest one, and then the "passage", and then the performance with "multi" is significantly worst (Table 2). We inferred there might be two reasons that caused this: the length of the answer in "multi" and each answer from a single question are not continuous in the paragraph. The fact that answers are not continuous in the text is the major issue for the question-answering model because they are designed to "extract" certain text from the paragraph, we need a generative model for tackling this task. To this end, we adopt the "summarization model", and used the t5-base (Raffel et al., 2020) model specifically.

3.3 Spoiler Classification

Ideally, we want to separate the methods for each type. Nevertheless, we can only have the tag label from the result of clickbait type classification. The classification results on three classes are not ideal, however, if we focus on the result of classifying "multi" and "non-multi" ("phrase" and "passage"), it is good enough for us to adopt it (see Model-1 in Table 5). We split the data into "multi" and "non-multi" ("phrase" and "passage"), and feed the "multi" data for t5-base (summarization model) for fine-tuning. Finally, we combine the prediction of "non-multi" ("phrase" and "passage") from the question-answering model and the prediction of "multi" from summarization and generate the final result. This improved by around 0.03 Meteor score for question-answering along. (see Table 3)

4 Experiments

4.1 Dataset

The dataset provided by the challenge consisted of a training set with 3200 samples and a validation set and test set with 400 samples each. In the training set for spoiler types (also known as "tags"), there was 1367 samples for "phrase", 1274 for "passage" and "559" for multi. For task 1, we trained the models using the "targetTitle". For task 2, clickbait

	phrase	passage	phrase+passage	multi	all
meteor	0.4298	0.2933	0.3633	0.15819	0.3202
squad/exact match	38.27	9.09	24.05	0.0	19.0
suqad/f1_score	50.23	32.1082	41.39	24.54	37.86

Table 2: The performance of RoBERTa-base on each type of spoiler.

	RoBERTa-base for all	task 1 result + RoBERTa-base for "non-multi" + t5-base for "multi"	Oracle (label) + RoBERTa-base for "non-multi" + t5-base for "multi"
phrase+passage	0.3633	0.3379	0.3633
multi	0.15819	0.4360	0.4557
All	0.3202	0.3521	

Table 3: For the first two columns, we show the combination of RoBERTa-base and t5-base with the result from task 1 is better than RoBERTa-base alone. For the second and the third columns, we compare our final solution to the potential best performance by using the golden label as the classification result.

spoiler generation, we extract "targetParagraphs", "targetTitle", and "spoiler" attributes for training and evaluation.

4.2 Evaluation Metrics

In the context of classification, usually accuracy can be a good metric. However, in the case of the dataset we have, it may not capture the performance of the model due to the fact that there is an imbalace in the dataset. For example, if there is a binary classifier with 80% of data in class 0 and 20% in class 1, a poor model which guesses everything as class 0 would achieve an accuracy of 80%. In such cases, F1 score is a good metric and thus in our classification tasks we have added, precision, recall and F1 score in addition to accuracy. For multi-class problems, precision, recall and F1 scores cannot be directly used as they are intended for binary classification problems, however we can use macro-averaged scores of the three. Macro-averaging takes into account the class imbalances and gives equal weightage to all classes(Opitz and Burst, 2021). For overall accuracy metric, we went with balanced accuracy that takes into account the class imbalance (Grandini et al., 2020) and was the way the challenge was judged(Fröbe et al., 2023).

For clickbait spoiler generation, to align with the Kaggle competition we attended, we majorly use Meteor (Banerjee and Lavie, 2005) metric for computing the spoiler prediction and the answer. SQuAD metric (Rajpurkar et al., 2016) is also adopted in some experiments as the comparison, it comes with the "exact match" and "f1 score". Notice that we modified the answers from "multi" so that the metric can perform without bias from the first answer. We concatenate all of the strings in "multi" into one long string as a single answer. See the following example for demonstrating what are the differences in the Meteor score if we concatenate the string or not:

Without Concatenation:
Prediction sentence: ['Elettra Wiedemann']
Reference sentence: [['Elettra Wiedemann', 'extra strength work, so weights, and quite a few planks for my core. My diet stayed pretty much the same, except I cut out sugar for the week of the shoot']]
Meteor Score: 0.9375

With Concatenation:
Prediction sentence: ['Elettra Wiedemann']
Reference sentence: [['Elettra Wiedemann extra strength work, so weights, and quite a few planks for my core. My diet stayed pretty much the same, except I cut out sugar for the week of the shoot']]
Meteor Score: 0.0559

4.3 The clickbait type classification

For classification, we initially trained 3 multinomial naive bayes models. One for unigrams, bigrams and a combination of both unigram and bigram models. All three of the models had very poor results and the results can be seen in Table 4. All the naive bayes models tend to overfit the training data to a very big extent but generalize very poorly.

	Balanced Accuracy	Overall (Macro)		
		Prec.	Recall	F1
Unigram	0.511	0.605	0.511	0.5140
Bigram	0.471	0.551	0.471	0.4790
Uni+bi	0.494	0.627	0.494	0.496

Table 4: Overall metrics for the Multinomial Naive Bayes models

	Accuracy	Prec.	Recall	F1
Model-1	0.830	0.621	0.488	0.546
Model-2	0.718	0.741	0.691	0.715

Table 5: The metrics score of two binary classification models used to create the Mixed-DeBERTa model.

From previous works (Thirumala and Ferracane, 2023; Fröbe et al., 2023) using a pretrained transformer model gave good performance. Our initial model which we named Multiclass-DeBERTa was a fine-tuned DeBERTa model for sequence classification in which we modified the model to classify for 3 classes. We used a standard initial learning rate of 3e-5 with AdamW as an optimizer. We trained it on 5 epoch and batch size of 6, with the model choosing the best weights at the end based on the validation loss. We provided a weight decay of 0.4 to prevent overfitting and 500 warm-up steps for the model.

For our other pretrained model which we refer to as Mixed-DeBERTa, we have used model hyperparameters which are identical to the one which our work was inspired from(Thirumala and Ferracane, 2023). Both had AdamW as an optimizer but having differing learning rate, with model-1 having a learning rate of 5.9e-6 and model-2 having a learning rate of 8.2e-6 and the decay rates were 0.4 and 0.5 respectively. Model-1 was trained on 4 epochs while model-2 was trained on 2 epochs and both sub-models had a batch size of 16. The results of each sub-model in Mixed-DeBERTa is give in Table 5 and the per class and overall results of the two final models are given in Table 6 and Table 7 respectively. For the Mixed-DeBERTa model, even-though the individual sub-models performed highly the final model did not perform as expected giving a lower accuracy than the Multiclass-DeBERTa model.

4.4 The clickbait spoiler generation

4.4.1 The question-answering method

For all of the models, we use the default setting for optimizer AdamW (Loshchilov and Hutter, 2019), which is the Adam optimizer with fixed weight decay, and chose recommended initial learning rate at 3e-5. For the training epoch, we empirically pick the range from 5 to 30, depending on the convergence of the fine-tuning based on the task.

We saw RoBERTa-based has the best performance among the three models we tried, hence, we further do experiments with it. According to the statics from the dataset (Hagen et al., 2022) (see Table 8), we see the average length of the paragraph is noticeably longer than the "max sequence length" as the hyperparameter. This means that when the paragraphs are preprocessed, they are split into several chunks, which is not ideal for the perception of the model, even with the window stride. To address this, we experimented with a longer "max sequence length" and a longer stride. The result showed that expanding the sequence length and the stride did not affect the performance (see Table 9).

4.4.2 The summarization method

To leverage both the title and the paragraph, we concatenate the title in front of the paragraph, making it the text to be summarized. Inspired by (Krog and Agirrezabal, 2023), if the title does not end with a question mark, we pad a semicolon and a space between the title and the paragraph. If the title ends with a question mark, we only pad a space. For example, give the title: "Wes Welker Wanted Dinner With Tom Brady, But Patriots QB Had A Better Idea" and the paragraph: "It'll be just like old times this weekend for Tom Brady and Wes Welker...", the input text for the model would become: "Wes Welker Wanted Dinner With Tom Brady, But Patriots QB Had A Better Idea: It'll be just like old times this weekend for Tom Brady and Wes Welker...".

According to the statistics from the data (Hagen et al., 2022), the length of the spoilers will mostly be clipped at seventy, therefore we chose this number as the max prediction length for the model hyperparameter.

At first, we tried the t5-base summarization model on all of the data, including "phrase", "passage", and the "multi", and this performed significantly poorer than the question-answering model doing the same job. (see Table 10) Therefore, we

	Phrase			Passage			Multi		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Mixed-DeBERTa	0.634	0.641	0.638	0.579	0.738	0.649	0.775	0.369	0.500
Multiclass-DeBERTa	0.632	0.679	0.6548	0.611	0.627	0.619	0.617	0.500	0.552

Table 6: Per class metrics score for the pretrained DeBERTa models

	Balanced Accuracy	Overall (Macro)		
		Prec.	Recall	F1
Mixed	0.583	0.662	0.583	0.595
Multiclass	0.602	0.620	0.602	0.608

Table 7: Overall metrics score and accuracy for the pretrained DeBERTa models

	Average Lenth (Std.Dev.)	
	Paragraph	Spoiler
phrase	505.9 (599.4)	2.8 (1.6)
passage	602.4 (774.0)	24.1 (18.1)
multi	889.8 (892.2)	33.9 (34.8)

Table 8: The statistics of the paragraphs and the spoilers.

	Meteor Score
Original Setting	0.3728
sequence length: 768	0.3708
sequence length: 1536	0.3728
stride: 256	0.3728

Table 9: Comparison of using different "max sequence length" or "stride"

	RoBERTa -base	t5-base (tuned on all)	t5-base (tuned only on “multi”)
phrase	0.4298	0.12048	0.2852
passage	0.2933	0.2683	0.2758
multi	0.1581	0.1851	0.4557
All	0.3202	0.1889	0.3174

Table 10: Compare the summarization model (t5-base) tuned on different targets. It shows that the t5-base is better suited for "multi".

	phrase+passage
RoBERTa-base (tuned on “non-multi”)	0.3728
RoBERTa-base (tuned on “all”)	0.3633
t5-base (tuned on “non-multi”)	0.2593
t5-base (tuned on “multi”)	0.2806

Table 11: Experimenting how well both the question-answering model (RoBERTa-base) and the summarization model (t5-base) tuned on "non-multi" instead.

decided to let the model only focus on "multi" type of spoiler. Remarkably, the results boost by around 0.3 Meteor score on the "multi" type of spoiler (see Table 10). This shows summarization method is better suited for longer, discontinuous text than the question-answering model.

4.4.3 Further experiments

For comparison, we also used the question-answering or summarization model fine-tuned only on "non-multi" ("phrase" and "passage"). The performance of the question-answering model showed a similar result, but the summarization model performed badly. (see Table 11)

Additionally, we feed the question-answering model with the output of the summarization model. The intuition behind this is that the paragraph often be long, and it would be helpful if the summarization model can extract the essential part of the

paragraph so that the question-answering model can focus more on the target. Surprisingly, this does not aid question-answering to perform better.

4.4.4 Qualitative Result from Spoiler generation

In this section, we show both good predictions and bad predictions from both models. Remarkably, the summarization model can predict a long sentence that is close to the references.

Question-Answering Model (RoBERTa-base):

- Good Example:
  - Title: "The Anytime Snack You Won't Feel Guilty About Eating (VIDEO)"
  - Prediction: "Smoky Paprika-Baked Garbanzo Beans"
  - Reference: "Smoky Paprika-Baked Garbanzo Beans"
- Bad Example:
  - Title: "Five Nights at Freddy's Sequel Delayed for Weird Reason"
  - Prediction: "too dark"
  - Reference: "some of the plot elements are so disturbing that they are making him feel sick"

Summarization Model (t5-base):

- Good Example:
  - Title: "14 Things That'll Happen In 2017 According To T̈he Simpsons̈"
  - Prediction: "1. Some alt-right guy will invent the make-up gun. 2. There will be a referendum on whether or not to deport illegal immigrants. 3. Greedy, corrupt energy firms will cause an environmental catastrophe, and a dome will be built over the contaminated site. 4. The chandelier in Elton John'",
  - Reference: "1. Some alt-right guy will invent the make-up gun. 2. There will be a referendum on whether or not to deport illegal immigrants. 3. Greedy, corrupt energy firms will cause an environmental catastrophe, and a dome will be built over the contaminated site. 4. The chandelier in Elton John's private jet will malfunction. 5. It will be made illegal to teach evolution in schools."
- Bad Example:
  - Title: "The oven cleaning hacks that will have your appliance sparkling in MINUTES without ANY scrubbing (from removing grease with oil to banishing bad smells with water)"
  - Prediction: "1. CREATE YOUR OWN CLEANER 2. CLEAN SHELVES IN 30 MINUTES 3. USE AN OLD TOOTHBRUSH 4. USE A GLASS SCRAPER 5. REMOVE OIL WITH OI"
  - Reference: "homemade oven cleaner place the shelves in a resealable plastic bag, spray with oven cleaner, seal the bag, then leave to soak old toothbrush is an essential oven-cleaning tool glass scraper is ideal for removing tough stains remove greasy build-up on the hood of your oven with oil"

5 Conclusion

In conclusion, for spoiler classification, training a pretrained BERT model produces the best result with a small dataset as the model would be able to generalise well and not overfit on the very few samples provided. In particular DeBERTa for sequence classification and having a three class classifier at the end worked the best. For the clickbait spoiler generation, we can conclude that question-answering is an excellent choice to do the work because both tasks share a similar nature. However, among three types of spoilers, the question-answering model struggled to predict well on the "multi" because it is a collection of answers from different parts of the article, which is against the mechanism of the question-answering model. Furthermore, we discovered the summarization model is a perfect fit for the "multi" type of spoiler due to its generative behavior.

5.1 What we learned

Exceptionally different from computer vision, NLP processes the input or output data in a unique way. In the assignments from this class, we learned about tokenizing, now we further learned how to deal with large-scale or long text from the real world. Those include splitting single text into multiple chunks of data, and mapping them back during training or validating. Additionally, we know how to cope with the issues created by Unicode encoding and decoding so that the data would not be con-

taminated by them. Fine-tuning pretrained models is a very powerful in getting a machine learning model to be fit onto a small but complex dataset. Even then learning a model for NLP tasks is quite challenging and the myriad of model parameters can highly dictate model performance. Furthermore, we encountered new metrics that can help judge model performance as well as metrics suited to different class distributions present.

5.2 Future Work

For future work, in task 1 we can leverage the larger and improved deBERTaV2 model on our training set and also experiment with larger batch sizes since we currently had lower computational resources. We also could try training three different binary classifiers in a "one-vs-all" approach and see how that would fare. To counteract the dataset imbalance another technique like addding class weights could also be explored. For task 2, we have an idea of "rephrasing" the title into a more straightforward "question", and hopefully question-answering model could perform even better on the spoiler extracting.

References

Hugging face models repository. https://huggingface.co/models. Accessed: August 4, 2023.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*, pages 9–16, San Francisco, CA, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Maik Fröbe, Benno Stein, Tim Gollub, Matthias Hagen, and Martin Potthast. 2023. SemEval-2023 task 5: Clickbait spoiling. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2275–2286, Toronto, Canada. Association for Computational Linguistics.

Fröbe, Maik and Gollub, Tim and Stein, Benno and Hagen, Mattias and Potthast, Martin. 2023. Clickbait Challenge at SemEval 2023 - Clickbait Spoiling. https://pan.webis.de/semeval23/pan23-web/clickbait-challenge.htmlrelated-work. Accessed: August 4, 2023.

Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. 2022. A survey on text classification algorithms: From text to predictions. *Information*, 13(2):83.

Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait spoiling via question answering and passage retrieval.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Niels Krog and Manex Agirrezabal. 2023. Diane simmons at SemEval-2023 task 5: Is it possible to make good clickbait spoilers using a zero-shot approach? check it out! In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 477–481, Toronto, Canada. Association for Computational Linguistics.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016)*, volume 1773 of *CEUR Workshop Proceedings*, Barcelona, Spain. CEUR-WS.org.

Juri Opitz and Sebastian Burst. 2021. Macro f1 and macro f1.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 2383–2392, Austin, Texas, USA. The Association for Computational Linguistics.

Victoria Rubin, Niall Conroy, and Yimin Chen. 2015. Towards news verification: Deception detection methods for news discourse. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS48) Symposium on Rapid Screening Technologies, Deception Detection and Credibility Assessment Symposium*, Kauai, Hawaii, USA.

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

Kai Shu, Suhang Wang, Thai Le, Dongwon Lee, and Huan Liu. 2018. Deep headline generation for clickbait detection. In *IEEE International Conference on Data Mining, ICDM 2018*, pages 467–476, Singapore. IEEE Computer Society.

Adhitya Thirumala and Elisa Ferracane. 2023. Clickbait classification and spoiling using natural language processing.

Peng Xu, Chien-Sheng Wu, Andrea Madotto, and Pascale Fung. 2019. Clickbait? sensational headline generation with auto-tuned reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 3063–3073, Hong Kong, China. Association for Computational Linguistics.