Clickbait Spoiler Classification and Detection using Transformers

Mamatha Yarramaneni
mamathay@buffalo.edu

Naman Khurpia
namankhu@buffalo.edu

Mohammed Junaid Shaik
mshaik6@buffalo.edu

Department of Computer Science and Engineering
State University of New York at Buffalo

Abstract

Clickbait links exploit users' curiosity as a strategy to promote a web page without giving a detailed summary of its content. The goal of clickbait spoiling is separated into two tasks - one is clickbait spoiler classification into phrase, passage, and multi, and the other task is clickbait generation, which generates spoilers for the given clickbait and its linked document. The dataset is collected from the Webis-Clickbait-22 corpus, which contains 4000 clickbait posts with manually curated versions of the related documents. A very efficient cleaning pipeline to extract, clean, tokenize and predict the outcome for both tasks is proposed. This final pipeline of classification and spoiler generation is used to predict 800 clickbait posts. RoBERTa-large outperformed the other models with a prediction accuracy of 73.875%, among the neural models used for spoiler classification. Moreover, for spoiler generations, RoBERTa base and sentence transformers outperformed other models.

1 Introduction

Clickbait detection is a challenge that involves developing applications that automatically detect them on various social media platforms and websites. These click baits may sometimes be trivial with less importance so the linked documents could be classified and summarized into shorter sentences. This is the main of the project - to classify the clickbait spoilers and generate spoilers for clickbait posts so that we can improve the ability to distinguish between genuine news and clickbait. To solve the clickbait challenge problem, the dataset from the Webis-Clickbait-22 corpus is considered. This dataset consists of click baits, linked documents, and manually curated spoilers collected from various social media platforms. This paper discusses the models that are developed to classify and generate the spoilers for this dataset. The classification and generation problems have been tested on a wide range of models from classical to state-of-the-art. The clickbait spoiler classification has been implemented using the text classification models, and the clickbait generation has been implemented using the question-answering and passage retrieval methods.

2 Related work

The challenge to find clickbait is not novel and has been a topic of research in several fields. Clickbait detection and personalized blocking by Chakraborty et al. (2016) proposed a solution to reduce the number of clickbait posts displayed to users by using the user's click history and personalized blocking. Klairith et. al. used Thai clickbait and non-clickbait headlines to train and evaluate different models, using various classification techniques. And, in Clickbait Challenge 2017, Potthast et al. (2018) developed a regression model for clickbait detection and strength by using sentiment analysis, readability, and n-grams.
Headline Generation for Clickbait Detection by Shu et al. (2018) proposed a novel approach for generating clickbait by using a bi-directional encoder and attention mechanisms. Xu et al. (2019) proposed a model for generating headlines and identifying clickbait using auto-tuned reinforcement learning. Hagen et al. (2022) proposed clickbait classification and spoiler generation models using RoBERTa, and DeBERTa and reported that the best results for clickbait classification have been done by the RoBERTa model, and for the clickbait generation, DeBERTa large outperformed other models.

3 Methods and Model Architecture

3.1 Spoiler type classification

3.1.1 Methods

The spoiler classification problem has been developed on a wide range of models like classi-

cal - Naive Bayes, SVM, Logistic Regression, and neural models - BERT base/large, DistilBERT, RoBERTa base/large, DeBERTa base/large. The task of clickbait spoiler classification is considered a text classification problem with input as clickbait post title for classification models and a concatenation of post title and post paragraphs for the neural models. The output of these models is one among three labels - phrase, passage, and multi. The models are trained on 3200 clickbait posts and tested on 800 posts.

3.1.2 Final/Best Model Architecture

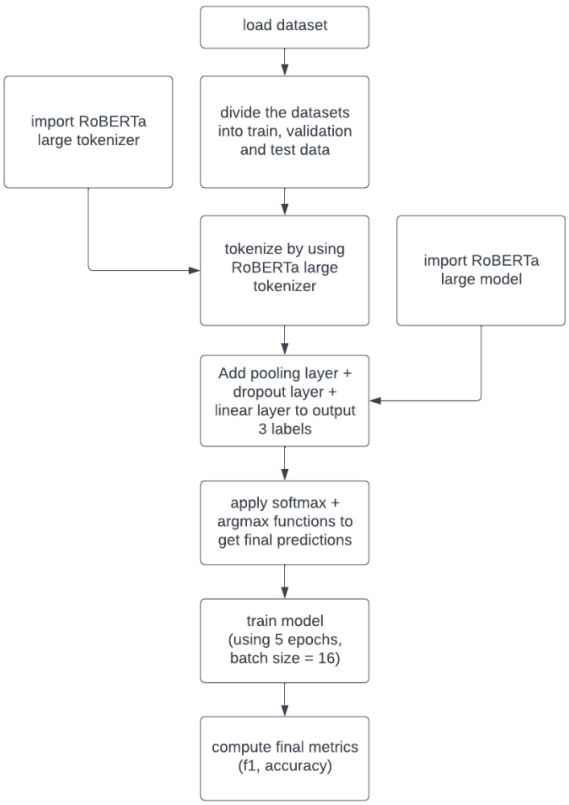

Figure 1: Spoiler classification architecture

Among all the models, RoBERTa large is chosen as the final model for clickbait spoiler classification. As shown in Figure 1, initially, the dataset is imported and split into train, test, and validation sets. These datasets contain 2560 training, 640 validation, and 800 clickbait posts for testing each.
The pre-trained RoBERTa large tokenizer is imported to tokenize the input text. The pre-trained RoBERTa large model is imported and the pooling layer, dropout layer, and linear layers are added to project the final classification to three dimensions to match the output dimensions. The training arguments are defined to fine-tune the models to match our problem using 3200 clickbait posts. The learning rate of the training process is set to 2e-5, with a training and validation batch size of 16 each. The weight decay is set to 0.01, and the model is trained for 5 epochs by saving every checkpoint at the end of each epoch.
After training the models, the best checkpoint in the training process is loaded for testing purposes. This model is used to predict the output for the test dataset with 335 phrases, 322 passages, and 143 multi-posts. Then, softmax and arg max functions are used to process the final model's predictions. Finally, macro-averaged F1 and accuracy metrics are computed.

3.2 Spoiler generation

3.2.1 Methods

The spoiler generation problem has been developed on a wide range of models like classical IR methods - using tf-df vectorization and cosine similarity of the vectors, and neural models - BERT base/large, DistilBERT, RoBERTa base/large, DeBERTa base/large, sentence transformers. The task of clickbait spoiler generation is done separately for the phrase, passage, and multi-type spoilers. The models are trained on 3200 clickbait posts and tested on 800 posts. The concatenation of the post title with post paragraphs is taken as the input for these models.

3.3 Phrase Spoiler generation

3.3.1 Models used

The models used for passage spoiler generation are - DistilBERT base, sentence-transformers using cosine similarities, sentence-transformers using dot scores, RoBERTa base, sentence-transformers with quora DistilBERT base, BERT large uncased, DeBERTa base trained on SQUAD.

3.3.2 Final/Best Model Architecture

Among all the models, the RoBERTa base, pre-trained on the SQUAD dataset is chosen as the final model for phrase clickbait spoiler generation. As shown in Figure 2, initially, the dataset is imported and split into train, test, and validation sets. These datasets contain 2560 training, 640 validation, and 800 clickbait posts for testing each.
The pre-trained RoBERTa base tokenizer is imported to tokenize the input text. The spoiler start and end points, and the overflow, offset management, and max length of the tokens are taken into

consideration while cleaning up the dataset. The training arguments are defined to fine-tune the models to match our problem using 3200 clickbait posts. The learning rate of the training process is set to 2e-5, with a training and validation batch size of 2 each. The weight decay is set to 0.01, and the model is trained for 2 epochs by saving every checkpoint at the end of each epoch.

After training the models, the best checkpoint in the training process is loaded for testing purposes. This model is used to predict the output for the test dataset with 335 phrases, 322 passages, and 143 multi-posts. Then, the context and questions from test datasets are passed to the final model. Finally, the BERT score, BLEU, and METEOR metrics are computed.

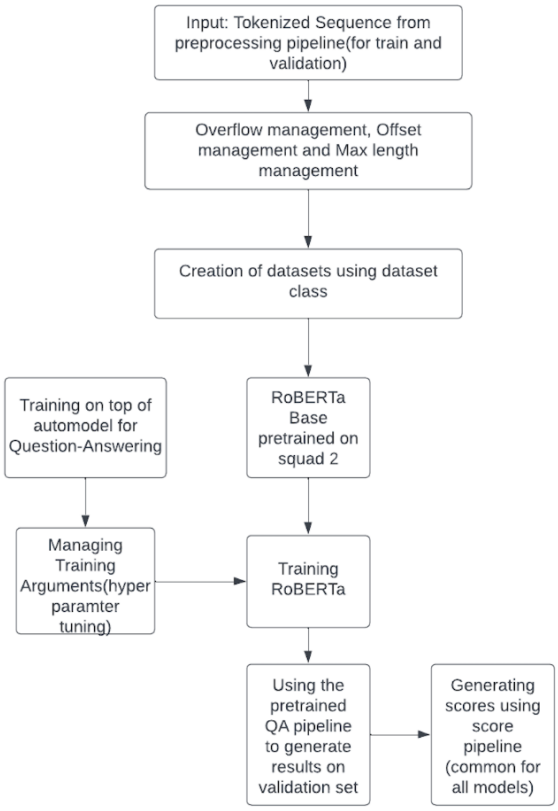

Figure 2: Phrase spoiler generation architecture

3.4 Passage Spoiler generation

3.4.1 Models used

The models used for passage spoiler generation are - DistilBERT base, sentence-transformers using cosine similarities, sentence-transformers using dot scores, RoBERTa base, sentence-transformers with quora DistilBERT base, BERT large uncased, DeBERTa base trained on SQUAD.

3.4.2 Final/Best Model Architecture

Among all the models, the sentence transformer model pre-trained on Wikipedia, BookCorpus, SNLI, and MNLI datasets is chosen as the final model for passage clickbait spoiler generation. As shown in Figure 3, initially, the dataset is imported and split into train, test, and validation sets. These datasets contain 2560 training, 640 validation, and 800 clickbait posts for testing each.

The pre-trained sentence transformers tokenizer is imported to tokenize the input text. MNLI datasets. The respective model's tokenizer is used to preprocess input text to tokens. The sentence transformers use the cosine similarity metric to calculate the similarity between the input question and the sentences in the context and return the maximum similarity

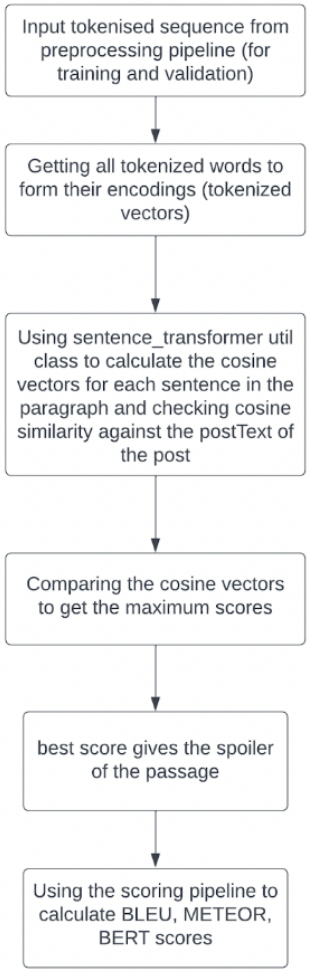

Figure 3: Passage spoiler generation architecture

sentence. The spoiler start and end points, and the overflow, offset management, and max length of the tokens are taken into consideration while cleaning up the dataset. The training arguments are defined to fine-tune the models to match our problem using 3200 clickbait posts. The learning rate of the training process is set to 2e-5, with a training and validation batch size of 2 each. The weight decay is set to 0.01, and the model is trained for 2 epochs by saving every checkpoint at the end of each epoch.

After training the models, the best checkpoint in the training process is loaded for testing purposes. This model is used to predict the output for the test dataset with 335 phrases, 322 passages, and 143 multi-posts. Then, the context and questions from test datasets are passed to the final model. Finally, the BERT score, BLEU, and METEOR metrics are computed.

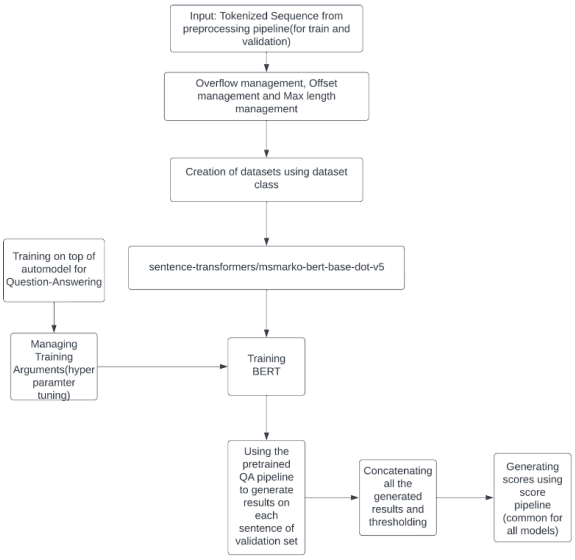

Figure 4: Multi spoiler generation architecture

3.5 Multi Spoiler generation

3.5.1 Models used

The models used for passage spoiler generation are - DistilBERT base, sentence-transformers using cosine similarities, sentence-transformers using dot scores, RoBERTa base, sentence-transformers with quora DistilBERT base, BERT large uncased, DeBERTa base trained on SQUAD, and sentence transformers with semantic search.

3.5.2 Final/Best Model Architecture

Among all the models, the sentence transformer model is pre-trained on the MS MARCO dataset for semantic search, which maps the output to 768-dimensional dense vector space and is chosen as the final model for multi-clickbait spoiler generation. As shown in Figure 4, initially, the dataset is imported and split into train, test, and validation sets. These datasets contain 2560 training, 640 validation, and 800 clickbait posts for testing each.

The pre-trained sentence transformers tokenizer is imported to tokenize the input text. MNLI datasets. The respective model's tokenizer is used to preprocess input text to tokens. The sentence transformers use the cosine similarity metric to calculate the similarity between the input question and the sentences in the context and return the maximum similarity sentence. The spoiler start and end points, and the overflow, offset management, and max length of the tokens are taken into consideration while cleaning up the dataset.

Specifically for multi, each sentence in the context that is passed to the question-answering pipeline discussed above. The start and end logits are used to find the maximum probability score for the final prediction output. If this probability score is greater than 0.1, then the answer is added to the final model. Similarly, this process is repeated for all the records in the test dataset. Then, the models are trained to fine-tune the pre-trained model for our problem, using this approach. The training arguments are defined to fine-tune the models to match our problem using 3200 clickbait posts. The learning rate of the training process is set to 2e-5, with a training and validation batch size of 2 each. The weight decay is set to 0.01, and the model is trained for 2 epochs by saving every checkpoint at the end of each epoch.

After training the models, the best checkpoint in the training process is loaded for testing purposes. This model is used to predict the output for the test dataset with 335 phrases, 322 passages, and 143 multi-posts. This final model is used to predict the outputs of the test dataset using the same threshold strategy. Then, the context and questions from test datasets are passed to the final model. Finally, the BERT score, BLEU, and METEOR metrics are computed.

4 Results

4.1 Spoiler type classification results

RoBERTa large outperformed other models, as evidenced by its impressive accuracy and F1 score of

Model name	Accuracy	F1 score
bert-base-cased	0.66625	0.6618399351725451
bert-large	0.68125	0.6821537241704321
distilbert-base-cased	0.62625	0.6136566019433247
roberta-base	0.6975	0.696888314371547
roberta-large	0.73875	0.7302641134426479
deberta-base	0.70875	0.7056227340408437
deberta-large	0.7325	0.7276887505552079

Figure 5: Spoiler classification scores

73%(shown in Figure 5). As a large variant of the pre-trained model BERT, RoBERTa has established itself as a leader in natural language processing tasks, demonstrating state-of-the-art performance across a wide range of applications. Its advantages include improved performance, transfer learning, large capacity, and multilingual support, making it an excellent choice for any NLP task that involves classification based on multiple tokens.

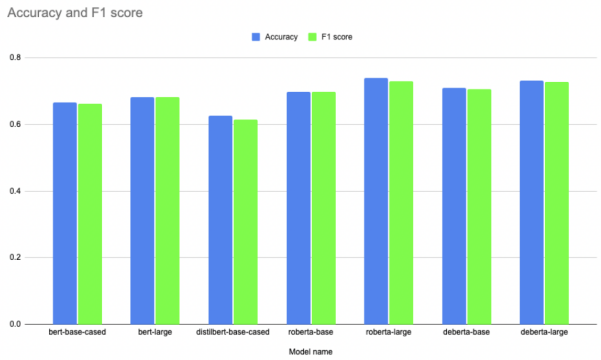

Figure 6: Spoiler classification scores comparison

4.1.1 Comparison of results - task 1

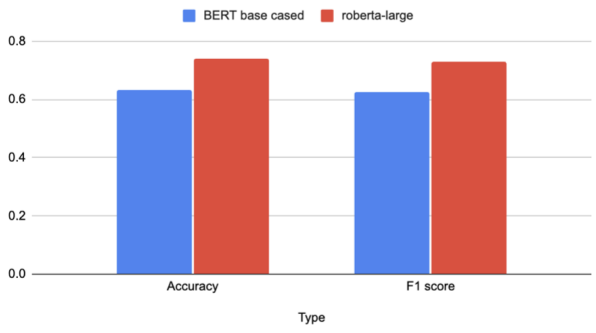

Figure 7: Comparison between final model's and milestone 2 baseline model's results - task 1

Model name	Accuracy	F1 score
BERT base cased	0.63125	0.6242281795
roberta-large	0.73875	0.7302641134

Figure 8: Comparison between final model's and milestone 2 baseline model's results -task 1

Model name	Accuracy
Roberta (ACL paper)	*0.7912*
roberta-large (Our model)	0.73875

Figure 9: Comparison between final model's and ACL paper's results -task 1

The final model's accuracy is 0.738, which is 6.6% lower than the result reported in the ACL paper(Figure 9). The ACL paper has only worked on two types namely phrase and passage. Our model's study also included multi-type spoilers.

4.2 Spoiler generation results

4.2.1 Phrase

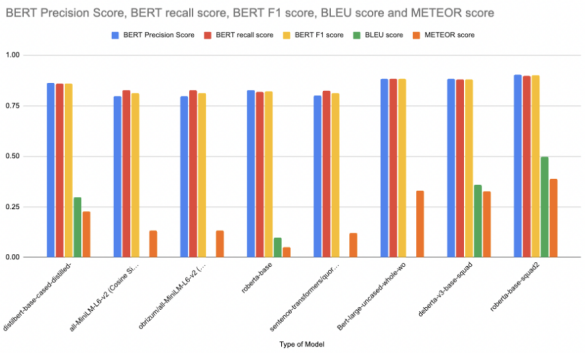

Figure 10: Phrase Spoiler generation scores comparison

From Figure 11, it is evident that Roberta-base-squad-d2 has surpassed all other models, as it achieved superior performance in multiple metrics. Specifically, in comparison to BERT, Roberta-base-squad-d2 obtained a precision score of 90%, a recall score of 89%, and an F1 score of 90%. Additionally, its BLEU score of 49% and METEOR score of 38% further highlight its exceptional performance. When analyzing phrases, Deberta performed comparably to Roberta. It is noteworthy that Roberta-base-squad-d2 was pre-trained on the SQUAD dataset, which contributed to its performance. Overall, these findings emphasize that

Type of Model	BERT Precision Score	BERT recall score	BERT F1 score	BLEU score	METEOR score
distilbert-base-cased-distilled-squad (with finetuning)	0.8611	0.861	0.8601	0.2963	0.2274
all-MiniLM-L6-v2 (Cosine Similarities)	0.7992	0.8281	0.8127	5.52E-233	0.1324
obrizum/all-MiniLM-L6-v2 (Dot scores)	0.7992	0.8281	0.8127	4.54E-232	0.1324
roberta-base	0.827	0.8184	0.8216	0.09807566	0.0491804
sentence-transformers/quora-distilbert-base	0.8019	0.8244	0.8123	4.18E-232	0.1195
Bert-large-uncased-whole-word-masking-finetuned-squad (cosine similarity)	0.883	0.8823	0.8821	1.42E-233	0.32956
deberta-v3-base-squad	0.8824	0.8811	0.881	0.35779428	0.3261704
roberta-base-squad2	0.9046	0.8972	0.9002	0.49842812	0.3897169

Figure 11: Phrase Spoiler generation scores

Roberta-base-squad-d2 is an outstanding model that offers excellent performance for phrase generation.

4.2.2 Comparison of results - Phrase

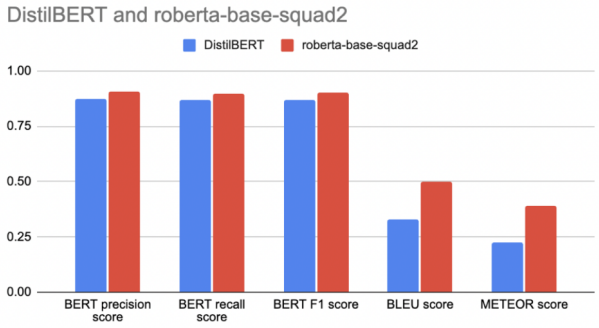

Figure 12: Comparison between final model's and milestone 2 baseline model's results - task 2 Phrase

Type	BERT precision s	BERT recall scor	BERT F1 score	BLEU score	METEOR score
DistilBERT	0.8717	0.8708	0.8704	0.3277	0.2225
roberta-base-squad2	0.9046	0.8972	0.9002	4.98E-01	0.3897169587

Figure 13: Comparison between final model's and milestone 2 baseline model's results - task 2 Phrase

Type of Model	BERT Precision Score	BLEU score	METEOR score
roberta-base-squad2 (Our Model)	0.9046	0.4984281217	0.3897169587
RoBERTa-large (ACL Pa	*0.8404*	*0.7947*	*0.7861*

Figure 14: Comparison between final model's and ACL paper's results - task 2 Phrase

A comparison between BERT precision score, the BLEU score and METEOR scores were conducted in this task. Our model achieved almost two-thirds score in the BLEU score and nearly half in the METEOR score (Figure 14). However, our model outperformed the benchmark by nearly 7% in the BERT score.

4.2.3 Passage

The all-MiniLM-L6-v2 sentence-transformers model has exhibited superior performance compared to other models(Figure 15). This model can map sentences and paragraphs to a 384-dimensional dense vector space, which is useful for tasks such as clustering or semantic search. In comparison to BERT, the all-MiniLM-L6-v2 sentence-transformers model achieved an impressive precision score of 85%, a recall score of 85%, and an F1 score of 85%. Its METEOR score of 25% further indicates its exceptional performance. Additionally, while utilizing dot scores to analyze passages was effective, cosine vectors produced more insightful results.

Type of Model	BERT Precision Score	BERT recall score	BERT F1 score	BLEU score	METEOR score
distilbert-base-cased-distilled-squad (with finetuning)	0.8403	0.8213	0.8304	0.0868	0.0569
all-MiniLM-L6-v2 (Cosine Similarities)	0.854	0.8565	0.8594	5.52E-233	0.2522
obrizum/all-MiniLM-L6-v2 (Dot scores)	0.854	0.8656	0.8594	5.52E-233	0.2522
roberta-base	0.8314	0.8177	0.8243	0.11739245	0.0516242
sentence-transformers/quora-distilbert-base	0.8558	0.8631	0.8592	4.41E-233	0.2229
Bert-large-uncased-whole-word-masking-finetuned-squad (cosine similarity)	0.8528	0.8339	0.8429	4.96E-236	0.11453
deberta-v3-base-squad2	0.8546	0.8314	0.8423	0.13267946	0.0956198
roberta-base-squad2	0.8492	0.8235	0.8358	0.12564868	0.1282661

Figure 15: Passage Spoiler generation scores

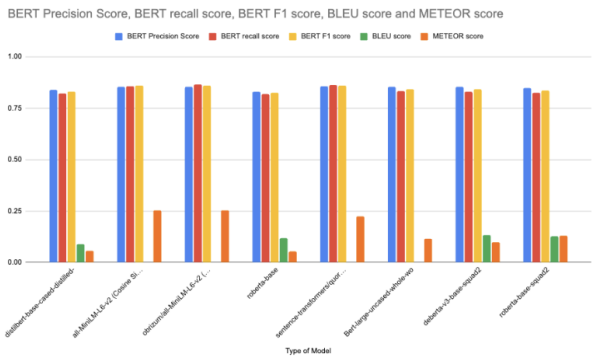

Figure 16: Passage Spoiler generation scores comparison

4.2.4 Comparison of results - Passage

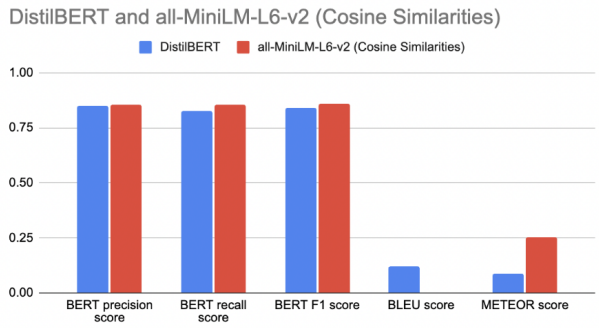

Figure 17: Comparison between final model's and milestone 2 baseline model's results - task 2 Passage

Type	BERT precision sc	BERT recall score	BERT F1 score	BLEU score	METEOR score
DistilBERT	0.8518	0.8273	0.839	0.1209	0.0868
all-MiniLM-L6-v2 (Cosine Similarities)	0.854	0.8565	0.8594	5.52E-233	0.2522

Figure 18: Comparison between final model's and milestone 2 baseline model's results - task 2 Passage

Type of Model	BERT Precision Score	BLEU score	METEOR score
MonoBERT (ACL Paper)	0.18	0.34	0.41
all-MiniLM-L6-v2 (Cosine Similarities) (Our Model)	0.854	5.52E-233	0.2522

Figure 19: Comparison between final model's and ACL paper's results - task 2 Passage

4.2.5 Multi

Type of Model	BERT Precision Score	BERT recall score	BERT F1 score	BLEU score	METEOR score
distilbert-base-cased-distilled-squad (with finetuning)	0.8402	0.8047	0.8217	0.0884	0.0728
all-MiniLM-L6-v2 (Cosine Similarities)	0.8319	0.8223	0.8268	6.62E-233	0.1116
obrizum/all-MiniLM-L6-v2 (Dot scores)	0.8319	0.8223	0.8268	6.62E-233	0.1116
roberta-base	0.8278	0.7962	0.8113	0.0979553497	0.03553560362
sentence-transformers/quora-distilbert-base	0.8321	0.8182	0.8249	4.39E-233	0.0936
Bert-large-uncased-whole-word-masking-finetuned-squad (cosine similarity)	0.8547	0.8132	0.833	2.18E-234	0.1213
deberta-v3-base-squad2	0.8519	0.8054	0.8275	0.0854094376	0.07868105274
roberta-base-squad2	0.8268	0.7834	0.8039	0.0740118560	0.08734685195
sentence-transformers	0.7946	0.8352	0.8141	0.0404891681	0.2292876338

Figure 20: Multi Spoiler generation scores

The Sentence Transformer-msmarco model has proven to outperform other models in various metrics (Figure 20, Figure 21). When compared to BERT, the model achieved a precision score of 79%, a recall score of 83%, and an F1 score of 81%. Its exceptional METEOR score of 22% further

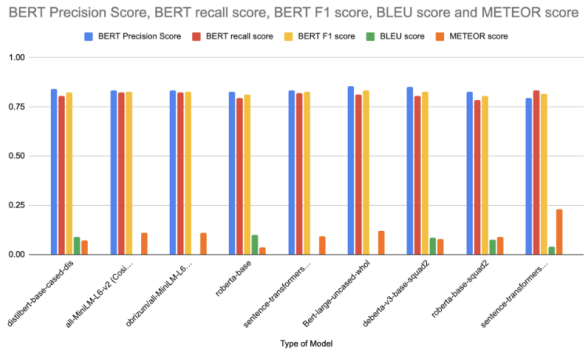

Figure 21: Multi Spoiler generation scores comparison

highlights its superior performance. In terms of multi, none of the other models came close to this approach. While other models performed well in BERT scores, the Sentence Transformer-msmarco model excelled in METEOR scores.

4.2.6 Comparison between final model and milestone 2 Baseline results - Multi

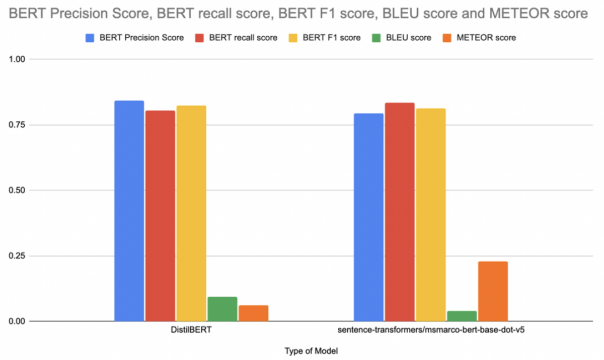

Figure 22: Comparison between final model's and milestone 2 baseline model's results - task 2 Multi

Type of Model	BERT Precision Score	BERT recall score	BERT F1 score	BLEU score	METEOR score
DistilBERT	0.8415	0.806	0.823	0.0929	0.0607
sentence-transformers	0.7946	0.8352	0.8141	0.0404891681	0.2292876338

Figure 23: Comparison between final model's and milestone 2 baseline model's results - task 2 Multi

Type of Model	BERT Precision Score	BLEU score	METEOR score
No results from Paper	NA	NA	NA
sentence-transformer MSMARCO	0.79	0.04	0.22

Figure 24: Comparison between final model's and ACL paper's results - task 2 Multi

5 Discussion and Error Analysis

For task 1, i.e., clickbait spoiler type classification, the RoBERTa large model has achieved almost similar performance compared to the ACL paper. There is a scope for improvement in the model by tokenizing sequences to a different length accumulating larger texts and analyzing parts of speech. Also, these results can be further improved by hyper-parameter tuning of training arguments with different parameters like batch size, number of epochs, etc. A more advanced pre-trained model and perform fine-tuning to improve the results further. Moreover, there is an ambiguity in task 1 classification where certain spoilers can be of passage type but are classified as phrase and multi. Also, due to data set imbalance, the model might be biased to the type that has more records. This model can be further improved by taking these factors into consideration.

Currently, there are a large number of spoiler-post text examples in the dataset, making the training process easier and yield better results for phrases. However, the count of multi-type posts examples is very low. Moreover, the accuracy may be low due to the hyperparameters in the training process. It is not appropriate to have the same values for all types of data.

Besides inconsistencies and errors, a better preprocessing pipeline can be generated that accurately labels and performs stemming, lemmatization, and tokenization for the correct tokens. Another approach is to incorporate more context, such as replacing similar words with synonyms and short forms with full forms and replacing any slang with the original word, removing any hyperlinks in the original target document. Additionally, a more advanced model like GPT can be utilized for a question-answering system. Developing a feedback loop can also help in improving accuracy as the model feeds its answer and asks for feedback that can be used to enhance its performance.

Overall, for task 2, the question-answering and retrieval pipeline can be improved by optimizing, pre-processing, incorporating more context, fine-tuning the language model, using more advanced models, implementing an ensemble approach, and developing a feedback loop.

6 Conclusion

The task of clickbait spoiler type classification and spoiler generation has been outperformed by the transformer models. The clickbait spoiler type classification is considered as a text classification task, while the clickbait spoiler generation is considered as a question answering or a passage retrieval task. These two tasks have been implemented by finetuning various state-of-the-art models that give remarkable results. The spoiler-type classification results almost match with the state-of-the-art results, while for spoiler generation, the phrase type gives good scores, while there is a scope for improvement for passage and multi-type spoiler generation.

7 Contributions

Name	Contributions
Mamatha Yarramaneni	Preprocessed the dataset for task 1 and task 2. Implemented task 1 using classical models and neural models. Implemented clickbait spoiler generation using IR methods. Built the best neural model using RoBERTa large for task 1 and using sentence transformers for a multi-spoiler generation.
Naman Khurpia	Implementation of Preprocessing the code, Implementation of Training the dataset for DistillBERT and predicting tokens, Implementation of Cosine similarities for passage retrival methods, Implementation of dot scores for passage retrival methods, Calculation of Scores via creating a scoring pipeline - BLEU, METEOR, BERT scores for task 2, creation of Github repository with all datasets.
Mohammad Junaid Shaik	Implementing the input pipeline for preprocessing, Implementation of Training the dataset for Roberta-base, desert-base and Bert-large-uncased for the best phrase retrieval. Validating the scoring pipeline using vanilla models.

Table 1: Contributions to the project

References

Dataset - webis clickbait spoiling corpus 2022.

Fine tuning transformer models for qa custom data - medium article.

Qa finetuning custom data - skandavivek.

Qa pre trained transformers pytorch custom data - towardsdatascience article.

Sentence transformers - huggingface.

Text classification - training and inference using huggingface.

Tf-idf vectorizer - sklearn.

Trainer huggingface.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait spoiling via question answering and passage retrieval. *arXiv preprint arXiv:2203.10282*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

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*, pages 311–318.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

(Zhu et al., 2020) (Devlin et al., 2018a) (Liu et al., 2019) (He et al., 2021) (Vaswani et al., 2017) (Devlin et al., 2018b) (Hagen et al., 2022) (Reimers and Gurevych, 2019) (Zhang et al., 2019) (Banerjee and Lavie, 2005) (Papineni et al., 2002) (Wolf et al., 2019) (Wolf et al., 2020) (Hug) (Skl) (med) (tow) (ska) (tra) (tex) (web)