Clickbait Spoiler Classification and detection using Transformers

Mohammed Junaid Shaik, Naman Khurpia, Mamatha Yarramaneni
School Engineering and Applied Sciences
State University of New York at Buffalo

Abstract

Clickbait spoiling was divided into two tasks of Spoiler classification and Spoiler generation of three different types - Phrase, Passage and Multi. For both the tasks we created a very efficient cleaning pipeline which would extract, clean, tokenise and massage the data which would later act as input to the classification and LLM models that we train. We have a dedicated train function which is based on various models as you would see below that would give us final spoiler predictions. From the various models we used for training our model, we found that there is no unique model to solve all the tasks, instead each model has a different Huggingface transformer that would give better results than the rest. The models that we have tested for task 1 include bert-base-cased, bert-large, distilbert-base-cased, roberta-base, roberta-large, deberta-base and deberta-large in which roberta-large came out on top. For Task 2, we trained models like distilbert-base-cased-distilled-squad, all-MiniLM-L6-v2, obrizum/all-MiniLM-L6-v2, roberta-base, sentence-transformers/quora-distilbert-base, Bert-large-uncased-whole-word-masking-finetuned-squad, deberta-v3-base-squad, roberta-base-squad2 and msmarco-bert-base-dot-v5 in which roberta-base-squad2, all-MiniLM-L6-v2 and msmarco-bert-base-dot-v5 were the best models for various types of spoilers. The remaining results are later compared to the baseline results of vanilla models and the ACL paper that we have used as the main project task.

1 Introduction

The challenge "Clickbait Detection and generation of spoiler" aims to develop and evaluate systems for automatically identifying clickbait headlines on news websites and social media platforms. Clickbait refers to headlines that use sensationalism, exaggeration, or misleading information to entice users to click on a link. The challenge provides a dataset of headlines labeled as either clickbait or not, and we have to create machine learning model that can accurately classify new headlines as clickbait or not. The goal is to improve the ability to distinguish between genuine news and clickbait.

As a solution we have created a pipeline to preprocess this data and then train various models that would use huggingface transformers to then take in this data and generate spoilers to the posts that were basically clickbait. We generated results from vanilla models in milestone 2 and for milestone 3 we added out own pipelines that would further enhance these results and we strived to achieve results as close as possible to the state of the art results provided in the ACL paper that we were provided with.

2 Related Work: A literature survery

We have researched the following sources and papers for our understanding of the task and the creation of the process and pipelines pertaining to the tasks asked:

- Clickbait detection and the personalized blocking by Chakraborty et al. (2016)
- Clickbait Challenge 2017, Regression Model for Clickbait detection and strength by Potthast et al. (2018)
- Headline Generation for Clickbait Detection by Shu et al. (2018)
- Headline Generation for Clickbait Detection using auto-tuned reinforcement learning by Xu et al. (2019)
- Clickbait Spoiler generation introduced by Hagen et al.(2022)
- Thai Clickbait Detection Algorithms Using Natural Language Processing with Machine Learning Techniques

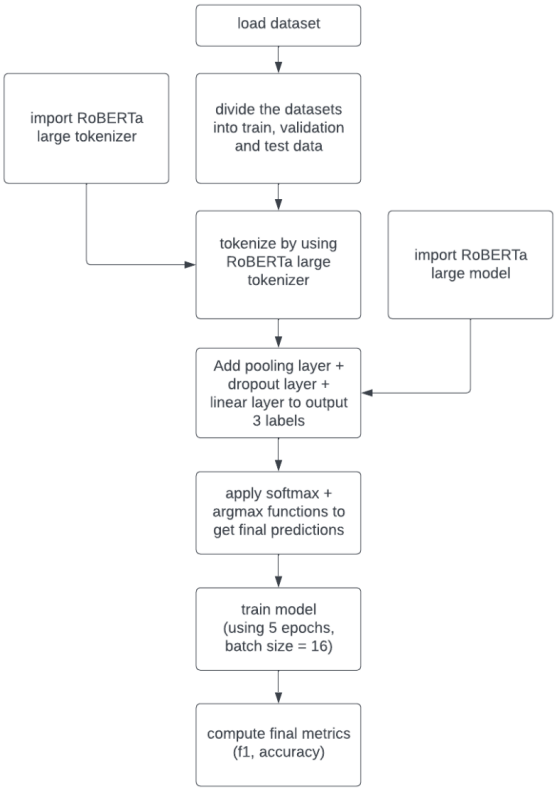

Figure 1: Task one Architecture - Classification of type of clickbait

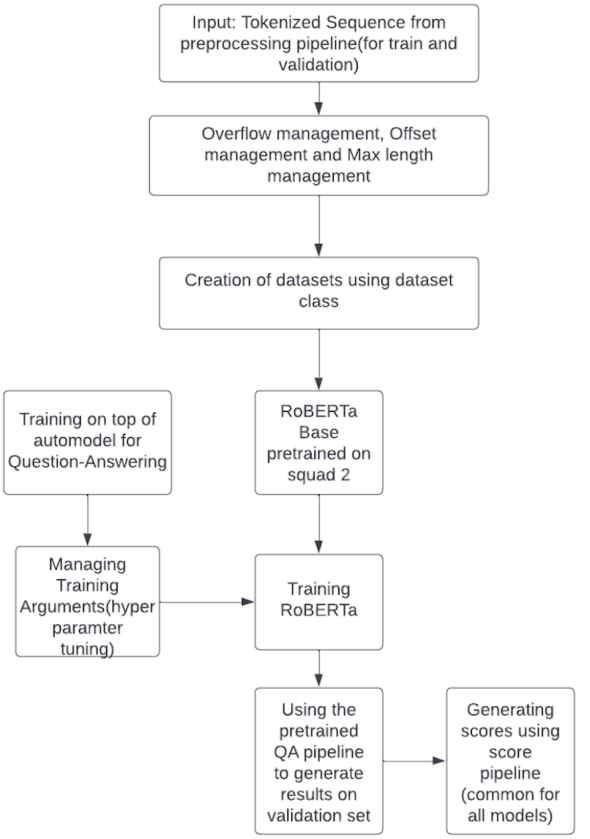

Figure 2: Task two Architecture - Generation of tokens for phrase type spoiler

- Question-Answering Models
- Clickbait spoiling as a question-answering problem
- SQuAD v1.1 (107,785 questions and answers) and TriviaQA (95000 questions and answers)
- BERT, Big Bird, DeBERTa, ELECTRA, MPNet, RoBERTa
- Passage Retrieval
- Benchmark - MS MARCO
- MonoBERT, MonoT5, BM25, Query Likelihood

3 Methods/Model Architecture

3.1 Task one (Spoiler type classification)

Regarding this procedure, see Figure 1. We used the test train split library to import the dataset and divide it into a train, test, and validation set. The question gave us the validation set to utilize in our accuracy calculations. There are 2560 records for training, 640 records for validation, and 800 records for testing in each dataset. The Roberta-large is the ultimate classification scheme for clickbait. The model is imported, and the pooling layer, dropout layer, and linear layers are added to reduce the model's final dimension to three to match the result. The pre-trained model is then modified using the newly discovered models to fit our circumstance. For the procedure's training and validation phases, a batch size of 16 is employed, and the learning rate is 2e-5

The weight decay of the model is set at 0.01. Throughout the model's five training epochs, every checkpoint is saved. At the conclusion, the very best and last checkpoint is loaded. This final model predicts the results of the test dataset. The softmax and argmax methods are then used to process the predictions of the finished model. In the end, accuracy and f1 are computed.

3.2 Task two (Generation of tokens for phrase type spoiler)

First, the dataset is imported and split into three sets (train, test, and validation) using the test train split

library from sklearn. The sizes of these sets are 2560, 800, and 640 records respectively. The final model used for producing clickbait phrase spoilers is the Roberta base, which was pre-trained using the SQUAD 2.0 dataset. To tokenize the input text into tokens, the appropriate tokenizer for the model is used. The pre-trained model is then fine-tuned for our specific situation using newly learned models. A batch size of 2 is used for both the training and validation stages of the process, and the learning rate is set at 2e-5, with a weight decay of 0.01. Checkpoints are saved at every epoch during the two epochs of training for this model. For a better understanding of the entire architecture step-by-step, refer to Figure 2.

3.3 Task two (Passage retrival using cosine similarities)

The cosine similarity between the input question (referred to as "posttext") and targeted paragraphs (referred to as "context") is calculated using the "use cosine similarity" method from the sentence transformer library. These sentences are then ranked in descending order based on their similarity score, and the sentence with the highest similarity score is returned. BERT, BLEU, and METEOR scores are then computed as metrics. For a clearer understanding of the architecture, please refer to Figure 3.

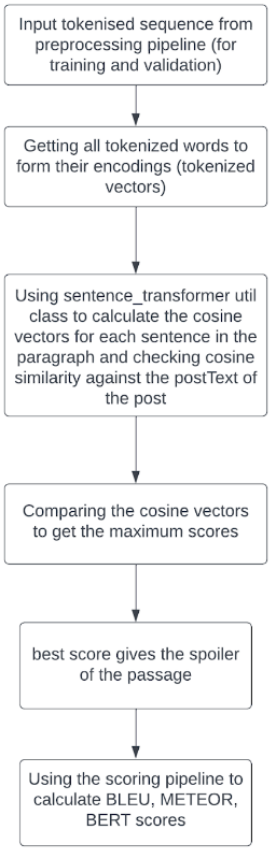

Figure 3: Task two Architecture - Paragraph similarity calculation and retrieval

3.4 Task two (Multi type spoiler token generation)

For the generation of multi-type spoilers, we used a sentence transformer model that was pre-trained on the MSMARCO dataset, which translates the input to a 768-dimensional dense vector space. The process for generating multi-type spoilers is no different from that for generating phrases, which was discussed in section 3.2. We first train our model using the same method, but during prediction, we pass the context as sentences that are split from targeted paragraphs. The answers generated have a threshold value, and this value is added to the final result if the probability score is greater than 5 percent. For a clearer understanding, please refer to Figure 4.

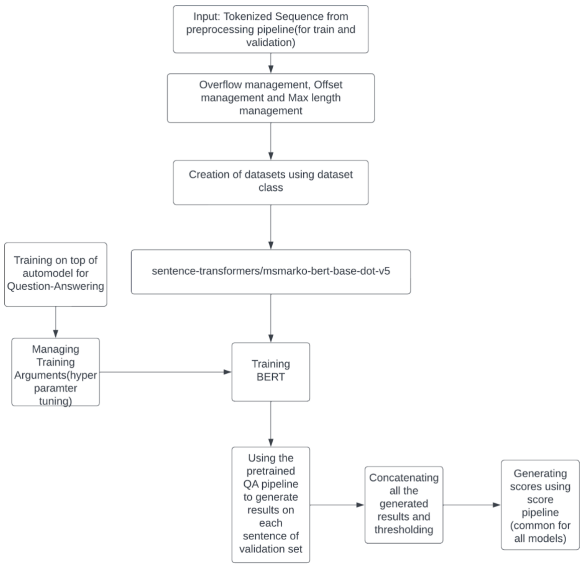

Figure 4: Task two Architecture - Multi type spoiler, generation of multiple tokens

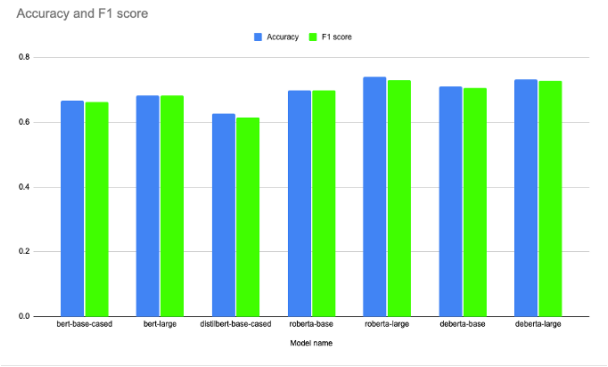

Figure 5: Task one results - classification of spoiler type

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 6: Task one results - classification of spoiler type

4 Results and comparison

4.1 Results of our experiment

4.1.1 Spoiler classification

From the results, we can observe that the Roberta large model outperformed all other models. We achieved an accuracy and F1 score of 73 percent using this model.

4.1.2 Phrase spoiler generation

The Roberta-base-squad-d2 model pre-trained on the SQUAD dataset performed better than any other model, achieving a BERT precision score of 90, a BERT recall score of 89, a BERT F1 score of 90, a BLEU score of 49, and a METEOR score of 38. For generating phrases, the Deberta model also performed similarly to Roberta. Please refer to Figures 7 and Figure 8 for a more detailed understanding.

4.1.3 Passage spoiler generation

In the case of passages, the calculation of dot scores was also a good approach, but using cosine vectors was produced better results.We can see from the results that the all-MiniLM-L6-v2 model outperformed all other models, achieving a BERT precision score of 85 percent, a BERT recall score of 85 percent, a BERT F1 score of 85 percent, and a METEOR score of 25 percent. Please refer to Images 9 and 10 for a more detailed understanding.

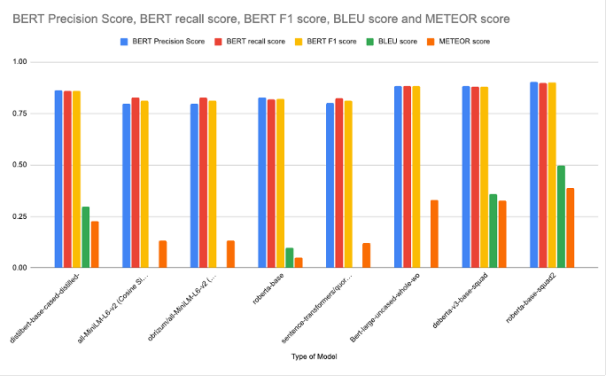

Figure 7: Task two results - generation of phrase type spoiler

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 8: Task two results - generation of phrase type spoiler

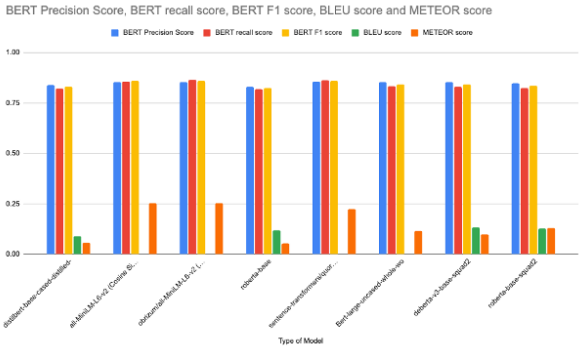

Figure 9: Task two results - passage retrieval

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 10: Task two results - passage retrieval

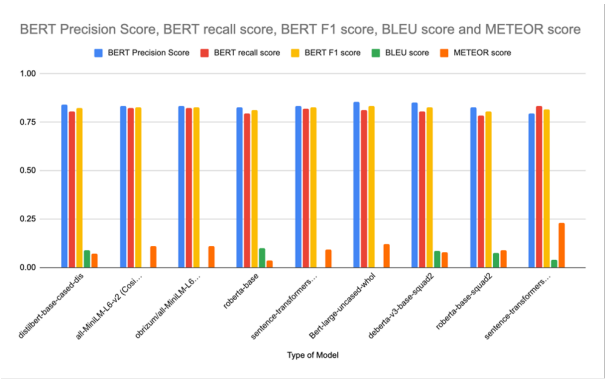

Figure 11: Task two results - generation of multi type spoiler

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 12: Task two results - generation of multi type spoiler

4.1.4 Multi spoiler generation

The Sentence transformer -msmarco is the best performing model in this case, achieving a BERT precision score of 79 percent, a BERT recall score of 83 percent, a BERT F1 score of 81 percent, and a METEOR score of 22 percent. We could not achieve results closer to this using any other model. For a more detailed understanding, please refer to Figure 11 and Figure 12.

4.2 Comparison of Baseline results (milestone 2) and milestone 3 results

4.2.1 Comparison between baseline and final results - for spoiler type classification

Significant jumps in performance and accuracy were achieved for milestone 3. Roberta Large trained on 160GB of text gave an accuracy jump of 73 percent and F1 score of 73 percent. Since it was trained on a large corpus of data we can attribute the increase in performance to it. Please refer to Figure 13 and Figure 14 for a graphical comparison.

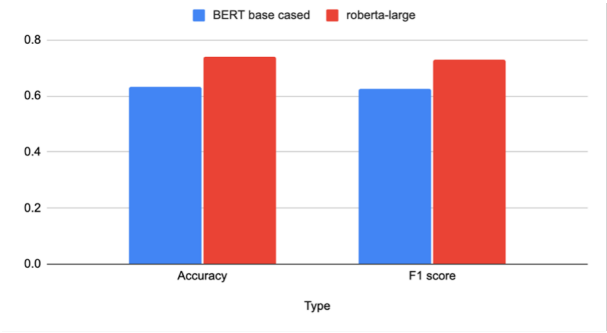

Figure 13: Task one results - Comparison between and final results - spoiler type classification

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

Figure 14: Task one results - Comparison between results and current results - spoiler type classification

4.2.2 Task two - Comparison between baseline and current results - phrase spoiler generation

A 16 percent increase in both meteor and accuracy can be attributed to the SQUAD2 dataset on which Roberta was trained on. Refer Figure 15 and Figure 16.

4.2.3 Task two - Comparison between baseline and current results - passage spoiler generation

This is practically a comparison between distilbert which was used to achieve the milestone 2 result and cosine similarity model used for the milestone 3. This change was inspired because with the similarity model we can compare synonyms and similar words. Refer Figure 17 and Figure 18.

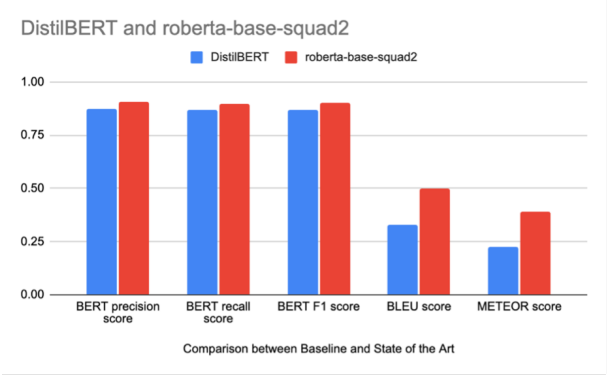

Figure 15: Task two - Comparison between baseline and current results - phrase spoiler generation

Type	BERT precision s	BERT recall score	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 16: Task two - Comparison between baseline and current results - phrase spoiler generation

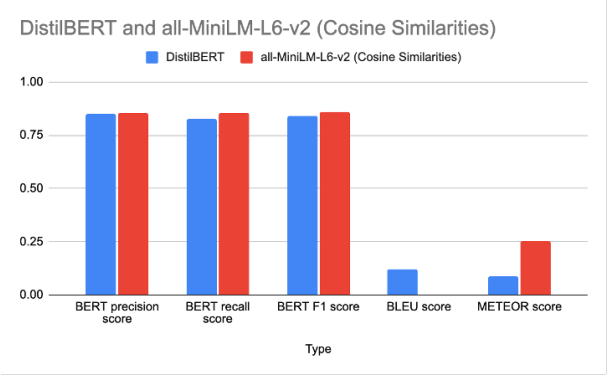

Figure 17: Task two results - Comparison between baseline and current results - passage spoiler generation

4.2.4 Task two - Comparison between baseline and current results - multi spoiler generation

We chose to go with a sentence transformer pre-trained on MSMARCO dataset which is specifically designed for information retrieval and question answering task. The results when compared to normal LLM transformers can be seen in Figure 19 and Figure 20.

4.3 Comparison of State of the Art results and our results

4.3.1 Comparison between State of the art results and our results - spoiler type classification

Results from the ACL paper that have the highest METEOR score are chosen to be the State of the art results and are further used to compare with our model. We see that our model has a Accuracy nearly 6.6 percent lower than the ACL paper result. Refer Figure 21 and Figure 22.

Type	BERT precision s	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: Task two results - Comparison between baseline and current results - passage spoiler generation

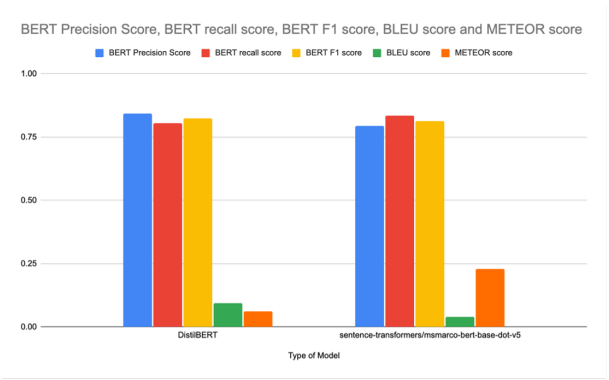

Figure 19: Task two results - Comparison between baseline and current results - multi spoiler generation

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 20: Task two results - Comparison between baseline and current results - multi spoiler generation

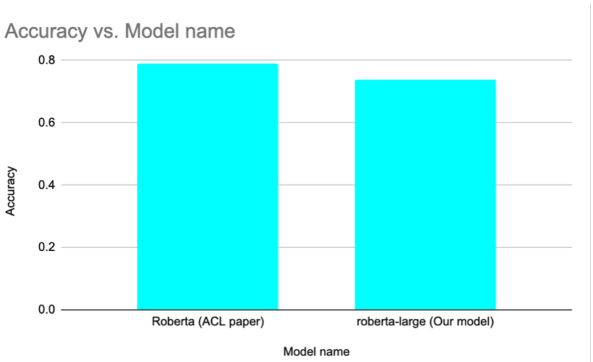

Figure 21: Task one results - Comparison between State of the art and our results - spoiler type classification

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

Figure 22: Task one results - Comparison between State of the art and our results - spoiler type classification

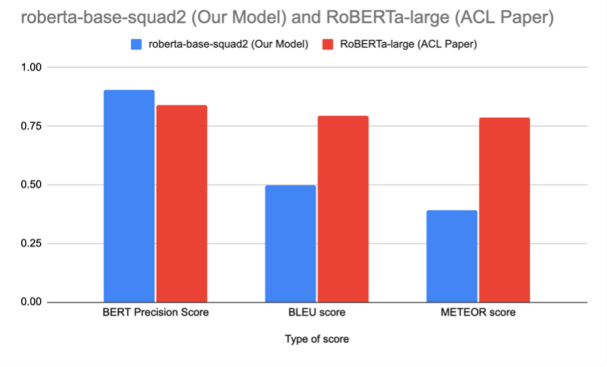

Figure 23: Task two results - Comparison between State of the art and our results - phrase spoiler generation

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 24: Task two results - Comparison between State of the art and our results - phrase spoiler generation

4.3.2 Comparison between State of the art and our results - phrase spoiler generation

Underperformance of one-thirds in the BLEU score and nearly half in the METEOR score can be observed when compared with our results. But, our model performs better than the ACL paper(for the particular model that we chose) in the BERT score by nearly 7 percent. This can be seen in Figure 23 and Figure 24

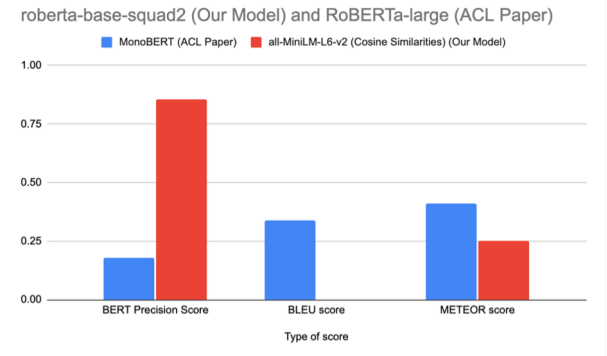

Figure 25: Task two results - Comparison between State of the art and our results - passage spoiler generation

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 26: Task two results -Comparison between State of the art and our results- passage spoiler generation

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 27: Task two results - Comparison between State of the art and our results - multi spoiler generation

4.3.3 Comparison between State of the art results and our results - passage spoiler generation

Major differences in the results was observed for passage spoiler generation when compared to the results in the ACL paper. Refer Figure 25 and 26.

4.3.4 Comparison between State of the art results and our results - multi spoiler generation

With no reference from the ACL paper in this context, we have mentioned our meteor score of 22 percent in Figure 27.

5 Discussion and Error Analysis

In this section, we would like to discuss on how we think this work can be improved to get better results. For task 1, we can use a more efficient pre-trained model combined with analysis of part of speech and tokenizing this. This model would require further tuning of hyper parameters, but it has the capability to give comparatively better results. After expansively testing of various models available we think the data sets can also be improved thus removing the ambiguity between various types of spoilers.

Another change would be increase in the number of multi type spoilers in the data set. This would lead to better training and better results as we see in the case of phrase spoilers. And, better parameter tuning could also help enhance the results.

In addition to the previously mentioned inconsistencies, there are some steps we can take to improve the pre-processing pipeline. For example, we can accurately label, stem, lemmatize, and tokenize the correct tokens. We can also incorporate more context by replacing similar words with synonyms, expanding short forms to their full forms, and replacing any slang with the original word. Using a more advanced model like GPT for a question answering system could also be beneficial. Another potential improvement is to develop a feedback loop where the model feeds its answer and asks for feedback, which can be used to improve the model's performance over time.

6 Conclusions

We can conclude that various types of spoiler generation need various models, and one model is not the solution to everything. Classical models have given very good performance for task 1, whereas different types of LLM models have been the solution to different type of task 2 problems. Another conclusion we can get is that an increase in the size and variety of the data set can possibly help us get better results. In the case of passage generation, we see that our pipeline with a LLM model gives a nearly comparable result, but our pipeline really shines through in the phrase generation. As for multi type spoiler generation, we have do not have a frame of reference to compare to.

7 Contributions

Name	Contributions
Mohammed Junaid Shaik	Implementing the input pipeline for preprocessing,Implementation of Training the dataset for roberta-base, deberta-base and Bert-large-uncased for the best phrase retrival. Validating the scoring pipeline using vanilla models.
Naman Khurpia	Preprocessing the input, Implementation of Training the dataset for DistillBERT, Cosine similarities and dot scores for the best passage retrieval methods. Creating a scoring pipeline for BLEU, METEOR, BERT scores for task 2.
Mamatha Yarramaneni	Preprocessed the dataset for task1 and task2. Implemented task1 using classical models and neural models. Implemented clickbait spoiler generation using IR methods. Built the best neural model for task 1 and for multi spoiler generation

Table 1: Contribution Table

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 6000-6010).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

He, P., Liu, X., Gao, J. and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.

Hagen, M., Fröbe, M., Jurk, A. and Potthast, M. (2022). Clickbait spoiling via question answering and passage retrieval. arXiv preprint arXiv:2203.10282.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Hugging Face. (n.d.). Transformer Models: Sentence-Transformers/multi-qa-MiniLM-L6-cos-v1. https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1.

scikit-learn. (n.d.). TfidfVectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

Hugging Face Forum. (n.d.). Trainer only doing 3 epochs no matter the TrainingArguments. https://discuss.huggingface.co/t/trainer-only-doing-3-epochs-no-matter-the-trainingarguments/19347/5.

Hugging Face. (n.d.). Preprocessing Input Data: Question Answering. https://huggingface.co/docs/transformers/tasks/question_answering#preprocess.

Towards Data Science. (n.d.). Fine-tune Transformer Models for Question-Answering on Custom Data. https://towardsdatascience.com/fine-tune-transformer-models-for-question-answering-on-custom-data-513eaac37a80.

Hugging Face. (n.d.). Custom Datasets. https://huggingface.co/transformers/v3.3.1/custom_datasets.html#qa-squad.

Towards Data Science. Question Answering with Pretrained Transformers using PyTorch. https://towardsdatascience.com/question-answering-with-pretrained-transformers-using-pytorch-c3e7a44b4012.

Towards Data Science. https://github.com/skandavivek/transformerQA-finetuning

Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q., and Artzi, Yoav. "BERTScore: Evaluating Text Generation with BERT." In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1499–1510, Online, 2020. Association for Computational Linguistics. arXiv preprint arXiv:1904.09675.

Banerjee, Satanjeev and Lavie, Alon. "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, USA, 2005. Association for Computational Linguistics.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. "BLEU: A Method for Automatic Evaluation of Machine Translation." In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 311-318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.

Wolf, Thomas, Debut, Lysandre, Sanh, Victor, Chaumond, Julien, Delangue, Clement, Moi, Anthony, Cistac, Pierric, Rault, Tim, Louf, Rémi, Funtowicz, Morgan, Davison, Joe, Shleifer, Sergey, von Platen, Clara, Ma, Caglar, Jernite, Yacine, Plu, Jean-Baptiste, Xu, Canwen, Le Scao, Yoann, Gugger, Sylvain, Drame, Quentin, Lhoest, Quentin, and Rush, Alexander M. "HuggingFace's 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, 2020. Association for Computational Linguistics. arXiv preprint arXiv:1910.03771.