Clickbait Spoiler Classification and Generation: A Comparative Analysis of Different Models for Clickbait Challenge 2023

Krishna Tarun Saikonda Srujani Vattikonda

ttsaikonda@uwaterloo.ca svattiko@uwaterloo.ca

Abstract

In this paper, we conduct a comparative analysis of different models for addressing the challenges of the second PAN Clickbait Challenge, Task 5 at SemEval 2023. Our focus is on supporting social media users by generating concise spoilers to close the curiosity gap induced by clickbait posts. For Subtask 1, we employ three approaches for spoiler type classification, assessing their performance in determining the suitable spoiler type for clickbait posts. In Subtask 2, we perform SQuAD analysis using three models to generate spoilers that complement clickbait posts. Our findings offer insights into each model's effectiveness, contributing to the advancement of clickbait research and user-friendly systems for detection and spoiler generation.

1 Introduction

Clickbait refers to online content, such as headlines or social media posts, deliberately designed to entice users to click on the link or engage with the content. It often employs sensationalized or misleading information, creating a "curiosity gap" that prompts users to click. While clickbait can boost website traffic and ad revenue, it is criticized for potentially delivering content that does not live up to its promises. As a result, it can negatively impact user trust and credibility in online sources. Striking a balance between attracting engagement and providing valuable content is essential for content creators and platforms to maintain user satisfaction and authenticity in the digital landscape.

Figure 1 shows four examples of clickbait on Twitter, along with spoilers. Generally, clickbait spoilers can be of 3 types: phrase, passage and multi-line based. As seen from the figure, all 3 spoiler types have different text structure, length etc. Figure 2 and Figure 3 represents the basic BERT model structure based on which we worked on different models in our classification and generation tasks.

Figure 1: Illustration of some example inputs and the expected output for clickbait spoiling

In the first task, RoBERTa(Liu et al., 2019),DistillBert(Sanh et al., 2020) and DeBERT(He et al., 2021) models were utilized for classification to determine the type of spoiler (i.e., "phrase," "passage," or "multi"). By leveraging the pre-trained representations of these transformer-based models, the aim was to predict the appropriate spoiler type and automatically generate concise spoilers that bridge the curiosity gap induced by clickbait posts, thereby delivering more informative content to social media users. The evaluation of the classification models was based on accuracy and F1 score metrics, providing insights into the overall correctness and balanced performance of the models in accurately classifying the various spoiler types. These evaluation metrics guided the selection of the most effective model for predicting spoiler types in clickbait posts, enhancing the user experience with more contextually relevant and engaging content.

In the second task, SQuAD analysis was conducted using pre-trained models, namely BERT-base(bert-base-uncased-squad v1, 2019), DeBERTa-v3-base-squad2(deberta-v3-base squad2, 2022), and RoBERTa-base-SQuAD2(base squad2, 2022). These models have undergone pre-training specifically for question answering tasks on the SQuAD dataset. The main objective was to generate spoilers for clickbait posts and evaluate the quality of the generated spoilers. Leveraging these pre-trained models, the spoiler generation process was conducted, aiming to produce informative spoilers that align contextually with the clickbait posts.

To assess the performance of the generated spoilers, three evaluation metrics were employed. METEOR measured the linguistic similarity between the generated spoilers and the ground truth spoilers. BLEU score quantified the overlap between the generated spoilers and the reference spoilers. Additionally, the exact match metric provided insights into the percentage of spoilers that exactly matched the ground truth spoilers. Through the comparative analysis of the BERT, Deberta, and RoBERTa models using these evaluation metrics, valuable insights were gained into the effectiveness of each model for the spoiler generation task. This comparative evaluation enabled the identification of the strengths and weaknesses of each model, aiding in selecting the most appropriate model for generating accurate and contextually relevant spoilers for clickbait posts.All our code is published in GitHub.[1]

2 Dataset

The dataset for the clickbait challenge comprises clickbait posts, manually cleaned versions of the linked documents, and extracted spoilers for each clickbait post, categorized into three types: short phrase spoilers, longer passage spoilers, and multiple non-consecutive pieces of text. The dataset is provided in JSON Lines format (.jsonl), where each line represents a dataset entry containing various fields. The main tasks are spoiler type classification (task 1) and spoiler generation (task 2).

For each entry in the training and validation dataset, the following fields are available: "uuid" (unique identifier of the entry), "postText" (the text of the clickbait post to be spoiled), "targetParagraphs" (paragraphs of manually extracted main content from the linked web page used for classification and generation), "targetTitle" (the title of the linked web page used for classification and generation), "targetUrl" (the URL of the linked web page), "humanSpoiler" (human-generated abstractive spoiler for the clickbait post, available in training and validation sets), "spoiler" (human-extracted spoiler for the clickbait post, available in training and validation sets), "spoilerPositions" (position of the human-extracted spoiler), and "tags" (spoiler type to be classified in task 1, available in training and validation sets).

[1]https://github.com/saikonda5468/MSCI-Project

Split	Count	Phrase	Passage	Multi
Train	3200	1367	1274	559
Val	800	335	322	143
Test	400	NA	NA	NA

Table 1: Train/Val/Test Splits

It's important to note that some fields, such as "postId," "postPlatform," "targetDescription," "targetKeywords," and "targetMedia," contain additional meta-information about the entry but are not used for the tasks. The dataset is divided into 3200 posts for training and 400 posts for validation.

The dataset provided for the clickbait challenge contains three distinct types of spoilers: "Phrase" spoilers, comprising one or a few words; "Passage" spoilers, which are complete sentences; and "Multi" spoilers, with diverse forms like relevant segments of article phrases, enumerations, or individual words.The Tabel 1 defines the Train,Validation and Test splits. It is essential to highlight that the distribution of these spoiler types in the training dataset is not even, with 1367 phrase spoilers, 1274 passage spoilers, and 559 multi spoilers. The imbalanced distribution poses a challenge in training models to accurately classify and generate spoilers for different types of clickbait posts.Systems participating in the challenge will be evaluated based on their performance on the Validation set.

3 System Overview

3.1 Task 1: Spoiler Type Classification

3.1.1 Transformer Approaches

In our study, we employed transformer-based models, namely RoBERTa(Liu et al., 2019), DistilBERT(Sanh et al., 2020), and DeBERTa(He et al., 2021), which leverage the powerful transformer architecture for natural language processing tasks. These models were pretrained on a large corpus of text data and then fine-tuned on the clickbait detection task. Through this process, they learned to effectively represent the input text and make accurate predictions for the target labels ('Phrase', 'Passage', 'Multi').

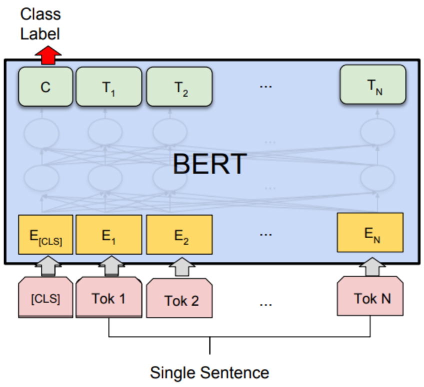

Figure 2: Bert-based Model architecture for Classification Task

The above figure below illustrates the architecture of the BERT-based model used in our clickbait detection task.The BERT(Devlin et al., 2019) model, based on the transformer architecture, forms the backbone of our classification model. It takes tokenized text as input, with special tokens such as [CLS] and [SEP] to mark the start and separation of sequences. BERT's stack of transformer encoder layers processes the input text, capturing contextual information from both the left and right contexts of words

RoBERTa(Robustly Optimized BERT): RoBERTa(Liu et al., 2019) is a variant of BERT(Devlin et al., 2019) (Bidirectional Encoder Representations from Transformers) that was introduced to address some limitations and improve the performance of the original model. Like BERT, RoBERTa is based on the transformer architecture, which is a neural network architecture known for its effectiveness in processing sequential data, such as natural language. RoBERTa is "robustly optimized" because it benefits from large-scale pretraining on a large corpus of diverse data.

During pretraining, RoBERTa is trained on a large corpus of unlabeled text in a masked language modeling task. It tries to predict missing words in a sentence, given the context of the surrounding words. This pretraining process allows RoBERTa to learn powerful contextual representations that capture the meaning of words in different contexts.

In the code, the RoBERTa model is loaded with the 'roberta-base' variant, which is a smaller version of RoBERTa. It is then fine-tuned on the clickbait detection task using a custom training loop with the Trainer class from the Hugging Face Transformers library.

DistilBERT (Distill and Improve BERT): DistilBERT(Sanh et al., 2020) is another variant of BERT(Devlin et al., 2019) designed to be more lightweight and computationally efficient while maintaining good performance. It achieves this by distilling the knowledge from the original BERT model into a smaller model while using a technique called knowledge distillation. The distilled model, DistilBERT, can perform similar tasks as BERT but with fewer parameters, making it more suitable for deployment in resource-constrained environments.

During training, DistilBERT is first pretrained on the masked language modeling task like BERT, and then the knowledge is distilled into a smaller model with fewer layers and parameters.

In the code, the DistilBERT model is loaded with the 'distilbert-base-uncased' variant. It is then fine-tuned on the clickbait detection task using the same custom training loop as used for RoBERTa(Liu et al., 2019).

DeBERTa (Decoding-enhanced BERT): DeBERTa(He et al., 2021) is another transformer-based model, similar to BERT(Devlin et al., 2019) and RoBERTa, with the key difference being the "decoding" enhancements that improve performance. DeBERTa incorporates decoding strategies to process the input text more efficiently and effectively. It uses techniques like two-stream attention and adaptive mask to better capture dependencies between words and generate more accurate representations.

During training, DeBERTa is pretrained on the masked language modeling task similar to BERT and RoBERTa.

Here, the DeBERTa model is loaded with the 'microsoft/deberta-base' variant. It is then fine-tuned on the clickbait detection task using the same custom training loop as used for RoBERTa and DistilBERT.

3.2 Task 2: Spoiler Creation

3.2.1 Transformer Approaches

The models we employed in our study are all based on transformer architectures and have undergone fine-tuning on the Stanford Question Answering Dataset (SQuAD). However, each model has unique characteristics that set them apart and make them suitable for specific tasks:

The bert-base-uncased-squad-v1(bert-base-uncased-squad v1, 2019) model is a variant of BERT(Devlin et al., 2019) that has been trained

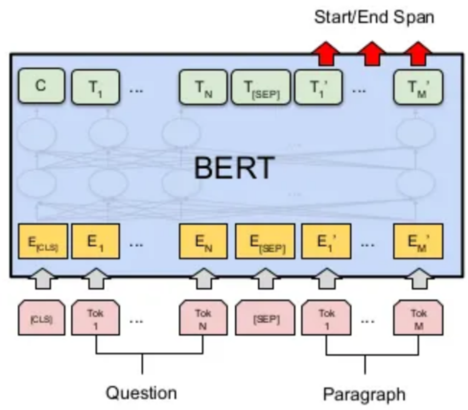

Figure 3: Fine-Tuning for Bert Base Model

with a 'base' configuration, including 12 transformer layers and 110M parameters, on uncased text. The model has been fine-tuned on the SQuAD v1 dataset, where every question corresponds to an answer within the provided text. BERT employs a bidirectional attention mechanism, enabling it to understand the context of a word based on both its preceding and succeeding words. This contextual understanding is crucial for question-answering tasks.

The deberta-v3-base-squad2(deberta-v3-base squad2, 2022) model is a DeBERTa(He et al., 2021) model also with a 'base' configuration, but it has been fine-tuned on the SQuAD v2 dataset. DeBERTa improves upon BERT and RoBERTa by disentangling content and position information in the self-attention mechanism, making it more interpretable and optimizable. Additionally, it introduces an "enhanced mask decoder" for refining the pre-training phase. Unlike SQuAD v1, the SQuAD v2 dataset includes questions that do not have an answer in the given passage, thus increasing the complexity of the task.

The last model, roberta-base-squad2(base squad2, 2022), is a RoBERTa(Liu et al., 2019) model that is fine-tuned on the SQuAD v2 dataset. RoBERTa, another variant of BERT, uses a byte-level Byte-Pair-Encoding (BPE) tokenizer and adjusts key hyperparameters in the model architecture and training process, resulting in improved performance over BERT. Like the deberta-v3-base-squad2 model, roberta-base-squad2 benefits from the SQuAD v2 dataset's challenging questions that may lack answers in the provided passage.

In our spoiler creation task, we utilized the question-answering capabilities of these models. For each post, we treated the question as the desired spoiler, and the models generated answers (spans) from the target article. To find the best answer, we calculated the sum of the start and end logit scores for each potential answer span. The answer span with the highest combined logit score was chosen as the most likely answer to the question in the post. By selecting the answer span with the highest combined logit score, we aimed to find the most confident and relevant response to the given question. This method allowed us to extract meaningful answers directly from the target article, providing accurate spoilers for the question-like posts in our task.

4 Experiment

4.1 Task-1:Spolier Classification

4.1.1 Preprocessing

In the preprocessing step, we organized the clickbait data into a structured format, making it easier to work with. We selected important details like the unique identifier, clickbait post text, title of the linked webpage, main content of the webpage, and spoiler text. To prepare the data for the models, we transformed the spoiler text and identified where it appears in the main content. Then, we converted the clickbait post text and main content into a numerical format, which is what the models understand. We ensured that the data's length met specific requirements and added padding to make processing more efficient. The final output was a set of tokenized inputs with their corresponding positions, ready for training the models.

4.1.2 Model Configurations

In Table 2, we present the hyperparameters used for the classification task in our study. These hyperparameters play a critical role in training the transformer-based models (RoBERTa, DistilBERT, and DeBERTa) for predicting the appropriate spoiler type for clickbait posts.

Table 2: Hyperparameters

Parameter	Value
epochs	5
classes	3
batch_size	RoBERTa and DeBERTa: 8, DistilBERT: 4
learning_rate	1e-5
model	RoBERTa: "roberta-base" DistilBERT: "distilbert-base-uncased" DeBERTa: "microsoft/deberta-base"

We conducted training for a total of 5 epochs, where each epoch represents one pass through the entire dataset during model training. The classification task involved predicting three classes corresponding to different spoiler types: "Phrase," "Passage," and "Multi."

To efficiently train the models, we used different batch sizes for each transformer model. For RoBERTa and DeBERTa, we used a batch size of 8, while for DistilBERT, a smaller batch size of 4 was employed. The batch size affects memory usage and training speed, and these specific sizes were chosen based on the models' characteristics.

For optimization during training, we set the learning rate to 1e-5 (0.00001). The learning rate controls the step size at which the model's parameters are adjusted based on the computed gradients during backpropagation, influencing the convergence speed and stability of the training process.

We fine-tuned the transformer models using the Hugging Face Transformers library, specifying the architecture for each model. RoBERTa was used with the "roberta-base" variant, DistilBERT with "distilbert-base-uncased," and DeBERTa with "microsoft/deberta-base." These models were selected due to their unique architectural characteristics and pre-training approaches, which made them well-suited for the clickbait classification task.

By configuring these hyperparameters, we effectively trained the transformer-based models for clickbait classification, resulting in accurate predictions of the spoiler types ("Phrase," "Passage," or "Multi"), contributing to the delivery of more informative content and enhancing user experience with clickbait posts.

4.1.3 Results

When comparing the three pre-trained models from Hugging Face's Transformers library for clickbait detection, we focused on evaluating their performance using the F1-score metric. The F1-score is a harmonic mean of precision and recall, providing a balanced measure of a model's effectiveness in correctly classifying clickbait posts.

After conducting preprocessing and training the models, we observed distinct patterns in their F1-score performance. The 'Deberta' model exhibited the highest F1-score among the three models, showcasing its superior ability to detect clickbait content, with a value of 75.5

On the other hand, the 'Distilbert' model demonstrated comparatively lower F1-score performance, with a value of 65.5

The 'Roberta' model showcased a balanced F1-score performance, with a value of approximately 73

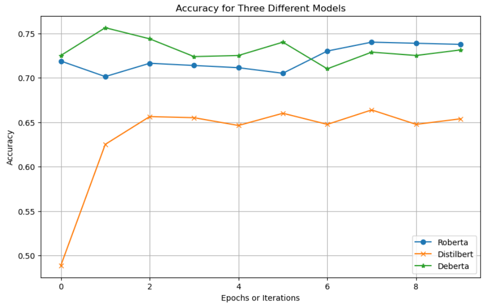

Figure 4: Accuracy plot of Roberta,Distilbert and Deberta Models for Classification Task

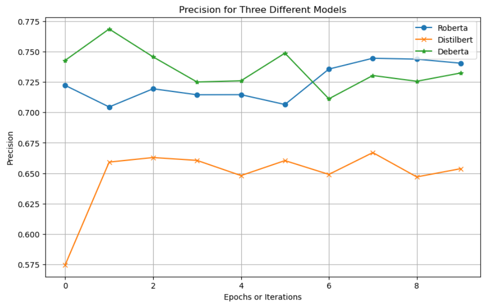

Figure 5: F1 score plot of Roberta,Distilbert and Deberta Models for Classification Task

Figure 4 illustrates the accuracy of all three models, RoBERTa, DistilBERT, and DeBERTa, used for clickbait detection. It clearly shows that the DeBERTa model achieved the highest accuracy, followed closely by the RoBERTa model, and finally the DistilBERT model with the lowest accuracy.

Figure 5 depicts the F1-score of the three models for clickbait detection. Similar to the accuracy results, the DeBERTa model again outperforms the other models in terms of F1-score, achieving the highest score. The RoBERTa model shows a balanced F1-score, while the DistilBERT model exhibits the lowest F1-score.

These graphs provide visual representations of the performance comparison among the models, showcasing the effectiveness of each model for clickbait detection. The DeBERTa model consistently performs the best, demonstrating its superiority in accurately classifying clickbait posts based on the F1-score metric. These insights help in selecting the most suitable model for clickbait detection, considering both accuracy and F1-score for robust and reliable performance.

Model	Huggingface Model Identifier	Accuracy	Recall	F1-Score
Roberta	roberta-base	73.1	73.1	73.0
Distilbert	distilbert-base-uncased	65.6	65.6	65.5
Deberta	**deberta-base**	**75.6**	**75.6**	**75.5**

Table 3: Evaluation Metrics of Classification Task

4.2 Task-2:Spolier Generation

4.2.1 Preprocessing

In the preprocessing step, we perform necessary transformations on the dataset to prepare it for a Question Answering (QA) task. The first part involves selecting important information from the original data, such as the unique identifier, the clickbait post text, the title of the linked web page, the main content of the webpage, and the spoiler text. We then combine these details to create a new column that pairs the spoiler text with its corresponding position within the main content.

In the second part, we further process the training examples to prepare them for the QA model. We convert the clickbait post text and the main content into a numerical format that the model can understand. We ensure that the tokenization process handles any potential overflow and returns the relevant tokens along with their positions. Using the spoilers, we identify the starting and ending positions of the answer span within the tokenized main content. If the spoiler is completely contained within the main content, we find the corresponding positions of the answer span. However, if the spoiler extends beyond the main content, we mark it as invalid for that context. The final output is a set of tokenized inputs, along with the start and end positions, which are essential for training the QA model effectively.

During the data preprocessing phase for the validation examples, we perform necessary transformations to make the clickbait post texts and their corresponding main content compatible with the Question Answering (QA) model. We apply a tokenizer to convert the text into a numerical representation suitable for the model, ensuring that the input length meets specific constraints and is adequately padded for efficient processing. To maintain the connection between the original examples and the tokenized inputs, we create a mapping that links them together using unique example IDs. Additionally, we accurately identify token positions within the main content and exclude non-content tokens, ensuring that the model generates relevant answers based on the original context. These processed inputs, along with their corresponding offset mappings and example IDs, help the QA model produce appropriate answers for the validation examples.I utilised code from HuggingFace tutorial(Hugging Face, 2022) after converting spoiler type data into SQUAD 2.0 format.

4.2.2 Model Configurations

Table 4: Hyperparameters

Parameter	Value
epochs	10
max_length	512
stride	128
n_best	25
max_answer_length	200
learning_rate	1e-5
batch_size	RoBERTa and BERT: 8, DeBERTa: 4
model	RoBERTa: "roberta-base" BERT: "distilbert-base-uncased" DeBERTa: "microsoft/deberta-base"

The clickbait Generation models were fine-tuned using three transformer models: RoBERTa, Bert, and DeBERTa. Each model was trained for 10 epochs, which means the entire dataset was passed through the models ten times during training. The maximum length of the input text sequence allowed during tokenization was set to 512 tokens, and a stride of 128 tokens was used to generate overlapping segments of the text. During evaluation, the models considered the top 25 candidate answers. The maximum allowed length for predicted answers during evaluation was set to 200 tokens to avoid excessively long or irrelevant responses.

For optimization during training, a learning rate of 1e-5 (0.00001) was used, which controls how much the model's parameters are adjusted based on the computed gradients. The batch size used during training varied depending on the model. For RoBERTa and DistilBERT, a batch size of 8 was used, while for DeBERTa, a smaller batch size of 4 was employed. Batch size affects memory usage and training speed, so different sizes were chosen for these models.

The specific transformer models used were

RoBERTa with the "roberta-base" architecture, DistilBERT with "distilbert-base-uncased," and DeBERTa with "microsoft/deberta-base." These models have different architectural characteristics and pre-training approaches, making them suitable candidates for the clickbait Generation task.I utilised code from HuggingFace tutorial(Hugging Face, 2022) after prediction of start and end logistics using the model and converting them to the readable format.

4.2.3 Post-Processing

The QA models predicts the probability of a token to be start and end index, so as output we get start and end logit probabilities for each token in answer content.So,We conducted post-processing to evaluate the performance of a Question-Answering (QA) model designed to answer questions based on a given context. The processing step involves using the model's predictions for the start and end positions of potential answers, along with information about the examples and ground truth answers. We systematically examine each example, leveraging the model's predictions to determine the most probable start and end positions for answers. We ensure that the predicted answers are valid by verifying they fit within the context and meet a specified maximum length criterion. To select the best answer for each example, we combine the start and end probabilities and choose the highest-scoring answer. In cases where the model cannot find a valid answer, we leave the prediction empty.

The primary goal of this processing is to assess the QA model's effectiveness by comparing its answers to the correct ones. Through this evaluation, we refine the model's predictions and comprehensively gauge its accuracy and linguistic quality in answering questions within diverse contexts.

4.2.4 Results

The table 5 presents evaluation results of three Hugging Face models on a validation set of 800 clickbait datasets using METEOR, BLEU-4, and Exact Match (EM) metrics. The Roberta model (roberta-base-squad2) achieves the highest scores in all metrics, with METEOR 45.8, BLEU-4 22.9, and Exact Match 35.8, showcasing its superior performance in accurately answering questions.The Deberta model (deberta-v3-base-squad2) also performs well with scores of METEOR 43.05, BLEU-4 18.5, and Exact Match 30.0, outperforming the Bert model (bert-base-uncased-squad-v1) with scores of METEOR 34.30, BLEU-4 15.5, and Exact Match 25.2. These results indicate that the Roberta model is the most suitable choice for precise question-answering tasks on clickbait datasets.

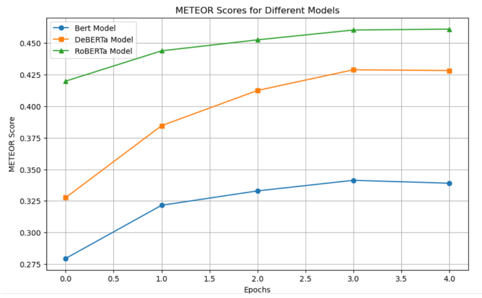

Figure 6: Meteor score of Roberta,Deberta and Bert Model

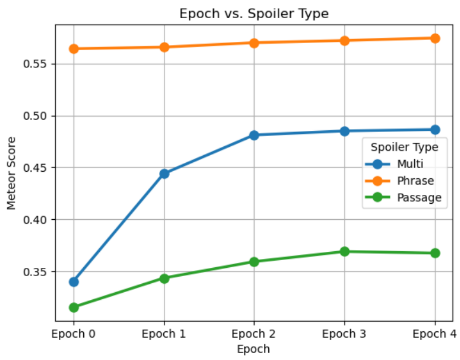

Figure 7: Meteor score of different Spoiler types using Roberta Model

In the study, we aimed to evaluate the performance of the RoBERTa(base squad2, 2022) model, which emerged as the top-performing model for clickbait detection. To better understand its effectiveness in predicting different spoiler types, we conducted data splitting based on spoiler types and individually trained the datasets. This approach enabled us to capture the nuances and context-specific features of different spoiler types, resulting in more precise and informative results. From Table 6 the obtained outcomes provide valuable insights into RoBERTa's performance across different spoiler types. For Multi-type spoilers, RoBERTa achieved an impressive Meteor Bleu-4 score of 48.6, highlighting its proficiency in handling complex and multifaceted textual information. Additionally, for Phrase-type spoilers, the model showcased remarkable capabilities, achieving an outstanding Meteor Bleu-4 score of 57.4, demonstrating its ability to accurately predict phrase-based spoilers. Although the Passage-type spoilers presented some

Model	Huggingface Model Identifier	Meteor	Bleu-4	Exact Match
Bert	(bert-base-uncased-squad v1, 2019)	34.30	15.5	25.2
Deberta	(deberta-v3-base squad2, 2022)	43.05	18.5	30.0
Roberta	(base squad2, 2022)	**45.8**	**22.9**	**35.8**

Table 5: Evaluation Metrics of Different SQUAD Models

Spoiler Type	Meteor	Bleu-4	Exact Match
Multi	48.6	34.0	51.7
Phrase	**57.4**	**34.2**	**57.3**
Passage	36.7	22.9	14.2

Table 6: Evaluation Metrics of Differnet Spoiler Types

challenges, the model still managed to deliver a competitive Meteor Bleu-4 score of 36.7. Overall, these findings underscore the effectiveness of RoBERTa in handling spoiler-type prediction tasks and emphasize the significance of specialized training on specific data subsets to optimize model performance.

5 Conclusion

This paper presents a comprehensive analysis of transformer-based models for addressing clickbait challenges on social media. Our primary goal is to generate concise spoilers that close the curiosity gap induced by clickbait posts and offer users more informative content. We explore three models, BERT, DeBERTa, and RoBERTa, for spoiler type classification. Deberta emerges as the most effective in detecting clickbait content, outperforming others in accuracy, recall, and F1-score.

Additionally, using BERT-base, Deberta-v3-base-squad2, and RoBERTa-base-SQuAD2 models, we conduct SQuAD analysis to generate spoilers for clickbait posts. Results show RoBERTa's excellence in producing accurate and contextually relevant spoilers, outperforming other models in metrics like METEOR, BLEU-4, and Exact Match. Moreover, specialized training on specific data subsets demonstrates RoBERTa's effectiveness in predicting different spoiler types, catering to various clickbait scenarios.

These findings hold significant implications for clickbait research and the development of user-friendly social media systems. Leveraging advanced transformer models enhances content authenticity and user experience, providing more valuable and engaging content to users while browsing social media feeds.

References

Roberta base squad2. 2022. Question answering model hugging face tutorial. *Hugging Face*.

bert-base-uncased-squad v1. 2019. Question answering model hugging face tutorial. *Hugging Face*.

deberta-v3-base squad2. 2022. Question answering model hugging face tutorial. *Hugging Face*.

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

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

Hugging Face. 2022. Question answering hugging face tutorial. *Hugging Face*.

Jüri Keller, Nicolas Rehbach, and Ibrahim Zafar. 2023. nancy-hicks-gribble at SemEval-2023 task 5: Classifying and generating clickbait spoilers with RoBERTa. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*.

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.

Ronghao Pan, José Antonio García-Díaz, Francisco García-Sánchez, and Rafael Valencia-García. 2023. Chick adams at SemEval-2023 task 5: Using RoBERTa and DeBERTa to extract post and document-based features for clickbait spoiling. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.