Spoiler Type Classification and Spoiler Generation for Clickbait Posts

Maria Saeed*
i200836@nu.edu.pk
FAST NUCES, Islamabad
Islambad, Pakistan

Manal Zehra*
i200828@nu.edu.pk
FAST NUCES, Islamabad
Islambad, Pakistan

Zuha Umar*
i200603@nu.edu.pk
FAST NUCES, Islamabad
Islambad, Pakistan

Abstract

In the digital age inundated with information, spoilers in clickbait posts disrupt narrative immersion. This paper presents a system utilizing natural language processing to identify and neutralize spoilers. Through a multi-model approach, our solution not only classifies spoiler types but also locates them precisely, offering a concise spoiler essence. Unlike traditional spoiler avoidance, our system empowers users to navigate narratives confidently. The paper explores transformer-based models, transitioning from non-transformer to hybrid approaches with LSTM, BiLSTM, and BERT, achieving a notable 62.0% accuracy in spoiler classification. Additionally, a T5 transformer model is introduced for spoiler generation. The results underscore the effectiveness of our system in addressing spoiler challenges, contributing to the evolution of spoiler classification and generation. This research advocates for active participation in narrative experiences, transforming frustration into informed engagement.

Keywords: natural language processing, spoilers, clickbait, transformer models, LSTM, BiLSTM, BERT, T5, spoiler classification, spoiler generation.

ACM Reference Format:
Maria Saeed, Manal Zehra, and Zuha Umar. 2023. Spoiler Type Classification and Spoiler Generation for Clickbait Posts. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In our information-saturated age, spoilers lurk like digital landmines, their cryptic headlines and misleading promises detonating narrative immersion. This paper tackles this frustration head-on, presenting a system that leverages natural language processing to disarm these literary booby traps. Through a rigorous exploration of diverse models and innovative combinations, we've developed a solution that not only identifies spoiler types within articles but also pinpoints their exact location, delivering the essence in a single, concise line. This isn't about spoiler avoidance, but about empowerment. Imagine confidently navigating the narrative landscape, selecting the spoilers you want to experience on your own terms. This research delves into the technical underpinnings of our multi-model approach, unveiling the mechanisms behind spoiler classification and generation. Join us as we chip away at the tyranny of clickbait and redefine the relationship between reader and story, transforming frustration into informed engagement. Let's not just consume narratives, but actively participate in their unfolding.

2 Literature Review

Here's the literature review we conducted before opting for a methodology and conducting the experiments.

2.1 Matt Bai at SemEval-2023 Task 5: Clickbait spoiler classification via BERT

In the study conducted by Tailor et al. (2023), the authors addressed the challenge of identifying and spoiling clickbait posts, focusing on the classification of spoiler types. The research utilized a dataset consisting of 5,000 clickbait posts with spoilers. The dataset included key information such as unique post IDs, platform sources, post texts, manually extracted titles, main content divided by paragraphs, optimized spoilers, spoiler positions, and spoiler types. The authors employed a BERT-base model, pre-trained on English using a masked language modeling (MLM) objective, and fine-tuned it specifically for the task of spoiler type classification. The dataset was divided into training, validation, and testing sets, with the model achieving a balanced accuracy of 0.63 and F1-scores of 0.69, 0.65, and 0.61 for classifying spoilers into phrases, passages, and multipart categories, respectively. Notably, the authors compared their results with non-transformer-based methods such as Naive Bayes, SVM, logistic regression, and transformer-based models like DeBERTa and RoBERTa. The findings indicated that

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

transformer-based models outperformed non-transformer-based ones, highlighting the significance of language understanding in the classification task. However, the authors acknowledged that their approach did not surpass the scores achieved by other papers who have used roBERTa attributing this difference to the exclusion of postTitles in their experimental setup. They emphasised that while the post content provided more details than the title, the title played a crucial role in effectively classifying the type of spoiler to be generated. The paper suggested that the inclusion of post titles in the model input could enhance performance, as titles could provide valuable contextual information for more accurate spoiler classification.

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

The work done by Keller et al. (2023) investigates clickbait spoiling and spoiler type classification using transformer-based text classification. Their study compares transformer models, specifically RoBERTa, with conventional classifiers. The authors propose an innovative question-answering approach for spoiler generation. Results indicate superior performance of transformer models in classification, achieving a 62% accuracy rate. The dataset includes social media posts with unevenly distributed spoiler types—phrase, passage, and multi. RoBERTa-based models outperform shallow-learned classifiers, such as XGBoost. Spoiler creation, adopting a question-answering strategy with RoBERTa, yields satisfactory results. Future directions include improving spoiler generation models, exploring diverse datasets, and investigating interpretability of transformer-based models in clickbait analysis.

2.3 Jack-flood at SemEval-2023 Task 5: Hierarchical Encoding and Reciprocal Rank Fusion-Based System for Spoiler Classification and Generation.

In the study conducted by Kumar et al. (2023), the importance of a technique known as hierarchical encoding for spoiler classification (phrase, multi, passage) was highlighted.This method involves breaking down information in a structured manner, somewhat like organizing it into a tree-like structure for better comprehension. The authors specifically advocate for the use of advanced models like BERT, RoBERTa, and DeBERTa in this process. These transformer models stand out for their exceptional ability to understand context, especially in longer pieces of text, by considering the intricate relationships between words. To further capture the sequential dependencies among sentences, Bidirectional Long Short-Term Memory (BiLSTM) was applied. Their study places emphasis on considering document length and word count overlap features, recognizing the significant impact of word count in spoiler-type classification. In essence, the use of these transformer models is favored over non-transformer models due to their prowess in capturing contextual nuances, making them akin to adept detectives for understanding the intricacies of textual information. For task 2, the authors adopt an information retrieval approach, initially utilizing the BM25 model for passage spoiler retrieval. Acknowledging the limitations of BM25, they enhance the retrieval process by incorporating pretrained language models (S-BERT, DeBERTa) for semantic similarity estimation. Additionally, the MonoT5 model is considered for document ranking in information retrieval tasks. The integration of Reciprocal Rank Fusion (RRF) further refines the passage spoiler retrieval, combining rankings obtained from different information retrieval systems.

3 Methodology

Now, let's talk about the methodologies we used in the two tasks.

3.1 Task 01: Spoiler Classification

Let's start off by talking about the different approaches we used for the spoiler classification task.

3.1.1 Exploration of Non-Transformer Models. Initially, our approach involved experimenting with non-transformer models for spoiler type classification. The models we used were SVM, Logistic Regression and Random Forest. However, the results were suboptimal, indicating the need for more sophisticated models to capture the nuanced features of clickbait posts.We chose this to verify claims made in the study by Tailor et al. (2023)about the fact that non-transformer based libraries do not perform well in this scenario and they were proven right. Our accuracy in all of them were below average. We realised that this was not the best approach so we decided to shift the approach to a transformers based model.

3.1.2 Transition to Transformer Models. To enhance the model's performance, we transitioned to transformer-based architectures, specifically BERT and RoBERTa models. The reason we chose these models was that consistently throughout the papers we saw how it outperformed most of the models and that is what we wanted to test. We tried the BERT model which was giving accuracy of around 42%. Still not satisfied with it we tried using a different approach using roBERTa but that just led to a lower accuracy. Not giving up we decided to use another approach: we decided that we can add titles in the feature extraction process as suggested by Tailor et al. (2023) . This had no real significance in the accuracy which was very disappointing as two research papers we read suggested this as a way of improvement. These models demonstrated improved capabilities in spoiler type classification, suggesting their effectiveness in capturing intricate contextual relationships within the input data.

3.1.3 Incorporating Sequential Models: LSTM and BiLSTM.

At that time we had been studying the LSTM model in class and we also had a consultation from a Deep Learning student. Both our instructor and the student emphasised the importance of LSTM and how it captures contextual information. Therefore, we extended our exploration to include Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) models. We were stuck at this point as our goal was to have a 60% or more accuracy. Yet we were unable to reach it.

3.1.4 Using Dual Model Approach.

A significant breakthrough came when we came across the idea of using hybrid models i.e. we use one model for feature extraction and another for training and testing for classification. Here we realised if we used BERT for training and testing as that is what got most research papers a good result but for feature extraction we can use LSTM as it has a good way of getting contextual information and this was our Holy Grail. We achieved our goal in terms of accuracy. Figure 1 summarizes our dual model approach.

3.2 Task 02: Spoiler Generation

In Task 2, dedicated to generating spoilers from clickbait posts and linked documents, a T5 transformer model was employed, leveraging its proficiency in conditional text generation. The approach involved treating the clickbait post as a question and utilizing the T5 model to extract pertinent information from the linked document, thereby generating tailored spoilers. The methodology comprised several key steps. Firstly, the T5 model and tokenizer were configured for conditional generation, capitalizing on the model's pretraining on tasks of similar nature. Subsequently, a dedicated function was established to harness the T5 model's capabilities for generating spoilers based on a constructed prompt. This prompt, formulated by framing the clickbait post as a question and incorporating relevant details from the linked document, served as input for spoiler generation. Finally, a loop was implemented to traverse through each entry in the training data. For each entry, the spoiler generation function was invoked, producing spoilers for both clickbait posts and linked documents. The results, inclusive of UUIDs and corresponding spoilers, were systematically stored for further analysis and evaluation.

4 Experiments

4.1 Task 01: Spoiler Classification

4.1.1 Non-Transformer Models.

Initially, our approach involved experimenting with non-transformer models for spoiler type classification. We started off by using the Support Vector Machine (SVM) model for this task. The post text and target title were treated as the features. Furthermore, each tag, including multi, phrase, and passage, was treated as a separate classification task. This model utilized TF-IDF features extracted from the concatenated post text and target title. Despite implementing this sophisticated approach, the SVM model achieved an accuracy of 33.0%, indicating the need for further enhancements.

We then switched to logistic regression by taking post text and target title as the features. We achieved an accuracy of 35.5% which showed room for further improvements yet again.

To further enhance the spoiler classification task, a transition to a more complex model, Random Forest, was made. The Random Forest model was trained using the same TF-IDF features from the post text and target title, with each tag being treated as a separate output in the MultiOutputClassifier. This refinement yielded a slight improvement in accuracy, reaching 37.3%.

4.1.2 LSTM and BiLSTM.

In our quest for an effective spoiler classification model, we leveraged the capabilities of Long Short-Term Memory (LSTM) networks using TensorFlow and Keras. The process began with preparing our textual data through tokenization and padding, crucial steps to ensure the data was ready for the LSTM model. Our model, constructed with Keras's Sequential API, included an Embedding layer, an LSTM layer with 100 neurons, and a Dense layer using the softmax activation function—ideal for multiclass classification tasks. The model was trained with categorical cross entropy as the loss function and the Adam optimizer.

This LSTM-based architecture significantly contributed to our spoiler classification pipeline, helping the model learn intricate contextual patterns within clickbait posts. The outcome was a notable accuracy of 53.9%, showcasing the effectiveness of the LSTM model in distinguishing spoilers based on provided tags. Furthermore, exploring Bidirectional LSTM (BiLSTM) improved our results even further, yielding an enhanced accuracy of 55.6%. This achievement underscored the significance of considering both past and future context in understanding the subtleties of spoiler content within posts.

Model Architecture	Components	Training Parameters
LSTM-based Spoiler Classification	Tokenization, Padding, Sequential API, Embedding, LSTM, Dense	Categorical Cross Entropy Loss, Adam Optimizer
Bidirectional LSTM (BiLSTM) Enhanced	Addition of Bidirectional LSTM Layer	Categorical Cross Entropy Loss, Adam Optimizer

Table 1. Summary of Model Architectures, Components, and Training Parameters

Performance Metrics:

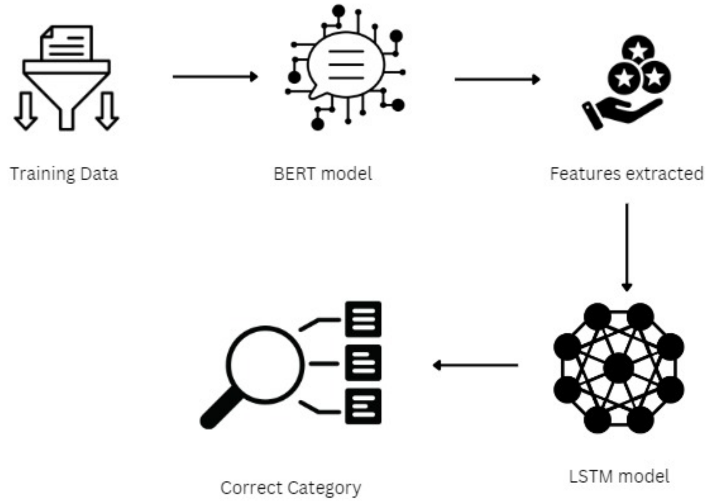

Figure 1. Workflow of Dual Model Approach

- **LSTM-based Spoiler Classification:** Accuracy: 53.9
- **Bidirectional LSTM (BiLSTM) Enhanced:** Enhanced Accuracy: 55.6

4.1.3 Hybrid Models.

In our pursuit of an advanced spoiler classification model, we embraced a hybrid approach by combining the power of BERT (Bidirectional Encoder Representations from Transformers) and LSTM (Long Short-Term Memory) networks. For the fusion of BERT and LSTM, we first employed the BERT tokenizer and model, transforming our text data into tokenized sequences with attention masks. These representations were then fed into an LSTM network. This fusion allowed us to benefit from both BERT's contextual understanding and LSTM's ability to capture sequential dependencies. The BERT model provided contextual embeddings for the input sequences, capturing intricate relationships between words. Subsequently, these embeddings were processed by the LSTM network, enhancing the model's capacity to understand and remember sequential patterns within the text data. The BERT model tokenized and encoded the training and validation data, extracting features that were then utilized for classification. The LSTM model, with a specific architecture tailored for the task, was trained on these features. The model was fine-tuned using a combination of cross-entropy loss and the Adam optimizer to achieve accurate spoiler type classification. This hybrid architecture not only leveraged BERT's contextual awareness but also incorporated the sequential memory of LSTM, resulting in a comprehensive understanding of the clickbait post data. By using this approach, we were able to achieve a total of 62.0% accuracy which showed significant improvement in comparison with the results of other models.

Model	Accuracy (%)
Logistic Regression	35.5
Support Vector Machine	33.0
Random Forest	37.3
BERT	42.0
LSTM	53.9
BiLSTM	55.6
BERT + LSTM	62.0

Table 2. Model Accuracy Comparison

4.2 Task 02: Spoiler Generation

For Task 2, dedicated to generating spoilers based on clickbait posts and linked documents, a T5 transformer model, renowned for its prowess in conditional text generation, was employed. The T5 model treated the clickbait post as a question and harnessed its ability to extract pertinent information from the linked document, ultimately generating spoilers tailored to the provided input. In order to generate spoilers we followed the following steps:

1. **T5 Model and Tokenizer Setup:** The T5 model, configured for conditional generation, was set up along with its corresponding tokenizer. The T5 model has been pre-trained on conditional generation tasks, aligning perfectly with the spoiler generation task at hand.
2. **Spoiler Generation Function:** A dedicated function, `generate_spoiler`, was defined to utilize the T5 model

for generating spoilers based on a provided prompt. This prompt is constructed by framing the clickbait post as a question and incorporating relevant information from the linked document.

3. **Spoiler Generation Loop:** A loop was implemented to traverse through each entry in the training data. For every entry, the `generate_spoiler` function was invoked, generating spoilers for clickbait posts and linked documents. The results, including UUIDs and the corresponding spoilers, were stored.

5 Conclusion

In conclusion, our research offers a robust solution to the pervasive issue of spoilers in clickbait content through a sophisticated system employing natural language processing and a multi-model approach. The transition from non-transformer to hybrid models, incorporating LSTM, BiLSTM, and BERT, resulted in a substantial 62.0% accuracy in spoiler classification. The introduction of a T5 transformer model for spoiler generation further enhances our system's capabilities. These findings contribute to the evolving landscape of spoiler classification and generation, emphasizing the importance of user empowerment in navigating digital narratives. Our work encourages a shift from passive consumption to active engagement, marking a significant step in transforming the frustration associated with spoilers into informed and empowered narrative experiences.

References

Sujit Kumar, Aditya Sinha, Soumyadeep Jana, Rahul Mishra, and Sanasam Ranbir Singh. 2023. "Jack-flood at SemEval-2023 Task 5: Hierarchical Encoding and Reciprocal Rank Fusion-Based System for Spoiler Classification and Generation." In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori (Eds.). Association for Computational Linguistics, Toronto, Canada, Jul 2023, 1906-1915. DOI: https://doi.org/10.18653/v1/2023.semeval-1.262

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 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori (Eds.). Association for Computational Linguistics, Toronto, Canada, Jul 2023, 1712-1717. DOI: https://doi.org/10.18653/v1/2023.semeval-1.238

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

Tailor, N., & Mamidi, R. (2023). *Matt Bai at SemEval-2023 Task 5: Clickbait spoiler classification via BERT. International Workshop on Semantic Evaluation.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: pre-training of deep bidirectional transformers for language understanding.* In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Maik Fröbe, Tim Gollub, Matthias Hagen, and Martin Potthast. 2023a. *SemEval-2023 Task 5: Clickbait Spoiling.* In *17th International Workshop on Semantic Evaluation (SemEval-2023).*

Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023b. *Continuous Integration for Reproducible Shared Tasks with TIRA.io.* In *Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023)*, Lecture Notes in Computer Science, Berlin Heidelberg New York. Springer.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. *Clickbait Spoiling via Question Answering and Passage Retrieval.* In *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 7025–7036. Association for Computational Linguistics.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.*

Jing Zhu. 2002. *Bleu: a method for automatic evaluation of machine translation.* In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, July 6-12, 2002, Philadelphia, PA, USA, pages 311–318. ACL.

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

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. *Transformers: State-of-the-art natural language processing.* In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.