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

Sujit Kumar, Aditya Sinha*, Soumyadeep Jana*, Rahul Mishra and Sanasam Ranbir Singh
Indian Institute of Technology Guwahati
University of Geneva
{sujitkumar, aditya2001, sjana, ranbir}@iitg.ac.in, rahul.mishra@unige.ch

Abstract

The rise of social media has exponentially witnessed the use of clickbait posts that grab users' attention. This paper presents our approach, which use different encoding techniques. We propose hierarchical encoding with count and document length feature-based model for spoiler type classification, which uses Recurrence over Pretrained Encoding. We also propose combining multiple ranking with reciprocal rank fusion for passage spoiler retrieval and question-answering approach for phrase spoiler retrieval. For multipart spoiler retrieval, we combine the above two spoiler retrieval methods. Based on the benchmark, the experimental results indicate that our proposed spoiler retrieval methods are highly effective in retrieving spoilers that are semantically very close to the ground truth spoilers.

1 Introduction

The rapid adoption of social media among the masses has pushed online publishers to stay relevant by making their content prominent on these platforms (Teixeira, 2014). This is mostly achieved through clickbait. This paper presents our proposed approach and detailed analysis for two subtasks of the clickbait spoiling challenge (Task 5) (Fröbe et al., 2023) Subtask 1: clickbait spoiler type classification, b) Subtask 2: clickbait spoiler generation. This contest is organized in the 17th International Workshop on Semantic Evaluation (SemEval 2023). This challenge addresses the issue of generating spoilers for clickbait. The importance of this task can be gauged from the fact that spoilers would preemptively warn the readers about the content of the clickbait posts, enabling the users to make an informed decision. This would also mitigate the spread of misinformation (Silverman, 2015) and the over-dependence of content publishers on sensational posts for prominence.

Our proposed methods for spoiler-type classification and spoiler generation rely on strength of pretrained language models such as BERT (Devlin et al., 2018), and its different variants like DeBERTa (He et al., 2021). We propose a hierarchical encoding-based model for spoiler-type classifications (subtask-1), which encodes different aspects of hierarchical information presented in the target paragraph along with document length and count overlap-based features. Considering the unique challenges and characteristics posed by passage, phrase and multipart spoiler generation, we adopt three different approaches for spoiler generation. We combine several ranking models using reciprocal rank fusion (RRF) for passage retrieval, a pretrained RoBERTa model over question answering task for phrase retrieval and combine methods of passage retrieval and phrase retrieval for multipart spoiler retrieval. Our code repository is available at[1] https://github.com/thesujitkumar/jack-flood-at-SemEval-2023-Task-5_Spoiler-Classification-and-Generation to recreate and reproduce the results presented in this paper.

2 Background

Initial studies (Blom and Hansen, 2015; Schaffer, 1995; Rony et al., 2017; Anand et al., 2017; Biyani et al., 2016; Agrawal, 2016; Chakraborty et al., 2016) in the literature focus on the detection of clickbait, which deals with identifying whether a post is clickbait or not. But such detection of clickbait does not impede the curiosity induced by clickbait. To fill this gap and overcome such limitations, the recent study(Hagen et al., 2022) curated a dataset for spoiler generation and proposed information retrieval and question-answering methods for spoiler generation, which helps to impede the curiosity induced by clickbait. Task 5 of SemEval 2023, Clickbait Spoiling (Hagen et al., 2022) aims

* Equal contributions

[1] Jack-flood code repository

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1906–1915

to generate spoilers for clickbait posts. Generating effective spoilers poses unique challenges, such as understanding the context of the title and its relevance with the post body. We propose novel approaches to solving this problem by harnessing the power of pretrained language models and the Reciprocal Rank Fusion technique to generate phrase, passage, and multipart-type spoilers.

3 Methodology

This section presents our proposed method for subtask 1: spoiler type classification, and subtask 2: spoiler generation.

3.1 Subtask 1: Spoiler Type Classification:

Given a pair of post text $\mathcal{C}$ and target paragraphs $\mathcal{P}$, the task is to identify the spoiler type $\mathcal{Y}$ where $\mathcal{Y} \in \{\text{phrase, passage, multipart}\}$.

Inspired by the superiority of hierarchical encoding in literature for large document classification (Yoon et al., 2019; Sun et al., 2021), we exploit the hierarchical structure of target paragraphs $\mathcal{P}$ by following the hierarchical structure of the target paragraph while encoding. We propose Recurrence over the pretrained Encoding RPE model. RPE captures the following key properties of hierarchical encoding (i) encoding of a sentence using a pretrained language model which includes BERT[2] (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021), which captures long-term dependencies between words within a sentence through multi-head attention component in the encoder of above-mentioned pretrained language models. (ii) target paragraph $\mathcal{P}$ consists of a sequence of sentences in which every sentence is contextually related to the previous and next sentence and conveys the sequential information present within the target paragraphs $\mathcal{P}$. To capture the sequential dependencies among the sentences of the target paragraphs $\mathcal{P}$. We apply $BILSTM$ (Hochreiter and Schmidhuber, 1997) over encodings of sentences, which effectively capture the context of the sentence from left and right. By encoding the sequence of sentences in $\mathcal{P}$ from left to right, RPE captures the context of sentences on the previous sentence, while encoding from right to left, RPE captures the context of a sentence from the next sentence. (iii) The task of spoiler-type classification is heavily influenced by the word count in the spoiler; for example, if a pair of post text $\mathcal{C}$ and target paragraph $\mathcal{P}$ is classified in passage type spoiler then the spoiler is going to be few sentences in length, if it is classified in a phrase type spoiler then spoiler will be few words and in case of multipart it is going to list of sentences from multiple segments of the document. Considering such properties of the spoiler type classification, RPE also considers document length and word count overlap features. Figure 1 presents a block diagram of *Recurrence over a pretrained Encoding* RPE model. Given a pair of post text $\mathcal{C}$ and target paragraphs, $\mathcal{P}$ the RPE model first splits the $\mathcal{P}$ into set sentences $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_n\}$. Next, encoded representation of post text $\mathcal{C}$ and sentences $\mathcal{S}_i$ are obtained using a pretrained language model as follows:

$$\mathbf{s}_i = \Big(\mathbf{PLM}\Big(\mathcal{S}_i\Big)\Big) \tag{1}$$

$$\mathbf{c} = \Big(\mathbf{PLM}\Big(\mathcal{C}\Big)\Big) \tag{2}$$

Ideally, any pretrained language model can be used to encode $\mathcal{S}_i$ and $\mathcal{C}$ but in this study, we use BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021). Considering the limited size of the available training dataset, we did not finetune BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021) for spoiler-type classification. Next, we apply BiLSTM (Hochreiter and Schmidhuber, 1997) over the encoded representation of sentences to obtain the encoded representation $\mathbf{p}$ of target paragraphs $\mathcal{P}$ using equation 3. The main motivation behind applying BiLSTM over the encoded representation of sentences is to capture the sequential dependencies among the sentences of $\mathcal{P}$. Here, sequential dependencies essentially mean that every sentence in a target paragraph is contextually related to its previous and next sentence.

$$\mathbf{p} = \mathbf{BiLSTM}\Big(\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n\Big) \tag{3}$$

Given a pair of post text $\mathcal{C}$ and target paragraphs $\mathcal{P}$, document length and count overlap features are estimated as follows: (i) **Document length features :** We extract the count of unigrams, the number of sentences, punctuation marks, and paragraphs present in $\mathcal{P}$. We also extract the same features for post text $\mathcal{C}$. We form a feature vector $\mathbf{d}$ by concatenating the document length feature as discussed above. (ii) **Count overlap features :** To extract the count overlap feature vector $\mathbf{o}$, we first

[2]BERT

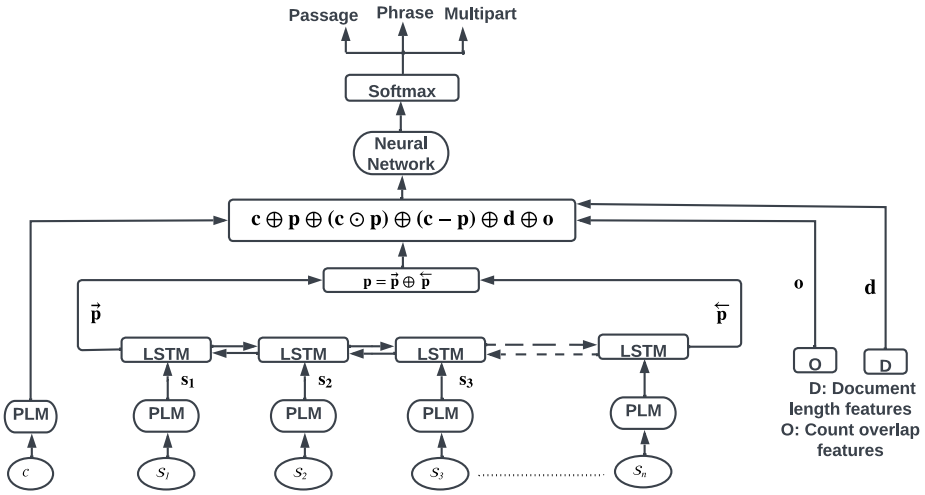

Figure 1: present the working diagram of the proposed model RPE for spoiler-type classification.

apply preprocessing over $\mathcal{C}$ and $\mathcal{P}$ to remove all stop words and special symbols, etc. Then we extract the frequency of each unigram from $\mathcal{C}$ in $\mathcal{P}$. Count overlap features essentially measure the similarity between $\mathcal{C}$ and $\mathcal{P}$ in terms of the number of unigram overlap and their overlap frequency. Subsequently, we adopted two different approaches for spoiler classification, namely (i) document classification and (ii) similarity-based approach. In the document classification approach, we simply concatenate the encoded representation $\mathbf{c}$ and $\mathbf{p}$ of post text $\mathcal{C}$ and target paragraphs, $\mathcal{P}$ respectively, along with feature vector $\mathbf{d}$ and $\mathbf{o}$ using the equation defined below.

$$\mathbf{f}=\Big(\mathbf{p} \oplus \mathbf{c} \oplus \mathbf{o} \oplus \mathbf{d}\Big) \tag{4}$$

In contrast, similarity-based approach, we form a feature vector by estimating the angle and difference between $\mathbf{c}$ and $\mathbf{p}$. This estimates how similar post text $\mathcal{C}$ and target paragraphs $\mathcal{P}$ are.

$$\mathbf{f}=\Big(\mathbf{p} \oplus \mathbf{c} \oplus (\mathbf{p}-\mathbf{c}) \oplus (\mathbf{p} \odot \mathbf{c})\ \mathbf{o} \oplus \mathbf{d}\Big) \tag{5}$$

Where $\oplus$ represents the concatenation of two vectors and $\odot$ represents the element-wise multiplication operation between two vectors. Next, we pass the feature vector $\mathbf{f}$ to a two-layer fully connected neural network for spoiler type classification. We use a cross entropy loss function to learn the parameters.

3.2 Subtask 2: Spoiler Generation:

The objective of this subtask is to generate or retrieve text from target paragraphs $\mathcal{P}$, which satisfies the curiosity persuaded by a clickbait post text $\mathcal{C}$. Based on the length and type of text that needs to be generated or retrieved from $\mathcal{P}$, the spoiler is classified into three categories, namely phrase spoiler, passage spoiler, and multipart spoiler. Each of these spoiler categories poses unique characteristics and challenges (Fröbe et al., 2023; Hagen et al., 2022). Considering such challenges, we adopt different approaches for phrase spoiler, passage spoiler, and multipart spoiler generations.

3.2.1 Passage Spoiler Retrieval

To extract the passage spoiler from target paragraphs $\mathcal{P}$, we adopt an information retrieval approach. Initially, we apply a probabilistic model BM25 (Robertson et al., 1994; Trotman et al., 2014) by considering post text $\mathcal{C}$ as a query and different sentences of target paragraphs $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_n\}$ as a set of documents. The objective is to retrieve the most relevant sentences as passage spoilers. Though BM25 is relatively fast and accurate, it is a bag-of-word-based model; hence it does not consider the sequential and contextual information present in $\mathcal{P}$ and $\mathcal{C}$. It also fails to capture semantic similarity. To overcome such limitations of BM25, we consider pretrained language models such Sentence-BERT (S-BERT) (Reimers and Gurevych, 2019, 2020) and DeBERTa (He et al., 2021) to encode post text $\mathcal{C}$ and sentences of target paragraphs $\mathcal{P}$ and then estimate the semantic similarity between encoded vector representations of post text $\mathcal{C}$ and encoded vector representation of sentences of target paragraphs $\mathcal{P}$. Finally, we select the top similar sentences of target paragraphs

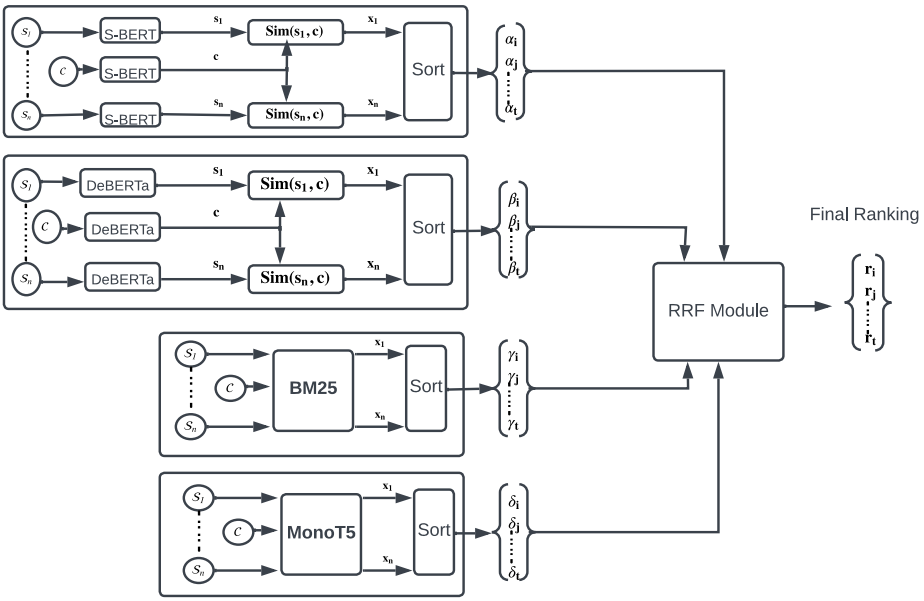

Figure 2: presents the working diagram of the proposed model for spoiler generation.

$\mathcal{P}$ as passage spoilers. Though the above-discussed approach based on a pretrained language model and semantic similarity improved the performance of passage spoiler retrieval over the BM25 model, this approach lacks training and fine-tuning on information retrieval tasks. To overcome these limitations, we consider the pretrained model MonoT5 [3] (Nogueira et al., 2020) on document ranking information retrieval task. Studies (Benham and Culpepper, 2017; Clipa and Di Nunzio, 2020) have shown that reciprocal rank fusion (RRF) (Cormack et al., 2009) is an effective method for combining document ranking obtained from different information retrieval systems. Studies (Benham and Culpepper, 2017; Clipa and Di Nunzio, 2020; Cormack et al., 2009) in the literature also suggest that combining ranks obtained from different information retrieval systems outperform any individual information retrieval system. Motivated by such observations, we combine the rank of sentences obtained using BM25, the pretrained language models with semantic similarity and MonoT5 using reciprocal rank fusion (RRF) (Cormack et al., 2009). Subsequently, we select the top-rank sentence from RRF as a passage spoiler. Figure 2 presents the working of the passage spoiler retrieval system. It first splits the target paragraph $\mathcal{P}$ into set sentences $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_n\}$. Next, encoded representation $\mathbf{c}$ and $\mathbf{s}_i$ of post text $\mathcal{C}$ and sentences $\mathcal{S}_i$ of target paragraphs are obtained using Equations 2 and 1 respectively. Here two different pretrained language models have been used, and based on that; we obtain two different rankings. We obtain the first ranking vector $\boldsymbol{\alpha}$ of sentences in $\mathcal{P}$ with $S-BERT$ and as pretrained language model in Equation 1 and 2 and Equation defined below.

$$\boldsymbol{\alpha}=\left(\mathbf{c}\ \mathbf{P}^{\intercal}\right) \tag{6}$$

Where $\mathbf{P}$ is a matrix of sentences encoding $\mathbf{s}_i$, and $\intercal$ is a matrix, transpose operator. Similarly, we also obtain a ranking vector $\boldsymbol{\beta}$ by considering $DeBERTa$ as a pretrained language model in Equation 1, 2 and following Equation 6. Next, we obtain the ranking of sentences $\boldsymbol{\gamma}$ using BM25 as defined in the equation below.

$$\boldsymbol{\gamma} = \left(\mathbf{BM25}\left(\mathcal{C}, \mathcal{S}_1, ..., \mathcal{S}_1\right)\right) \tag{7}$$

Similarly, we further obtain the ranking of sentences $\boldsymbol{\delta}$ using Mono-T5 as defined in the equation below.

$$\boldsymbol{\delta} = \left(\mathbf{MonoT5}\left(\mathcal{C}, \mathcal{S}_1, ..., \mathcal{S}_1\right)\right) \tag{8}$$

Next, we combine the ranking of sentences $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ using RRF as defined in Equation 9. Our RRF ranking Equation is similar to the equation defined in study (Cormack et al., 2009).

$$\mathbf{x}_i=\mathbf{RRF}\left(\mathcal{S}_i \in \mathcal{P}\right) = \left(\sum\nolimits_{\mathbf{r}\in\mathbf{R}} \frac{1}{\mathbf{k} + \mathbf{r}\left(\mathcal{S}_i\right)}\right) \tag{9}$$

[3] MonoT5 Source

We set k=60 and number of ranking R= 4 namely α, β, γ and δ. We select the top sentence ranked by RRF as passage spoiler.

3.2.2 Phrase Spoiler Retrieval

The key objective in phrase spoiler generation is to extract the group of words from target paragraph $\mathcal{P}$, which satisfy the curiosity induced by a clickbait post text $\mathcal{C}$. Considering such challenging nature of phrase spoiler generation, we first apply the word-level information retrieval system discussed in subsection 3.2.1 to retrieve words from target paragraphs. But we observe that such a system retrieves words from different parts of target paragraphs $\mathcal{P}$, which lacks correlations and context among them. Considering such limitations of a word-level information retrieval system, we apply a question-answering approach that considers post text $\mathcal{C}$ as a query and target paragraphs $\mathcal{P}$ as a corpus from where answers need to be extracted. We consider two pretrained language models trained on question-answering tasks, RoBERTA[4] (Liu et al., 2019) and DeBERTa[5] (He et al., 2021). We also tried a system by combining (concatenation, union or intersection) the output of RoBERTA (Liu et al., 2019) and DeBERTa (He et al., 2021) for phrase spoiler generation, but we observed that RoBERTA (Liu et al., 2019) alone outperforms DeBERTa (He et al., 2021) and combinations of RoBERTA (Liu et al., 2019) and DeBERTa (He et al., 2021).

3.2.3 Multipart Spoiler Retrieval

The main objective of multipart spoiler retrieval is to extract more than one passage or phrase from target paragraphs $\mathcal{P}$ to satisfy the curiosity persuaded by a clickbait post text $\mathcal{C}$. These extracted passages or phrases could be non-consecutive or from different segments of target paragraphs $\mathcal{P}$. Considering the characteristics of a multipart spoiler, that multipart spoiler could be either passages or phrases, or both. We combine our approach proposed in subsection 3.2.1 for passage spoiler retrieval and the phrase retrieval approach discussed in section 3.2.2 for multipart spoiler retrieval. Given a post text $\mathcal{C}$ and target paragraphs $\mathcal{P}$ first we apply our passage spoiler retrieval system presented in section 3.2.1 and apply the sorting over the rank generated by RRF and select the top k sentences. If any numeric figure is mentioned in post text $\mathcal{C}$, then we consider that numeric figure as the value of k; otherwise, we set the value of k to 5. Next, we feed each pair of post text $\mathcal{C}$ and sentence $\mathcal{S}_i$ from top k selected sentences to pretrained RoBERTa model trained for question answering task as defined in blow Equations.

$$\mathcal{M}_1 = \Big(\textbf{RoBERT}\Big(\mathcal{C}, \mathcal{S}_1\Big)\Big) \tag{10}$$

$$\mathcal{M}_k = \Big(\textbf{RoBERT}\Big(\mathcal{C}, \mathcal{S}_k\Big)\Big) \tag{11}$$

Subsequently, we combine answers $\mathcal{M}_1$ to $\mathcal{M}_k$ generated by RoBERTa model as defined in Equations 10 and 11 for multipart spoiler generation.

4 Experimental Setup

Table 3 presents the details of hyperparameters used to produce the results presented in this paper. This study uses the dataset provided by (Hagen et al., 2022) to analyze the performance of proposed systems. Further details of experimental setup in presented in section A.2.

5 Results and discussion

Table 6 present the performance comparison between the different setups of the proposed model Recurrence over Pretrained Encoding RPE over the validation set. Extensive study and analysis of experimental results over validation set are presented in section A.3. Comparing the performance of different setups of RPE from Table 6 following observations can be made: (i) performance of the RPE model with the similarity-based approach is superior compared to the RPE model with document classification approach. This indicates spoiler types also rely on the similarity between post text and the target paragraph. (ii) performance RPE model with $RoBERTa$ as the encoder is superior to RPE model with $BERT$ and $DeBERTa$. (iii) adding feature F (count overlap and document length) boost the performance of the RPE model. This validates our intuition behind considering document length features and counts overlap features that spoiler types heavily rely on length. However, the RPE model with a similarity-based approach and features $RPE(RoBERT, S, F)$ outperform all other setups of the RPE model. Accordingly, we submitted $RPE(RoBERT, S, F)$ for evaluation over the test set. Table 1 presents the performance of the $RPE(RoBERT, S, F)$ model over the test dataset for spoiler-type classification.

[4]Pretrained RoBERTa model on question answering task
[5]Pretrained DeBERTa model on question answering task

Table 1: Overview of the effectiveness in spoiler type prediction (subtask 1 at SemEval 2023 Task 5) measured as balanced accuracy over all three spoiler types and precision (Pr.), recall (Rec.), and F1 score (F1) for the phrase, passage, and multi spoilers on the test set.

Submission			Accuracy	Phrase			Passage			Multi		
Team	Approach	Run		Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1
jack-flood	upload	2023-02-12-16-10-51	0.52	0.57	0.56	0.56	0.55	0.66	0.60	0.61	0.34	0.44

Table 2: Overview of the effectiveness in spoiler generation (subtask 2 at SemEval 2023 Task 5) measured as BLEU-4 (BL4), BERTScore (BSc.) and METEOR (MET) over all clickbait posts respectively those requiring phrase, passage, or multi spoilers on the test set.

Submission			All			Phrase			Passage			Multi		
Team	Approach	Run	BL4	BSc.	MET	BL4	BSc.	MET	BL4	BSc.	MET	BL4	BSc.	MET
jack-flood	upload	2023-02-12-16-12-01	0.18	0.88	0.18	0.32	0.89	0.14	0.08	0.87	0.21	0.05	0.85	0.16

Table 5 presents the performance of the different combinations of systems for passage, phrase, and multipart spoiler generation. As mentioned in 3.2.1 $S-BERT$ has been used to encode a sentence for generating ranking score $\boldsymbol{\alpha}$, $DeBERTa$ has been used to encode a sentence for generating ranking score $\boldsymbol{\beta}$, ranking score $\boldsymbol{\gamma}$ is obtained using BM25 and ranking score $\boldsymbol{\delta}$ obtained using $MonoT5$. Accordingly, in Table 3.2.1 $S-BERT$, and $DeBERTa$, BM25 and $MonoT5$ indicated that passage spoiler are obtained based on $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ rank respectively. Similarly, RRF indicates our proposed system for passage spoiler retrieval in Figure 2. $RoBERTa$ and $DeBERTa$ in phrase columns indicate that pretrained $RoBERTa$ and $DeBERTa$ models, respectively, over question answering task is considered for phrase retrieval. Considering the BSc. score of systems from Table 5, it is apparent that our proposed system is able to retrieve the passage, phrase, and multipart spoiler, which is semantically very close to ground truth. Considering the performance measures BL4 and MET from Table 5, it is evident that the performance of our proposed system is average. BL4 and MET, which relies on the token overlap between generated text and ground truth(Zhang et al., 2019), and BSc, which relies on semantic similarity between generated text and ground truth. Study (Zhang et al., 2019) also suggest token overlap based evaluation methods fail to consider meaning-preserving tokens and compositional diversity. Our error analysis for different performance measures for the spoiler generation task also suggests that the BERTScore is more suitable for comparing ground truth spoilers and generated spoilers. Section A.1 presents the details of our error analysis over different performance measures in spoiler generation tasks. It is apparent that though our system performance is average over BL4 and MET, our system is able to retrieve text which is semantically very close to or similar to spoiler ground truth. Table 2 presents the performance of our system (RRF for passage, RoBERTa for phrase and RRF+RoBERTa for multipart spoiler) over the test set. Considering the BSc score in Table 2 it is apparent that our system is able to generate a spoiler that is semantically very close to ground truth.

6 Conclusion

We propose a hierarchical encoding-based model recurrence over Pretrained Encoding RPE for spoiler-type classification. This paper adopts reciprocal rank fusion (RRF) for passage retrieval and adopt question answering based approach for phrase spoiler retrieval, and combine the passage and phrase retrieval model for multipart spoiler retrieval.Our results suggest that the proposed system effectively identifies spoiler types and retrieves spoilers from the target paragraph, which are semantically very close to the ground truth spoiler. In future work, we plan to explore extractive summarization methods for spoiler generation.

References

Amol Agrawal. 2016. Clickbait detection using deep learning. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 268–272.

Ankesh Anand, Tanmoy Chakraborty, and Noseong Park. 2017. We used neural networks to detect click-

baits: You won't believe what happened next! In *Advances in Information Retrieval: 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings 39*, pages 541–547. Springer.

Rodger Benham and J Shane Culpepper. 2017. Risk-reward trade-offs in rank fusion. In *Proceedings of the 22nd Australasian Document Computing Symposium*, pages 1–8.

Prakhar Biyani, Kostas Tsioutsiouliklis, and John Blackmer. 2016. " 8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Jonas Nygaard Blom and Kenneth Hansen. 2015. Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics*, 76:87–100.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, pages 9–16. IEEE.

Teofan Clipa and Giorgio Maria Di Nunzio. 2020. A study on ranking fusion approaches for the retrieval of medical publications. *Information*, 11(2):103.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

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

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

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.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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. *arXiv preprint arXiv:1907.11692*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Md Main Uddin Rony, Naeemul Hassan, and Mohammad Yousuf. 2017. Diving deep into clickbaits: Who use them to what extents in which topics with what effects? In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 232–239.

Deborah Beth Schaffer. 1995. Shocking secrets revealed! the language of tabloid headlines.

Craig L. Silverman. 2015. Lies, damn lies and viral content.

Qian Sun, Aili Shen, Hiyori Yoshikawa, Chunpeng Ma, Daniel Beck, Tomoya Iwakura, and Timothy Baldwin. 2021. Evaluating hierarchical document categorisation. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*, pages 179–184.

Thales S. Teixeira. 2014. The rising cost of consumer attention: Why you should care, and what you can do about it.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 58–65.

Seunghyun Yoon, Kunwoo Park, Joongbo Shin, Hongjun Lim, Seungpil Won, Meeyoung Cha, and Kyomin Jung. 2019. Detecting incongruity between

news headline and body text via a deep hierarchical encoder. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 791–800.

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

A Appendix

A.1 Error Analysis of BERTScore vs BLEU-4 vs Meteor

Figure 3 presents a comparison between the performance of the BLEU-4 score and the BERTScore for comparing the relation between ground truth spoiler and generated spoiler. The following observations can be made from Figure 3. If we consider example 4, it is evident that the ground truth spoiler and the generated spoiler are exactly similar but the BLEU-4 score is low, and the BERTScore is 1. Examples 2, 3, and 5 show that generated spoilers convey the same meaning by describing and elaborating the phrase in ground truth spoiler but even in this case, the BLEU-4 score is low, and the BERTScore is high. Similarly, for example 1, it is evident that both ground truth spoilers and generated spoilers are semantically similar, but due to a lack of lexical overlap between them, BLEU-4 score is low, but the BERTScore is high. From Figure 3 and the above discussion, it is established that the BERTScore is more suitable for the evaluation of spoiler generation rather than other BLEU and Meteor scores.

A.2 Details of Experimental Setups, Experimental Datasets and Hyperparameters

Table 3 presents the details of hyperparameters used to produce the results presented in this paper. We use a cross-entropy loss function with a learning rate of 0.01 to learn the parameters. We consider at most 35 sentences in the target paragraph $\mathcal{P}$. We discard the sentences if it $\mathbf{P}$ has more than 35 sentences and apply padded random vector if it $\mathbf{P}$ has more than 35 sentences. This study uses the dataset provided by (Hagen et al., 2022) to analyze the performance of proposed systems. The no of posts for each training, validation and test split are shown in Table 4. From Table 4, it is apparent that the dataset has an imbalance where the multipart class is over-represented by phrase and passage class. It is to be noted that no extra data was used for training our models. We used the validation split to choose the most effective model for both tasks.

Table 3: Details of Experimental Setups and Hyperparameters

	Values
Epoch	500
Batch Size	5
Learning Rate	0.01
Cell State Dimension of LSTM	100
Hidden State Dimension of LSTM	100
Number of Layer in MLP	2
Maximum #sentence in $\mathbf{P}$	35

Table 4: Characteristics of experimental datasets for spoiler type classification and generation.

	Phrase	**Passage**	**Multipart**
Train	1367	1274	559
Valid	335	322	143
Test	423	403	174

A.3 Analysis of Experimental Results on Validation Set

Table 6 present the performance comparison between the different setups of the proposed model Recurrence over Pretrained Encoding RPE over the validation set. Our proposed RPE model differs in three parameters: (i) encoding of post text $\mathcal{C}$ and sentences $\mathcal{S}_i$ of target paragraph using BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019) or DeBERTa (He et al., 2021), (ii) document classification $\mathcal{D}$ or similarity-based approach $\mathcal{S}$, (iii) $\mathcal{F}$ denotes that count overlap and document overlap feature are considered in the model. For example, $RPE(BERT, D, F)$: indicate that $\mathcal{C}$, $\mathcal{S}_i$ is encoded using BERT (Kenton and Toutanova, 2019), D indicates that document classification approach and F denote that count overlap and document length features are considered. Comparing the performance of different setups of RPE from Table 6 following observations can be made: (i) performance of the RPE model with the similarity-based approach is superior compared to the RPE model with document classification approach. This indicates spoiler types also rely on the similarity between post text and the target paragraph. (ii) performance RPE

Sl. No.	Post Text	Ground Truth Spoiler	Generated Spoiler	BLEU-4 Score	BERT Score	Meteor Score
1	Explainer: Is There Any Science Behind Astrology?	Although astrologers seek to explain the natural world, they don't usually attempt to critically evaluate whether those explanations are valid and this is a key part of science.	There haven't been many studies that investigate the science behind astrology, but of the few that have, the results have failed to support the validity of astrological views.	1.07e-231	0.96	0.11
2	The one morning work mistake you can't recover from	starts later	The researchers concluded that employees who can opt for flexible work schedules should shift their schedules earlier, not later, to account for managers' morning bias.	8.06e-232	0.95	0.10
3	We've finally figured out when the moon formed	4.47 billion years ago	The researchers deduced that the moon-forming impact occurred about 4.47 billion years ago, in agreement with many previous estimates.	2.25e-78	0.97	0.69
4	Guess who Obama just dined with in Vietnam	Anthony Bourdain	Anthony Bourdain	1.22e-77	1.0	0.93
5	Scientists say this behavior can make men more attractive to women	altruism	Both men and women rated the altruistic people as more attractive for long-term relationships - but women showed a stronger preference for altruism than men did.	9.50e-232	0.96	0.13

Figure 3: Present a comparison between the performance measures BLEU-4, BERTScore, and Meteor for different types of samples.

model with $RoBERTa$ as the encoder is superior to RPE model with $BERT$ and $DeBERTa$. (iii) adding feature F (count overlap and document length) boost the performance of the RPE model. This validates our intuition behind considering document length features and counts overlap features that spoiler types heavily rely on length. However, the RPE model with a similarity-based approach and features $RPE(RoBERT, S, F)$ outperform all other setups of the RPE model.

Table 5: presents the performance of different setups of proposed systems for spoiler generation over the validation set.

System			All			Phrase			Passage			Multipart		
Passage	Phrase	Multipart	BL4	BSc	MET	BL4	BSc	MET	BL4	BSc	MET	BL4	BSc	MET
S-BERT	RoBERTa	S-BERT+ RoBERTa	0.189	0.878	0.176	0.332	0.899	0.135	0.103	0.870	0.216	0.047	0.847	0.140
DeBERTa	RoBERTa	DeBERTa+ RoBERTa	0.190	0.878	0.178	0.332	0.900	0.143	0.102	0.868	0.205	0.053	0.847	0.162
MonoT5	RoBERTa	MonoT5+ RoBERTa	0.200	0.880	0.192	0.335	0.901	0.138	0.125	0.873	0.239	0.053	0.848	0.152
BM25	RoBERTa	BM25+RoBERTa	0.185	0.877	0.175	0.331	0.900	0.145	0.096	0.866	0.208	0.044	0.848	0.144
RRF	**RoBERTa**	**RRF+RoBERTa**	0.196	0.879	0.184	0.337	0.901	0.145	0.117	0.871	0.226	0.042	0.846	0.141
S-BERT	DeBERTa	S-BERT+DeBERTa	0.124	0.812	0.152	0.167	0.791	0.075	0.116	0.839	0.213	0.042	0.800	0.126
DeBERTa	DeBERTa	DeBERTa+DeBERTa	0.127	0.811	0.154	0.168	0.792	0.078	0.116	0.837	0.204	0.054	0.801	0.142
MonoT5	DeBERTa	MonoT5+DeBERTa	0.137	0.814	0.168	0.170	0.792	0.079	0.140	0.843	0.233	0.051	0.801	0.141
BM25	DeBERTa	BM25+DeBERTa	0.122	0.811	0.154	0.166	0.792	0.077	0.110	0.835	0.213	0.043	0.802	0.135
RRF	DeBERTa	RRF+DeBERTa	0.132	0.813	0.161	0.173	0.792	0.081	0.131	0.840	0.223	0.041	0.799	0.157

Table 6: presents the performance of different setups of RPE model for spoiler-type classification. Here D and S indicate document classification and similarity-based approach, respectively, F indicates the count overlap and document length feature.

	Acc	**Phrase**			**Passage**			**Multipart**			**All**		
		Pr	**Rec**	**F1**	**Pr**	**Rec**	**F1**	**Pr**	**Rec**	**F1**	**Pr**	**Rec**	**F1**
$RPE(BERT, D)$	0.49	0.52	0.42	0.47	0.46	0.62	0.53	0.55	0.36	0.44	0.50	0.49	0.49
$RPE(RoBERTa, D)$	**0.55**	0.56	**0.60**	**0.58**	**0.54**	0.56	0.55	**0.51**	**0.38**	**0.44**	**0.55**	**0.55**	**0.54**
$RPE(DEBERTa, D)$	0.54	**0.59**	0.45	0.51	0.52	**0.70**	**0.60**	0.46	0.34	0.39	0.54	0.54	0.53
$RPE(BERT, S)$	0.50	0.57	0.35	0.43	0.48	**0.67**	0.56	0.45	0.43	0.44	0.51	0.50	0.49
$RPE(RoBERTa, S)$	**0.55**	0.56	**0.60**	**0.58**	**0.54**	0.56	0.55	**0.51**	0.38	0.44	**0.55**	**0.55**	**0.54**
$RPE(DeBERTa, S)$	0.54	**0.61**	0.43	0.50	0.53	0.66	**0.59**	0.46	**0.50**	**0.48**	0.55	0.54	0.53
$RPE(BERT, S, F)$	0.46	0.54	0.19	0.29	0.43	**0.81**	0.56	0.52	0.27	0.35	0.49	0.46	0.41
$RPE(RoBERTa, S, F)$	**0.59**	**0.60**	**0.63**	**0.62**	**0.56**	0.64	**0.59**	**0.69**	**0.39**	**0.50**	**0.60**	**0.59**	**0.59**
$RPE(DeBERTa, S, F)$	0.55	0.57	0.50	0.54	0.51	0.69	0.58	0.76	0.36	0.49	0.58	0.55	0.55