Clickbait Spoiling via Question Answering and Passage Retrieval

Matthias Hagen[1] **Maik Fröbe**[1] **Artur Jurk**[1] **Martin Potthast**[2]

[1]Martin-Luther-Universität Halle-Wittenberg
[2]Leipzig University

Abstract

We introduce and study the task of clickbait spoiling: generating a short text that satisfies the curiosity induced by a clickbait post. Clickbait links to a web page and advertises its contents by arousing curiosity instead of providing an informative summary. Our contributions are approaches to classify the type of spoiler needed (i.e., a phrase or a passage), and to generate appropriate spoilers. A large-scale evaluation and error analysis on a new corpus of 5,000 manually spoiled clickbait posts—the Webis Clickbait Spoiling Corpus 2022—shows that our spoiler type classifier achieves an accuracy of 80%, while the question answering model DeBERTa-large outperforms all others in generating spoilers for both types.

1 Introduction

Clickbait is the term used to describe posts in social media that are intended to inappropriately entice their readers to visit a web page. This is achieved through formulations such as sensationalism or cataphors that are believed to create a so-called curiosity gap: "a form of cognitively induced deprivation that arises from the perception of a gap in knowledge or understanding" (Loewenstein, 1994). Clickbait is perceived as inappropriate since its resolution is usually ordinary or trivial, comprising little more than a phrase, short passage, or a list of things that could just as easily have been included in the post. This observation motivates us to introduce the task of clickbait spoiling: identifying or generating a spoiler for a clickbait post.

Figure 1 shows four examples of clickbait on Twitter, along with spoilers. The first two tweets explicitly or implicitly promise a surprising resolution to spark curiosity, but their spoilers are brief and trivial. The linked page of the first tweet adds almost nothing, and the spoiler of the second is common sense. The third spoiler is a passage from the linked page, and the fourth is a list of things.

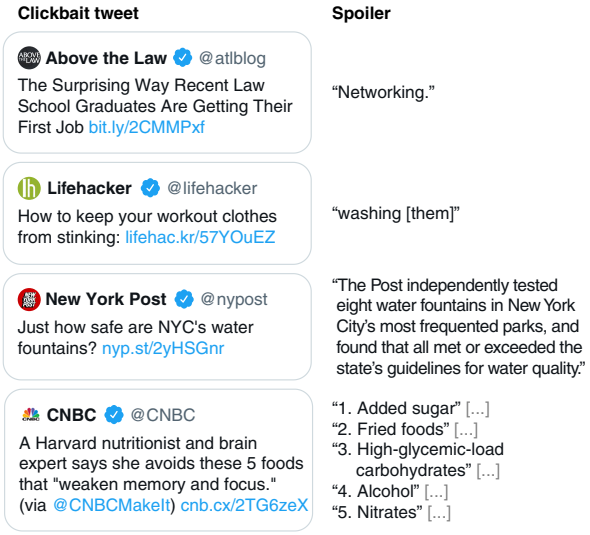

Figure 1: Examples of clickbait tweets and spoilers for them extracted from the respective linked web page.

Even though there are length limits to the informativeness of tweets, the spoilers in all examples could easily have been part of the original tweets.

This paper reports about our investigation into clickbait spoiling and the following contributions: (1) The Webis Clickbait Spoiling Corpus 2022 (Webis-Clickbait-22), consisting of 5,000 clickbait posts, their linked pages and a spoiling piece of text therein.[1] (2) A two-step approach to clickbait spoiling that first classifies a clickbait post according to its spoiler type (phrase or passage), and then treats spoiling either as a question answering or as a passage retrieval task. (3) A systematic evaluation of state-of-the-art methods for spoiler type classification, question answering, and passage retrieval.[2] Although the first step of spoiler type classification is not necessary, our results suggest that it can be helpful. Even more so, as we have not yet tackled multipart spoilers (bottom example in Figure 1; 876 cases also part of our corpus) that probably require a different spoiling approach.

[1]Data: https://webis.de/data.html?q=clickbait
[2]Code: https://github.com/webis-de/ACL-22

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7025 - 7036

2 Related Work

Following an overview of research on clickbait and its operationalization so far, models of question answering and passage retrieval are examined.

2.1 Clickbait and its Operationalization

The underlying assumption of most research on clickbait is that it is a form of data-driven optimization of social media posts to exploit the curiosity gap described by Loewenstein (1994). At least that's what Peter Koechley (2012), the CEO of Upworthy, claimed. Upworthy became one of the first major spreaders of clickbait on Facebook, and their success has prompted Facebook to change its news recommendation algorithms to curb the amount of clickbait, twice (El-Arini and Tang, 2014; Peysakhovich and Hendrix, 2016).

Exploratory and theoretical studies of clickbait and its impact on journalism analyzed its prevalence for more than 150 publishers (Rony et al., 2017); its economics for the news market (Munger, 2020); its impact on perceptions of credibility and quality (overall negative) (Molyneux and Coddington, 2020); and noted a slow decline over the past decade (Lischka and Garz, 2021).

Journalistic studies of this kind rely on clickbait detection technologies. Originally proposed by Rubin et al. (2015) but not followed up, Potthast et al. (2016) and Chakraborty et al. (2016) independently developed the first detectors. Starting from a shared task organized by Potthast et al. (2018) shortly after, more than 50 approaches have been contributed to date. An overview is beyond the scope of our work, but transformer models dominate this task as well. For the clickbait generation task, preceded by a rule-based generator (Eidnes, 2015), only Shu et al. (2018) and Xu et al. (2019) have presented more advanced models, while Karn et al. (2019) generate teaser headlines that are explicitly not meant to be clickbait. So far, no attempt has been made to generate spoilers for clickbait.

2.2 Question Answering

If one considers clickbait spoiling as a question answering problem, there are numerous possible solutions. Among the available question-answering benchmarks (Dzendzik et al., 2021), we select two to choose appropriate state-of-the-art models for our evaluation: (1) SQuAD (Rajpurkar et al., 2016) compiles 107,785 questions and answers based on 536 Wikipedia articles. Although a wide range of questions and answers are included, the vast majority of 93.6% are factual (32% names, 31.8% noun phrases, 19.8% numbers, 5.5% verb phrases, and 3.9% adjective phrases), while the remainder are descriptive (3.7% clauses and 2.7% other). We use SQuAD v1.1, not the v2.0 superset (Rajpurkar et al., 2018), which contains unanswerable questions, since we do not expect clickbait to be "unspoilable". (2) TriviaQA (Joshi et al., 2017) contains 95,000 question–answer pairs, mostly dealing with trivia questions that are supposed to be particularly difficult to answer. These are comparable to clickbait in that many of them address rather trivial things (see Figure 1).

The question answering models used in our experiments are ALBERT (Lan et al., 2020), AllenAI-Document-QA (Clark and Gardner, 2018), BERT (cased/uncased) (Devlin et al., 2019), Big Bird (Zaheer et al., 2020), DeBERTa (large) (He et al., 2021), ELECTRA (Clark et al., 2020), Funnel-Transformer (Dai et al., 2020), MPNet (Song et al., 2020), and RoBERTa (base/large) (Liu et al., 2019). Many of them are or were state of the art on the above benchmarks and implement various different architectural paradigms.

2.3 Passage Retrieval

Passage retrieval relaxes the question answering task a bit in the sense of allowing longer passages of text as answers (e.g., one or more sentences), rather than exact phrases or statements. Neural retrieval models, as surveyed by Guo et al. (2020) and Lin et al. (2021), have been successfully applied to passage retrieval. One of the most important passage retrieval benchmarks is part of MS MARCO, a series of challenges whose first edition was a large question answering task (Nguyen et al., 2016). A passage retrieval dataset of 8.8 million passages was derived for the underlying set of 100,000 questions originally submitted to Bing. This dataset formed the basis for two consecutive shared tasks at the TREC 2019 and 2020 Deep Learning tracks (Craswell et al., 2019, 2020).

The passage retrieval models used in our experiments are MonoBERT (Nogueira and Cho, 2019; Nogueira et al., 2019) and MonoT5 (Nogueira et al., 2020) (both topped the MS MARCO passage retrieval leaderboard once), and the classic baseline models BM25 (Robertson and Zaragoza, 2009) and Query Likelihood (Ponte and Croft, 1998), implemented in Anserini (Yang et al., 2017).

3 Webis Clickbait Spoiling Corpus 2022

To tackle clickbait spoiling for the first time, we created the Webis Clickbait Spoiling Corpus 2022 (Webis-Clickbait-22), a collection of 5,000 clickbait posts and their associated spoilers.

3.1 Corpus Construction

Our corpus is primarily based on five social media accounts on Twitter, Reddit, and Facebook that manually spoil clickbait: r/savedyouaclick, @HuffPoSpoilers, @SavedYouAClick, @UpworthySpoiler, and @StopClickBaitOfficial. With the goal of collecting 5,000 "spoilable" clickbait posts at an expected rejection rate of around 10% of unusable posts, 5,555 were initially collected from the accounts. Each of them was manually reviewed, and those that turned out not to be spoiled clickbait were removed (e.g., funny posts not intended to be spoilers, or posts with unavailable linked documents). The rejection rate was higher than expected, and only 4,204 posts remained.

To reach our goal of 5,000 posts, we then sampled from the Webis-Clickbait-17 corpus used in the Clickbait Challenge 2017 (Potthast et al., 2018). The corpus contains 38,517 tweets, each of which was rated by 5 annotators on a 4-point Likert scale for clickbaitiness: "no clickbait," "slight clickbait," "considerable clickbait," and "heavy clickbait." Of the tweets, 1,845 scored an average of 0.8 or higher and can safely be considered clickbait. We selected tweets from this subset and manually spoiled them based on the linked document until our target size of 5,000 posts was reached.

Thus, our final corpus consists of 4,204 posts from Twitter, Reddit, and Facebook that were spoiled by a third party specializing in this task, and 796 tweets from the Webis-Clickbait-17 corpus with an average clickbaitiness of at least 0.8 that we spoiled ourselves. For each of the 5,000 clickbait posts, we also reviewed and corrected erroneous spoilers and labeled their exact positions in the linked documents. Our internal guidelines dictated that a spoiler should be as short as possible (i.e., if one word is enough, not a whole sentence should be chosen). Since the underlying annotation task is simple, one main annotator was sufficient. Nevertheless, randomly selected as well as ambiguous cases were discussed with two additional experts among the co-authors. No systematic errors or unforeseen difficulties in solving the annotation task were identified during these discussions.

During our annotation, we found that none of the common approaches to main content extraction worked reliably for all the documents linked in the clickbait posts. Yet, clean content is a prerequisite for research on clickbait spoiling to eliminate as many confounding variables as possible. To ensure a clean corpus, one annotator manually extracted the main content of the linked documents, removing (inline) advertisements, links to related articles (e.g., "READ ALSO: [...]" or "Also from CNBC [...]"), credits (e.g., "Image credit: [...]" or "Photo by [...]"), and social media links (e.g., "Subscribe to [...]" or "Follow us on [...]"). A random selection was reviewed to ensure high quality.

Moreover, during spoiler annotation, it turned out that there are basically three types of spoilers: (1) *phrase spoilers* consisting of a single word or phrase from the linked document (e.g., the first two spoilers in Figure 1, but often named entity spoilers as well), (2) *passage spoilers* consisting of one or a few sentences of the linked document (e.g., the third spoiler in Figure 1), and (3) *multipart spoilers* consisting of more than one non-consecutive phrases or passages of the linked document (e.g., the fourth spoiler in Figure 1). Spoiler types were also annotated by the main annotator, and randomly checked by the other two.

In sum, each of the 5,000 posts in our corpus consists of a unique ID, the platform from which it was taken, the respective platform's post ID, the post's text (i.e., the "clickbait"), the URL to the linked document, the manually extracted title and paragraph-divided main content of the linked document, the manually optimized spoiler, the spoiler's character position in the main content, and the type of spoiler (phrase, passage, or multipart). In total, the annotation took about 560 hours, which marked the limit of our budget dedicated for this step.

3.2 Corpus Statistics

Table 1 summarizes the main statistics of our corpus. Most spoiled clickbait posts come from Twitter (47.5%) and Reddit (36%), whereas the Facebook account contributes less (16.5%). Most spoilers are phrases (42.5%) and passages (40%). That there are fewer multi-part spoilers could be due to the fact that spoiler account operators prefer to spoil "simpler" clickbait posts. For the corpus, we also provide a fixed random 80/20/20 train/validation/test split to ensure future reproducibility and comparability with our results.

Table 1: Key statistics of the Webis Clickbait Spoiling Corpus 2022 (Webis-Clickbait-22).

Source	Spoiler	Entries	Average text length ± Std.Dev.			Corpus splits			Top source	
			Post	Document	Spoiler	Train	Val.	Test	Name	Count
Facebook	Phrase	342	13.4 ±3.6	433.7 ±347.9	3.0 ±1.6	221	45	76	Stop Clickbait	342
	Passage	388	13.4 ±4.0	490.9 ±351.5	24.9 ±20.0	231	73	84	Stop Clickbait	388
	Multipart	94	14.2 ±4.1	651.8 ±545.2	28.5 ±33.0	68	12	14	Stop Clickbait	94
Reddit	Phrase	688	13.2 ±4.0	584.6 ±798.6	2.8 ±1.6	455	109	124	savedyouaclick	688
	Passage	859	13.1 ±4.0	657.2 ±1004.7	25.4 ±20.3	533	148	178	savedyouaclick	859
	Multipart	250	12.8 ±4.4	991.7 ±899.5	32.7 ±36.2	162	46	42	savedyouaclick	250
Twitter	Phrase	1,095	11.0 ±3.4	479.1 ±502.9	2.7 ±1.7	691	181	223	HuffPoSpoilers	794
	Passage	752	10.3 ±4.2	597.4 ±605.8	22.3 ± 13.5	510	101	141	HuffPoSpoilers	328
	Multipart	532	11.5 ±3.8	884.0 ±930.3	35.4 ±34.4	329	85	118	HuffPoSpoilers	148
$\sum$	Phrase	2,125	12.1 ±3.8	505.9 ±599.4	2.8 ±1.6	1,367	335	423	HuffPoSpoilers	794
	Passage	1,999	12.1 ±4.3	602.4 ±774.0	24.1 ±18.1	1,274	322	403	savedyouaclick	859
	Multipart	876	12.2 ±4.1	889.8 ±892.2	33.9 ±34.8	559	143	174	savedyouaclick	250

4 Type-dependent Clickbait Spoiling

Our approach to clickbait spoiling is based on the observation that there are three types of spoilers: (1) phrase spoilers, (2) passage spoilers, and (3) multipart spoilers. We assume that different tailored approaches will work best for each spoiler type. However, an important prerequisite for this is the corresponding classification of clickbait. Therefore, we first investigate how well the spoiler type of a clickbait post can be predicted (Section 4.1).

The generation of phrase and passage spoilers for a given clickbait post is similar in that the solution to the problem in both cases amounts to extracting a coherent piece of text from the linked document. To this end, there are a variety of existing approaches in related disciplines whose output is either a phrase or a passage, and which may be adapted to clickbait spoiling. We therefore investigate whether phrase spoilers can be identified by conventional question answering methods (i.e., we treat a clickbait post as a "question" to which a phrase of the linked document should be returned as the "answer"; Section 4.2), and whether passage spoilers can be identified by conventional passage retrieval methods (i.e., we treat a clickbait post as a "query" and the paragraphs of the linked document as the collection from which to retrieve the best "passage"; Section 4.3). In our evaluation, we focus on phrase and passage spoilers and also examine the abilities of the above question answering and passage retrieval methods to serve as one-size-fits-all solutions for phrases and passages. For multipart spoilers, a novel approach will be needed, which is beyond the scope of our current work but an interesting direction for the future.

4.1 Spoiler Type Classification

For the spoiler classification subtask, we experimented with classic feature-based models (Naïve Bayes, Logistic Regression, SVM) and the neural models BERT-, DeBERTa-, and RoBERTa.

As feature types for the classic models, we use tf- and $tf \cdot idf$-weighted word and POS tag uni- and bigrams from the clickbait post and $tf \cdot idf$-weighted word and POS tag uni- and bigrams from the linked document. We include features from the linked document, since it has to be analyzed for the spoiler generation anyway. The idf values are calculated on the OpenWebText corpus (Gokaslan and Cohen, 2019) to prevent any bias from the comparatively small size of our corpus.

The input for the neural models is a post concatenated with the main content of the linked document.

4.2 Phrase Spoiler Generation

Viewing a clickbait post for which a phrase spoiler should be derived as a "question" and the linked document as potentially containing an "answer", phrase spoiler generation can be tackled by question answering methods. We therefore employ ten state-of-the-art question answering methods trained on the SQuAD data and fine-tune them on our new clickbait spoiling training set: ALBERT, BERT (cased/uncased), BigBird, DeBERTa (large), ELECTRA, FunnelTransformer, MPNet, and RoBERTa (base/large).

4.3 Passage Spoiler Generation

Treating the clickbait post whose spoiler type is a passage as a "query" for which the "most relevant" passage from the linked document is to be

Table 2: Effectiveness of spoiler type classification in the multi-class (first column) and one-vs-rest settings on 1000 test posts (training: 3200; validation: 800).

Model	**Balanced accuracy** (0, 1, 2 indicate class labels)			
Phrase	0	1	0	0
Passage	1	0	1	0
Multipart	2	0	0	1
Naïve Bayes	56.15	65.03	62.50	64.82
SVM	59.62	68.03	68.70	70.28
Log. Regression	60.04	68.04	69.33	71.26
BERT	67.84	74.06	75.70	75.56
DeBERTa	**73.63**	78.39	78.65	77.93
RoBERTa	71.57	**80.39**	**79.30**	**79.12**

retrieved, passage spoiler generation can be tackled by passage retrieval methods. We therefore use ten state-of-the-art passage retrieval approaches trained on the MS MARCO data: BM25 and QLD in four variants each (alone or with RM3/Ax/PRF query expansion), MonoBERT, and MonoT5. In addition, we also adapt all of the above question answering models to retrieve passages by simply considering the passage as the returned result from which the question answering model extracts its answer.

5 Evaluation of Spoiler Type Classification

In our evaluation, we assume a setup in which a previous clickbait detection would have (perfectly) identified posts as clickbait. To then evaluate the effectiveness of spoiler type classification on such detected clickbait posts, we conduct three experiments: (1) multi-class, (2) one-vs-rest, and (3) one-vs-one for the types of phrase and passage spoilers.

In all cases, the hyperparameters of the six studied classifiers were optimized based on the validation set of our corpus. For the three feature-based approaches, a chi-square feature selection step selected all post-based features and 70% of the document-based features. The post-based features are weighted 4-times higher than the document-based features. Most hyperparameters of the transformer models were left at their default values, but a grid search was used to find the most effective combination of learning rate (1e-5, 4e-5, 1e-4), warm-up ratio (0.02, 0.06, and 0.1), stack size (8, 16, and 32), number of epochs (1 to 10), and maximum sequence length (256, 384, 512).

Table 2 shows the balanced accuracy of the six classifiers. All are less effective in the multi-class setting than in the one-vs-rest settings and the transformer-based classifiers are clearly more effective than the feature-based ones; DeBERTa is best in the multi-class setting (accuracy of 73.63) and RoBERTa in the one-vs-rest ones (79.12 to 80.39).

Table 3: Effectiveness of spoiler type classification in the one-vs-one (phrase-vs-passage) setting on 826 test posts (training: 2,641; validation: 657).

Model	**Effectiveness**				
	TP	TN	FP	FN	Acc.
Naïve Bayes	298	256	147	125	67.07
SVM	311	264	139	112	69.61
Log. Regression	306	273	130	117	70.10
BERT	315	315	88	108	76.27
DeBERTa	318	335	68	105	79.06
RoBERTa	332	332	71	91	**80.39**

Table 3 shows the accuracy of the six classifiers on the 826 test posts with phrase and passage spoilers (almost balanced setup, since there is hardly any class imbalance). Again, the transformer-based classifiers clearly are more effective than the feature-based ones; with RoBERTa achieving the best accuracy of 80.39.

The substantial improvements of DeBERTa and RoBERTa over the feature-based classifiers in all settings (about 9–10 accuracy points) indicates that classifying the clickbait spoiler type requires more advanced language "understanding" than what is encoded in the basic features that the Naïve Bayes, SVM, or logistic regression classifiers used.

6 Evaluation of Spoiler Generation

To assess the effectiveness of the question answering and passage retrieval methods for clickbait spoiling, we evaluate both for their respective intended spoiler types, but each also for the respective other spoiler type. Multipart spoilers are deferred to future work. We continue to assume that prior clickbait detection (perfectly) identifies clickbait posts as such. Our evaluation of the generated spoilers includes quantitative and qualitative assessments (Section 6.1). In a pilot study with ten question answering and ten passage retrieval models at their default settings, two models in each category dominate the respective others (Section 6.2). The computationally expensive step of hyperparameter optimization is restricted to these four models plus two baselines (Section 6.3). Then, the effectiveness of spoiling clickbait posts dependent on spoiler type is evaluated (Sections 6.4 and 6.5), and compared to an end-to-end clickbait spoiling setup independent of spoiler type (section 6.6).

6.1 Quantitative and Qualitative Assessment

We introduce the measures used to evaluate generated spoilers and how we manually determined thresholds for them above which a generated spoiler is considered as "correct".

Evaluation measures. To assess the quantitative correspondence between a derived spoiler and the ground truth, we use three question answering-oriented and one passage retrieval-oriented measure: BLEU-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) in its extended version of Denkowski and Lavie (2014), BERTScore (Zhang et al., 2020), and Precision@1.

The three question answering-oriented measures each calculate a (penalized) harmonic mean of measure-specific definitions of precision and recall when comparing a generated spoiler to the ground truth. In case of BLEU-4, the overlap of word 1- to 4-grams is determined (if the length n of a generated spoiler is less than 4 words, we compute BLEU-n), in case of METEOR the overlap of word 1-grams, and in case of BERTScore the best matching embeddings of word pairs. Note that in their original formulation, BLEU-4 and METEOR penalize the score, the more the n-gram order differs. To arrange the measures on a spectrum from calculating predominantly syntactic (BLEU-4) to predominantly semantic similarity (BERTScore), we omit METEOR's penalization term.

The question answering-oriented measures are not really suited to assess the effectiveness of passage retrieval models since a retrieved passage is often longer than the ground truth spoiler. Therefore, we also use Precision@1 to measure whether the top-ranked passage contains the ground truth spoiler (all phrase spoilers and 98% of the passage spoilers come from a single passage; for the other passage spoilers, we consider all containing passages as relevant). To calculate the Precision@1 of question answering models, we use the first passage that contains the returned spoiler.

High-confidence thresholds. Candidates with higher scores on the question answering-oriented measures BLEU-4, METEOR, and BERTScore are closer to the ground truth. However, it is unclear what score threshold a particular spoiler candidate has to exceed so that it would be considered a true positive in a manual analysis. Determining such thresholds enables "high confidence" estimations of how many correct spoilers an approach generates without having to manually check its outputs each time with each new variant.

Table 4: Manually determined numbers of false positives/negatives (FP/FN) on 500 sampled clickbait posts with phrase spoilers and 500 with passage spoilers for question answering (top row group) and passage retrieval models (bottom row group), dependent on score threshold (Thresh.), spoiler type, and effectiveness measure (BL4 = BLEU-4, MET = METEOR, BSc. = BERTScore). The thresholds selected for subsequent assessment are indicated by bold FP/FN numbers.

Thresh.	Phrase Spoilers						Passage Spoilers					
	BL4		MET		BSc.		BL4		MET		BSc.	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
10%	11	11	18	7	238	0	5	44	168	15	399	0
20%	7	14	16	7	234	0	3	48	67	27	325	3
30%	7	14	14	9	165	1	**1**	**51**	31	35	134	21
40%	2	27	8	13	59	6	0	55	15	39	18	38
50%	**2**	**27**	2	28	24	14	0	60	9	42	5	51
60%	2	30	3	31	11	25	0	64	4	57	**1**	**59**
70%	1	33	**2**	**31**	6	36	0	66	**1**	**54**	0	66
80%	1	34	0	37	**1**	**40**	0	66	0	61	0	73
5%	8	40	28	64	208	0	**0**	**95**	225	10	355	0
10%	**4**	**104**	**8**	**108**	180	60	0	95	140	30	355	0
20%	0	184	0	164	44	144	0	95	35	65	305	15
30%	0	188	0	184	**0**	**176**	0	105	**5**	**90**	145	55
40%	0	188	0	188	0	188	0	115	5	105	20	95
50%	0	192	0	188	0	192	0	120	5	110	**5**	**105**
60%	0	192	0	192	0	192	0	125	0	120	5	130

In a pilot study, we thus determined such thresholds by running all question answering models (cf. Section 4.2 and 4.3) on a random sample of 500 clickbait posts with phrase spoilers and 500 with passage spoilers. For each post, a random spoiler generated by a question answering model and a random spoiler generated by a passage retrieval model were manually checked for whether they could be viewed as correct. Table 4 shows the number of manually determined false positives and false negatives for different thresholds of BLEU-4, METEOR, and BERTScore. The manually selected subjective thresholds (FP/FN in bold) for each combination of measure, spoiler type, and model type (question answering or passage retrieval) minimize the false positives at a rate where being more strict would incur too many false negatives. For instance, for phrase spoilers and BLEU-4, we set the question answering model threshold at 50% since a more strict threshold of 60% does not reduce the false positives but increases the false negatives.

In addition to reporting quantitative mean effectiveness scores, applying the determined thresholds helps to estimate how many of the spoilers of a

Table 5: Pilot study spoiling effectiveness of question answering and passage retrieval models on 200 validation posts (models ordered lexicographically). The bracketed numbers indicate the expected number of true positives as per our pre-determined high-confidence score thresholds; P@1 is the Precision@1. The models DeBERTa-large and RoBERTa-large, as well as MonoBERT and MonoT5 are the most effective in their groups.

Type	Model	Phrase Spoilers (n = 97)				Passage Spoilers (n = 103)			
		BLEU-4	METEOR	BERTScore	P@1	BLEU-4	METEOR	BERTScore	P@1
Question Answering	ALBERT	63.82 (50)	55.97 (49)	74.07 (46)	63.64	24.51 (33)	38.42 (27)	44.61 (24)	38.71
	BERT-cased	60.27 (49)	58.87 (47)	73.55 (44)	59.09	17.65 (22)	28.09 (20)	40.30 (16)	27.96
	BERT-uncased	62.36 (49)	53.17 (47)	75.87 (47)	60.23	18.05 (22)	32.50 (20)	39.86 (18)	32.26
	Big Bird	69.21 (55)	64.80 (54)	77.39 (49)	63.64	23.89 (30)	36.20 (28)	44.55 (27)	43.01
	DeBERTa-large	70.19 (57)	65.08 (56)	78.02 (50)	65.91	29.52 **(38)**	43.72 **(36)**	49.63 **(37)**	**48.39**
	ELECTRA	69.10 (55)	65.97 (53)	79.26 (51)	65.91	25.78 (32)	39.87 (30)	46.64 (27)	43.01
	Funnel-Transf.	68.31 (54)	63.89 (53)	78.78 (51)	64.77	28.59 (36)	40.95 (32)	47.93 (29)	40.86
	MPNet	72.92 (58)	65.90 (57)	80.26 (55)	69.32	30.16 (36)	40.68 (35)	50.07 (32)	40.86
	RoBERTa-base	73.02 (59)	65.56 (57)	80.39 (54)	65.91	27.61 (35)	41.55 (35)	48.76 (30)	44.09
	RoBERTa-large	79.47 **(66)**	78.61 **(61)**	84.04 **(58)**	**70.45**	29.58 (35)	43.49 (32)	48.65 (32)	44.09
Passage Retrieval	BM25	3.49 (10)	3.67 (10)	17.73 **(2)**	5.68	11.49 (22)	22.64 (21)	36.80 (12)	9.68
	BM25+Ax	3.39 (10)	3.57 (9)	18.07 **(2)**	5.68	11.27 (21)	22.46 (19)	36.51 (12)	9.94
	BM25+PRF	3.25 (10)	3.21 (9)	18.03 **(2)**	5.13	9.68 (20)	21.10 (17)	35.44 (11)	8.84
	BM25+RM3	3.43 (10)	3.62 (9)	17.14 **(2)**	5.13	10.06 (21)	21.03 (20)	35.56 (11)	8.84
	MonoBERT	3.42 **(11)**	4.13 **(12)**	18.32 (1)	**32.95**	14.55 **(29)**	26.86 (25)	38.10 (15)	**31.18**
	MonoT5	3.16 (9)	4.19 (11)	18.30 (0)	31.82	14.27 **(29)**	26.70 **(26)**	38.94 **(17)**	29.03
	QLD	2.51 (7)	2.69 (7)	17.24 (0)	12.50	10.94 (25)	17.80 (18)	36.70 (11)	19.35
	QLD+Ax	2.61 (7)	2.71 (7)	17.10 (0)	12.50	9.68 (20)	17.84 (18)	36.68 (11)	8.84
	QLD+PRF	2.60 (7)	2.70 (7)	17.13 (0)	11.94	10.86 (25)	17.52 (18)	36.46 (11)	17.67
	QLD+RM3	2.41 (7)	2.54 (7)	16.97 (0)	11.39	10.66 (25)	17.54 (18)	36.13 (11)	17.12

model would be perceived as "good" by human readers. This corresponds to a conservative assessment, since we believe that a model should only be deployed to production if it has been tuned to not return a spoiler if in doubt about its correctness; also probably somewhat minimizing the otherwise possible spread of auto-generated misinformation.

6.2 Pilot Study for Model Selection

In a pilot study on 1,000 clickbait posts (800 training, 200 validation), we compare ten question answering and ten passage retrieval models (cf. Table 5) at their default settings to select models for subsequent experiments with more extensive (and expensive) hyperparameter tuning. The question answering models were or are among the most effective in the SQuAD and TriviaQA question answering benchmarks. In our setup, they return a piece of text from the linked document as an "answer" to the clickbait post as the "query". As passage retrieval models, we empoly MonoBERT and MonoT5 using their PyGaggle[3] implementations, and eight variants of the popular baseline retrieval models BM25 and QLD using their Anserini implementations (Yang et al., 2017). These models return the most "relevant" paragraph from the linked document for the clickbait post as the "query".

[3] https://github.com/castorini/pygaggle

Using Nvidia A100 GPUs, the question answering models were first fine-tuned on SQuAD v1.1 and then on the pilot training data. This was the most effective setup from an ablation study with other fine-tuning regimes (e.g., the phrase spoiler BERTScore for RoBERTa-large dropped from 84.04 to 69.91 when only fine-tuned on our pilot study data, to 64.61 when only fine-tuned on SQuAD, and to 46.60 without fine-tuning). Interestingly, the models' SQuAD effectiveness does not predict their spoiling effectiveness (e.g., RoBERTa-base and FunnelTransformer were tied on SQuAD, but RoBERTa-base is more effective at spoiling). This indicates the importance of the pilot study.

Table 5 shows the pilot study effectiveness of all models on the 200 validation posts. RoBERTa-large (for phrasal spoilers) and DeBERTa-large (for passage spoilers) are the most effective. Among the passage retrieval models, MonoBERT and MonoT5 achieve the best scores. Contrary to our original assumption that passage retrieval models might be particularly well-suited to identify passage spoilers, MonoBERT and MonoT5 have similar Precision@1 scores on both phrase and passage spoilers and are substantially less effective than the best question answering models (e.g., DeBERTa-large has a Precision@1 of 48.39 for passage spoilers compared to 31.18 for MonoBERT).

Table 6: Effectiveness on the 826 test clickbait posts with phrase and passage spoilers. The bracketed numbers indicate the expected number of true positives as per our pre-determined high-confidence score thresholds; P@1 is the Precision@1. Overall, DeBERTa-large and RoBERTa-large are the most effective models.

Type	Model	Phrase Spoilers (n = 423)				Passage Spoilers (n = 403)			
		BLEU-4	METEOR	BERTScore	P@1	BLEU-4	METEOR	BERTScore	P@1
Question Answering	BERT (baseline)	58.89 (257)	56.75 (266)	71.06 (215)	66.67	21.59 (110)	35.49 (100)	44.38 (109)	42.43
	DeBERTa-large	**68.80 (300)**	**67.93 (298)**	**77.03 (250)**	**75.65**	**31.44 (157)**	**46.06** (142)	**51.06** (161)	**54.84**
	RoBERTa-large	65.70 (290)	66.15 (293)	74.81 (233)	72.58	29.61 (148)	45.20 **(145)**	49.99 **(167)**	53.85
Passage Retrieval	BM25 (baseline)	3.40 (55)	5.06 (83)	19.94 (12)	8.27	7.91 (53)	20.19 (61)	34.71 (42)	4.22
	MonoBERT	4.20 (72)	6.12 (103)	20.66 (11)	42.08	10.43 (74)	22.37 (75)	36.58 (46)	26.05
	MonoT5	4.95 (82)	6.47 (115)	20.98 (16)	43.97	10.58 (74)	22.02 (74)	36.70 (46)	29.03

6.3 Tuning the Selected Models

Given the pilot study results, six models are selected for a more extensive hyperparameter tuning: the best two question answering models (DeBERTa-large was best for phrase spoilers, RoBERTa-large for passage spoilers) plus BERT as baseline, as well as the best two passage retrieval models (MonoBERT and MonoT5) plus BM25 as baseline.

As the ablation study in our pilot study showed that fine-tuning the question answering models on SQuAD first and then on our corpus works best, we apply this fine-tuning regime to DeBERTa-large, RoBERTa-large, and BERT using the clickbait spoiling training data (depending on the experiment, either only the phrase spoilers, only the passage spoilers, or both combined). Most hyperparameters of DeBERTa-large, RoBERTa-large, BERT, MonoBERT, and MonoT5 are left at their defaults, but a grid search is run to find the most effective combination of learning rate (1e-5, 4e-5, 1e-4), warmup ratio (0.02, 0.06, 0.1), batch size (8, 16, 32), number of epochs (1 to 10), and maximum sequence length (256, 384, 512). For BM25, we try combinations of k_1 from 0.1 to 0.4 and b from 0.1 to 1.0 with a step size of 0.1.

6.4 Effectiveness on Phrase Spoilers

The 'Phrase Spoilers' column group in Table 6 shows the effectiveness of the selected question answering and passage retrieval models on the 423 test clickbait posts with phrase spoilers. Given the ground-truth spoiler, we report the predicted spoilers' average BLEU-4, METEOR, BERTScore, and Precision@1 (using 1,367 posts with phrase spoilers for training and 335 posts for validation to tune the hyperparameters; cf. Table 1).

Overall, DeBERTa-large is the most effective model for phrase spoilers. Based on our high-confidence score thresholds, it generates the correct spoiler for 250–300 of the 423 test posts (i.e., for about 60–70% of the cases) according to a BERTScore or BLEU-4 evaluation. Similar to our pilot study, the passage retrieval models are comparably ineffective in identifying phrase spoilers. Among them, MonoT5 achieves the highest scores but is even substantially less effective than the question answering baseline BERT. For instance, with a BLEU-4 of 58.89 and probably 257 correct spoilers (61% of the 423 test posts), BERT is way ahead of MonoT5 with a BLEU-4 of 4.95 and only 82 probably correct spoilers (19% of the 423 posts).

6.5 Effectiveness on Passage Spoilers

The 'Passage Spoilers' column group in Table 6 shows the effectiveness of the selected passage retrieval models on the 403 test clickbait posts with passage spoilers (using 1,274 and 322 posts for training and validation). The numbers of probably correct spoilers are lower for all models compared to the phrase spoilers (even the higher amount of probably correct passage spoilers of the passage retrieval models according to their BERTScore threshold are still worse than the estimated probably correct phrase spoilers according to BLEU-4 or METEOR). Similar to the pilot study, all question answering models are also substantially more effective on passage spoilers than the passage retrieval models. Overall, DeBERTa-large and RoBERTa-large achieve the highest Precision@1 scores and the highest amount of probably correct passage spoilers (about 35–41% of the passage spoilers are correctly identified according to our high-confidence thresholds).

6.6 Effectiveness of the End-to-End System

We evaluate the entire spoiling pipeline using all 826 phrase and passage test posts by comparing two-step pipelines that first classify the spoiler type

Table 7: End-to-end effectiveness on the 826 phrase and passage test posts. Spoiling models that classify the spoiler type to then select an appropriately trained spoiler model ('Classif.', using the most effective spoiler type classifier), models without spoiler type classification ('None'), and unrealistic models with perfect-accuracy type classification ('Oracle').

	Model	End-to-End Effectiveness			
		BLEU-4	METEOR	BERTScore	P@1
Classif.	BERT	35.95 (311)	34.25 (303)	53.86 (294)	52.66
	DeBERTa	44.98 (392)	44.32 (377)	59.18 (378)	63.44
	RoBERTa	42.70 (374)	43.23 (356)	58.01 (361)	61.86
None	BERT	38.85 (346)	37.80 (330)	54.60 (314)	55.33
	DeBERTa	**46.16 (409)**	**47.01 (407)**	**60.43 (382)**	64.16
	RoBERTa	44.69 (400)	44.72 (395)	59.51 (375)	**65.13**
Oracle	BERT	40.69 (367)	39.02 (366)	58.05 (324)	54.84
	DeBERTa	**50.58 (457)**	**49.40 (440)**	**64.36 (411)**	**65.50**
	RoBERTa	48.10 (438)	48.57 (438)	62.71 (400)	63.44

to then select an appropriately trained spoiler model (trained on the respective type) and single-step approaches that skip the spoiler type classification and simply run the same spoiler model on all posts (trained on the complete training data). For the two-step pipelines, we experiment with two variants: (1) using an artificial classifier that returns perfect oracle-style answers about a post's type, and (2) using the best RoBERTa-based phrase-vs-passage classifier from Section 5.

Since the passage retrieval models were less effective in our spoiler experiments (cf. Table 6), we report results only for pipelines with question answering models. In the two-step pipelines the respective question answering models are fine-tuned on the respective spoiler types, in the single-step approach on the combined training data.

Table 7 shows the achieved end-to-end effectiveness values. The individual two-step pipelines with oracle type classification (row group 'Oracle') are substantially more effective than their single-step counterparts without type classification (row group 'None') that again are more effective than the respective two-step pipelines with "real" RoBERTa-based type classification (row group 'Classif.'). Overall, the DeBERTa pipeline with oracle classifier achieves an estimated amount of about 50–55% correctly spoiled posts (i.e., 411 to 457 of 826). This result confirms that classifying the required spoiler type can be beneficial for clickbait spoiling. Still, among the currently realistically applicable end-to-end spoiling approaches (with RoBERTa type classification or without spoiler type classification), the one-step DeBERTa approach without spoiler type classification is the most effective according to the number of probably correctly spoiled posts (382 to 409 of the 826 posts, i.e., 46–50%). This indicates that the currently best RoBERTa-based spoiler type classifier with its accuracy of 80.39% is still not good enough to result in an end-to-end system that actually benefits from spoiler type classification.

Our results show that effectively spoiling clickbait with question answering models is possible in practice but also that there is still room for improvements (e.g., improved spoiler type classification, improved spoiler generation for the individual types, and taking multipart spoilers into account).

7 Conclusion

Clickbait spoiling is a new task to help social media users who do not want to be manipulated into falling for clickbait links. Unlike clickbait detection, which often involves filtering out clickbait posts from users' timelines, clickbait spoiling subverts the curiosity triggered by clickbait, presenting users with the withheld "punchline" in advance.

We compile the first large resource for clickbait with associated spoilers. By interpreting clickbait spoiling as either a question answering task or a passage retrieval task, many possible approaches are available to extract from the linked document of a clickbait post the phrase or passage that spoils it. We have explored the effectiveness of a number of state-of-the-art solutions for both tasks in a large-scale experiment, including fine-tuning the respective models on our resource to determine their effectiveness for type-specific clickbait spoiling. Our experimental setup considers type-specific spoiling on the one hand, but on the other hand it also includes an end-to-end configuration for comparison. Overall, our results show that type-agnostic question answering-based spoiling is the most effective yet, but also that spoiler type-specific solutions have the potential to outperform them.

In addition to the possibilities explored, there might also be other approaches to clickbait spoiling: for example, paraphrasing technology could be used to directly transform a clickbait post into a version that contains its own spoiler. With respect to multipart spoilers, the use of summarization models could be an interesting direction to select the different parts of the linked document of a clickbait post that make up its multipart spoiler.

Acknowledgement

We thank Tim Gollub, and our students Jana Puschmann and Bagrat Ter-Akopyan, who helped to create earlier versions of the dataset.

Ethics Statement

The spread of clickbait on social media by news publishers to promote click-through to their websites has been empirically found to decrease their perceived credibility in readers (Molyneux and Coddington, 2020). There is, of course, nothing wrong with monitoring and optimizing the effectiveness of marketing a newly published news article, especially in cases where the editors make an honest effort to reach and inform their target audience. But the clickbait in our corpus mostly spreads trivial facts that could have been easily fitted into the length limits of a social media post, which is why we consider these posts to fall short of the journalistic ideal. However, it is as of yet unclear, in terms of journalism ethics, whether clickbait is an acceptable means to an end for publishers (i.e., whether it is "necessary in driving audiences to the journalism they need by giving them the journalism they seem to want."), or whether it serves to "crowding out «real» journalism by reducing quality in favor of the need for a click-through at whatever cost" (Harte, 2021).

Facebook intervened twice with algorithmic filters to reduce the amount of clickbait that people are exposed to in their timelines—even though this probably also lowered Facebook's user engagement metrics. Our technology demonstrates another, complementary way of relatively simply circumventing the purported exploitation of the curiosity gap by giving the audience a choice on whether or not they wish their cognitive "loopholes" to be exploited. If a sufficiently large portion of people decide to adopt spoiling tools, that would send a clear message to publishers and social media platforms alike. Spoiling clickbait, as opposed to removing it, however, still gives publishers the benefit of the doubt, since, as the publishers claim, there are people who enjoy these kinds of trivia.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

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 2016, San Francisco, CA, USA, August 18-21, 2016*, pages 9–16.

Christopher Clark and Matt Gardner. 2018. Simple and Effective Multi-Paragraph Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 845–855. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Zihang Dai, Guokun Lai, Yiming Yang, and Quoc Le. 2020. Funnel-Transformer: Filtering out Sequential Redundancy for Efficient Language Processing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Michael J. Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 376–380. The Association for Computer Linguistics.

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.

Daria Dzendzik, Jennifer Foster, and Carl Vogel. 2021. English Machine Reading Comprehension Datasets: A Survey. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8784–8804. Association for Computational Linguistics.

Lars Eidnes. 2015. Auto-Generating Clickbait With Recurrent Neural Networks. https://web.archive.org/web/20220223161935/http://larseidnes.com/2015/10/13/auto-generating-clickbait-with-recurrent-neural-networks/.

Khalid El-Arini and Joyce Tang. 2014. News Feed FYI: Click-baiting. http://web.archive.org/web/20150529104738/http://newsroom.fb.com/news/2014/08/news-feed-fyi-click-baiting/.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2020. A Deep Look Into Neural Ranking Models for Information Retrieval. *Inf. Process. Manag.*, 57(6):102067.

David Harte. 2021. Clickbait and Banal News. In *The Routledge Companion to Journalism Ethics*, pages 346–353. Routledge.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Sanjeev Kumar Karn, Mark Buckley, Ulli Waltinger, and Hinrich Schütze. 2019. News Article Teaser Tweets and How to Generate Them. 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 3967–3977. Association for Computational Linguistics.

Peter Koechley. 2012. Why The Title Matters More Than The Talk. http://web.archive.org/web/20150611110506/http://blog.upworthy.com/post/26345634089/why-the-title-matters-more-than-the-talk.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Juliane A Lischka and Marcel Garz. 2021. Clickbait News and Algorithmic Curation: A Game Theory Framework of the Relation between Journalism, Users, and Platforms. *New Media & Society*.

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.

George Loewenstein. 1994. The Psychology of Curiosity: A Review and Reinterpretation. *Psychological Bulletin*, 116(1):75–98.

Logan Molyneux and Mark Coddington. 2020. Aggregation, Clickbait and Their Effect on Perceptions of Journalistic Credibility and Quality. *Journalism Practice*, 14(4):429–446.

Kevin Munger. 2020. All the News That's Fit to Click: The Economics of Clickbait Media. *Political Communication*, 37(3):376–397.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR*, abs/1901.04085.

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, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 708–718. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *CoRR*, abs/1910.14424.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Alex Peysakhovich and Kristin Hendrix. 2016. Further Reducing Clickbait in Feed. https://web.archive.org/web/20210207042429/https://about.fb.com/news/2016/08/news-feed-fyi-further-reducing-clickbait-in-feed/.

Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 275–281. ACM.

Martin Potthast, Tim Gollub, Matthias Hagen, and Benno Stein. 2018. The Clickbait Challenge 2017: Towards a Regression Model for Clickbait Strength. *CoRR*, abs/1812.10847.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait Detection. In *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 2016)*, volume 9626 of *Lecture Notes in Computer Science*, pages 810–817, Berlin Heidelberg New York. Springer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Md M. U. 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, Sydney, Australia, July 31 - August 03, 2017*, pages 232–239. ACM.

Victoria Rubin, Niall Conroy, and Yimin Chen. 2015. Towards News Verification: Deception Detection Methods for News Discourse. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS48) Symposium on Rapid Screening Technologies, Deception Detection and Credibility Assessment Symposium*, Kauai, Hawaii, USA.

Kai Shu, Suhang Wang, Thai Le, Dongwon Lee, and Huan Liu. 2018. Deep Headline Generation for Clickbait Detection. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 467–476. IEEE Computer Society.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Peng Xu, Chien-Sheng Wu, Andrea Madotto, and Pascale Fung. 2019. Clickbait? Sensational Headline Generation with Auto-tuned Reinforcement Learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3063–3073. Association for Computational Linguistics.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1253–1256. ACM.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.