Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 515–523
Marseille, 20-25 June 2022

Know Better – A Clickbait Resolving Challenge

Benjamin Hättasch, Carsten Binnig
Technical University of Darmstadt (TU Darmstadt)
Germany
{benjamin.haettasch, carsten.binnig}@cs.tu-darmstadt.de

Abstract

In this paper, we present a new corpus of clickbait articles annotated by university students along with a corresponding shared task: clickbait articles use a headline or teaser that hides information from the reader to make them curious to open the article. We therefore propose to construct approaches that can automatically extract the relevant information from such an article, which we call *clickbait resolving*. We show why solving this task might be relevant for end users, and why clickbait can probably not be defeated with *clickbait detection* alone. Additionally, we argue that this task, although similar to question answering and some automatic summarization approaches, needs to be tackled with specialized models. We analyze the performance of some basic approaches on this task and show that models fine-tuned on our data can outperform general question answering models, while providing a systematic approach to evaluate the results. We hope that the data set and the task will help in giving users tools to counter clickbait in the future.

Keywords: Information Extraction, Question Answering, Corpus, Summarization, Natural Language Generation, Clickbait, Shared Task

1. Introduction

Nearly everyone knows headlines like "The hidden secret of Kermit the Frog" or "15 hacks that will change your life". If you open the article, you will usually find completely trivial and well-known information, or you will find that the headline was completely exaggerating or misleading. In short, you have fallen for clickbait—a term that refers to a certain style of a headline or other teaser text designed to "bait" the reader into clicking a link to a full article.

Unfortunately, clickbait is annoying but effective. Just as with tabloid headlines, people's curiosity is exploited to get them to open the article and read it. And since their curiosity is so strong, they spend time reading the articles—-even though they usually know very well that the article is clickbait and that the "shocking" news will turn out to be unsurprising in the end. A good analysis of how marketing experts use the fear of missing out on information (Loewenstein, 1994) and turn this into a curiosity gap by intentionally making headlines really irresistible, can be found in the recent work by Scott (2021). She identifies patterns in these headlines and shows that, e.g., certain groups of adjectives are used significantly more often in clickbait headlines than in general.

Websites that use such techniques usually focus on attracting people to the page and, in the best case, making them stay as long as possible in order to "play out" as many ads as possible. As such, clickbait is a billion-dollar business today and thus many more advanced techniques are being developed. For example, with the help of social media and article recommender systems, owners of these websites "made for advertising" try to get the largest possible share from the estimated $ 115 billion that will be spent for displaying ads in the US in 2022 alone (Barwick, 2021). In many of these cases, clickbait is not used to (legitimately) refinance the costs of running a news page or a social network, but these websites are solely built to generate profit by tricking their users.

Sadly, clickbait not only causes people to waste time. A study by Gabielkov et al. (2016) shows that almost 60 percent of the links shared on social media were not clicked before being shared. Clickbait articles with provocative titles (containing, e.g., exaggerations or rhetoric questions) might spread false information to those who do not read the full articles even if the misinformation is corrected in the article itself at some point (which might be page 30 of an image gallery with ads on every other page). Sometimes, clickbait headings do not point to an article at all but to webpages distributing malware or collecting personal data (e.g., with fake sweepstakes).

So can the NLP community help reducing the amount of clickbait on the web? Current scientific work on clickbait mainly focuses on detecting clickbait using linguistic analyses of headlines, learned models for classification, or regression approaches to determine the degree to which an article is clickbait. This information is then used to hide clickbait articles from webpages and timelines. Completely hiding contents is effective—but like with every other filter approach, there might be false positives. Simply marking articles as clickbait but still showing them solves the issue of filtering out important articles, but may not be sufficient, since many people open clickbait articles despite knowing they are clickbait. We therefore suggest working on approaches that can automatically extract the teased information from the article text and thus fill the curiosity gap by displaying this information next to the headline without requiring the user to click the link.

Contributions In this paper, we hence present a dataset and a shared task to *complement* existing approaches on *clickbait detection* to enable the development of new approaches that will allow users to better deal with clickbait. To be more precise, we present a corpus of clickbait samples (titles/teasers and texts) together with their resolutions, which were annotated by university students. We define a *clickbait resolving* task based on the dataset and will maintain a leaderboard for approaches submitted to that task.

To allow a direct application of the resulting approaches, submissions have to provide/implement an interface that allows running the model on new data (i.e., resolving a given clickbait article). We also expect authors to open source their code and pre-trained models (after they successfully published their work). The leaderboard, evaluation scripts and other code, instructions how to submit and obtain the corpus (including additional silver data), and baseline implementations including pre-trained models can be found at: `https://link.tuda.systems/clickbait-resolving-challenge`

Finally, it is important to note that this paper is meant as a starting point. We hope that the resulting approaches (e.g., browser plugins or the adaption by social networks) will ultimately reduce the amount of clickbait articles and additionally help to increase media literacy.

Outline We first give an overview of related work, previous tasks, and datasets related to clickbait in Section 2. Afterwards, we describe the construction process (Section 3 and the properties of our corpus (Section 4). Then we define the task and important metrics in Section 5. Finally, we analyze the usefulness of our data using several baseline approaches (Section 6) before wrapping up our contribution in Section 7.

2. Related Work & Existing Datasets

"The term *clickbait* refers to social media messages that are foremost designed to entice their readers into clicking an accompanying link to the posters' website, at the expense of informativeness and objectiveness" (Potthast et al., 2018a). Early work by Vijgen (2014) and Blom and Hansen (2015) studied the phenomenon from a linguistic perspective and detected homogeneous structures (e.g., headlines starting with a number leading to *listicles*, i.e., articles only consisting of long lists or image galleries) and the use of certain patterns and expressions (e.g., "This will blow your mind.").

Current work in this area mostly focuses on *detecting* clickbait articles to warn users or hide those headlines and teasers from them. Agrawal (2016) presented a convolutional neural network for classification whether headlines are using clickbait techniques or not. Chakraborty et al. (2016) evaluated different techniques to create such a model, too. Additionally, they propose to ask users to mark contents they perceive as clickbait and infer from that to block similar contents. Finally, they integrated this approach into a browser extension that can mark and hide clickbait contents on several media sites and ran a field study. Rony et al. (2017) trained embeddings for classification based on a large corpus of social media posts.

To evaluate all these and some other approaches, different corpora containing clickbait articles were created by the authors, a good overview of these corpora can be found in Potthast et al. (2018b). In that paper, the authors describe how they constructed a new corpus of clickbait teasers based on Twitter posts for the *clickbait challenge 2017* (Potthast et al., 2018a). That challenge is the first that phrases the problem as a regression task which tries to assign a score for the *strength* of clickbait. It received 13 submissions during the original shared task period, but is still open for further submissions. The best scoring submission was able to improve on the F1 score by nearly 20 percentage points compared to the baseline.

After these classification and regression approaches, we now propose to go one step further by creating models that *generate or extract text*. Our task is related to several NLP disciplines: one of them are topic-focused single document summarization approaches, that try to extract the most important parts of a text with regard to a certain topic (the so-called content selection). A systematic evaluation of such approaches was already performed in the DUC2005 challenge (Dang, 2006), more recent ways to frame this task were, e.g., proposed by Narayan et al. (2018) or Deutsch and Roth (2019).

Moreover, our task can be seen as a specialized question answering (QA) task, particularly as textual QA which aims to answer a given question based on unstructured textual documents. That field in turn integrates with neural machine reading comprehension (MRC). Traditional approaches tried to tackle the problem using different components dealing with question analysis, classification, document retrieval and answer extraction but are nowadays mostly replaced by neural end-to-end models. A good overview of existing approaches in these fields can be found in the recent paper by Zhu et al. (2021), an extensive analysis of transformer based language models and their preparation for different downstream task was recently presented by Kalyan et al. (2021).

Even though the general task description (find a resolution to a short text snippet in a longer text) matches the one of textual question answering, there are differences: Teasers might be formulated as (rhetoric) questions but usually do not have a question format, and they may contain certain expressions like "will change your life" that are not actually useful to find the resolution. Furthermore, the presented texts do not always contain a real resolution or the resolution may at least not match the detail level promised in the teaser. We will further discuss these challenges in the next sections. Taking this into account, we think it is reasonable

to consider clickbait resolving as a task that should be tackled with dedicated approaches.

3. Corpus Construction

The two most straightforward approaches to construct a corpus of clickbait articles and their resolutions are a) annotating clickbait articles with the resolution manually or b) finding combinations of articles and resolutions and manually checking and confirming them. We decided to go with the second way, hoping that this will result in a higher linguistic variability of the resolutions since they are written by lots of different authors, and also a higher quality since the authors wrote them with the intrinsic motivation of helping other people. This however required finding suitable data sources as well as a careful checking of the resulting samples.

We evaluated multiple possible sources, starting with Twitter accounts like *@SavedYouAClick* and *@WeHateClickbait* that post resolutions to clickbait articles. Unfortunately (at least for our use case) they often do only include a screenshot of the headline but no link to the original source, the overall amount of tweets is limited to a few hundred tweets, or many of the tweets deal with other topics.

Therefore, we settled on the Subreddit *Saved you a click* (not related to the Twitter account) which was created in June 2014 and has nearly 1.8 million members. From April to September 2021, we downloaded 4870 posts containing article links from that Subreddit and determined Web Archive links for the URLs if necessary.

We then crawled and parsed the pages to extract full article texts. On purpose, we did not remove sentences like "Get the latest from xyz Sign up for our newsletter" since they will be contained when retrieving page contents in real world usage, too. However, it might be interesting to train or adapt models that remove these parts and detect the *real* content automatically, and one might use them as part of your processing pipeline when submitting to the shared task (see Section 5 for details. We will curate a public list of preprocessing models and other systems working on/with the data on the webpage for the challenge, and welcome submissions to this independently of a submission to the task itself.

To compile a corpus out of the raw data, university students manually checked and annotated the data following a list of guidelines. They first checked whether the resolution from Reddit is suitable to answer the kind of question raised in the teaser and afterwards determined their correctness based on teaser, resolution and the full text. Thereby, they also determined whether the text contains the relevant information to produce the proposed resolution. More than half of the samples had to be discarded in this step due to quality problems. Even though we wanted to keep the linguistic variability high, the students were advised to reformulate the resolutions in some cases, e.g., to remove long sequences of exclamation marks, sarcastic comments regarding the articles beyond the resolution, and other additions like the amount of clicks the author needed to get to the teased information. Also, for articles which could simply be summarized with "yes" or "no", we marked this as additional information and replaced resolutions like "Nope" or "Yeah".

Finally, we split the resulting samples into train, dev and test set.

As a result of our approach, we created a corpus with articles from many different sources, which contains resolutions written by different authors and consists of manually confirmed entries only. The input texts correspond to what one could expect when running a crawler on an arbitrary page to resolve the clickbait for a user that just spotted the headline of that article.

4. Dataset Statistics

Our corpus consists of 2635 samples for English clickbait articles with their resolutions. It is split into a public training set (2108 samples / $80\,\%$), a public dev set (264 samples / $10\,\%$) and a test set (263 samples / $10\,\%$) that we keep private for evaluation.

For each sample, we provide a title/teaser, the article text and the resolution, as well as some meta information: the URL to the full article, a timestamp when the resolution was created, and the score the resolution post on Reddit achieved (which might hint on the quality of the uncleaned answer but is also influenced by the overall interest in the topic and other subjective factors). Finally, we manually annotated whether a clickbait headline is hinting at a simple yes/no answer and normalized the resolution for those cases. This applies to about $3.9\,\%$ of the samples.

The average title has a length of 68.7 characters or 13.4 words. The texts are on average 3582.2 characters or 716.6 words long. They were written in the years 2016 to 2021.

5. Task definition

On top of the proposed dataset, we define a task for finding resolutions to clickbait articles, which is evaluated using the public dev and private test set. We first discuss the metrics used for evaluation, and afterwards describe details for task and submission.

5.1. Metrics

Already the first shared task on question answering (Voorhees and Tice, 2000) raised the question whether metrics known from the field of information retrieval are suitable for this kind of task (i.e., really resemble human judgement of correctness and equivalence). This is particularly a problem for free form question answering, where simple metrics like precision or recall cannot be employed directly but the semantic equivalence of strings has to be (automatically) evaluated to estimate the correctness of an answer. Like in many tasks of the NLP community, this is a

hard problem, e.g., due to ambiguity, synonyms, and context-depending meanings. Even human annotators might disagree, particularly because of different previous knowledge or different interpretations of the question. Another problem is different levels of granularity, which poses in fact a typical issue with clickbait: the headline promises a detailed answer, but the article then just presents somewhat common knowledge or things that could be easily guessed (e.g., "you won't believe what the other kids call Prince George" and the answer is just "George"). But what effect does this have on the evaluation of resolution correctness—should answers of another detail level be treated as similar or not?

These difficulties lead to the development of a range of metrics with different properties: metrics working only on the syntactical level might both under- and overestimate similarity (e.g., sentences consisting of synonyms will get a low similarity score, but sentences like "They said yes" and "They said no" or "We did start the fire" and "We did not start the fire" will get high scores even though they express the absolute opposite). Trained models might be able to better capture semantic meanings, but this highly depends on the data they were trained on, and it still cannot be guaranteed that the contexts are correctly interpreted and that the background knowledge incorporated in the language model is valid for this specific pair of texts.

Recent papers like Chen et al. (2019) and Si et al. (2021) evaluated different classic and transformer based metrics, but found that none of them can resemble human judgement in every case. We therefore decided to measure and report multiple metrics in our task at once. That way, we can take different aspects of similarity (e.g., syntactic equivalence and semantic correspondence) into account. Potential users of the resulting models can then choose the model to use for their application based on these aspects.

The evaluation of the task will use the following metrics:

Exact Match This metric measures whether the predicted resolution matches the human written one character by character.

Recall-Oriented Understudy for Gisting Evaluation ROUGE (Lin, 2004) was developed to evaluate the quality of a summary by comparing it to human created gold summaries. There are different variants: ROUGE-N measures the n-gram recall, precision and F1 score between a text and the gold standard. We use ROUGE-2 (based on bigrams) and report the F1 score. We also report the ROUGE-L F1 scores, which are based on the longest common subsequence.

Bilingual Evaluation Understudy BLEU (Papineni et al., 2001) was developed to score the similarity between machine and human translations of a given text but may also be used to evaluate text similarity in general. This metric was one of the first to show a high correlation with human judgment. It is precision-based, uses cumulative n-grams, and works best if there are multiple reference translations (which is unfortunately not the case for our data). We use BLEU-2 which incorporates the unigram and bigram overlap in one single score.

Metric for Evaluation of Translation with Explicit ORdering METEOR (Banerjee and Lavie, 2005) again was originally designed for evaluation of machine translation. It works on unigram level but allows generalization by taking not only the surface forms but also stemmed forms and meanings into account. METEOR creates an alignment between the tokens of the texts to compare, scores that alignment using precision and recall, and combines these scores in an F-measure with a higher weight on recall.

BERTScore BERTScore (Zhang* et al., 2020) was developed as a robust automatic metric for text generation. Similar to the previous metrics, it scores the similarity of the tokens in candidate and reference texts. But to do so, it uses the cosine similarity between pre-trained contextual BERT embeddings instead of the surface forms and sums them up to a single score. Studies show that this score better aligns with human judgment than other metrics in many cases.

5.2. Task Details

A submission to the *clickbait resolving challenge* should produce resolutions for given texts and teasers of clickbait articles. We do not give constraints on the implementation. Both generative models, which produce a new text to do this, and extractive models, which extract part of the original text, can be used. There is also no specification for the maximum length of the resolution, but since the expected resolutions are always only a few words to a few sentences at most, unnecessarily long predictions will automatically result in poor scores. The meta-information described in Section 4 can also be taken into account, but the approach should be robust to a lack of certain information (e.g., the Reddit score). It is allowed to use other resources (e.g., ontologies, models for pre- and post-processing). Approaches that access online resources to produce the resolution (e.g, API calls to lexical resources) are listed in a separate leaderboard.

As described in Section 3, we call for supporting models and approaches (e.g., for pre-processing) to be submitted to us as well, so that we can promote them prominently on the project page.

In a recent paper (Hättasch et al., 2021) we discussed the importance of not only reproducibility but applicability of results from shared tasks. We therefore require the implementation of a small python interface that can be used to predict resolutions for single or multiple new samples. That interface should be published together with code and pre-trained model dumps (if necessary).

5.3. Leaderboard & Submission

All details regarding evaluation and submission procedure can be found on the project website.
We publish an evaluation script that can be used to evaluate an approach on the dev set. The test set is kept privately by us and used to finally evaluate submissions. Each approach will be evaluated on the test set only once.
We will maintain a leaderboard for models/approaches for our task as part of the webpage. It will report both the dev and test scores for all submitted approaches. Submissions under review may show up as anonymous on the board, but we place great value on reproducibility. It is therefore required to open source code, model dumps and the above-mentioned code snippet to generate new predictions to stay in the leaderboard once an approach was successfully published somewhere.

6. Experimental Results & Discussions

6.1. Baselines

To prove the usefulness of the data for the proposed task but also show that the task cannot be trivially solved with existing approaches, we evaluated our data using the following approaches:

First & Last Sentence As two trivial baselines, we extract the first or last sentence of an article and treat that sentence as resolution. This approach neither uses the training data nor incorporates the teaser information and is mainly used to "calibrate" the scores.

Longformer2Roberta Summarization[1] This is a EncoderDecoder model based on Longformers[2] and the RoBERTa-base[3] model fine-tuned for summarization. The use of longformer models with a maximum input size of 4096 characters allows us to load the full text of nearly all input samples. This approach again does not make use of the teaser information.

BART SQuADv2[4] A BART-LARGE (Lewis et al., 2020) model fine-tuned on the second version of the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016). This is a seq2seq model that can handle sequences up to 1024 characters, longer input was truncated. Both the teaser/title and the text are used for generation.

Sentence Transformers (S-BERT) QA[5] This sentence transformer model (Reimers and Gurevych, 2019) was fine-tuned on $215\,M$ question-answer-pairs. It works in an extractive fashion and uses a vector space designed for semantic search.

T5 SQuAD[6] The T5 base model (Raffel et al., 2020) fine-tuned on the SQuAD dataset. This is again a seq2seq model incorporating both the teaser and the text for producing the resolution.

T5 SQuAD fine-tuned For the final two baselines, we fine-tuned existing models using our training set. First, we took the T5-based QA model (see above) for that and ran a standard fine-tuning on all 2108 training data samples.

T5 SQuAD augmented + fine-tuned To better level between the size of the SQuAD dataset the model was originally trained on and the amount of data available for fine-tuning, we additionally created silver data using an augmentation step and fine-tuned the model with both the training data and the automatically created silver data. For the augmentation step, we used NLPAug (Ma, 2019) to create two artificial teasers for each training sample based on WordNet. The texts were not modified. Hence, that model was fine-tuned on $2108 + 2 \cdot 2108 = 6324$ samples.

6.2. Results

The resulting scores from running the baselines on dev and test set can be found in Table 1. In Table 2 we show the resulting ranks of the different approaches for each metric.
Most important, it can be seen that the approach using our dataset the most (namely using title and context for prediction itself, and being fine-tuned on the training data) performs better than all other approaches regardless of the metric. This is also remarkable because the underlying (non-refined) model performs worse than the BART model also trained on the SQuAD data for most metrics (except Exact Match, and BERTScore on the dev set), i.e., this is achieved although not even a superior respectively the best model was refined.
For the most part, the extractive baselines do not perform well and, depending on the metric, are even beaten by the LongformerSummary model which does not include the title when generating the response. This is certainly partly due to the fact that the resolutions were written by hand and not generated by selecting existing text blocks. When applying metrics that primarily measure the overlap of tokens or n-grams, it may thus not even be possible to achieve a perfect score. We therefore also report the extractive upper bound in Table 1, i.e., the highest possible score that could be achieved if always the one sentence best matching the answer was selected. As described earlier, we also have to assume that the information needed to resolve a clickbait can often not be found in a single sentence, but rather spans a longer range, which might be another tripping stone for extractive approaches.

[1] https://huggingface.co/patrickvonplaten/longformer2roberta-cnn_dailymail-fp16
[2] https://huggingface.co/allenai/longformer-base-4096
[3] https://huggingface.co/roberta-base
[4] https://huggingface.co/a-ware/bart-squadv2
[5] https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1
[6] https://huggingface.co/valhalla/t5-base-squad

Approach	ExactMatch	Rouge-2	Rouge-L	Meteor	BLEU-2	BERTScore
		Dev				
Extractive Upper Bound	.0000	.1351	.2090	.1852	.0957	.1760
First Sentence (E)	.0000	.0151	.0657	.0638	.0128	.0214
Last Sentence (E)	.0000	.0070	.0386	.0455	.0058	.0553
Longformer Summary (S)	.0000	.0298	.1109	.0612	.0123	.0533
BART SQuADv2 (S)	.0038	.0565	.1030	.1164	.0508	.0476
S-BERT QA (E)	.0000	.0178	.0747	.0697	.0131	.0472
T5 (S)	.0189	.0394	.0907	.1074	.0423	.0730
T5 fine-tuned (S)	.0455	.0716	.1568	.1891	.0737	.1567
T5 augmented+fine-tuned (S)	**.0720**	**.0870**	**.1846**	**.2296**	**.0910**	**.2089**
		Test				
Extractive Upper Bound	.0000	.1029	.1616	.1456	.0686	.1662
First Sentence (E)	.0000	.0187	.0517	.0582	.0125	.0514
Last Sentence (E)	.0000	.0043	.0306	.0404	.0035	.0045
LongformerSummary (S)	.0000	.0202	.0816	.0450	.0064	.0594
BART SQuADv2 (S)	.0038	.0600	.1034	.1089	.0610	.1276
S-BERT QA (E)	.0000	.0161	.0592	.0573	.0094	.0705
T5 (S)	.0114	.0275	.0849	.0952	.0250	.1124
T5 fine-tuned (S)	.0342	.0716	.1534	.1790	.0681	.2137
T5 augmented+fine-tuned (S)	**.0456**	**.0688**	**.1702**	**.2038**	**.0690**	**.2523**

Table 1: Results of the baseline models evaluated with different metrics (Exact Match, Rouge-2 & Rouge-L F1, Meteor, BLEU-2, and BERTScore) on the dev and test sets of our corpus. Additionally, we show the scores for the extractive upper bound (i.e., selecting the one sentence corresponding best with the manually written answer). Values between 0 and 1, and higher is better for all metrics. (S) marks Seq2Seq models, (E) marks models working extractively. The fine-tuned versions of T5 outperform all other approaches regardless of the metric.

Approach	ExactMatch	Rouge-2	Rouge-L	Meteor	BLEU-2	BERTScore
		Dev				
First Sentence (E)	5	7	7	6	6	8
Last Sentence (E)	5	8	8	8	8	6
Longformer Summary (S)	5	5	3	7	7	7
BART SQuADv2 (S)	4	3	4	3	3	4
S-BERT QA (E)	5	6	6	5	5	5
T5 (S)	3	4	5	4	4	3
T5 fine-tuned (S)	2	2	2	2	2	2
T5 augmented+fine-tuned (S)	1	1	1	1	1	1
		Test				
First Sentence (E)	5	6	7	5	5	7
Last Sentence (E)	5	8	8	8	8	8
LongformerSummary (S)	5	5	5	7	7	6
BART SQuADv2 (S)	4	3	3	3	3	3
S-BERT QA (E)	5	7	6	6	6	5
T5 (S)	3	4	4	4	4	4
T5 fine-tuned (S)	2	2	2	2	2	2
T5 augmented+fine-tuned (S)	1	1	1	1	1	1

Table 2: Resulting ranks of the baseline approaches based on the different metrics. 1 is the best rank. The two fine-tuned versions of T5 rank on the first two ranks regardless of the metric. The summary approach that does not take the teaser into account as well as the extractive approaches land on the back ranks for all metrics.

A qualitative analysis of the cases where our augmented and fine-tuned T5 baseline fails on the dev set according to the Rouge-2 metric shows several patterns: in some cases, the metric does not reflect that the most important aspect of the answer was correctly extracted (e.g., for entry 17669 with the gold answer "'The Intelligent Investor' by Benjamin Graham, it advises to buy stocks when they are low and hold them and also ways to avoid huge mistakes." the answer "The Intelligent Investor" was produced. Yet, since it is a considerably shorter subset, the score is low. In many other cases where only a subset of the expected answer is returned, this low score seems however to be justified. "130/80" is indeed an important part of the resolution to entry 13522, but without the information that this is the new definition for high blood pressure, it will probably not be understandable on its own. The same applies for "cold" (entry 10009) with the expected answer "his Burger King food was cold". Similarly, the baseline approach often extracts something that is related to the answer, but on another detail level and thus may be too generic to really satisfy the information need. For example, it returns "fear" as "the sad reason half of Americans don't take all their paid vacation"—which is true but not as detailed as "they believe they'll be replaced" (entry 15745). Finally, the approach often only repeats a central phrase from the title, e.g., "the Iron Throne" for "I sat on the actual Iron Throne from 'Game of Thrones'—here's what it was like" (entry 21787).

To summarize: Measured with different metrics, patterns in rankings emerge, but the differences make the use of different scores seem justified and provide intuition about to which aspects (e.g., customized wording) certain approaches perform particularly well. Generative approaches (seq2seq) seem more promising than extractive approaches. Most neural models clearly outperform the trivial baselines, but there are still a lot of cases where they are not able to produce the correct answer, and several patterns for such cases can be found. For the models tested, generic QA approaches cannot match the quality of an approach specifically refined on the data. Finally, by means of augmentation, the quality of the approach can be boosted even more, without having to manually annotate further data.

7. Conclusion

In this paper, we presented a new corpus for clickbait resolving and established a corresponding task. We showed that our dataset is suitable to train approaches for that task using several baselines and evaluating with different metrics—building metrics for free form question answering evaluation is thereby treated as orthogonal problem, but we will happily include new metrics resembling human judgment that are developed in the next years into our evaluation procedure. The leaderboard and all important details to train, evaluate, and submit own approaches can be found on the project webpage. We hope that our dataset and our task will help to preserve people from wasting their time, improve their media literacy, and in the end reduce the amount of clickbait on the internet.

8. Acknowledgements

We thank Max Doll, Kathrin Ferring, Martin Otterbein, and Jan-Hendrik Schmidt for the countless hours they spent reading hardly bearable texts.

This work has been supported by the German Research Foundation as part of the Research Training Group *Adaptive Preparation of Information from Heterogeneous Sources (AIPHES)* under grant No. *GRK 1994/1*, as well as the German Federal Ministry of Education and Research and the State of Hesse through the National High-Performance Computing Program.

9. Bibliographical References

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

Banerjee, S. and Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics.

Barwick, R. (2021). Brands are still playing ball with clickbait ad sites, advertising's roach that will survive the bomb. `https://www.morningbrew.com/marketing/stories/2021/09/08/brands-still-playing-ball-clickbait-ad-sites-advertisings-roach-will-survive-bomb`.

Blom, J. N. and Hansen, K. R. (2015). Click bait: Forward-reference as lure in online news headlines. In *Journal of Pragmatics*, volume 76, pages 87–100.

Chakraborty, A., Paranjape, B., Kakarla, S., and Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '16, page 9–16. IEEE Press.

Chen, A., Stanovsky, G., Singh, S., and Gardner, M. (2019). Evaluating Question Answering Evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124. Association for Computational Linguistics.

Dang, H. T. (2006). DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering - SumQA '06*, page 48. Association for Computational Linguistics.

Deutsch, D. and Roth, D. (2019). Summary Cloze: A New Task for Content Selection in Topic-Focused

Summarization. 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)*, pages 3720–3729. Association for Computational Linguistics.

Gabielkov, M., Ramachandran, A., Chaintreau, A., and Legout, A. (2016). Social Clicks: What and Who Gets Read on Twitter? In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pages 179–192. ACM.

Hättasch, B., Geisler, N., and Binnig, C. (2021). Netted?! How to Improve the Usefulness of Spider & Co. *2nd International Conference on Design of Experimental Search & Information REtrieval Systems (DESIRES)*, page 6.

Kalyan, K. S., Rajasekharan, A., and Sangeetha, S. (2021). AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing. arXiv: 2108.05542.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Loewenstein, G. (1994). The psychology of curiosity: A review and reinterpretation. In *Psychological bulletin*, volume 116, pages 75–98. American Psychological Association.

Ma, E. (2019). NLP augmentation. `https://github.com/makcedward/nlpaug`.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311. Association for Computational Linguistics.

Potthast, M., Gollub, T., Hagen, M., and Stein, B. (2018a). The Clickbait Challenge 2017: Towards a Regression Model for Clickbait Strength. arXiv: 1812.10847.

Potthast, M., Gollub, T., Komlossy, K., Schuster, S., Wiegmann, M., Garces Fernandez, E. P., Hagen, M., and Stein, B. (2018b). Crowdsourcing a large corpus of clickbait on Twitter. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1498–1507, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*, volume 21, pages 1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (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, 11.

Rony, M. M. U., Hassan, N., and Yousuf, M. (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*, ASONAM '17, page 232–239, New York, NY, USA. Association for Computing Machinery.

Scott, K. (2021). You won't believe what's in this paper! Clickbait, relevance and the curiosity gap. In *Journal of Pragmatics*, volume 175, pages 53–66.

Si, C., Zhao, C., and Boyd-Graber, J. (2021). What's in a name? answer equivalence for open-domain question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9623–9629, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Vijgen, B. (2014). The Listicle: An exploring research on an interesting shareable new media phenomenon. In *Studia Universitatis Babes-Bolyai - Ephemerides*, volume 59, pages 103–122. Studia Universitatis Babes-Bolyai.

Voorhees, E. M. and Tice, D. M. (2000). Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 200–207. Association for Computing Machinery.

Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations 2020*.

Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., and

Chua, T.-S. (2021). Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. arXiv: 2101.00774.