Jack-Ryder at SemEval-2023 Task 5: Zero-Shot Clickbait Spoiling by Rephrasing Titles as Questions

Dirk Wangsadirdja **Jan Pfister** **Konstantin Kobs** **Andreas Hotho**
Julius-Maximilians-Universität Würzburg (JMU)
{surname}@informatik.uni-wuerzburg.de

Abstract

In this paper, we describe our approach to the clickbait spoiling task of SemEval 2023. The core idea behind our system is to leverage pre-trained models capable of Question Answering (QA) to extract the spoiler from article texts based on the clickbait title without any task-specific training. Since oftentimes, these titles are not phrased as questions, we automatically rephrase the clickbait titles as questions in order to better suit the pretraining task of the QA-capable models. Also, to fit as much relevant context into the model's limited input size as possible, we propose to reorder the sentences by their relevance using a semantic similarity model. Finally, we evaluate QA as well as text generation models (via prompting) to extract the spoiler from the text. Based on the validation data, our final model selects each of these components depending on the spoiler type and achieves satisfactory zero-shot results. The ideas described in this paper can easily be applied in fine-tuning settings.

1 Introduction

The objective of the *Clickbait Spoiling* task (Fröbe et al., 2023a) is to find/extract the spoiler from a linked article that satisfies the reader's curiosity that was induced by a clickbait title (e.g. "You won't believe how well this SemEval clickbait spoiling system performs!"). The challenge differentiates between three types of spoilers: phrase (the spoiler is a short phrase), passage (a sentence or multiple sentences that span from a beginning position to an end position without any gap), and multi (the spoiler consists of multiple disjoint phrases/passages).

In this paper, we describe our system for this challenge that is based only on pre-trained models and no fine-tuning. We model spoiler extraction as a Question Answering (QA) task and explore four different pre-trained architectures: Two extractive QA using BERT models and two prompt-based text generation models (with two prompts each). In order to make the clickbait title more similar to the questions the QA models have been trained on, we propose to automatically rephrase them into a question form. To ensure that the model receives the most relevant article text passages in its limited context, we propose to filter and sort the article text using a pre-trained semantic similarity model. We evaluate our pipeline on the validation dataset and combine the best components for each spoiler type as our final model. We find that rephrasing the clickbait title usually improves performance, while relevance-based context ordering does not. Our final model makes use of clickbait title rephrasing for phrase and multi-spoiler types, a DeBERTa QA model for phrase and passage-type spoilers, and a Flan-T5 model for multi-type spoilers. The results on the test set show satisfactory results, considering that our system is not fine-tuned in any way. Our code is publicly available[1].

2 Background

Research on clickbait texts can be divided into mainly three categories: Clickbait detection, type classification, and spoiling. Clickbait detection tries to predict whether a news article's headline is a clickbait title. For this, several feature-based machine learning models have been developed (Potthast et al., 2016; Cao et al., 2017; Genç and Surer, 2021). Clickbait type classification and spoiling are the tasks of this challenge, which try to detect the type of spoiler (phrase, passage, multi) and extract the spoiler from the text, respectively. For clickbait type classification, transformer-based models have shown the best performance (Hagen et al., 2022). The information about the type of spoiler can potentially be used in the spoiler extraction model. Clickbait spoiling is mainly done using fine-tuned extractive and abstractive Question Answering (QA) models (Hagen et al., 2022; Johnson

[1] https://github.com/lingsond/semeval-23-task-5

July 13-14, 2023 ©2023 Association for Computational Linguistics

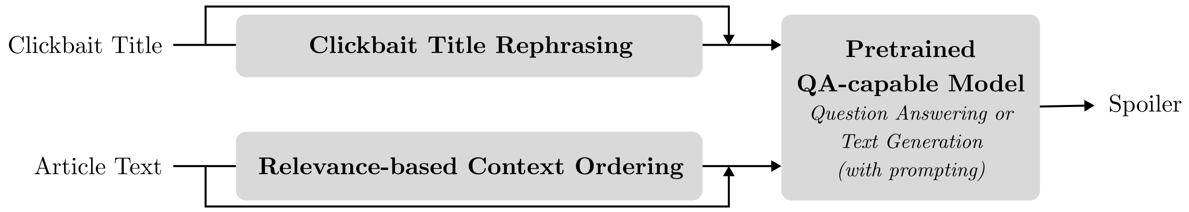

Figure 1: Overview of our system. It consists of mainly three components that are enabled based on the spoiler type. Given a clickbait title and its corresponding article text, the title is potentially rephrased into a question. To ensure that the article text fits into the context size of the following QA-capable model, we optionally use a SentenceBERT model to sort the text's sentences by relevance. The question and text are used in a pre-trained QA-capable model, which is either a QA model or a text generation model that uses prompting to extract the spoiler.

et al., 2022; Heiervang, 2022), where extractive models are generally more suitable (Johnson et al., 2022). Zero-shot spoiling using pre-trained QA models is inferior to fine-tuned models (Heiervang, 2022). All of the available approaches use the clickbait title without modification as the "question" input, which might not be beneficial for the performance of the model, since the clickbait title of news articles is oftentimes not phrased as a question (e.g. "You won't believe what happened after the event!"). In this paper, we show for the spoiling task that rephrasing the clickbait title as a question usually improves the performance of zero-shot QA models. While this still not achieves comparable performance to fine-tuned models, exploring easy-to-implement steps to improve zero-shot performance can easily be adapted to fine-tuned settings.

For the task, a training and validation set are provided, with each example consisting of a clickbait title (`postText`; the text that is posted on social media), article text (`targetTitle` and `targetParagraphs`), the extracted spoiler (`spoiler`; not available during testing) with its positions (`spoilerPositions`; not available during testing), as well as the type of spoiler (`tags`). Additional meta-information is provided but not used by our system. All submissions are submitted to and evaluated on TIRA (Fröbe et al., 2023b).

3 System Overview

Our system is shown in Figure 1. It only leverages pre-trained models (question answering, text generation, and semantic similarity models) to extract the spoiler from a given text without the need for any training. While it is certainly beneficial to employ fine-tuning as done by previous work, we experiment with other ideas that have not been proposed in the clickbait spoiling literature, leading to new insights into what works and what does not for this kind of task. The elimination of training enables faster model development and fast updates to the model pipeline when a better component is available. Note that our system can also be used in fine-tuning settings without any modifications, so our proposed additions to the pipeline could be beneficial to fine-tuned approaches.

Our pipeline consists of mainly three components that are introduced based on the following key ideas: 1. QA models are trained on questions, but clickbait titles often are not questions. Thus, we use a pre-trained model to rephrase clickbait titles into questions. 2. The article texts are oftentimes too long for the model input size. Therefore, we order each article's sentences by relevance using semantic similarity to the title, selecting the most important sentences as model input. 3. Recently, zero-shot capabilities of large language models are getting massive research interest, but have not been investigated for clickbait spoiling. Therefore, we evaluate text generation-based models with different prompt templates in addition to QA models, calling them collectively "QA-capable models". We now go through each of these three components (Clickbait Title Rephrasing, Relevance-based Context Ordering, and Pre-trained QA-capable Model) and describe how they work. We later select the model to use and whether to use clickbait title rephrasing and relevance-based context ordering based on the validation results for each spoiler type (phrase, passage, multi).

3.1 Clickbait Title Rephrasing

In order to use pre-trained Question Answering (QA) models to extract spoilers to a clickbait title from its corresponding article text, we hypothesize that rephrasing the clickbait title as a question overall improves performance. For this, we check whether the clickbait title is already a question by looking for an ending question mark. If not, we let a pre-trained language model, i.e. Flan-T5 (Chung et al., 2022), rephrase the provided clickbait post text into a question using prompting (e.g. “The anytime snack you won’t feel guilty about eating” is rephrased to the question “What snack is the best for when you want to eat something without feeling guilty?”). We create the following prompt template: `Change the headline to question form. Headline: {clickbait title}`, with `{clickbait title}` being the article headline.

In order to ensure that the rephrased question is still similar to the original clickbait title, the output of the model is compared to the original clickbait title with a semantic similarity model. For this, we embed both texts as vectors using the `all-MiniLM-L6-v2` model from Sentence-BERT (Reimers and Gurevych, 2019) and compute the cosine similarity between them. If the similarity is above 0.7, we continue using the newly created question, otherwise, the original clickbait title is used. This procedure catches failed rephrasings, i.e. incorrect or unrelated questions derived from the clickbait title. For example “This popular soda could cure your hangovers scientists say” is rephrased to “What is the best way to cure hangovers?”, which does not capture the “soda” part of the original clickbait title and is thus not used. From now on, we refer to the output of this system component as the “question”, regardless of whether it is the clickbait title or a rephrased version.

3.2 Relevance-based Context Ordering

For the QA-capable models used, we need to provide a context that contains the answer to the question, i.e. the spoiler to the clickbait title. In general, we create such context by concatenating the article headline to the article text, if the headline is not already in the article. Note that the article headline and the clickbait title (which is often posted to social media to deceive people into clicking the link) might be different, so it is useful to consider the headline as part of the context. One issue with the application of QA-capable models to these texts is that the article text is usually longer than the supported input size. Consequently, we propose to select the most relevant sentences of the text using a semantic similarity model.

This process is again implemented via Sentence-BERT: we split the article into sentences, embed all sentences separately, and sort them by embedding similarity to the original clickbait title. The approach is based on the assumption that the sentence that contains the answer to a question is semantically more similar to the clickbait title than other possibly unrelated sentences in the article. We sort all sentences in decreasing order and concatenate them to create the new context. We then take as many tokens as possible from the beginning that fit into the context size of the QA-capable model.

3.3 Pre-trained QA-capable Model

Given the question and context of the previous components, we can apply pre-trained QA-capable models to extract the spoiler. Similarly to related work, we use Question Answering models that take the question and the context as input. In light of the recent rise of language models that are capable of showing impressive performance in many tasks in a zero-shot setting (Chung et al., 2022; Kojima et al., 2022), we also test such models. We call them Text Generation (TG) models from now on.

Question Answering Models We can apply pre-trained QA models without any modifications since we mostly use questions as input. Concretely, we follow related work and experiment with DeBERTa[2] (He et al., 2021) and RoBERTa[3] (Liu et al., 2019) for question answering.

Text Generation Models with Prompting We also evaluate text generation models that take the question and context in a prompt template and generate the output according to their text completion objective. For the models, we experiment with Flan-T5 (Chung et al., 2022) and UnifiedQA (Heiervang, 2022). We also evaluate different prompt templates that are more focused on the spoiler extraction or the question-answering task. The spoiler-focused prompt template (*Spoil-prompt*) is:

```
From the following Context and Clickbait,
extract the spoiler.
Clickbait: {question}
Context: {context}
Spoiler:
```

The question answering focused prompt template (*QA-prompt*) is:

```
Answer the following question based on
the given context.
question: {question}
```

[2] https://hf.co/deepset/deberta-v3-large-squad2

[3] https://hf.co/deepset/roberta-large-squad2

Table 1: BLEU-4 scores (in percent) without and with clickbait title rephrasing (separated by /) for all models, prompts, and the relevance-based context ordering step. The best value between clickbait title rephrasing and without rephrasing in each cell is given in bold. The best combination for a given spoiler type is shaded green.

			w/o relevance-based context ordering				**w/ relevance-based context ordering**			
			All	**Phrase**	**Passage**	**Multi**	**All**	**Phrase**	**Passage**	**Multi**
QA	DeBERTa		23.70 / **25.97**	39.69 / **46.41** 🟩	**15.01** 🟩 / 14.33	**5.81** / 4.33	23.18 / **24.34**	38.54 / **42.47**	**14.75** / 14.26	**6.15** / 4.56
	RoBERTa		23.68 / **25.05**	40.89 / **45.47**	**13.83** / 12.58	**5.56** / 5.28	21.53 / **22.32**	36.26 / **40.43**	**13.60** / 11.27	**4.84** / 4.77
TG	Flan-T5	*Spoil-prompt*	6.72 / **10.95**	6.41 / **15.38**	8.55 / **9.25**	3.31 / **4.40**	7.11 / **11.21**	6.05 / **16.30**	**10.06** / 9.16	2.93 / **3.92**
		QA-prompt	16.91 / **21.15**	29.54 / **37.01**	9.26 / **10.90**	4.58 / **7.08** 🟩	15.30 / **20.32**	29.17 / **38.06**	5.99 / **8.17**	3.78 / **6.14**
	UnifiedQA	*Spoil-prompt*	10.97 / **15.62**	20.09 / **30.10**	5.08 / **6.14**	2.90 / **3.08**	11.05 / **16.18**	19.82 / **31.59**	**5.52** / 5.47	2.94 / **4.19**
		QA-prompt	5.97 / **14.38**	10.57 / **27.84**	3.26 / **5.77**	1.30 / **2.23**	5.70 / **13.44**	10.40 / **26.65**	2.97 / **4.91**	0.85 / **1.70**

Table 2: Fraction of examples where the ground truth spoiler is present in the context given to the model. All values are given in percent.

Models	**w/o relevance-based context ordering**	**w/ relevance-based context ordering**
DeBERTa	92.00	87.38
RoBERTa	92.38	87.63
Flan-T5 (*Spoil-prompt*)	89.38	83.63
Flan-T5 (*QA-prompt*)	89.38	84.38
UnifiedQA (*Spoil-prompt*)	89.38	83.63
UnifiedQA (*QA-prompt*)	89.38	84.38

```
context: {context}
answer:
```

Note that line breaks are used between parts of the prompt, following the work of UnifiedQA (Heiervang, 2022). The sorted article text from the previous step is at the end of the prompt, which leads to cutting off only context whenever the prompt is getting too large for the model to process.

4 Experiments and Results

We evaluate our proposed system on the validation data of the challenge and give the BLEU-4 score for all and spoiler-type specific examples in Table 1. Here, we observe multiple findings that we use to build the final model for our submission.

Clickbait Title Rephrasing improves results We propose to rephrase the clickbait title as a question in order to better reflect the type of texts the QA-capable models were trained on. In Table 1, each cell states the performance without and with the clickbait title rephrasing component, separated by a slash. We typeset the better performance of the two in bold. Overall, the clickbait title rephrasing component mostly improves performance on the dataset, especially for the phrase spoiler type.

Relevance-based Context Ordering usually does not help We propose to sort the sentences of the article text by the semantic similarity to the clickbait title, presumably giving the most relevant context into the QA-capable models. As comparing the left and right parts of Table 1 suggest, this approach usually leads to worse performance than feeding the model the beginning of the article text. This can have multiple reasons: For one, the spoiler that needs to be extracted could not be present in the constructed context, which means that our approach does not select the most relevant sentences. Another reason might be that due to the reordering of the sentences, the content of the text is misleading or confusing. To analyze this, we compute the fraction of examples in which the desired spoiler text can be found in the context that we give to the QA-capable model. These fractions in Table 2 are lower for the relevance-based ordered context, meaning that the desired spoiler text is often found in the first few sentences of the article (around 90%) and that our approach does not find the most relevant sentences. We assume that the semantic similarity based on SentenceBERT is not a good measure for relevance, since the similarity between a question and its answer might often be low.

Question Answering models generally work better than Text Generation models While previous work exclusively uses QA models, we also experiment with more general text generation models that are instructed by prompts. Table 1 shows a clear tendency in this regard: Dedicated QA models generally work better than text generation models on this task. We argue that TG models are also capable of solving this task, however, they usually do not reach the performance of QA models. A reason for why this is the case is that QA models have exactly one task they were trained on. TG models are usually more general and can be instructed to perform tasks using prompts.

The choice of these prompts seems to have a large influence on the performance. We find that for the different models, different prompts work best:

Table 3: Final test results given as BLEU-4, BERTScore (BERTSc), and METEOR (MET) over all clickbait posts respectively those requiring phrase, passage, or multi spoilers. We report all three selected models (according to Table 1) as well as our final hybrid model (subsection 4.1).

Model	All			Phrase			Passage			Multi		
	BLEU-4	BERTSc	MET	BLEU-4	BERTSc	MET	BLEU-4	BERTSc	MET	BLEU-4	BERTSc	MET
DeBERTa w/o clickbait title rephrasing	0.25	0.89	0.28	0.43	0.91	0.31	0.14	0.87	0.30	0.07	0.85	0.20
DeBERTa w/ clickbait title rephrasing	0.27	0.89	0.29	0.48	0.92	0.36	0.13	0.87	0.30	0.07	0.86	0.22
Flan-T5 w/ clickbait title rephrasing & *QA-prompt*	0.22	0.88	0.23	0.41	0.91	0.27	0.08	0.86	0.23	0.06	0.86	0.21
Final hybrid model (sec. 4.1)	0.27	0.89	0.31	0.48	0.92	0.36	0.14	0.87	0.30	0.06	0.86	0.21

the UnifiedQA model works best with the spoiling-focused prompt (*Spoil-prompt*), while Flan-T5 works better with the QA-focused prompt (*QA-prompt*). We hypothesize that the better prompt better resembles prompts from training.

4.1 Final Hybrid Model

Based on our observations above, we build a hybrid model making use of the best model for every spoiler type (shaded green in Table 1). This means our final hybrid model uses clickbait title rephrasing for phrase and multi-spoiler types, no relevance-based context ordering, the DeBERTA QA model for phrase and passage-type spoilers, as well as the Flan-T5 model with *QA-prompt* for multi-type spoilers. Overall, we thus evaluate the DeBERTa model with and without clickbait title rephrasing as well as the Flan-T5 model with clickbait title rephrasing and *QA-prompt* as the prompt template on the test dataset of the challenge as well their combination depending on the spoiler type.

4.2 Challenge Results

Table 3 shows the BLEU-4 score, micro BERTScore, and METEOR score (Banerjee and Lavie, 2005) for all models we used to determine the final test results for our system. Overall, our system achieves satisfactory results, given the fact that no fine-tuning was conducted. For phrase and passage spoilers, the best validation models are also the models with the best test performance. For multi-type spoilers, however, the Flan-T5 model does not perform as well as the DeBERTa model with clickbait title rephrasing, indicating that pre-trained QA models might be enough for this task.

Note that the shown models have been submitted to TIRA after the official deadline. The official submissions contain components derived from an incorrect interpretation of Table 1. Overall, however, the new results are very similar to the official submissions. According to the organizers, our official submissions were among the top ten teams for most metrics and for phrase spoiling even among the top five teams for all three metrics.

5 Conclusion

In this paper, we have presented our approach to the clickbait spoiling challenge. We have explored multiple strategies to improve the performance of pre-trained models without any fine-tuning. The results are not as good as fine-tuned models (see other competitors and the challenge baseline), but we have found that clickbait title rephrasing usually improves performance, which can potentially be applied to fine-tuned models as well.

One strategy that has not worked was the relevance-based context ordering, which mostly decreases performance. We suspect that our measure of relevance is not reliable for question and answer pairs. Exploring more reliable and effective measures might be interesting. Another possible focus for future work might be to experiment with more prompts for text generation models. We have shown that different prompts can have a severe impact on the model's performance, so more careful prompt engineering might lead to similar performance as for the QA models and can then get fine-tuned. Also, using larger and thus more powerful text generation models such as the hosted GPT-3[4] (Brown et al., 2020) or the open source OPT-175B (Zhang et al., 2022) are options that showed promising results in preliminary experiments, but were not further explored due to monetary and computational limitations.

Acknowledgements

This work is partially supported by the MOTIV research project funded by the Bavarian Research Institute for Digital Transformation (bidt), an institute of the Bavarian Academy of Sciences and Humanities. The authors are responsible for the content of this publication.

[4] https://openai.com/

References

Satanjeev Banerjee and Alon Lavie. 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, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Xinyue Cao, Thai Le, Jason, and Zhang. 2017. Machine Learning Based Detection of Clickbait Posts in Social Media. ArXiv:1710.01977 [cs].

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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

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

Şura Genç and Elif Surer. 2021. ClickbaitTR: Dataset for clickbait detection from Turkish news sites and social media with a comparative analysis via machine learning algorithms. *Journal of Information Science*, page 01655515211007746. Publisher: SAGE Publications Ltd.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait Spoiling via Question Answering and Passage Retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7025–7036, Dublin, Ireland. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-Enhanced BERT with Disentagled Attention. In *International Conference on Learning Representations*.

Markus Sverdvik Heiervang. 2022. Abstractive title answering for clickbait content. Master's thesis, University of Oslo, Department of Informatics.

Oliver Johnson, Beicheng Lou, Janet Zhong, and Andrey Kurenkov. 2022. Saved You A Click: Automatically Answering Clickbait Titles. ArXiv:2212.08196 [cs].

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

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.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait Detection. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 810–817, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pretrained Transformer Language Models.