Click-BERT: Clickbait Detector with Bidirectional Encoder Representations from Transformers

Changyuan Qiu Chenshu Zhu Haoxuan Shan
University of Michigan
{peterqiu, jupiterz, shanhx}@umich.edu

Abstract

Clickbait is a rampant problem haunting online readers. Nowadays, clickbait detection in tweets remains an elusive challenge. In this paper, we propose Clickbait Detector with Bidirectional Encoder Representations from Transformers (Click-BERT), which could effectively identify clickbaits utilizing recent advances in pre-training methods (BERT and Longformer) and recurrent neural networks (Bi-LSTM and Bi-GRU) with a parallel model structure. Our model supports end-to-end training without involving any hand-crafted features and achieved state-of-the-art results on the Webis 17 dataset.

1. Introduction

Clickbait refers to a certain kind of headline that attracts people to click but gives something uncorrelated in that link. The motivation behind clickbaiting is to boost site traffic (and therefore, advertisement revenue) by exploiting the curious nature of human readers. The clickbait technique works by dangling a hyperlink with enticing headlines to lure people into clicking; and then redirect them to the publishers' own websites which are uncorrelated to the headline. The discrepancy between the headline and the destination content wastes online readers significant amount of time on contents of which they have no interest.

To address this problem, we propose Click-BERT*: (**Click**bait Detector with **B**idirectional **E**ncoder **R**epresentations from **T**ransformers), which could effectively identify clickbaits utilizing state-of-the-arrt pre-training methods and self-attentive network.

We approach this task as a regression problem in our two parallel baseline models for benchmarking with previous models. The model takes the post title and the linked content as input and will output a clickbait score in the range of [0, 1] with 0 indicating non-clickbait and 1 indicating clickbait. By training on a large twitter posts corpus with annotations of their 'clickbaitness' on a scale of [0, 1], we expect our model to be capable of capturing clickbait patterns in the headline and the content.

Main challenges of the clickbait detection problems lies in how well our model capture the meaning and correlation of the input headline/content (which differs greatly in length) and how properly the followed analysis get performed. And our main contributions include:

1. We applies advanced pre-trained models BERT and Longformer to extract sequence(headline/content) embedding to form a better understanding of the headline and the content.

2. We proposes a parallel model structure to integrate both prediction of whether the headline is luring people to click and prediction of whether the headline is related with the content into final judgement.

2. Related Works

Attempts to detect clickbaits have been made by many online media agencies. Facebook, for instance, launched a war against clickbaits in 2017. Their main strategy was to utilize users feedback to identify and block domains that are notorious for producing clickbaits.

In the academia, initial work in this domain can be traced back to [4], which relies heavily on feature engineering and does not generalize well [12]. Chakraborty et al. [7] took the lead to explore machine learning models (mainly SVM) in clickbait detection in 2016 and create a browser extension 'Stop Clickbait' for deterring clickbaits, yet they called for future work devoted to improving the classification performances.

[11] crowdsourced a standard dataset of tweets (Webis 17) as a clickbait detection benchmark. [1] proposed the first neural netowrk based approach in this field on the Webis 17 dataset. They employed various recurrent neural network architectures to model sequential data and its dependencies, taking as its inputs a concatenation of the word and

*Our codes are available at https://github.com/PeterQiu0516/Click-BERT (incomplete at the time of writing yet)

character-level embeddings of the headline. Their experiments yielded that bidirectional Long Short Term Memory (Bi-LSTM [14]) were much better at this task than previous SVM S.O.T.A by [7]. [15] built Bi-LSTMs to model each textual attribute of the post (post-text, target-title, target-paragraphs, target-description, target-keywords, post-time) available in the corpus [11], concatenating their outputs and feeding it to a fully connected layer to classify the post. Recently, attention mechanisms [2] become popular for various text classification tasks, utilizing this technique, [16] deployed a self-attentive bidirectional GRU to infer the importance of each tweet token and model the annotation distribution of headlines in the corpus, and achieved S.O.T.A on the Webis 17 dataset so far.

Moreover, pre-training methods which are based on transformer have revolutionized NLP over the past few years. Task-agnostic objectives such as autoregressive (GPT-2 [13], GPT-3 [6]) and masked language modeling (BERT [8]) have scaled across many orders of magnitude in compute, model capacity, and dataset, which leads to steady improvement in capabilities. Flagship systems like GPT-3 and BERT are now competitive across many tasks in NLP with bespoke models while requiring little to no dataset specific training data. As a result, we decide to utilize theses pre-training models for feature extraction from headlines and contents to achieve a better performance for our downstream clickbait detection task.

3. Data-preprocessing

3.1. Data Set

We perform experiments on the Webis 17 [10] dataset. [11] crowd-sourced the annotation of 38,517 tweets they had curated into various levels of their clickbait nature. These tweets contained the title and text of the article and also included supplementary information such as target description, target keywords and linked images. The data set is already split into train set (19,538 posts with 4761 being clickbaits and 14,777 non-clickbaits) and test set (18,979 posts). More detailed information about the dataset can be found in this website: Webis Clickbait Detection Challenge. Here we also included the structure of the dataset in the Table 1.

3.2. Data Selection

We focus on "postText", "targetParagraphs" and "truthMean" for each data. We excluded the media information since some of the posts do not have corresponding media annotations and thus it could hinder us from developing a model that can generalize well. For other information like post-id, target-keywords and post-timestamp, we think they are not highly related with whether a post is clickbait or not and we dispose them.

3.3. Data Cleaning

In our data cleaning process, we also decided it would be beneficial to exclude data that come with annotation (or label) of low confidence. The raw clickbait scores ('truthMean' label) are in a range of [0, 1], as an average of 5 scores coming from 5 annotators, and we decided to exclude tweets with a mean clickbait score that deviate no more than 0.2 from 0.5, or in the range of [0.3, 0.7]. These annotations shows little confidence to be distinguished as either clickbait or not, and might confuse our model. After the cleaning process, we obtained a total of 12963 valid examples from training set.

3.4. Train/Validation Split

We divide the dataset into a training set of 11663 tweets and a validation set of 1300 tweets. We use the validation set for preliminary performance evaluation and model selection and we report our results on the final test set. The statistics of the dataset can be found in Table 2.

4. Methodology, Training & Evaluation

4.1. Word Embedding

Word embedding is a text feature extraction technique that encodes the meaning of the word into a vector, thus obtaining the representation of words for text analysis and enables computers to calculate distances between words and measure similarity [9].

In this project, we used both BERT [8] embedding and Longformer [3] embedding. BERT provides representations for text extracted from headlines using a transformer architecture. We choose this embedding because it has shown significant capacity of text feature extraction and achieved state-of-the-art performance on many NLP downstream tasks. Longformer is used for learning a informative and concise representation of the content texts. It is chosen for its reputable ability in encoding long texts.

4.2. Baseline Model Architecture

We construct 2 baseline models in our project:

1. **Naive Cosine Similarity with BERT**. In the first baseline model, we use BERT-base to encode both the headline and content texts. The attention network outputs a $\mathbb{R}^{768}$ embedding vector for every word w. Then we average the word embedding for the sequences for both the headline and the content to attain two vector embedding $\bar{x}_{headline} \in \mathbb{R}^{768}$, $\bar{x}_{content} \in \mathbb{R}^{768}$ for headline and content respectively. Then, we apply element-wise inverse exponential mapping $x' = e^{-x}$ to each test embedding so that each element in the embedding is strictly positive. Finally, the two

Table 1. Webis 17 dataset structure [10]

Variable	Corresponding Data & Format
id	instance id
postTimestamp	⟨weekday⟩⟨month⟩⟨day⟩⟨hour⟩ : ⟨minute⟩ : ⟨second⟩⟨time offset⟩⟨year⟩
postText	⟨text of the post with links removed⟩
postMedia	⟨path to a file in the media archive⟩
targetTitle	⟨title of target article⟩
targetDescription	⟨description tag of target article⟩
targetKeywords	⟨keywords tag of target article⟩
targetParagraphs	⟨text of the ith paragraph in the target article⟩
targetCaptions	⟨caption of the ith image in the target article⟩
truthJudgments	clickbait scores provided by 5 annotators, fall in to 4 categories: 0 - "not clickbaiting", 0.33 - "slightly clickbaiting", 0.66 - "considerably clickbaiting", 1 - "heavily clickbaiting"
truthMean	mean of the truthJudgements
truthMedian	median of the truthJudgments
truthMode	if truthMean $>$ 0.5, 0 - non-clickbait; else, 1 - clickbait
truthClass	"non-clickbait" or "clickbait"

Table 2. Dataset statistics

	#tweets	#clickbaits	#non-clickbaits
Training	11663	2027	9636
Validation	1300	203	1097

embedding vectors are passed through a naive cosine similarity network to output the clickbait score as $y = \frac{\bar{x'}_{headline} \cdot \bar{x'}_{content}}{||\bar{x'}_{headline}||_2 \cdot ||\bar{x'}_{content}||_2}$ (cosine similarity fall in range [0,1]). The model diagram is given in Figure 1.

Figure 1. Baseline Model 1 - Naive Cosine Similarity with BERT.

2. **Headline BERT with RNN Network**. In the second baseline, we feed only the headline text into BERT to attain a sequence of word embedding. This sequence of embedding vectors are then fed into an recurrent network (we used bidirectional LSTM and GRU). The output from RNN is mapped by a fully connected layer to 1 output node. Here we train the network supervisedly, using the $Sigmoid$ activation function after the output node so that the predicted score falls in range [0,1]). The model diagram is given in Figure 2.

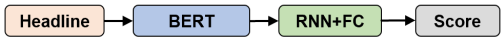

Figure 2. Baseline Model 2 - Headline BERT with RNN Network.

4.3. Our Method

BERT and Longformer with Parallel Structure. We build up our final model on top of the two baseline models. The first baseline model captures the relation between the headline and the contents. The second baseline model focuses on interpreting the headline. Under the clickbait definition outlined in our introduction section, both aspects should be taken into consideration when deciding whether a tweet is a clickbait or not. Therefore, we arrange the two models in a parallel structure, and concatenate the outputs from both. In doing so, we essentially attain an ensemble of the two baseline models. Finally, the outputs are again mapped by a fully connected layer, activated by the $Sigmoid$ function to get the clickbait score. To overcome the input length restriction of the BERT model, we changed the encoding layer in the second baseline model into Longformer, which performs well on long texts. Here, we only tried both Bi-LSTM in the recurrent neural network block since the previous test result shows the performance difference between Bi-LSTM and Bi-GRU is trivial. The model diagram is given in Figure 3.

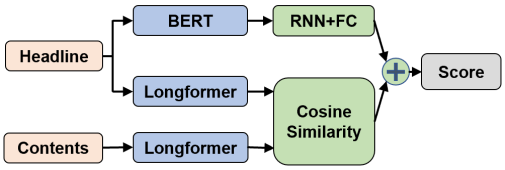

Figure 3. Our Method: BERT and Longformer with Parallel Structure.

4.4. Training

During training, we used Mean Squared Error(MSE) as the loss function and Adam Optimizer with a starting learn-

Table 3. Model Hyperparameters

	Hyperparameters
Bi-LSTM & Bi-GRU	Hidden dimension: 50 Layers: 2 Dropout: 0.2
FC Layer	Hidden dimension: 64 Dropout: 0.2
Loss Function	Mean Squared Error (MSE)
Weights Intialization	Xavier
Adam Optimizer	Learning rate : 10^{-4} Weight Decay: 10^{-3}
Learning rate scheduler (`ReduceLROnPlateu`)	Patience : 2 Weight Decay Factor: 0.25
Mini-batch size	8
Epochs	20

ing rate of 10^{-4}, notice we applied an adaptive learning rate with decay by factor of 0.25 and patience of 2. We use Xavier weights initialization. More detailed settings are shown in Table 3. Our baseline 1 model does not involve any training. Our baseline 2 model takes 12 hours to train on one Tesla T4 GPU and our BERT + Longformer model takes 4.5 days on one Tesla P100 GPU hosted on Google Colab.

4.5. Evaluation

We are evaluating the performance of our model on 2 tasks: binary classficiation task and regression task, the evaluation metrics include:

- For the binary classification task, we used accuracy and F1 score as our evaluation metric, which is consistent with prior works ([16, 12, 15]) on this dataset for comparison.
- For the regression model, we used Mean Squared Error (MSE) as our metric, which is consistent with prior works ([16, 12, 15]) on this dataset for comparison.

5. Experiments

5.1. Model Performance Benchmark

We perform the experiments on the pre-processed dataset. The results are shown in Table 4 and 5.

5.2. Performance Analysis

Our first baseline model, or Naive Cosine Text Similarity with BERT, involves no explicit training process. One peculiar observation is that the output of this baseline model is almost always around 0.5, which indicates little confidence in almost all predictions. As a result, the model only gets 26.9% accuracy on the dataset, even falls below a plain random classifier. The first baseline model demonstrates that:

1. Average pooling across word embeddings fails to represent the headline/content properly, this could result from the large length gap between headline and content.
2. The 26.9% accuracy might be related with the imbalanced training data that the raio of clickbait:non-clickbait is approximately 1:5.5.

Surprisingly, our second baseline model beats the previous S.O.T.A on both accuracy and MSE [16] on Webis 17 dataset with a simple Bi-LSTM structure even without fine-tuning on the raw BERT architecture, and this demonstrates that:

1. The BERT embedding is significantly powerful in the problem setting and provides salient representation of headlines.
2. The RNN(Bi-LSTM) architecture fits this downstream task well, and we substitute LSTM with GRU for ablation study and get similar performance ($\sim 1\%$ worse accuracy).

And our proposed model, BERT and Longformer with Parallel Structure, pushing our Baseline 2 even forward and surpasses previous S.O.T.A by $\sim 3\%$ in terms of classification accuracy and obtains a $\sim 25\%$ less regression MSE. This demonstrates:

1. The effectiveness of the parallel model structure we propose.
2. Longformer's strong capacity to process long text.

Model	Accuracy	F1-score
Headline only, Bi-LSTM Regression with BERT (ours)	85.83%	0.674
Headline only, Bi-GRU Regression with BERT (ours)	84.79%	0.658
Headline & Content, Naive Cosine Text Similarity with BERT (ours)	26.9%	0.166
Headline & Content, Bi-LSTM Regression with BERT + Longformer in parallel (ours)	**88.72%**	**0.715**
Headline only, Feature Engineering S.O.T.A. [12]	83.24%	0.550
Headline only, multi-classification with self-attentive Bi-GRU (prev S.O.T.A) [16]	85.6%	0.683
Headline & Content & Tags & Time, Concatenated NN Architecture [15]	74.00%	0.390

Table 4. Model Performance Benchmark (Classifcation)

Model	MSE
Headline only, Bi-LSTM Regression with BERT (ours)	0.031
Headline only, Bi-GRU Regression with BERT (ours)	0.032
Headline & Content, Naive Cosine Text Similarity with BERT (ours)	0.269
Headline & Content, Bi-LSTM Regression with BERT + Longformer in parallel	**0.025**
Headline only, Feature Engineering S.O.T.A. [12]	—
Headline only, multi-classification with self-attentive Bi-GRU (prev S.O.T.A) [16]	0.033
Headline & Content & Tags & Time, Concatenated NN Architecture [15]	0.045

Table 5. Model Performance Benchmark (Regression)

5.3. Error Analysis

We perform analysis of the data that our proposed model misclassifies. The statistics are shown in Table 6.

Corpus \ Clickbait score	$y < 0.3$	$y > 0.7$
Cleaned Corpus	17.20%	82.80%
Failure Corpus	24.50%	75.50%

Table 6. Failure cases statistics

We could observe that our model are better at detecting non-clickbait examples, that could be partially due to the imbalanced dataset that the raio of clickbait:non-clickbait is approximately 1:5.5 and thus our model get more 'experienced' on detecting non-clickbait headlines.

We also perform analysis on the data that get cleaned out in Section 3.3. The statistics are shown in Table 7.

Corpus	Classification Accu.
Cleaned Corpus ($y < 0.3 \vert y > 0.7$)	88.72%
Cleaned-out Corpus ($0.3 \leq y \leq 0.7$)	81.53%

Table 7. Cleaned-out cases statistics

We could observe that the model get $\sim 6\%$ worse accuracy on the cleaned-out corpus, this is intuitive since that:

1. The cleaned-out corpus with $0.3 \leq y \leq 0.7$ are harder to classify even for human since their clickbait score is close to 0.5
2. We did not train our model on the cleaned-out corpus.

5.4. Case Study

We also perform case study on both success and failure cases, a pair of success cases is shown in Figure 4 and a pair of failure cases is shown in Figure 5.

- For the success case:
  - The left non-clickbait example is highly correlated with the content that they both mention "Tokyo's subway is shut down", "missle attack-/launch", and our model assigns a score pretty close to 0 (0.074) to it and correctly detects it as non-clickbait.
  - While for the right clickbait example, the content isn't about the 26 interesting pictures and the poster only wants to absorb more followers to their account, and our model correctly detects it as clickbait with score of 0.895.
- For the failure case:
  - The left non-clickbait example use the idiom "Testing the waters", which stands for "gather feedbak on product prototypes before the official product launch" here and I guess our model failed here due to misunderstanding of this idiom.
  - While for the right clickbait example, I believe there is a human label error because both the headline and the content are talking about "18 uplifting documentaries", and it is indeed 'non-clickbait', while it has a label score of 0.73 and is labeled 'clickbait'

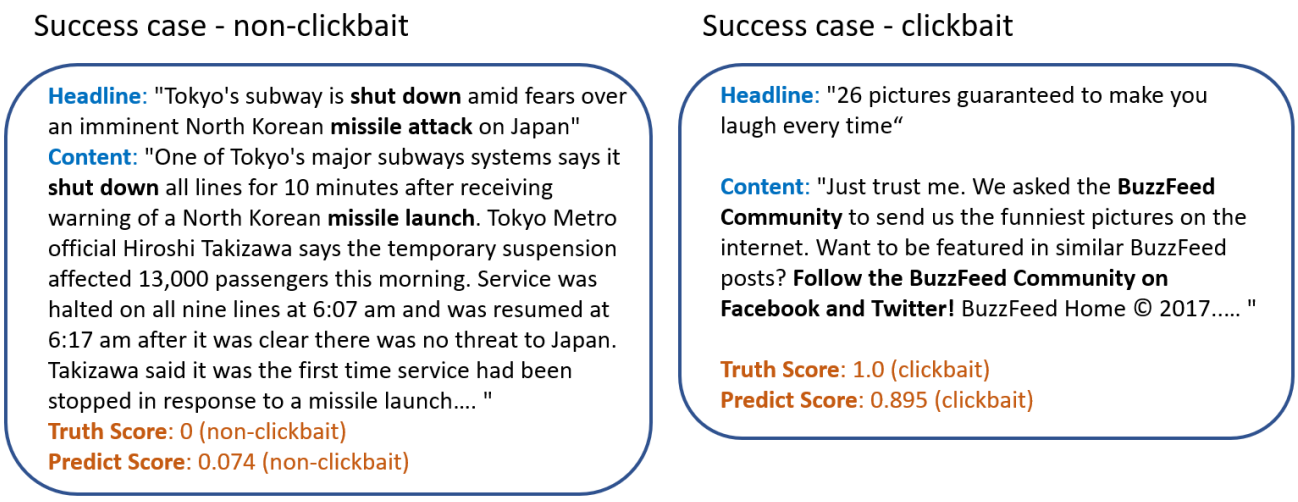

Figure 4. Case Study - Success cases

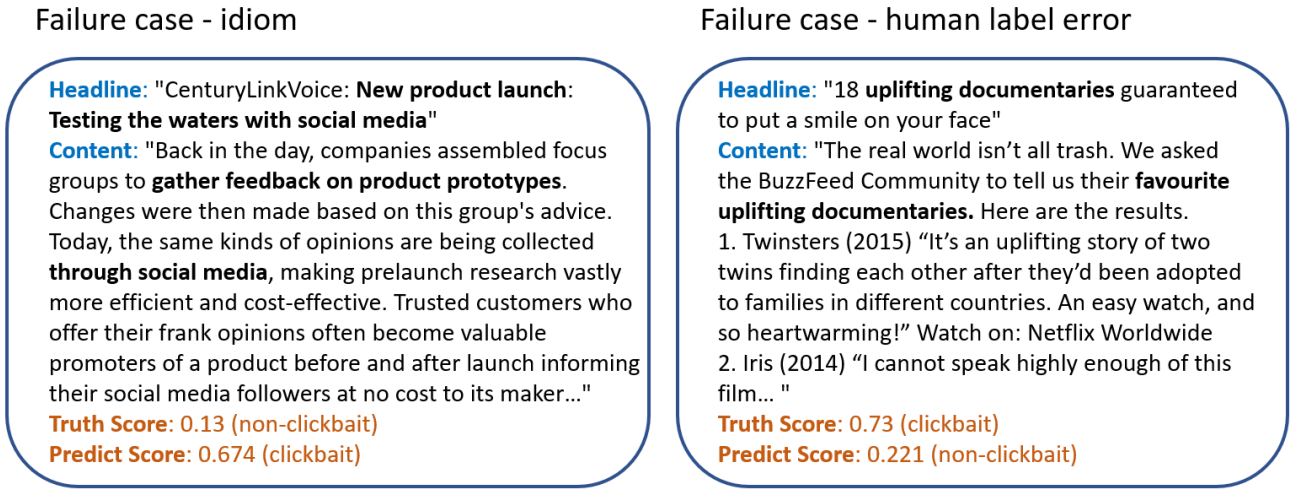

Figure 5. Case Study - Failure cases

6. Conclusion

In this project, we proposed Click-BERT: Clickbait Detector with Bidirectional Encoder Representations from Transformers. It could be trained end-to-end without involving any manual feature engineering. It achieved state-of-the-art performance on the Webis 17 dataset and is able to effectively identify clickbaits and non-clickbaits with high accuracy.

7. Future Work

While our proposed model achieve S.O.T.A on accuracy and MSE, there are still rooms for improvement, possible directions for future work are listed below.

1. Apply fine-tuning. When using BERT and Longformer techniques, we are directly using the pre-trained models to do the task. However, fine-tuning these models with data before the usage can improve their ability for embedding the headline and contents. We didn't carry out this procedure due to the limitations of compute resources.

2. Improve language understanding abilities for media idioms.

3. Incorporate feature engineering. We have built a classifier with features like readability score and the amount of special characters. However, our result shows that this method's performance drops sharply when the dataset size increase and get surpassed by our baseline 2 models. However, we expect that integrate the extracted features with our current model could produce a better performance.

4. As we can observe from Table 2, 6 that the dataset is imbalanced — there are approximately 5.5 times non-clickbaits training data than clickbait ones and that imbalancy do leads to some performance difference on classifying clickbait vs. non-clickbait examples. We may seek to sampling approaches that help with imbalanced regression like Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise (SMOGN [5]) and compare with the baseline.

5. As we can observe from Table 7 that our model performs worse on the "Cleaned-out" data, and we could consider how to make use of them without confusing the model.

References

[1] Ankesh Anand, Tanmoy Chakraborty, and Noseong Park. We used neural networks to detect clickbaits: You won't believe what happened next! In Joemon M. Jose, Claudia Hauff, Ismail Sengör Altingövde, Dawei Song, Dyaa Albakour, Stuart N. K. Watt, and John Tait, editors, *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*, volume 10193 of *Lecture Notes in Computer Science*, pages 541–547, 2017. 1

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. 2

[4] Prakhar Biyani, Kostas Tsioutsiouliklis, and John Blackmer. "8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 94–100. AAAI Press, 2016. 1

[5] Paula Branco, Luís Torgo, and Rita P. Ribeiro. SMOGN: a pre-processing approach for imbalanced regression. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications, LIDTA@PKDD/ECML 2017, 22 September 2017, Skopje, Macedonia*, volume 74 of *Proceedings of Machine Learning Research*, pages 36–50. PMLR, 2017. 7

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[7] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. *CoRR*, abs/1610.09786, 2016. 1, 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 2

[9] Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009. 2

[10] Martin Potthast, Tim Gollub, Matthias Hagen, and Benno Stein. The clickbait challenge 2017: Towards a regression model for clickbait strength. *CoRR*, abs/1812.10847, 2018. 2, 3

[11] Martin Potthast, Tim Gollub, Kristof Komlossy, Sebastian Schuster, Matti Wiegmann, Erika Patricia Garces Fernandez, Matthias Hagen, and Benno Stein. 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, Aug. 2018. Association for Computational Linguistics. 1, 2

[12] Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. Clickbait detection. In *European Conference on Information Retrieval*, pages 810–817. Springer, 2016. 1, 4, 5

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[14] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, 1997. 2

[15] Philippe Thomas. Clickbait identification using neural networks. *CoRR*, abs/1710.08721, 2017. 2, 4, 5

[16] Yiwei Zhou. Clickbait detection in tweets using self-attentive network. *CoRR*, abs/1710.05364, 2017. 2, 4, 5