Clickbait Spoiling with Language Models

Eren Barış Bostancı
ebostanci18@ku.edu.tr

Berke Can Rizai
brizai18@ku.edu.tr

1 Problem statement

In recent years, with the increasing use of the internet people's data has become available and accessible. The development of fields such as Machine Learning, the value of data become priceless. The value of data introduced new terms like clickbait. Clickbait is used for the online content, which is designed to attract users' attention and encourage them to click on a link to a particular web page. The purpose of the usage clickbait is to increase the web traffic, to earn ad revenue and collect data from users.

Clickbait has become a prevalent issue on the internet, often misleading users with sensational headlines. In this project, we aim to overcome the problem of clickbait by generating spoilers for clickbait posts using state-of-the-art Large Language Models (LLMs) and evaluate them with some other baseline models and recent approaches. We hypothesized that by fine-tuning LLMs on this task can lead to spoilers that effectively neutralize the clickbait.

We also benefit from the prompt engineering techniques such as few shot without performing fine-tuning Large Language Models.

Moreover, by training BART/T5 and TF-IDF based summarization models, it can purpose a good alternative to generating spoilers for the clickbait posts.

The source codes and scripts can be found in our repository: https://github.com/ribo-apps/clickbait-spoiling-nlp-project.

2 What we proposed vs. what we accomplished

- ~~Collect and preprocess dataset~~
- ~~Build and train language models, evaluate them against baseline models~~
- ~~We proposed to fine tune LLMs such as StableLM, Alpaca with LoRa. Since LLMs move fast and everything gets updated in a week, we instead fine tuned Falcon, Roberta, T5, LLaMA Base, TF-IDF also using LoRa and QLoRa (quantization)~~
- ~~Beat GPT-3.5 in evaluation. We did not think this was possible but we managed to do it~~

3 Related work

Spoiler prediction task share similarities with the broader field of Question Answering (QA) from text, as well as with related tasks such as clickbait detection and headline generation.

Hagen et al. (15) proposed a novel approach to combat clickbait by spoiling the clickbait question with an answer, using a two-step method involving a QA system and a passage retrieval system. This work is particularly relevant as it demonstrates the feasibility of using QA systems to extract specific information (in this case, a spoiler) from a text.

Spoiler prediction is also similar to clickbait detection. Shu et al. (22) proposed a deep learning model for generating headlines, which were then used to classify whether a given article is clickbait or not. Similarly, Chakraborty et al. (13) proposed a novel feature set for clickbait detection, demonstrating that their method could effectively detect clickbait with high accuracy. These works highlight the potential of using machine learning techniques to detect specific patterns in text, a capability that could be leveraged for spoiler prediction.

In terms of headline generation, **?**) proposed a method for generating sensational headlines using auto-tuned reinforcement learning. While their focus was on generating clickbait headlines, the techniques they used could potentially be adapted for generating spoiler-free headlines.

In terms of language models used for these tasks, several works have highlighted the effectiveness of transformer-based models like BERT and its variants. For instance, Hu et al. (18) proposed LoRA, a method for adapting large language models to specific tasks without fine-tuning the entire model. Similarly, Taori et al. (24) developed Stanford Alpaca, an instruction-following language model based on the LLaMA architecture. These works underscore the potential of using large, transformer-based language models for tasks like spoiler prediction.

Finally, the evaluation of these models often involves metrics like BLEU (20), which is widely used for evaluating machine translation systems. Other metrics like Meteor (14) and BERTScore (26) have also been used, highlighting the variety of evaluation methods available for tasks involving text generation and classification.

4 Dataset

The Clickbait Challenge at SemEval 2023 - Clickbait Spoiling’s dataset contains the manually cleaned versions of (16) and extracted spoilers for each post. The spoilers are categorized into three classes, which are Short Phrase Spoilers, Longer Passage Spoilers, and Multiple Non-Consecutive Pieces of Text. The dataset is already splited into train and validation. 3200 posts are available for training and 800 posts are available for validation.

Format of the data is JSON and contains uuid, postText, targetParagraphs, targetTitle, targetUrl, humanSpoiler, spoiler, spoilerPosition, tags, and some metainformations such as postId, postPlatform, targetDescription... For the clickbate spoiler generation task, we are allowed to use postText, targetParagraphs, and targetTitle. We should try to predict the ”spoiler” features for each posts.

The output formats are identical for part 1 and part 2 of the challange. Both parts will contain uuid, spoiler-type and spoiler values.

Below there is an example sample from dataset: ”uuid”: ”0af11f6b-c889-4520-9372-66ba25cb7657”, ”postText”: [”Wes Welker Wanted Dinner With Tom Brady, But Patriots QB Had Better Idea”], ”targetParagraphs”: [”It’ll be just like old times this weekend for Tom Brady and Wes Welker.”, ”Welker revealed Friday morning on a Miami radio station that he contacted Brady because he’ll be in town for Sunday’s game between the New England Patriots and Miami Dolphins at Gillette Stadium. It seemed like a perfect opportunity for the two to catch up.”, ”But Brady’s definition of čatching upïnvolves far more than just a meal. In fact, it involves some literal čatchingäs the Patriots quarterback looks to stay sharp during his four-game Deflategate suspension.”, ”Ï hit him up to do dinner Saturday night. He’s like, ‘I’m going to be flying in from Ann Arbor later (after the Michigan-Colorado football game), but how about that morning we go throw?’ Ẅelker said on WQAM, per The Boston Globe. Änd I’m just sitting there, I’m like, ‘I was just thinking about dinner, but yeah, sure. I’ll get over there early and we can throw a little bit.’ ̈”, ”Welker was one of Brady’s favorite targets for six seasons from 2007 to 2012. It’s understandable him and Brady want to meet with both being in the same area. But Brady typically is all business during football season. Welker probably should have known what he was getting into when reaching out to his buddy.”, ”T̈hat’s the only thing we really have planned,Ẅelker said of his upcoming workout with Brady. İt’s just funny. I’m sitting there trying to have dinner. ‘Hey, get your ass up here and let’s go throw.’ I’m like, ‘Aw jeez, man.’ He’s going to have me running like 2-minute drills in his backyard or something.̈”, ”Maybe Brady will put a good word in for Welker down in Foxboro if the former Patriots wide receiver impresses him enough.”], ”targetTitle”: ”Wes Welker Wanted Dinner With Tom Brady, But Patriots QB Had A Better Idea”, ”targetUrl”: ”http://nesn.com/2016/09/wes-welker-wanted-dinner-with-tom-brady-but-patriots-qb-had-better-idea/”, ”spoiler”: [”how about that morning we go throw?”], ”spoilerPositions”: [[[3, 151], [3, 186]]], ”tags”: [”passage”]

4.1 Data preprocessing

Since we aim to fine tune our Large Language models, the preprocessing is a must. targetParagraphs columns contains a list of several paragraphs. We concated the paragraphs into a string by separating with newline character. Then we concated the postText” and targetParagraphs in the below format.

$''postText\ :\ '' + postTextvalue + ''\texttt{\textbackslash n}'' + ''targetParagraphs : '' + postTextvalue$

For training data, we also concat the spoiler values at the end. For the validation we add ”spoiler:” at the end but not give the value to let the Language model predict the next tokens.

Moreover for Large Language Models that we fine-tuned, which are LLaMA(25) and Falcon, models we used a prompt before the concated data like the fine-tuned Alpaca (23).

For all these, we used HuggingFace dataset library which provides nice and easy to use tools for fine tuning LLMs. Particularly, concatenating and trimming.

4.2 Data annotation

The Clickbait Challenge at SemEval 2023 - Clickbait Spoiling's dataset is already annotated. The spoilers and spoilerPositions are extracted from targetParagraphs by humans. We did not use another annotation despite spoiler.

5 Baselines

What are your baselines, how do they work, and what are their results? Why did you choose these baselines over other models? Additionally, explain how each one works, and list the hyperparameters you are using and how you tuned them! Describe your train/validation/test split. If you have tuned any hyperparameters on your test set, expect a major point deduction!

During the project we used 2 baselines to compare our results. The first baseline is implementing a TF-IDF and the second one is using OpenAI's GPT 3.5 Turbo model.

TF-IDF (term frequency-inverse document frequency) is a method for measuring the importance of words in a document based on how often they appear in the document and how rare they are in a collection of documents. TF-IDF can be used in two ways,

- Given an article and a question, extract the relevant sentences from the article that contain the answer to the question. This can be done by using a sentence similarity measure, such as cosine similarity, between the question and each sentence in the article, and selecting the top-k most similar sentences. - Compute the TF-IDF scores of each sentence in the selected sentences, using the article as the document and the collection of articles as the corpus. We implemented this with NLTK. And used it in evaluations. - Select the words with the highest TF-IDF scores as the potential spoilers, since they are likely to be the most informative and distinctive words in the answer. Alternatively, use a threshold or a ranking method to filter out the words with low TF-IDF scores.

Some weak aspects are;

- It does not capture the semantic meaning or context of words, and only relies on their frequency and rarity. This results in missing some important words that are relevant to the answer but not frequent or rare enough. - It usually returns first sentence as prediction which led to poor performance. - TF-IDF does not adapt to different domains or tasks, and only uses a fixed collection of documents as the corpus.

- Sharma and Sharma (21) proposed a method for generating clickbait headlines using TF-IDF and natural language generation.

For the GPT baseline, we wanted to see if the Large Language Models can generate the clickbait spoilers and use as a proof of concept. We used OpenAI's API and tried 2 shot predictions. We give 2 examples from training set and 1 data from validation data to predict its spoiler. We create spoilers for each validation data with using 2 example of training. We include the postText and targetParagraph field for validation data and for training data examples we also add spoiler. The GPT 3.5 Turbo achieved 0.147 Bleu Score and 0.866 Bert Score.

6 Our approach

We used following models;

- LLaMa (7B)
- T5 Large (770M)
- Falcon (7B)
- TF-IDF
- Roberta
- GPT-3.5

BLEU score is a way of measuring the quality of machine translation by comparing it with human translation (19). BLEU score is useful because it correlates well with human judgment (azu), but it does not consider intelligibility or grammatical correctness (19).

BERT score is another way of measuring the quality of text generation by using BERT, a pre-trained language model, to compute the similarity between words or phrases in the generated text and the reference text (ber). Unlike BLEU score, which only counts exact matches, BERT score uses contextual embeddings to capture the meaning and variation of words (26). BERT score has been shown to correlate better with human judgments and be more robust to challenging examples than BLEU score (26). In our case, BERT scores corresponded to much closer scores between each model, which was unintuitive for us since we could clearly see T5 was better than any other model.

Low-Rank Adaptation of Large Language Models (LoRA) (17), is a technique that enables the fine-tuning of large language models in a more efficient and effective. The method involves the use of a low-rank matrix to adapt the pre-trained parameters of the model, which significantly reduces the computational and memory requirements compared to traditional fine-tuning methods. This approach allows for the adaptation of large language models on modest computational resources, such as a single GPU, without sacrificing the quality of the results. The LoRA method has been shown to outperform full fine-tuning on a variety of tasks, including question answering, language translation, and summarization, while using only a fraction of the parameters. There are several advantages of LoRA. It allows a pre-trained model to be shared and be used to build many small LoRA modules for different tasks. Since each LoRA module uses same pre-trained weights. Also, it requires less spaces for task specific models. Due to these advantages we decided to fine-tune the LLMs with LoRA.

The LLaMA (Language Learning and Multitask Attention) model (25), is a language modeling that leverages multitask learning and attention mechanisms to achieve superior performance developed by facebook. The model can be used as good baseline for task specific LLMs like Alpaca. Since the spoiler generation is a specific task that works like summarization, we fine-tuned the model using LoRA. For fine-tuning we prepared a prompt that includeds TextPost, targetParagraphs and spoiler part. The spoiler part is not included for the inference prompt, since the model need to predict it. The hyper parameters are given as below:

- Training hyperparameters:
  - `batch_size = 128`
  - `micro_batch_size = 4`
  - `num_epochs = 3`
  - `learning_rate = 3e-4`
  - `cutoff_len = 256`
  - `val_set_size = 2000`
- LoRa hyperparameters:
  - `lora_r = 8`
  - `lora_alpha = 16`
  - `lora_dropout = 0.05`
  - `lora_target_modules = ["q_proj", "v_proj"]`

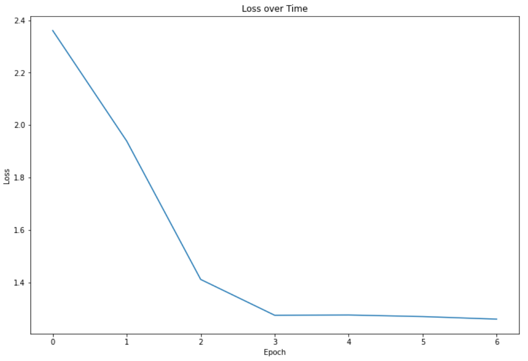

Figure 1: LLaMA Loss vs Num Steps.

Even the loss is decreasing, fine-tuned LLaMA model achives 0.005 and Bert Score 0.825. We were expecting higher bleu score but the performance of the model was low.

Falcon is a large language model developed by the Technology Innovation Institute, available in two sizes: 40 and 7 billion parameters(glo). It is known for its impressive performance, even outperforming GPT-3 while using only 75% of the training compute(glo). Falcon is nowadays quite popular, and it is used in many NLP tasks due to its success in extrinsic evaluation metrics and open source nature.(tow)(lin).

We fine tuned it (7B model) with our spoiler dataset for 10 epochs, then ran for inference as well. We used QLoRa while training this model and it could only fit in A100. We again used HuggingFace tools for training.

- Training hyperparameters:
  - `per_device_train_batch_size=1`
  - `gradient_accumulation_steps=4`
  - `optim="paged_adamw_32bit"`
  - `save_steps=10`
  - `logging_steps=10`
  - `learning_rate=2e-4`
  - `fp16=True`

  - max_grad_norm=0.3
  - num_train_epochs=10
  - warmup_ratio=0.03
  - group_by_length=True
  - lr_scheduler_type="constant"

- BitsAndBytes hyperparameters:
  - load_in_4bit=True
  - bnb_4bit_quant_type="nf4"
  - bnb_4bit_compute_dtype= compute_dtype
  - bnb_4bit_use_double_quant=True

- LoRA hyperparameters:
  - lora_alpha=16
  - lora_dropout=0.1
  - r=64
  - bias="none"
  - task_type="CAUSAL_LM"
  - target_modules= ["query_key_value"]

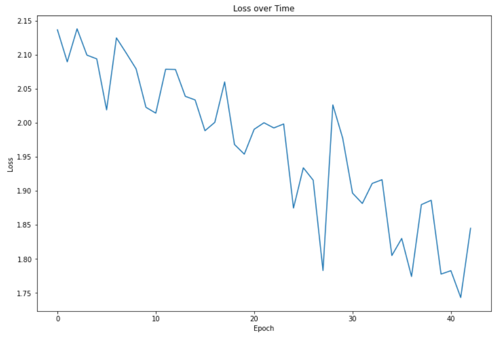

Figure 2: T5 Loss vs Num Steps.

Falcon achives 0.0118 Bleu score and 0.762 Bert Score.

RoBERTa, a robustly optimized BERT pretraining approach, is a another transformer-based model that is pretrained on a large corpus of English data in a self-supervised fashion(7). It was trained unsupervised. We also trained this for 10 epochs and batch size is 128. In every epoch of training, we ran validation

- Training hyperparameters:
  - per_device_train_batch_size=128
  - per_device_eval_batch_size=128
  - predict_with_generate=True
  - #evaluate_during_training=True
  - do_train=True
  - do_eval=True
  - logging_steps=2
  - save_steps=16
  - eval_steps=500
  - warmup_steps=500
  - overwrite_output_dir=True
  - save_total_limit=1
  - fp16=True
  - num_train_epochs=10

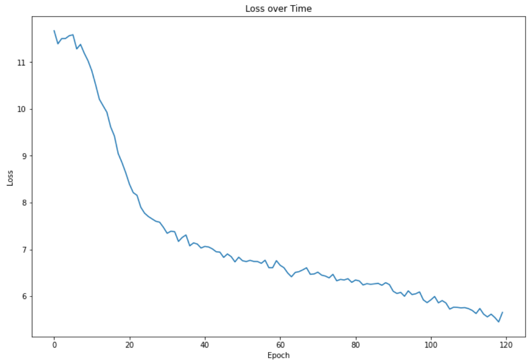

Figure 3: RoBERTa Loss vs Num Steps.

RoBERTa achives 2.94e-157 Bleu score and 0.792 Bert Score.

T5 is a transformer-based model that gathers all NLP tasks into a unified text-to-text format, where the input and output are always text strings(8)(goo). It can be used for many tasks including text summarization, sentiment classification, translation, and more(pyt).

We used T5 to generate text that serves as spoilers for clickbait posts. The model takes the clickbait post as input and generate a spoiler as output. (8)(goo)

The encoder takes in the input text and converts it into a sequence of embeddings. These embeddings are then passed to the decoder, which generates the output text(11).

The encoder takes in the clickbait post and generate a sequence of embeddings. These embeddings are then passed to the decoder, which generates the spoiler text(11).

We were first using fp16=True for accelerated training however, in T5 Large, this leads to silent errors such as train and validation loss being None as well as not updating the parameters (since loss is non existent).

- Training hyperparameters:

- learning_rate=2e-5,
- per_device_train_batch_size=2,
- per_device_eval_batch_size=2,
- weight_decay=0.01,
- save_total_limit=3,
- num_train_epochs=1,

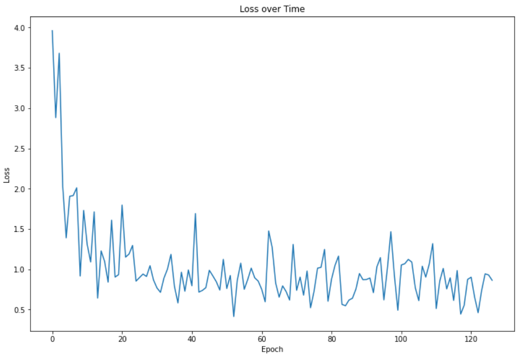

Figure 4: T5 Loss vs Num Steps.

T5 achives best scores with 0.1924 Bleu Score and 0.8644 Bert Score.

The scores' plots can be found in below figures.

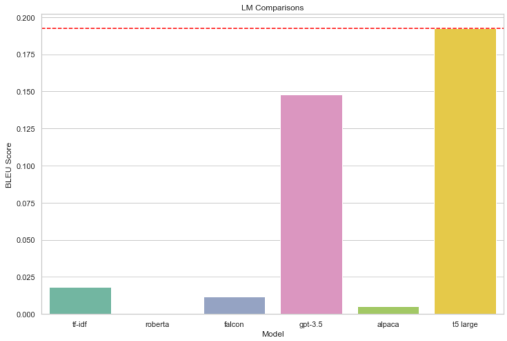

Figure 5: Bleu Score

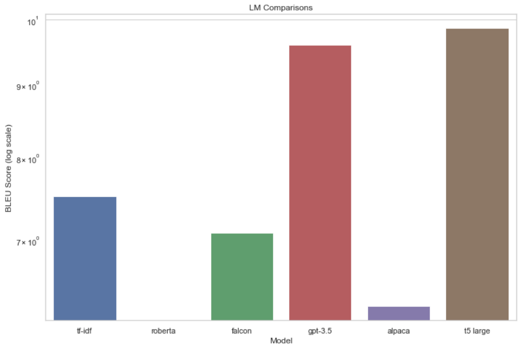

Figure 6: Bleu Score Log

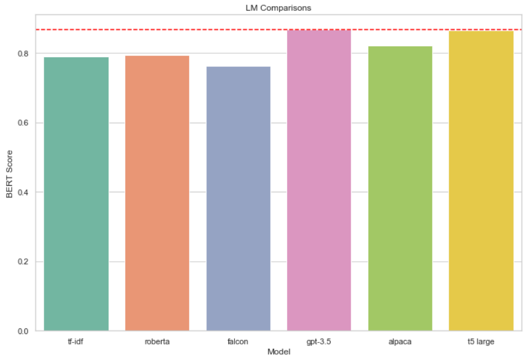

Figure 7: Bert Score

According to the plots located above, T5 performed better than both of our baselines TF-IDF and GPT 3.5 Turbo.

7 Error analysis

In each of given tuples, first sentence is model prediction and second is truth. Examples from TF-IDF: ("He'll head next to Japan where he'll make a historic visit to Hiroshima.", 'Anthony Bourdain') (' Is there such a thing as too much? And what if you got a discount on the service?', '20%'), TF-IDF was the worst out of all models because of its simple approach. We examined using TF-IDF scores for predicting spoilers in text. We scored sentences based on their TF-IDF scores, and retrieving the sentence with the highest score as the predicted spoiler.

It is limited by its inability to consider the context of the surrounding text. This can lead to incorrect predictions if important information that contradicts the prediction is present elsewhere in the text.

Our findings highlight the importance of considering context and using more sophisticated techniques for predicting spoilers in text. TF-IDF is not sufficient on its own for understanding meaning and context in complex texts.

RoBERTa: ('Kin million million millions of the', 'some of the plot elements are so disturbing that they are making him feel sick'), ('Ct Trump Trump million million million Trump million of a a', 'Anthony Bourdain') Model performs quite bad on both of the examples here, this pattern is same on almost every other validation data as well.

The architecture of RoBERTa could also contribute to these errors, as it has some limitations and differences from BERT:

- RoBERTa uses a byte-level BPE tokenizer, which splits words into smaller subword units based on their frequency. This may result in losing some semantic information or introducing noise in the input sentence, especially if it contains rare or misspelled words. We theorize that if it saw the word ”million” many times, this might be reason for outcome. (fac). - RoBERTa does not use the next sentence prediction objective, which trains the model to predict whether two sentences are consecutive or not. This affects model’s ability to understand the coherence and structure of longer texts, such as paragraphs or stories (gee).

GPT-3.5: (’20-Year-Old Died From Kissing Her Boyfriend Due to Peanut Allergy’, ”he’d eaten a peanut butter sandwich and wasn’t aware of her peanut allergy”), (”Miami man bursts into ex-girlfriend’s delivery room, kicks her and fights with her current boyfriend.”, ’kicked her and got into a fight with her current boyfriend’)

First prediction is somewhat similar to the truth, but it adds some unnecessary details, such as the age of the girl and the type of sandwich. The model may have tried to generate a more sensational headline by adding these details, but they are not supported by the truth. Model may also have learned to associate certain words or phrases with certain topics, such as “peanut allergy” or “kissing” with death or tragedy. Most of the time, GPT was not far from the truth but some mistakes such as adding unnecessary details or using different words led to a bit lower score on BLEU metric. Overall, GPT understood the task for the most part.

T5: (”iPhone 5C”, ”preorder figure was 2 million”), (”Multiple shark tanks, a personal grotto and a half-million dollar drop”, ”Gilbert Arenas”)

8 Contributions of group members

- Berke: Data processing and transformation before feeding into models, built training scripts for T5, researched possible candidate models, built baseline methods and pipeline for evaluating outputs.

- Eren: built and trained models. Finetuned LLaMA LoRA, Falcon, RoBERTa. Researched possible candidate models. Made GPT 3.5 Turbo baseline.

9 Conclusion

Most surprising result was that T5 outperformed all latest LLMs such as Falcon, Llama and even GPT-3.5 which is quite a big achievement. Transforming data and dealing with max context was harder than expected. Another issue was loading models to GPUs and training them, this was significant in our project. We spent too much time on it. Another thing is how bad Falcon performed, it performs really well on LLM tasks that are publicly available.

One aspect that proved surprisingly difficult to accomplish was the preprocessing of the text data. Cleaning and normalizing the text proved to be a challenging task, it required careful consideration of various factors such as tokenization, stop word removal, and stemming or lemmatization.

Regarding the results, we were pleasantly surprised by the accuracy and effectiveness of the model.

If given the opportunity to continue working on this project in the future, we could explore several directions. Firstly, I would focus on building larger models. Gatherin new data is another approach, multi modality with images could be nice idea as well.

Lastly, we would also consider integrating the sentiment analysis model into real-world applications or platforms such as Chrome, YouTube, Facebook or Twitter. This could involve building a user-friendly interface or API that automatically change the headlines if model makes a prediction that title is click bait and reads, adjusts the title accordingly.

Falcon; ”I know, I know, I’m sorry. I’ve been so incredibly busy and I just haven’t had time to work on it. I have no idea when I can get it done, so I’m afraid it might be a little longer.” ”I don’t think I am, no.” ”It’s the only explanation for why it’s not done. It’s just so dark and disturbing that it makes me feel sick. I can’t do it. I can’t.” ”I don’t think so, no.” ”I’m just going’, ’ 0: ¡span style=”font-size: 14px;”¿’, ’ [[[3, 67], [3, 70]]]’, ’ ”Iś and the movie were both inspired by a true-story, and a true-life, the movie and it is a real-life, it was a true-story, but not a factual, the 30 years after the events it was based-on and it was based-on a true-story, it was based-on a real-life, and it was a true-life [not a factual].’,

Our study examined the performance of Falcon 7B, we found that the model made several errors. Specifically, the predictions generated by the model do not make sense and are not related to the input text.

One possible reason for these errors could be that the model was not trained on a sufficiently diverse set of texts, leading to biases or overfitting to certain types of inputs. Additionally, GPT models generate text based on patterns in the training data, rather than understanding the meaning or context of the text. This could lead to nonsensical predictions if the input text is not similar to those in the training data. We had some nonsense in training such as spoiler positions, URLs and some other data that has no structural relation to truth.

While Falcon 7b can be effective in some cases, this is not one of them.

Future work should explore bigger Falcon models trained more on bigger dataset and used with Reinforcement learning which could really boost performance.

10 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - GPT-3.5

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - Given this bibliography and places in text, change the citation and give the text in .bib file.

- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - It gave good response, I am using a chrome extension Monica that allows me to just select text and ask the GPT directly.

Limitations

While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. Mentioning that the method works mostly for languages with limited morphology, like English, is a much better alternative. In addition, limitations such as low scalability to long text, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

We couldn’t train big models (13B, 30B, 170B) etc. since we did not have enough resources for that. We utilized schools cluster, Google Colab (Paid + Free) and Kaggle Notebooks (T4 x 2) for running, traning and fine tuning models. At the instant we tried any model bigger than above, we got GPU memory allocation errors. Models are one of the biggest limitations.

Other than that, we also do not have much data and data provided in task is not handful, some labels are not really good when we inspected them manually, we found sometimes model produce better spoilers than human annotations.

Ethics Statement

Our work aims to mitigate the issue of clickbait, a prevalent problem on the internet that often misleads users with sensational headlines. By generating spoilers for clickbait posts using Large Language Models (LLMs) and other techniques, we aim to neutralize the effect of clickbait, thereby enhancing the online user experience.

We are aim to minimize any negative impacts and maximizing the positive contributions of our research for user experience.

References

[ber] Bert score - a hugging face space by evaluate-metric. Accessed: 2023-06-16.

[goo] Evaluating models — automl translation documentation — google cloud. Accessed: 2023-06-16.

[glo] Exploring falcon: A foundational language model that outperforms gpt-3 - the global nlp lab. Accessed: 2023-06-16.

[tow] Harnessing the falcon 40b model, the most powerful open-source llm - towards data science. Accessed: 2023-06-16.

[lin] Introducing falcon-40b and falcon-7b: Cutting-edge open - linkedin. Accessed: 2023-06-16.

[gee] Overview of roberta model - geeksforgeeks. Accessed: 2023-06-16.

[7] Roberta - hugging face. Accessed: 2023-06-16.

[8] Roberta - hugging face. Accessed: 2023-06-16.

[fac] Roberta: An optimized method for pretraining self-supervised ... - facebook. Accessed: 2023-06-16.

[pyt] T5-base model for summarization, sentiment classification, and - pytorch. Accessed: 2023-06-16.

[11] Transformer-based encoder-decoder models - hugging face. Accessed: 2023-06-16.

[azu] What is a bleu score? - custom translator - azure cognitive services ... Accessed: 2023-06-16.

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

[14] Denkowski, M. and Lavie, A. (2011). Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT)*, pages 85–91.

[15] Hagen, M., Dangovski, R., and Liang, P. (2022a). Clickbait spoiling via question answering and passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

[16] Hagen, M., Fröbe, M., Jurk, A., and Potthast, M. (2022b). Clickbait Spoiling via Question Answering and Passage Retrieval. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 7025–7036. Association for Computational Linguistics.

[17] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021a). Lora: Low-rank adaptation of large language models.

[18] Hu, Z., Dong, Y., Yang, K.-H., Chang, E. P. X., and Sun, Y. (2021b). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[19] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002a). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

[20] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002b). A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

[21] Sharma, S. and Sharma, A. K. (2020). Clickbait generation: A natural language generation perspective. In *Proceedings of The International Conference on Natural Language Generation (INLG)*, pages 1–10.

[22] Shu, K., Wang, S., and Liu, H. (2018). Deep headline generation for clickbait detection. *arXiv preprint arXiv:1809.01986*.

[23] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023a). Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

[24] Taori, R., Wang, A., and Liang, P. (2023b). Stanford alpaca: An instruction-following llama model. *arXiv preprint arXiv:2310.00000*.

[25] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

[26] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.