On LLM Augmented AB Experimentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Automated experimentation methods to evaluate user preferences and engagement is a key cornerstone in the current digital landscape. Most such systems rely on marketers and creators to design the content before deployment. However, with the advent of Large Language Models (LLMs) the feedback cycle is considerably shortened while the experimentation space expands significantly, necessitating novel and efficient ways to assess user engagement. In this paper, we experiment with using LLMs as simulators or treatment raters in an A/B testing application without running an A/B test.

1 Introduction

Widespread adoption of mobile devices and increased internet access has led to a significant increase in digital content consumption. To maximize customer engagement, businesses constantly aim to optimize the content and user experience. For example, news media industries constantly strive to come up with attractive headlines and cover images (Coenen, 2019) to drive customer engagement. The standard practice to find attractive headlines is to use A/B testing. However, this is inefficient for applications surrounding social-media, news and related sectors; as news and trends have short lifetimes and might become irrelevant by the time a standard A/B test finishes. This problem is further aggravated due to significant democratization of content creation, which has led to shorter feedback cycles and increasing amount of content which needs to be experimented. Thus, in industries, where newer content constantly comes up, there is a great need for more-efficient engagement evaluation.

Large language models (LLMs) have been demonstrated to have significant potential for processing natural language text, following human instructions and generating high-quality responses (OpenAI, 2024). This has spurred their use in many applications such as tool learning (Qu et al., 2024) and information retrieval (Zhu et al., 2024b). Given that LLMs have even demonstrated the ability to mimic human preferences and behavior in a variety of consumer research tasks (Li et al., 2024; Brand et al., 2023); a natural question is 'how useful LLMs can be for content optimization?'.

Contribution In this work, we focus on the problem of using Large Language Models (LLMs) to bypass current A/B testing practices. Specifically we focus on using LLMs to identify appealing content. For concreteness, we will consider writing headlines for articles as our running example. As such we will use the terms content/article/prompt and the terms treatment/headline interchangeably. LLMs can be used in multiple ways for the purpose of rating treatments. In this paper, we explore in-context learning, embedding based methods, and generative model based evaluation using wo benchmark datasets from real-life A/B tests. In our experiments, we find that using LLMs as few-shot learners for treatment rating is significantly less effective than training models using LLM-based representations. The accuracy of using in-context learning is only slightly higher than random guessing. Furthermore, for methods which use LLM-based embeddings, the accuracy is not high enough to be used as a standalone treatment evaluation method. Finally, we tried to use a generative

Submitted to NeurIPS 2024 Workshop on Causality and Large Models (CaLM). Do not distribute.

approach by fine-tuning the LLM to produce more engaging headlines. We found using such a tuned generative model to be a more promising methodology for rating headlines.

2 Preliminaries

2.1 Learning from A/B Test

The language model is considered as a policy function π which observes a prompt x and produces a textual response a by sampling from a distribution $y \sim \pi(\cdot \mid x)$. We are given a dataset of $\mathcal{D}_{\text{pref}} = \{(x, a^{+}, a^{-})\}$ of prompts and labeled response pairs. Here, a^{+} is a positive response and a^{-} is a negative response. Consider the example of A/B testing different summaries or headlines for a given content. The preference data is obtained by exposing the incoming traffic to one of two possible treatments/headlines (a or b) and the effective engagement (measured as clicks, screen time or any other chosen metric) was monitored. The option with higher engagement is considered as the positive sample a^{+} while the other is considered as a^{-}.

Offline RLHF (Christiano et al., 2017; Ye et al., 2024a; Ouyang et al., 2022) deals with the problem of aligning a policy network, using $\mathcal{D}_{\text{pref}} = \{(x, a^{+}, a^{-})\}$. Given the context/prompt x, a pair of outputs are sampled from $\pi_{\mathsf{ref}}(\cdot \mid x)$ and then arranged as per preference function (typically implicitly given by human annotation). RLHF methods (Christiano et al., 2017; Ouyang et al., 2022) seek to obtain a policy $\hat{\pi}$ that is aligned with the preference data. This is done, by first estimating a reward function r from $\mathcal{D}_{\text{pref}}$ using maximum likelihood. Then one uses RL based optimization methods like PPO to maximize the learnt reward with an additional regularization term.

$$\hat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \mathbb{E}_{\pi} \left[r(x, y) - \beta \log \frac{\pi(a \mid x)}{\pi_{\mathsf{ref}}(a \mid x)} \right].$$

Overoptimization The phenomena of overoptimization and reward-hacking in alignment literature is well documented (Guo et al., 2024; Song et al., 2024). This problem can be alleviated when access to the underlying system is available, as data with the policy can be collected from the policy as it gets optimized (Gao et al., 2024; Guo et al., 2024). However, in the context of A/B testing, these methods are inapplicable, as the only way to collect data from the newer policy is deploying it in the field, i.e. another A/B test which defeats the purpose of using LLMs to bypass A/B testing.

2.2 Related Work

Researchers are increasingly trying to utilize LLMs for emulating human behaviour (Ziems et al., [n. d.]; Kim and Lee, 2023; Park et al., 2023). The idea of using AI agents to simulate users has a long history of research in information systems(Carterette et al., 2011; Mostafa et al., 2003). LLM based user simulators has been studied for evaluating task-oriented dialogue systems and recommender systems (Balog and Zhai, 2023, 2024). Chen et al. (2024) have demonstrated the potential of using self-play between LLMs for developing recommendation systems. Recent works have also suggested using LLM based models to warm start bandit based methods (Ye et al., 2024b) for A/B testing. However, concerns about the reliability of such simulations have also been raised (Zhu et al., 2024a).

3 Our Work

3.1 Direct Evaluation with LLM

LLMs have proven themselves to be good as both *embedding models* (Ethayarajh, 2019) and *task learners* (Brown et al., 2020). We consider both of these possible ways to develop LLM based baselines for rating treatments/headlines.

- Direct Prompting: We treat the LLM as an evaluator, provide it the article in the prompt and instruct it to rate the different headlines as more engaging. This effectively uses the LLM as a zero-shot classifier, and can directly measure the accuracy. We call this method *PromptOnly*.

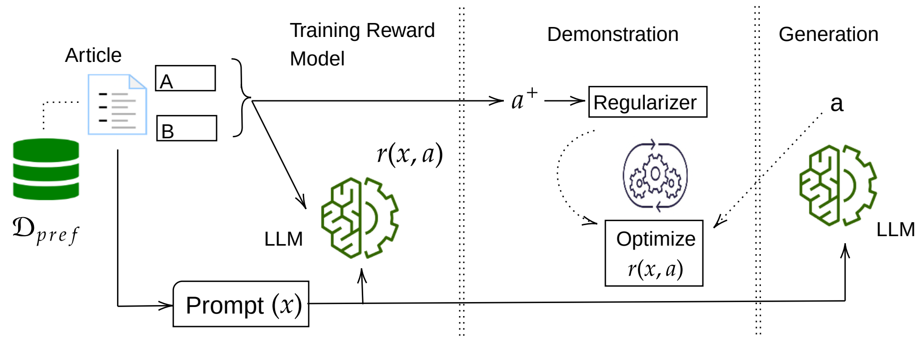

Figure 1: Overview of the proposed generative approach. The reward model r is obtained by tuning an LLM on the preference data $\mathcal{D}_{\text{pref}}$, which consist of tuples of contexts/articles along with two treatments arms (a^+, a^-). Given a prompt x (which includes the context/article along with instructions) the generator LLM produces an output a. The pair x, a^+ is considered as a demonstration for the generator to match and improve using the reward model r.

- In-context learning: Similar to direct prompting except that the LLM is also provided with a few in-context examples (or demonstrations) to learn from and choose the correct answer. We refer to these as *ICL* methods.
- Blackbox Embedding: We train an MLP based which used the LLM embeddings of the combined text of the article and headline to pick the better answer.
- Finetuning: We fine-tune an opensource LLM based on the data, to prick the better answer. This is similar to the blackbox embedding approach, except that has a frozen LM, whereas we allow the LM to be updates. Furthermore as both our compute resources and the amount of data is limited, we take LORA (Hu et al., 2022) approach. We call these FT methods.

In the experiment section, we will present the results from all of these methods. We found that prompting and in-context learning based methods are significantly worse (< 65%) than fine-tuning based approaches (∼80% accuracy). These results are qualitatively in line with other recent works focusing on using prompting and ICL based methods to classify content (Zhou et al., 2024).

3.2 Generative Evaluation with LLM

We also propose a method based on finetuning an LLM to generate engaging treatment arms (or content) using the results from the logged A/B testing data. Note that the goal here is not necessarily to use the generative model to generate new treatments, but instead use it to rate content based on model likelihood. We use RLHF (Ouyang et al., 2022) as our starting point. However we found that this model can overfit easily and is only slightly better than few-shot learning based approach. As such we modify the standard RLHF procedure to address the overfitting caused by nuances specific to the A/B testing. The overall schematic is presented in Figure 1 Most of the ideas in our approach can also be applied with other learning paradigms such as DPO (Rafailov et al., 2024), and CPO (Xu et al., 2024).

Ensemble reward model In the direct evaluation methods, we found the ensemble model from GPT embeddings to be a cheap and accurate model in predicting positive treatment arms. As such we leveraged GPT embeddings to train a reward model. However, following (Coste et al., 2024), we created an ensemble model E with different subsets of the data to help reduce reward overoptimization.

Regularizing Objectives Compared to the standard RLHF framework of (Ouyang et al., 2022) we make the following changes to the loss objective:

- We include an additional term of $\frac{\pi(a|x)}{\pi_{\text{ref}}(a|x)}$ as a regularizer in the objective. This terms more strongly penalizes deviations of π from π_{ref} than just the KL divergence. An astute reader might also note that this term is equivalent to regularizing with the order-2 Tsallis divergence.

- We also prevent the model from exploring a space which might be far from the space where the reward model is certain. Since $\mathcal{D}_{pref}$ forms the training data for the ensemble reward model, a y which is too far from the training data i.e. $P(y|x) << P(\mathcal{D}_{pref}|x)$ is low, can be considered to be a situation where the reward model is unreliable. Penalizing with the corresponding density ratio prevents the model from going too far out of the training support of the reward model. In our experiments, we estimate this ratio via prompting. Specifically, we provide GPT-3 with a prompt that describes the articles and example prompts, then follow the prompt with the current text sample to estimate the support the sample may have in data. We clip the log-density-ratio at δ to avoid drawing only training samples.

Combining these we get the following maximization objective

$$\hat{\pi} = \underset{\pi \in \Pi}{\operatorname{argmax}} \, \mathbb{E}_{\pi} \left[r_m(x, a) - \beta \log \frac{\pi(a \mid x)}{\pi_{\mathsf{ref}}(a \mid x)} - \beta \frac{\pi(a \mid x)}{\pi_{\mathsf{ref}}(a \mid x)} + \lambda \log_{+} \left(\frac{P(a \mid x)}{P(\mathcal{D} \mid x)}, \delta \right) \right] \tag{1}$$

where $r_m(x, a) = \frac{1}{E} \sum_{r \in E} r(x, a)$ is the ensemble reward, P is given by GPT probabilities, $P(\mathcal{D}|x) = P(a^{+}|x) + P(a^{-}|x)$, $\log_{+}$ is the clipped $\log$ functio, and β, λ are hyperparameters.

4 Experiments

Datasets We experiment with two public datasets obtained from real-life A/B testing scenarios.

- Upworthy: This dataset records a sample of A/B tests conducted by Upworthy Matias et al. (2021). The data consists of several versions of headlines created by an editorial teams for various articles. We only considered text only content and restricted to those treatments/headlines which were assessed to have statistically different CTRs (at p=0.10).
- Tweet Popularity: Tan et al. (2014) studied the effect of wording of a statement on retweeting. Their tweet popularity dataset is similar to an A/B test with a total of 13k tweet pairs, which are matched by the topic and the author; where the positive sample is considered as the one to receive more retweets. We apply a similar pre-processing as Upworthy.

Evaluation Assessing performance of direct models is straightforward, we simply analyse whether the model correctly classified test set examples. Evaluating the generative model is more nuanced. The generative method is assessed based on whether the likelihood of the positive answer is higher than of the negative answer i.e. whether $\pi(a^{+} \mid x) > \pi(a^{-} \mid x)$.

Table 1: Accuracy for different approaches on the different A/B testing datasets. † represents generative models evaluated on better treatment's likelihood. * denotes that the prompts triggered a safety check which were ignored in accuracy calculation

Model	Upworthy	Tweet
GPT-4 PromptOnly	56.6	47.1 *
Claude PromptOnly	58.1	49.6*
Llama-3-8b PromptOnly	55.7	45.3
GPT-4 ICL	64.2	61.3*
Claude ICL	60.1	56.8*
Llama-3-8b ICL	60.7	58.5
OpenAI text-embedding-3-large	82.5	79.9*
Llama-3-8b embedding	74.0	76.5
FT Single	82.8	79.4
FT Ensemble	83.6	80.2
DPO	72.8†	76.1†
Ours	**84.5**†	**81.6**†

Results Our results are presented in Table 1. From these results we can see that prompting based methods, both direct prompting and few-shot learning, are just a little better than average guessing. Specifically we see no model better than ($\sim 65\%$) accuracy. We also found that giving a few examples for ICL leads to slightly better performance than pure prompt based method ($\sim 60\%$ vs $\sim$55%).

Next we also see that methods based on training a model on the data using LLM as representation functions performs much better. With most embedding models we see accuracy of 74% or higher, which is significantly better than prompt based models. However, the best performance was obtained by fine-tuning these models; Fine-tuned Llama models outperforms GPT embedding models. These results are consistent between both the Upworthy and

Tweet dataset. We also note that the tweet dataset contain samples which triggered safety violations. With GPT/Claude we excluded these results, and so these numbers are not exactly comparable to each other.

Finally, we tried to rate content by first training an LLM to produce more engaging content, and using its likelihood as a measure of rating. The results indicate that by suitably training the Llama model to align with the preferences implicitly given by our dataset, we can match or outperform all the earlier approaches. This suggests that LLMs can potentially be used as generators of the treatments for an A/B test. However since we did not perform a human evaluation of its outputs, and more research is needed in this direction.

Analysis We delve further into the behaviours of the different models. As a representative of the direct evaluation method we chose the GPT based embedding model, and compared it to the generative model described earlier. We focus on the Upworthy dataset here as we have significantly more number of tests than Tweet. In Figure 3 we plot the calibration chart i.e. a comparison of the model accuracy and the predicted probability of treatment A better than treatment B. For the generative approach we normalized the probability of the two considered options instead of using the output likelihood. The ideal line is of a an oracle calibrated model whose output probabilites will match its accuracy. From Figure 3 one can see that the embedding model is overconfident in its predictions, and modern neural networks are known to suffer from this (Caruana et al., 2015; Guo et al., 2017). However, surprisingly the generative approach seems conservative in its predictions.

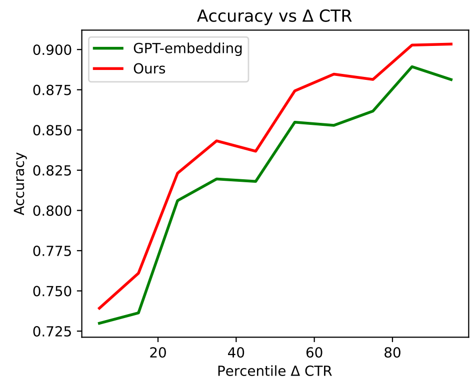

Figure 2: Plot of model accuracy against the absolute value of the mean difference in click rates i.e. $r(a^{+}, x) - r(a^{-}, x)$. Both models performs better when the underlying rates are different, but the generative approach outperforms the embedding based model.

We further analyse how the model performance varies across the difficulty of samples. Difficulty in this context is measured based on the difference in click-through (CTR) rates. In terms of downstream impact, having the right decision when the underlying click rates are different, is more important than when differences are lower. In Figure 2, we plot the accuracy of the the GPT3 embedding model against the percentile of the click rates. We can see that both models are more accurate as the underlying mean difference increases. This supports the idea that the LLM based evaluation can supplement A/B testing at low risk, as it is less error-prone when underlying costs of error are higher.

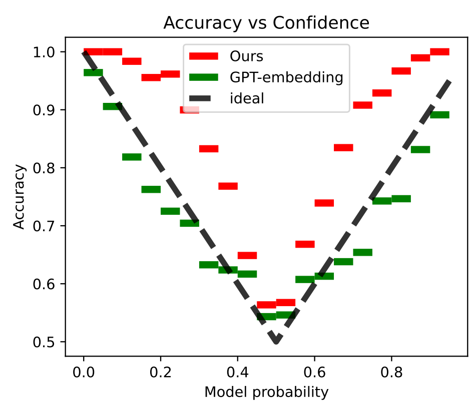

Figure 3: Model accuracy vs model confidence for the generative and embedding approaches. The embedding approach is overconfident while generative model is underconfident in picking the better arm.

5 Conclusion

We propose an approach to leverage LLMs for content experimentation in digital platforms. We first examined how well LLMs can predict appealingness of content. First we find that purely prompt based methods improve over random chance only by a small factor, suggesting that these methods are not suitable for predicting engagement. We also try fine-tuning based approaches to classify content and find that these are significantly better. Next, we try to see whether an LLM fine-tuned to produce engaging content can be used to rate the treatments. We find that suitably regularized generative model performs better than the best fine tuned ensemble models.

References

Krisztian Balog and ChengXiang Zhai. 2023. User simulation for evaluating information access systems. In *Proceedings SIGIR*. 302–305.

Krisztian Balog and ChengXiang Zhai. 2024. User Simulation for Evaluating Information Access Systems on the Web. (2024).

James Brand, Ayelet Israeli, and Donald Ngwe. 2023. Using gpt for market research. *Available at SSRN 4395751* (2023).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Neurips* (2020).

Ben Carterette, Evangelos Kanoulas, and Emine Yilmaz. 2011. Simulating simple user behavior for system effectiveness evaluation. In *Proceedings of CIKM*.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1721–1730.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models. arXiv:2401.01335 [cs.LG] https://arxiv.org/abs/2401.01335

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences.

Anna Coenen. 2019. How The New York Times is Experimenting with Recommendation Algorithms. https://open.nytimes.com/how-the-new-york-times-is-experimenting-with-recommendation-algorithms-562f78624d26

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2024. Reward Model Ensembles Help Mitigate Overoptimization. arXiv:2310.02743 [cs.LG] https://arxiv.org/abs/2310.02743

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing BERT, ELMo, and GPT-2 Embeddings. In *EMNLP)*. 55–65.

Zhaolin Gao, Jonathan D. Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J. Andrew Bagnell, Jason D. Lee, and Wen Sun. 2024. REBEL: Reinforcement Learning via Regressing Relative Rewards.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. 2024. Direct Language Model Alignment from Online AI Feedback. arXiv:2402.04792 [cs.AI] https://arxiv.org/abs/2402.04792

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Junsol Kim and Byungkyu Lee. 2023. AI-Augmented Surveys: Leveraging Large Language Models and Surveys for Opinion Prediction. *arXiv preprint arXiv:2305.09620* (2023).

Peiyao Li, Noah Castelo, Zsolt Katona, and Miklos Sarvary. 2024. Frontiers: Determining the Validity of Large Language Models for Automated Perceptual Analysis. *Marketing Science* (2024).

J Nathan Matias, Kevin Munger, Marianne Aubin Le Quere, and Charles Ebersole. 2021. The Upworthy Research Archive, a time series of 32,487 experiments in US media. *Scientific Data* 8, 1 (2021), 195.

Javed Mostafa, Snehasis Mukhopadhyay, and Mathew Palakal. 2003. Simulation studies of different dimensions of users' interests and their impact on user modeling and information filtering. *Information Retrieval* 6 (2003), 199–223.

OpenAI. 2024. New embedding models and API updates. `https://openai.com/index/new-embedding-models-and-api-updates/` Accessed: 2024-05-28.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of ACM USIT*.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Tool Learning with Large Language Models: A Survey. arXiv:2405.17935 [cs.CL] `https://arxiv.org/abs/2405.17935`

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: your language model is secretly a reward model *(NIPS '23)*.

Yuda Song, Gokul Swamy, Aarti Singh, J. Andrew Bagnell, and Wen Sun. 2024. The Importance of Online Data: Understanding Preference Fine-tuning via Coverage. arXiv:2406.01462

Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter. arXiv:1405.1438 `https://arxiv.org/abs/1405.1438`

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation. arXiv:2401.08417 [cs.CL] `https://arxiv.org/abs/2401.08417`

Chenlu Ye, Wei Xiong, Yuheng Zhang, and Tong Zhang. 2024a. Online Iterative RL from Human Feedback with General Preference Model. arXiv:2402.07314

Zikun Ye, Hema Yoganarasimhan, and Yufeng Zheng. 2024b. LOLA: LLM-Assisted Online Learning Algorithm for Content Experiments. arXiv:2406.02611

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. 2024. Hypothesis Generation with Large Language Models. arXiv:2404.04326 [cs.AI] `https://arxiv.org/abs/2404.04326`

Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024a. How Reliable is Your Simulator? Analysis on the Limitations of Current LLM-based User Simulators for Conversational Recommendation. arXiv:2403.16416

Yutao Zhu, Huaying Yuan, Shuting Wang, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2024b. Large Language Models for Information Retrieval. arXiv:2308.07107 [cs.CL] `https://arxiv.org/abs/2308.07107`

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. [n. d.]. Can large language models transform computational social science? *Computational Linguistics* ([n. d.]).

Appendix

The prompt used for in-context learning is given in Example 1. Instructions from this are also used in the context when tuning the models.

```
Instruction Prompt
You are an expert marketing writer for a newspaper company. You  you are excellent at choosing which
headlines are likely to get more clicks for an article.
You will be given an article context and two headlines, from which you determine which headline was
clicked more often.
You are given the headlines as "Headline _" where _ is either 1 or 2.  Give your final answer in the
following format:
"Answer: Headline _"

User Prompt
Here are some previous examples to help you:
··· more examples here ···
Which of the following headlines has more clicks:
Article: <context>
Headline 1: <headline_1>
Headline 2: <headline_2>

Think step by step, and explain your reasons
Step 1:  Look at the new pair of headlines and compare them with the examples associated with each
pattern.Step 2:  Find the set of examples that is closest to the given pair of headlines, and pick the
pattern associated with that set of examples.
Step 13: Think about which one out of the pair of headlines will get more clicks.
Step 14: Give your final answer.
```

Example 1: Zero/Few-shot Inference.

We further analyse how the model performance varies across the the statistical significance of the difference in the CTR rates [1]. The significance score is done by a Welch-t test (two-sample uncommon variance t-test).

This is different from Figure 2 as difficulty in this context is measured based on the statistical significance of the difference in the CTR rates. A high difference in click through rates need not mean high significance, as the difference is adjusted for the variance and/or the number of impressions for computing the significance. Note that since we already filtered out non-conclusive tests, we are considering only low p-value samples. The result of accuracy on the test-set for is presented in Figure 4. We can see that both models in general are more accurate as the significance increases (p-value decreases).

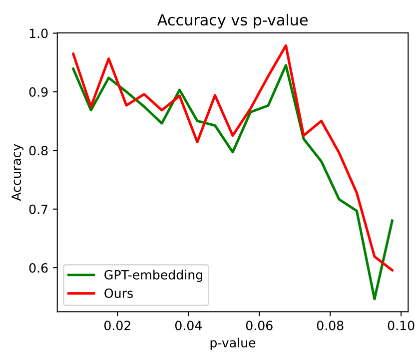

Figure 4: Plot of model accuracy against the percentiles of the click-rates difference of different arms.

[1]Since the significance score is also dependent on number of impressions which get influenced by the experimenter decisions, the difference in rates is not an ideal measure of difficulty.