Fame Bias – Large Language Models Change Their Judgement Depending on Personal Name

Huizhong Ji[1][0009−0002−0962−6686] and Rafal Rzepka ✉[2][0000−0002−8274−0875]

[1] Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan
huizhong.ji.w4@elms.hokudai.ac.jp
[2] Faculty of Information Science and Technology, Hokkaido University, Sapporo, Japan
rzepka@ist.hokudai.ac.jp

Abstract. This paper focuses on bias caused by changing personal names in the input of proprietary LLMs, fine-tuned language models, and lexicon-based sentiment analysis tool. It extracts examples from NLI and story cloze task datasets with personal names, replaces them with names of celebrities, ordinary people or mixtures. Results show that altered names significantly influence model assessments, especially with more than two choices, indicating potential real-world application hazards. A mitigation method for "fame bias" is proposed but the problem requires further research.

Keywords: large language models · fine-tuned language models · sentiment analysis tools · personal name bias

1 Introduction

In this paper, we examine how changing a personal name in a scenario character affects a large language model's judgement. For our experiment, we prepare two lists of names, namely *Famous Names List* and *Ordinary Name List*, extract instances containing names from a story cloze [1] and XNLI [2] datasets, and randomly replace the original names with these from our lists according to gender. The results show that name changes affect the models' output.

2 Related work

Our work can be categorized as a part of research on bias in LLMs. The closest example we found is by Jha et al. [3], who conducted a study on using large language models for spatial reasoning tasks. They generated prompts with different person names and input them into the model, such as "Samantha is on the right of Emma. Samantha is on the left of Olivia. Then, we can make a new inference that Emma is on the left of". It should be obvious that the accuracy of the model's answers do not fluctuate due to name changes. But contrary to

expectations, the choice of name significantly impacted the inference accuracy of the BLOOM large language models. Their test data were too homogeneous and did not explore fame's influence. Our experiments use more diverse and realistic scenarios to evaluate this aspect comprehensively.

3 Test Data Preparation

In this section, we provide a description of the data creation methods.

3.1 Famous and Ordinary Names Lists

For the *Famous Names List*, we have chosen famous people Wikipedia popular people page[3]. This list comprises of the top 100 famous names with a total search volume from December 1, 2007 to January 1, 2023, totaling 106 people. Subsequently, we removed some names of people known for their nicknames or stage names like "Snoop Dogg" or "Prince Philip, Duke of Edinburgh".
For the *Ordinary Names List*, we selected the top 100 family names from the Geneanet[4] website, as well as a maximum of the top ten first names for each family names. From the total of 1,000 complete names, in accordance with the gender ratio of the *Famous Names List*, we randomly selected names from the list of ordinary name, resulting in 64 ordinary male names and 36 ordinary female names. For example, the resulting sets contain *Thomas Barker* and *Jeanne Martin* as ordinary names, and *Joseph Stalin* and *Taylor Swift* as famous ones. Table 1[5] shows a portion of the list.

3.2 Test Datasets

For our experiments, we use ROCStories dataset for story cloze test and XNLI dataset for Natural Language Inference (NLI).

ROCStories [1] comprises 5,313 independent stories and this corpus is unique because it (1) captures rich causal and temporal relationships between daily events, and (2) is a high-quality collection of daily life stories useful for story generation. These advantages make it adequate for testing whether a personal name change affects a model's judgment.

XNLI [2] is a popular evaluation corpus with examples also including personal names. The developers of XNLI use fine-grained voting rules to ensure that the annotation results are unbiased to the greatest extent possible. This corpus is chosen to see the impact of name change on the model's judgments considering it has more than two options unlike the story task.

[3] `https://en.wikipedia.org/wiki/Wikipedia:Popular_pages#People`
[4] `https://en.geneanet.org/genealogy`
[5] `https://github.com/JiHuizhong549/Fame-Bias` (All figures, tables and the appendix for this article have been published on this website.)

3.3 Processing of test datasets

We first use spaCy[6] tool to extract examples with one or more names from the dataset. Next, we use the "Gender-guesser"[7] tool to determine the gender of the person's name, and then randomly select a person's name from the *Famous Names List* to replace the original name based on gender. Figure 1 shows the detailed process of processing data. We used the same method for the XNLI dataset and ultimately created eight datasets to test different models. Table 2 shows which datasets are used for each experiment.

4 Experiment and results

4.1 Zero-shot on proprietary Large Language Models

For the experiments with proprietary LLMs, we use GPT series models from OpenAI and Gemini 1.0 Pro model from Google. In the experiment, we use default temperature (0.2). We used prompts from Wei et al. [4] for story cloze task and from Muenighoff et al. [5] for XNLI task to ensure that the method is the same as in previous studies. For specific examples of prompts, see Table 11 and Table 12 in the Appendix. During the testing process, there were 2%-3% cases where the model refused to answer questions, and we decided to remove these samples from the test set. We simultaneously use the bootstrap method to determine whether the model's responses on famous names, ordinary names, and the combination of famous and ordinary names differ significantly from its responses on ordinary names. To implement the bootstrap algorithm, we chose the 'stats.bootstrap' function from the Scipy library and set the number of random samples to 1,000. The results of the test results for the story cloze task (see Table 3) show that in most cases, changing the names of people in the scene will reduce the accuracy of language model judgment, and the accuracy of using famous people names will decrease even more. At the same time, the experimental results show that changing the names of people has significant differences from the original example test results. In the NLI task, experiment also showed similar results (see Table 4).

4.2 Few-shot on open-source LLMs

For this experiment, we choose to test the Llama 2 model. We use two sizes of Llama 2-Chat optimized for dialogue scenarios: Llama 2-Chat 7b and Llama 2-Chat 13b. When we initialized the experiments, we found that the model was unable to complete the zero-shot task, hence we used three random examples to guide the model. In spite of changing the prompting strategy, there were still many instances of the model refusing to answer during the experiments. Based on the above issues, we have decided to use the f-score for evaluation.

[6] `https://github.com/explosion/spaCy?tab=readme-ov-file`
[7] `https://pypi.org/project/gender-guesser/`

Models output and correct answer are considered true positive. Models output but incorrect answer is considered as false positive. When the models does not output any answer, such case is treated as false negative.

As shown in Table 5, in small-size open-source language models, even when tested with few shots, the language model still exhibits significant bias towards different examples of changing names.

4.3 Experiments on fine-tuned language models

In the preceding two sections, we introduced how large language models can exhibit varying degrees of bias due to changes in personal names. In this section, we investigate whether fine-tuned models experience significant differences in their responses when faced with personal name changes. In this experiment, to ensure that the model achieves good prediction accuracy, we select two datasets, SCT (Story Cloze Test)-v1.0 and SCT-v1.5, totaling 5,313 data points, for fine-tuning the story ending prediction task on the distilbert-base-cased and bert-large-cased. The training set size is 4,782, and the validation set size is 533. Subsequently, experiments are conducted on the test sets listed in Table 2, and the results are presented in Table 6. When faced with the task of two options, although the accuracy of the fine-tuned model decreases after changing the name of the person, there is no significant difference from the original test data. We believe that during the fine-tuned process, there is an uncontrollable connection between certain names and the story ending, which leads to a decrease in the accuracy of the model.

For the XNLI dataset, we utilize various fine-tuned HuggingFace models to test whether they experience any biases due to personal name changes. These fine-tuned models are obtained from the HuggingFace platform and are publicly accessible [6] [7]. In the three option task, the example of changing the character name not only reduced the accuracy of the fine-tuned models' judgment, but also showed significant differences from the original example after testing with the bootstrap method (0.8378 vs 0.8216* vs 0.8378* vs 0.5621, see Table 7).

4.4 Results of sentiment analysis

The models we tested in Sections 4.1, 4.2 and 4.3 are all based on the transformer framework and use the attention mechanism. All these models have an impact on the accuracy of changes in the names of people in the sentences, hence we decide to experiment also with VADER[8], a lexicon and rule-based sentiment analysis tool.

Although VADER cannot perform tasks related to natural language understanding, it is able to output sentiment scores for text input, where values less than -0.05 represent negative, greater than -0.05 and less than 0.05 represent neutral, and greater than 0.05 represent positive sentiment. First, we input the original sentences with names in a dataset and use output scores as the baseline.

[8] `https://github.com/vaderSentiment/vaderSentiment`

Then we perform name replacements as in previous experiments and measure the sentiment score differences (details are shown in Table 2). The results after replacement are compared with the baseline results, and the comparison results are shown in Table 8. The comparison results show that the similarity between the test data results of changing names and the original data results is high (all similarities exceed 97%), and there is no significant change.

5 Discussion

Judging from the results presented in Section 4, whether it is a large language model or a fine-tuned model, changes in names caused changes in the results. At the same time, we also used the bootstrap method to demonstrate that there is a significant difference between the answers generated by the language model for sentences with changed character names and the original data.

Although using ordinary names has a slighter impact than using famous names, there are still potential risks in the real world. For example in a housekeeping scenario, if any name of the caller or a guest changes the behavior of an artificial agent, this is far from desirable for its user.

As mentioned earlier, celebrity names can have a significant impact on the model's judgment. This presents a detrimental effect on lay individuals who share their names with prominent public figures. For example, when we talk about Michael Jordan, the first thing that comes to mind is the basketball legend of the Chicago Bulls, and the large language model might have the same association. According to the Geneanet website, there are nearly 4,000 people in the United States named Michael Jordan, each with different personal background.

Mitigation methods include using lexicon and rule-based tools (imperfect) and introducing user personal information (with limitations due to RAG methods). In the next section we test a mitigation method using random placeholders for personal names.

6 Fame bias mitigation experiment

To address the bias of LLMs towards personal names, we perform experiments with randomly generated strings. These personal name placeholders were generated by randomly selecting ten letters, resulting in strings as "vipkp qixby." Subsequently, we evaluated influence of these random names using models fine-tuned on the IMDb dataset [8], including *bert-based-cased*[9] and *distilbert-base-cased*[10]. Thanks to being fine-tuned on IMDb data, these models can directly output sentiment scores. We selected randomly generated names with sentiment score deviations of less than 5 for both positive and negative sentiments to ensure that these pseudo-names do not carry any polarized sentiment. Following the process outlined in Section 3.2, we replaced the original personal names in the existing

[9] `https://huggingface.co/google-bert/bert-base-cased`
[10] `https://huggingface.co/distilbert/distilbert-base-cased`

test dataset with these random strings and repeated experiments. However, these random names did not eliminate the bias of language models towards personal names. Random names also confused the models resulting in visible differences between original names and placeholder. Tables 9 and 10 present a comparison of the test results between the random names and the original names across multiple models.

7 Conclusion and Future Work

This paper describes our investigation on the susceptibility of current NLP methods to changes in personal names. Experiments in story cloze and NLI tasks showed significant impacts on language models, and possible solutions like lexicon-based methods or random string replacement did not work well, indicating potential real-world risks. As the effect is only confirmed for English test data, will expand the study to multiple languages. As the mitigation test is rather simplistic, we plan to propose a new algorithm or framework to reduce the impact of personal names and enable models to focus more on semantic and contextual content without being influenced by a famous name.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number 22K12160 and by JST, CREST Grant Number JPMJCR20D2, Japan.

References

1. Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., Allen, J.: A corpus and evaluation framework for deeper understanding of commonsense stories. arXiv preprint arXiv:1604.01696 (2016)
2. Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H., Stoyanov, V.: XNLI: Evaluating cross-lingual sentence representations. arXiv preprint arXiv:1809.05053 (2018)
3. Jha, S.K., Ewetz, R., Velasquez, A., Jha, S.: What's in a name? the influence of personal names on spatial reasoning in bloom large language models (2022)
4. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652 (2021)
5. Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T.L., Bari, M.S., Shen, S., Yong, Z.X., Schoelkopf, H., et al.: Crosslingual generalization through multitask finetuning. arXiv preprint arXiv:2211.01786 (2022)
6. Laurer, M., Atteveldt, W.V., Casas, A.S., Welbers, K.: Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI. Preprint (Jun 2022), `https://osf.io/74b8k`, publisher: Open Science Framework
7. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems **33**, 5776–5788 (2020)
8. Dodds, K.: Popular geopolitics and audience dispositions: James Bond and the internet movie database (IMDb). Transactions of the Institute of British Geographers **31**(2), 116–130 (2006)