PULSE – Polling Using LLM-based Sentiment Extraction

Christos Karanikolopoulos
University of Ioannina
Greece
chkaranikolopoulos@cs.uoi.gr

Panagiotis Papadakos
ICS - FORTH
Greece
papadako@ics.forth.gr

Panayiotis Tsaparas
University of Ioannina
Greece
tsap@uoi.gr

***Abstract*—We introduce PULSE (Polling Using LLM-based Sentiment Extraction), a platform for simulating public polling using large language models (LLMs). PULSE leverages LLMs as aggregators of public knowledge and sentiment to forecast public opinion across various domains, such as elections and policy debates. The platform provides a flexible pipeline that uses carefully designed prompts to specify the target population and the polling question, and elicits preferences using next token probabilities. We demonstrate the platform's functionality with a case study on predicting the outcome of the 2024 U.S. presidential election.**

***Index Terms*—Virtual Polling, Forecasting, LLMs**

I. Introduction

The past few years have been marked by the rise of Generative AI, and the accelerated development and deployment of Large Language Models (LLMs). Trained on massive textual corpora, LLMs are able to encode human knowledge and experience, and use this encoded knowledge to answer questions, engage in conversations, and compose original textual content. With carefully designed *prompts*, LLMs can be placed within a context, allowing their output to be tailored to user specifications. For example, we can ask the LLM to adopt a specific identity with certain characteristics and produce content that reflects this identity. The ability of LLMs to produce human-like text has attracted significant research interest in building agents that can seamlessly interact with humans [1], [2], as well as in simulating human behavior across different settings, including games, cognitive tests, opinion surveys and social network interactions [3]–[8].

In this work, we present PULSE, a platform that leverages large language models (LLMs) to conduct virtual polling and forecast public opinion across diverse issues. PULSE provides a flexible pipeline for defining polling scenarios through prompts. Using system and user prompts, users can specify the target population (e.g., nationality, location, demographics), as well as the polling question.

To extract responses, we adopt a principled methodology that supplies an answer prefix in the assistant prompt, along with multiple completions, each representing a distinct viewpoint. By leveraging the LLM next-token probabilities, we can elicit population preferences across these viewpoints. Employing multiple completions enables exploration of nuanced positions while reducing noise. The tool also supports the crafting and filtering of answer completions. Designed as a general-purpose framework, PULSE can be applied to a wide range of public opinion studies. We illustrate its capabilities with a case study on the 2024 U.S. Presidential Election.

LLMs have been previously applied to public opinion research. In [9] they use LLMs to simulate individual population samples ("silicon samples") for opinion polling, and study the algorithmic fidelity between simulated and real opinions. They adopt an approach similar to ours, but they consider individual responses rather than aggregate, and they do not explore the answer space, limiting the generalizability of their approach. The work in [10] also generates individual silicon samples, and uses ChatGPT answers, instead of next-token probabilities, to estimate public opinion on political issues and elections. In [11] they consider virtual polls for climate change, while [12], [13] investigate ethics and performance issues in AI polling.

PULSE extends this line of work, presenting a general-purpose, easy-to-use platform for running polls across multiple models, covering diverse issues and target populations. It is is designed for a broad, interdisciplinary audience, including researchers and practitioners in data science conducting experiments with LLMs, as well as applied social and political scientists interested in using virtual polling as an auxiliary tool for studying public opinion, behavior, and potential LLM biases. The code for PULSE and a video demonstration are publicly available[1].

II. The PULSE tool: Overview and Methodology

We now provide an overview of the proposed tool and the underlying methodology. Consider an issue with two opposing sides, A and B, and assume that we wish to forecast which side the public will support. We use the *system prompt* of the LLM (or the *user prompt* if the model does not support a system prompt) to target a specific population by assigning a persona to the LLM. For example, this could represent the citizens of a country (*"You are a citizen of the U.S."*), or a demographic group (*"You are a male"*). Then, we pose the

This work has been implemented within the framework of the H.F.R.I call "Basic Research Financing" (H.F.R.I. Project Number: 016636), under the National Recovery and Resilience Plan "Greece 2.0", funded by the European Union - NextGenerationEU.

[1]https://github.com/elidek-themis/pulse

question regarding the issue at hand in the *user prompt*. For example, *"Who will you vote for in the 2024 U.S. presidential election?"*, or *"What is your opinion on abortion?"*.

To elicit a preference, we set the *assistant prompt* to the response prefix. For example, *"I will vote for "*, or *"I believe that abortion should be "*. We then provide possible response completions in the form of pairs $p = (c^{\mathrm{A}}, c^{\mathrm{B}})$, where completion c^{A} supports side A, while completion c^{B} supports side B. The pairs are constructed so that they are comparable and compatible, in terms of length and content, while expressing opposite semantics. For the examples above, the completions may be *("the Democratic candidate", "the Republican candidate")* and *("legal", "illegal")*, respectively. We compute the next-token probabilities for c^{A} and c^{B}, and compare them to determine which side is supported. Multiple such completions can be used, with each completion "voting" with some confidence for one side or the other. These votes are then aggregated to obtain the final forecast.

Formally, let X denote the input prompts to the model, including the system, user, and assistant prompts. For a completion string c, we obtain the negative log-likelihood $\mathrm{NLL}(c \mid X)$ of the model generating c conditioned on X, and the corresponding completion probability $P(c \mid X) = \exp(-\mathrm{NLL}(c \mid X))$. Given a completion pair $p = (c^{\mathrm{A}}, c^{\mathrm{B}})$, and the completion probabilities $P(c^{\mathrm{A}} \mid X), P(c^{\mathrm{B}} \mid X)$, we compute the *normalized completion probabilities* as

$$P_N\left(c^{\mathrm{S}} \mid X\right) = \frac{P\left(c^{\mathrm{S}} \mid X\right)}{P\left(c^{\mathrm{A}} \mid X\right) + P\left(c^{\mathrm{B}} \mid X\right)}$$

for $\mathrm{S} \in \{\mathrm{A}, \mathrm{B}\}$. These are the conditional probabilities of strings c^{A} and c^{B}, conditioned on the the pair p and input prompts X. Note that since we use the *raw next-token probabilities* and we do not sample tokens, the probabilities we compute are deterministic and independent of decoding hyperparameters, such as temperature scaling and nucleus sampling.

We use the difference of the normalized probabilities $\mathrm{diff}(p) = P_N(c^{\mathrm{A}} \mid X) - P_N(c^{\mathrm{B}} \mid X)$ as a predictor for the poll for the pair $p = (c^{\mathrm{A}}, c^{\mathrm{B}})$. Given a collection of completion pairs, $\mathcal{P} = \{p_1, \ldots, p_k\}$, we compute the mean difference value $\overline{\mathrm{diff}}(\mathcal{P}) = \frac{1}{|\mathcal{P}|}\sum_{p \in \mathcal{P}} \mathrm{diff}(p)$, and use it as the poll predictor. A positive value indicates a prediction for side A, while a negative value indicates a prediction for side B.

Note that our tool is designed to predict which side the majority will support, rather than the exact level of support for each side. Consequently, $\overline{\mathrm{diff}}(\mathcal{P})$ should not be interpreted as an estimate of the actual difference in support percentages, but rather as a measure of the strength of the prediction signal. To guide interpretation, we also compute the standard error of the mean value, which serves as an estimate of the confidence of the prediction. Predictions are considered more reliable when $\overline{\mathrm{diff}}(\mathcal{P})$ has high absolute value, and the standard error is low.

Comparing the probabilities of the different completions assumes that $P(c^{\mathrm{A}} \mid X)$, $P(c^{\mathrm{B}} \mid X)$ are sufficiently large. Otherwise, we are extracting conclusions from noise. To avoid this case, our tool provides statistics about the completions, allowing the user to filter out noise. We also assist the user in creating the completions, by enabling an interactive exploration of the completions space.

III. PULSE Demo: The 2024 U.S. Presidential Election Case Study

We will now demonstrate the functionality of the PULSE tool, using the 2024 U.S. Elections as the case study, where the goal is to forecast the results for different demographic groups.

A. Connection and Navigation

The first step is to establish a connection with the LLM host. On the left panel of the starting page (Fig. 1), the user provides the host URL and an API key. Once connected, a drop-down menu is populated with the available models, from which the user selects the desired one. The tool is model-agnostic: it can interface with any open-source model that supports the OpenAI API protocol[2].

In our case study, we employ `Llama-3.1`[3], a model pre-trained on up to 15 trillion tokens from publicly available sources. Its training data has a knowledge cutoff of December 2023 – prior to both the 2024 U.S. elections and the withdrawal of Joe Biden from the presidential race – ensuring no information leakage in our predictions. We also experimented with additional models, including `Gemma-2`[4] and `Phi-4`[5].

After connecting, users can navigate across three main pages corresponding to the tool's core functionalities: *Polling* for creating or loading a poll, *Results* for viewing the results of a poll, and *Explorer* for exploring the completion space. We describe these functionalities below.

B. Creating a Virtual Poll

Fig. 1 shows the *Polling* screen of PULSE. To create a new poll the user needs to specify the following: (1) The target population; (2) The polling question; (3) The answer prefix; (4) The completion pairs. This information is entered in the middle panel of the page (Fig. 1).

The target population: The target population is defined by assigning a persona to the LLM. A persona is specified in the *Persona* box, and its text becomes the system prompt, or part of the user prompt. For example: *"You are a citizen of the U.S."*

Users may wish to poll multiple complementary groups (e.g., genders, regions, or countries). Instead of running separate polls for each group, PULSE supports batch polling. In this mode, the persona prompt includes a placeholder, which is populated with a set of values. For example, to compare gender differences in voting, the prompt might be *"You are a {gender} U.S. citizen."*, with values {`male, female`} for the *gender* placeholder.

[2]https://docs.vllm.ai/en/latest/serving/openai_compatible_server.html
[3]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[4]https://huggingface.co/google/gemma-2-27b-it
[5]https://huggingface.co/microsoft/phi-4

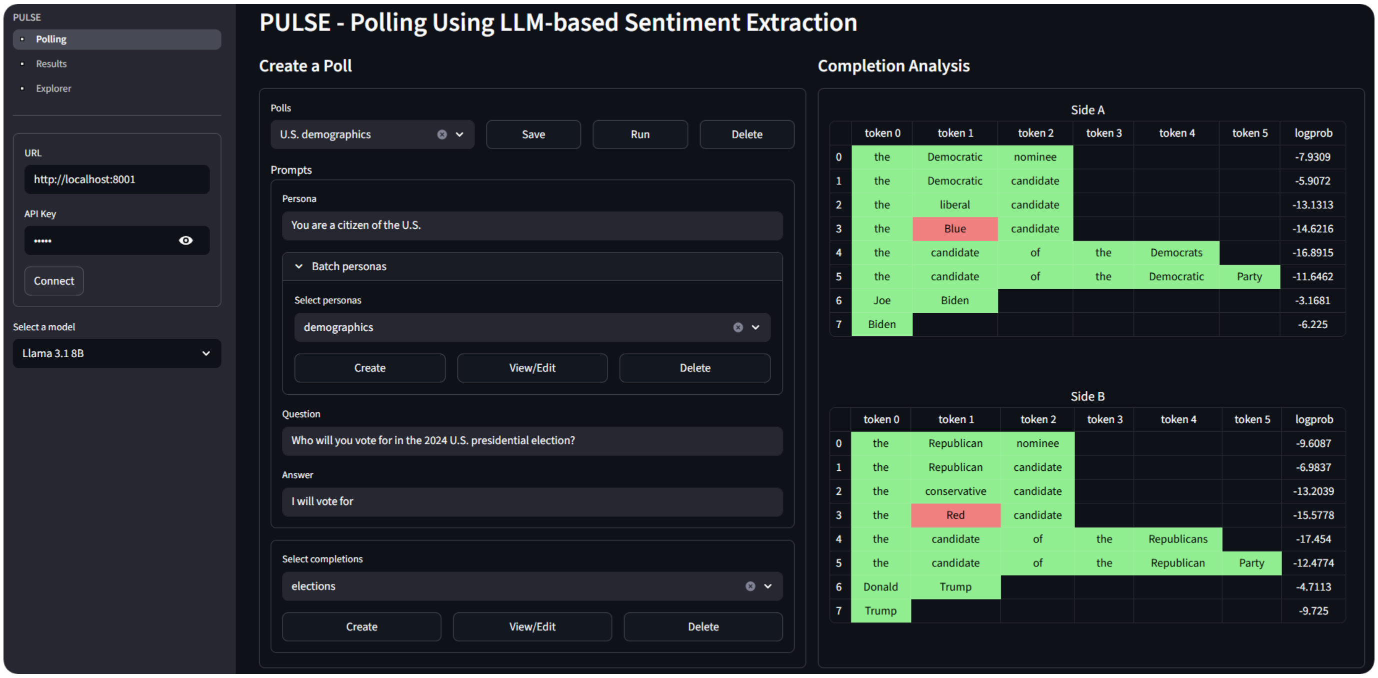

Fig. 1. The *Polling* page including the connection and navigation panel, the poll creation, and completions analysis.

To create batch personas, the user selects the *Batch Personas* option, clicks *Create*, and enters values for the placeholder in a pop-up window – or imports them from a CSV/JSON file. Each value is assigned an alias for easier interpretation of the results. The configuration is stored in a *persona file*, in the PULSE tool, which can be edited or reused in other polls.

For validation, persona files may also include ground-truth data for the A, B options. In elections, this could mean adding official results, or exit-poll percentages for demographic groups. These values enable benchmarking of PULSE predictions against existing polls or actual outcomes.

The polling question: The polling question is entered in the *Question* box, and it becomes part of the user prompt. For example: *"Who will you vote for in the 2024 U.S. Presidential Elections?"*.

The answer prefix: The answer prefix is entered in the *Answer* box, and becomes part of the assistant prompt. For example: *"I will vote for "*.

The completions: The completions are entered in the *Completions* panel. Recall that the completions are a collection of pairs $\mathcal{P} = \{(c_i^{\mathrm{A}}, c_i^{\mathrm{B}})\}$, for which we will compare the completion probability. Similar to persona creation, the user clicks *Create*, and enters completion pairs in a pop-up window – or imports them from a CSV/JSON file. The configuration is stored in a *completion file*, in the PULSE tool, which can be edited or reused.

C. Completion Analysis

When the user selects a completion collection ($\mathcal{P}$), the tool uses the defined prompts and displays an analysis of $\mathcal{P}$ in the right panel (Fig. 1). For each position in a completion, the tool retrieves the ranked list of tokens in descending order of probability. Tokens outside the nucleus top-99% set, i.e., tokens whose inclusion would push the cumulative probability beyond the 0.99 quantile, are highlighted in red. The log-probability of the entire completion is also shown.

These statistics help identify and filter unreliable completions. For example, a completion pair $(c^{\mathrm{A}}, c^{\mathrm{B}})$ in which both c^{A} and c^{B} have very low probability carries little value for comparison or forecasting. Likewise, a completion containing a token with very low rank may be noisy or erroneous. In such cases, users can edit or remove noisy completions. In our case study, we excluded the completion pair (*"the Blue candidate"*, *"the Red candidate"*), which we consider to be uninformative. For the selection we used the general *citizen of the U.S.* persona, which serves as an aggregate of the different personas.

D. Results

The user clicks on *Run* to execute the poll. The results are stored in PULSE, and can be viewed on the *Results* page (Fig. 2). For each persona value, the tool reports the forecast outcome between the two options, the mean difference $\overline{\mathrm{diff}}(\mathcal{P})$ for the completion collection $\mathcal{P}$, and the Standard Error (SE) of the mean. The $\overline{\mathrm{diff}}(\mathcal{P})$ value determines the forecast winner, while the SE specifies the confidence in the forecast, as discussed in Section II.

PULSE also visualizes the results in a point plot (Fig. 2), which is particularly useful in batch polling. Each point in the plot corresponds to a persona value, showing the average difference and SE. The dotted line marks the zero value. Positive values (option A) are colored blue, while negative values

Fig. 2. The *Results* page for the Elections poll across demographics.

(option B) are colored red. When ground-truth percentages are available, they are indicated with a star.

Fig. 2 shows the results for our virtual election poll across different demographic groups (e.g., male/female, LGBT/non-LGBT). The prompts and completions are shown in Fig. 1. Ground truth values are obtained from exit polls of the 2024 U.S. Elections[6]. Side A (blue) corresponds to the Democrats, and side B (red) to the Republicans. The virtual poll successfully captures known trends, such as the dichotomy in voting between men and women, white and colored, or LGBT and non-LGBT. Confidence is high for strongly partisan demographic groups (e.g., LGBT, Christian, or non-religious voters), but lower for more borderline cases (e.g., high income voters, or voters without college degree). Note that PULSE forecasts which side a group supports, not the exact support percentages.

E. Explorer

On the *Explorer* page (Fig. 3) the user can explore the space of possible completions, and design new ones. Given the Persona, Question, and Answer prompts, along with a partial completion, clicking *Sample* generates a ranked list of the top-k most likely tokens predicted by the LLM, where k is user-specified. For each token, their probability, and the cumulative probability at their rank is displayed. Reviewing this list, the user can select the next token to add to the completion and then resample to continue the process. In this way, the *Explorer* allows users to iteratively build completions, guided by the probabilities output by the LLM.

IV. Conclusion

We presented PULSE, a tool that leverages LLMs to forecast the public opinion on different polling issues, through appropriate prompting, and next-token probabilities. Our demo focuses on polls with binary options, but it can be easily extended to the case of multiple selections. In the future, we plan to use our tool for tasks beyond polling, such as, measuring bias in LLM responses.

[6]https://edition.cnn.com/election/2024/exit-polls/national-results/general/president

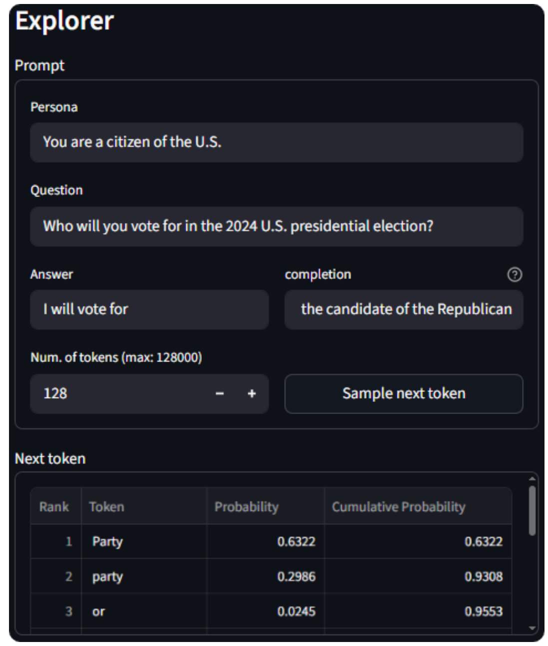

Fig. 3. The *Explorer* page of the PULSE tool.

References

[1] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, 2024.

[2] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *Science China Information Sciences*, vol. 68, no. 2, 2025.

[3] F. Jia, Z. Ye, S. Lai, K. Shu, J. Gu, A. Bibi, Z. Hu, D. Jurgens, J. Evans, P. H. Torr, B. Ghanem, G. Li, C. Xie, and C. Chen, "Can large language model agents simulate human trust behavior?" in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[4] Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno, "Llms and generative agent-based models for complex systems research," *Physics of Life Reviews*, 2024.

[5] C. Gao, X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, F. Xu, and Y. Li, "Large language models empowered agent-based modeling and simulation: A survey and perspectives," *Humanities and Social Sciences Communications*, vol. 11, no. 1, pp. 1–24, 2024.

[6] E. Junprung, "Exploring the intersection of large language models and agent-based modeling via prompt engineering," *arXiv:2308.07411*, 2023.

[7] C. Chen, B. Yao, Y. Ye, D. Wang, and T. J.-J. Li, "Evaluating the llm agents for simulating humanoid behavior." 1st HEAL Workshop at CHI Conference on Human Factors in Computing Systems, 2024.

[8] J. S. Park, L. Popowski, C. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Social simulacra: Creating populated prototypes for social computing systems," in *ACM UIST*, 2022, pp. 1–18.

[9] L. P. Argyle, E. C. Busby, N. Fulda, J. R. Gubler, C. Rytting, and D. Wingate, "Out of one, many: Using language models to simulate human samples," *Political Analysis*, vol. 31, no. 3, p. 337–351, 2023.

[10] S. Jiang, L. Wei, and C. Zhang, "Donald trumps in the virtual polls: Simulating and predicting public opinions in surveys using large language models," *arXiv preprint arXiv:2411.01582*, 2024.

[11] S. Lee, T.-Q. Peng, M. H. Goldberg, S. A. Rosenthal, J. E. Kotcher, E. W. Maibach, and A. Leiserowitz, "Can large language models estimate public opinion about global warming? an empirical assessment of algorithmic fidelity and bias," *PLOS Climate*, vol. 3, no. 8, 2024.

[12] Y. Qu and J. Wang, "Performance and biases of large language models in public opinion simulation," *Humanities and Social Sciences Communications*, vol. 11, no. 1, pp. 1–13, 2024.

[13] R. Cerina and É. Rouméas, "The democratic ethics of artificially intelligent polling," *AI & SOCIETY*, pp. 1–15, 2025.