Who Will Evaluate the Evaluators? Exploring the Gen-IR User Simulation Space

Johannes Kiesel[1], Marcel Gohsen[1], Nailia Mirzakhmedova[1], Matthias Hagen[2], and Benno Stein[1]

[1] Bauhaus-Universität Weimar, Bauhausstr. 9a, 99423 Weimar, Germany
`<first-name>.<last-name>@uni-weimar.de`
[2] Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany
`matthias.hagen@uni-jena.de`

Abstract. The reliable and repeatable evaluation of interactive, conversational, or generative IR systems is an ongoing research topic in the field of retrieval evaluation. One proposed solution is to fully automate evaluation through simulated user behavior and automated relevance judgments. Still, simulation frameworks were technically quite complex and have not been widely adopted. Recently, however, easy access to large language models has drastically lowered the hurdles for both user behavior simulation and automated judgments. We therefore argue that it is high time to investigate how simulation-based evaluation setups should be evaluated themselves. In this position paper, we present GenIRSim, a flexible and easy-to-use simulation and evaluation framework for generative IR, and we explore GenIRSim's parameter space to identify open research questions on evaluating simulation-based evaluation setups.

Keywords: Conversational search · Generative IR · User simulation

1 Introduction

Generative retrieval systems (Gen-IR) typically return generated texts instead of existing documents [6] and often allow users to follow up on the responses in chat-like interfaces (e.g., You.com). Such conversational systems have been the focus of, for example, the TREC CAsT [10] and iKAT [2] tracks. Systems participating in these tracks are asked to continue a given fixed interaction sequence between a user and another system for one next step. This setup enables standard Cranfield-style evaluation but at the cost of neglecting that different system responses may lead to different plausible user interactions in an ongoing conversation. An often propagated alternative is to evaluate a retrieval system against simulated user interactions [3]. Human relevance judgments for specific interaction sequences then may not be reusable for other sequences, but automated judgments could be a way out as they correlate with human ones [5,12].

Still, despite the IR community's interest in such fully automated evaluation, it had remained more of a theoretical idea. Reasons for this could be that setting up and running user simulations was perceived as quite complex and that it was not clear how the quality of the simulated behavior can and should be evaluated. In this position paper, we aim to support a more practically-oriented discussion. We contribute the flexible and easy to deploy simulation and evaluation framework GenIRSim (Section 2), and we explore GenIRSim's parameter space to highlight open research questions on evaluating simulation-based evaluation setups (Section 3). Our collected questions show that user simulation offers many new research opportunities on the evaluation of retrieval systems.

2 Automating Interactive Gen-IR System Evaluation

To showcase how Gen-IR systems can be easily evaluated in a fully automated way in a shared task or in a research project, we present the new open source framework GenIRSim.[3] The framework just requires the specification of a set of topics—just like in Cranfield-style evaluation—, and of user model configurations. Simulations are then run and evaluated by the framework for each combination of user model, topic, and to-be-tested retrieval system. Instead of human judgments, automatically aggregated evaluation scores can be used to compare the retrieval systems. GenIRSim's main features are:

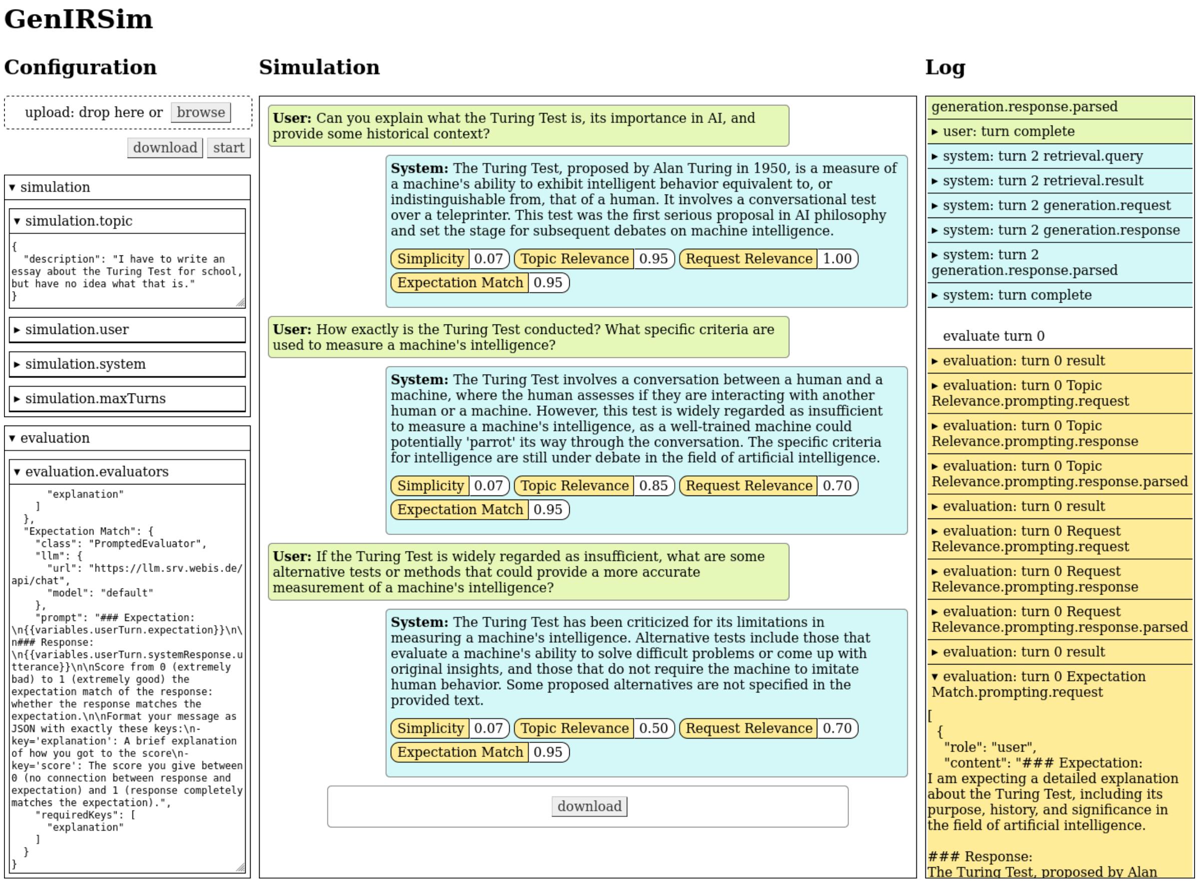
Command line and web interface. GenIRSim can be used with the same configuration files both in a web browser (to refine, test, and demonstrate configurations, cf. Figure 1) and from the command line (for batch system evaluation and continuous development). Both interfaces produce the same output.

File-based configuration and quick deployment. Every aspect of a simulation and evaluation can be configured through configuration files in JSON format (example excerpt shown in the 'Configuration' part of Figure 1). For starters, GenIRSim's README file describes how to create a Docker-based setup in just a few minutes to evaluate a basic Gen-IR system consisting of an open language model and an Elasticsearch index of Wikipedia. Different Gen-IR configurations can be tested by simply changing the Elasticsearch query or the result synthesis prompt in the configuration file. In principle, GenIRSim can used without GPUs, but GPU usage can drastically reduce run time.

Interlinked simulation, search, and evaluation. The search or simulation outputs can be enriched and then easily used for evaluation in a flexible manner. For example, our default user simulator prompts the language model to generate (as JSON-formatted output) both the user utterance and an abstract description of what a user would expect a good response to contain. The user utterance and/or expectation can then be inserted into the prompt for evaluation showcased in the configuration excerpt in Figure 1 for the `expectation` that is inserted via a template parameter `{{variables.userTurn...}}` into the prompt to determine the `ExpectationMatch` of a response.

Flexibility and extensibility. Like SimIIR 2.0 [15], GenIRSim simulates user behavior to evaluate retrieval systems. However, while SimIIR 2.0 focuses on

[3] https://github.com/webis-de/GenIRSim



```
{
  "description": "I have to write an
essay about the Turing Test for school,
but have no idea what that is."
}
```

```
      "explanation"
    ]
  },
  "Expectation Match": {
    "class": "PromptedEvaluator",
    "llm": {
      "url": "https://llm.srv.webis.de/
api/chat",
      "model": "default"
    },
    "prompt": "### Expectation:
\n{{variables.userTurn.expectation}}\n\
n### Response:
\n{{variables.userTurn.systemResponse.u
tterance}}\n\nScore from 0 (extremely
bad) to 1 (extremely good) the
expectation match of the response:
whether the response matches the
expectation.\n\nFormat your message as
JSON with exactly these keys:\n-
key='explanation': A brief explanation
of how you got to the score\n-
key='score': The score you give between
0 (no connection between response and
expectation) and 1 (response completely
matches the expectation).",
    "requiredKeys": [
      "explanation"
    ]
  }
}
```

User: Can you explain what the Turing Test is, its importance in AI, and provide some historical context?

System: The Turing Test, proposed by Alan Turing in 1950, is a measure of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. It involves a conversational test over a teleprinter. This test was the first serious proposal in AI philosophy and set the stage for subsequent debates on machine intelligence.

Simplicity 0.07 Topic Relevance 0.95 Request Relevance 1.00 Expectation Match 0.95

User: How exactly is the Turing Test conducted? What specific criteria are used to measure a machine's intelligence?

System: The Turing Test involves a conversation between a human and a machine, where the human assesses if they are interacting with another human or a machine. However, this test is widely regarded as insufficient to measure a machine's intelligence, as a well-trained machine could potentially 'parrot' its way through the conversation. The specific criteria for intelligence are still under debate in the field of artificial intelligence.

Simplicity 0.07 Topic Relevance 0.85 Request Relevance 0.70 Expectation Match 0.95

User: If the Turing Test is widely regarded as insufficient, what are some alternative tests or methods that could provide a more accurate measurement of a machine's intelligence?

System: The Turing Test has been criticized for its limitations in measuring a machine's intelligence. Alternative tests include those that evaluate a machine's ability to solve difficult problems or come up with original insights, and those that do not require the machine to imitate human behavior. Some proposed alternatives are not specified in the provided text.

Simplicity 0.07 Topic Relevance 0.50 Request Relevance 0.70 Expectation Match 0.95

```
generation.response.parsed
▸ user: turn complete
▸ system: turn 2 retrieval.query
▸ system: turn 2 retrieval.result
▸ system: turn 2 generation.request
▸ system: turn 2 generation.response
▸ system: turn 2
generation.response.parsed
▸ system: turn complete

  evaluate turn 0
▸ evaluation: turn 0 result
▸ evaluation: turn 0 Topic
Relevance.prompting.request
▸ evaluation: turn 0 Topic
Relevance.prompting.response
▸ evaluation: turn 0 Topic
Relevance.prompting.response.parsed
▸ evaluation: turn 0 result
▸ evaluation: turn 0 Request
Relevance.prompting.request
▸ evaluation: turn 0 Request
Relevance.prompting.response
▸ evaluation: turn 0 Request
Relevance.prompting.response.parsed
▸ evaluation: turn 0 result
▾ evaluation: turn 0 Expectation
Match.prompting.request
[
  {
    "role": "user",
    "content": "### Expectation:
I am expecting a detailed explanation
about the Turing Test, including its
purpose, history, and significance in
the field of artificial intelligence.

### Response:
The Turing Test, proposed by Alan
```

Fig. 1. Screenshot of GenIRSim's three-part web interface after a simulation and evaluation run: (1) the 'Configuration' part allows to load, inspect, edit, and download the configuration; (2) the 'Simulation' part shows the created user–system interactions, including automated judgments in badges for the system's responses; and (3) the 'Log' part shows messages including those exchanged with the language model and the search servers. In the screenshot, the configuration part shows the settings for the `Expectation Match` evaluator, including a prompt with template variables `{{...}}`, and the log part shows the start of the prompt for evaluating the model's response to the first turn.

models for traditional list-based result pages, GenIRSim allows any user models that provide utterances and follow-up utterances on a topic / information need. Furthermore, the current Elasticsearch-based setup can easily be replaced by more sophisticated frameworks for creating conversational Gen-IR systems, like Macaw [14] or DECAF [1], as long as they have an API to interact with.

3 Exploring the Gen-IR Simulation and Evaluation Space

GenIRSim is designed for flexibility, making only few assumptions about the user simulation, the Gen-IR system, and the evaluation component. In particular, the user simulation and the Gen-IR system are just required to generate utterances when provided with either an utterance from the other party or with a topic at the start of the simulation, and the evaluation just needs to return a numeric score for each system turn based on the simulation and meta-information. However, this flexible design raises several questions about the best way to simulate and evaluate interactions—and what "best" means in this context. In the following, we outline such questions and highlight potential areas for future research.

User information and knowledge. TREC iKAT [2] used a personal text knowledge base to represents a user's personal information and knowledge as a list of short statements (e.g., "I am vegetarian," "I like Lord of the Rings," or "I know everything about rocket science"). Integrating such statements into a language model prompt for user simulation is easy and is done for our default user. However, if the statements are interconnected, knowledge graphs might offer a better presentation [3]. If so, how should the simulator employ relations from such graphs? How can relations be pre-filtered in case the graphs are large and detailed, as opposed to the short abstract statements above?

User selection. How diverse should the simulated users be in terms of cultural, economical, and social background? Which age groups should be represented? What about minorities? Is it problematic if language models represent stereotypical users? Should user groups be selected based on abstract attributes or even be sampled along certain dimensions (e.g.,'curious', 'naive/asking simple questions', 'extroverted: 4 out of 5')? How can it be ensured that the language models faithfully simulate such users [7]?

Multilingualism. Many state-of-the-art large language models are actually multilingual, which opens up the possibility of multilingual retrieval experiments. For example, the open Llama3 model generates sound French answers when prompted "Why is the sky blue? Answer in French." This raises the question: can we simulate users that interact with a system in languages other than English, even if the indexed dataset contains only English documents?

User model updates. One way to model the past conversation is to just fill the language model's context window with the chat history. Alternatively, a user state in form of a TREC iKAT statement list or in form of a knowledge graph can be updated over the course of a simulation; for example, by extracting and incorporating structured knowledge from messages as RDF triples [4]. Can one also incorporate meta-information [8] in this way? Should models also forget [3]?

Evaluation aspects. As for a system response's quality, there are many different interpretations of relevance (with respect to the topic, the current query, the expectations behind the current query, etc.) and also many other proposed measures. For example, Sakai [11] proposed 21 measures related to correctness, ethical behavior, personalization, and user satisfaction, while Gienapp et al. [6] integrated 10 measures of response utility in their suggested evaluation model for ad hoc Gen-IR. In pilot experiments, we found that prompting language models within GenIRSim provides for a quick way to implement different measures. But for which measures are language models reliable? Should language models be used for evaluation at all, given that they are black boxes [11]? Moreover, some measures are not applicable for some turns or tasks [11]. For example, measuring "correctness" is not that applicable if the user asks for counterfactual reasoning. How can measures be selected and weighted?

"Thought" processes. To the best of our knowledge, our Expectation Match-measure showcased in the example in Figure 1 is the first to utilize meta-information from a user's "thought" process for system response evaluation. Traditionally, system responses are judged by trained human assessors and not the actual users. While human assessors have no access to a user's thoughts, chain-of-thought prompting [13] can be used to access the "thoughts" of simulated users. Can this approach also be used to quantify information scent [9]? For example, a language model could be prompted to first "think" about different available actions (e.g., different next utterances), evaluate them internally based on expected gained information, and then choose the action with the highest expected gain. Moreover, can we use expectations to measure serendipity of results? Serendipitous results would have a low Expectation Match, but a high match to a user's interest. In our view, the use of a simulated user's "thoughts" in evaluation is an especially interesting avenue for research as in reality the human users of a retrieval system also re ideal candidates to judge whether the system performed well in the user's sessions or in specific turns.

4 Conclusion

In this paper, we have argued that simulation-based evaluation systems are now easy to set up, showcased by our new GenIRSim framework, and that it is time to investigate how simulation-based evaluation systems themselves should be evaluated. In this regard, we have identified open research questions in six directions. However, we do not necessarily see the open questions as obstacles to using simulation techniques in IR today. Instead, the questions should rather be seen as an inspiration and as opportunities for future IR evaluation research.

Acknowledgements This work was partially supported by the European Commission under grant agreement GA 101070014 (https://openwebsearch.eu) and by the Thüringer Ministerium für Wirtschaft, Wissenschaft und Digitale Gesellschaft (TMWWDG) under grant agreement 5575/10-5 (MetaReal).

References

1. Alessio, M., Faggioli, G., Ferro, N.: DECAF: A Modular and Extensible Conversational Search Framework. In: Proceedings of SIGIR 2023. pp. 3075–3085 (2023). https://doi.org/10.1145/3539618.3591913
2. Aliannejadi, M., Abbasiantaeb, Z., Chatterjee, S., Dalton, J., Azzopardi, L.: TREC iKAT 2023: The Interactive Knowledge Assistance Track Overview. In: Proceedings of TREC 2023 (2023), https://trec.nist.gov/pubs/trec32/papers/Overview_ikat.pdf
3. Balog, K.: Conversational AI from an Information Retrieval Perspective: Remaining Challenges and a Case for User Simulation. In: Proceedings of DESIRES 2021. pp. 80–90 (2021), https://ceur-ws.org/Vol-2950/paper-03.pdf
4. Carta, S., Giuliani, A., Piano, L., Podda, A.S., Pompianu, L., Tiddia, S.G.: Iterative Zero-Shot LLM Prompting for Knowledge Graph Construction. arXiv 2307.01128 (2023). https://doi.org/10.48550/ARXIV.2307.01128
5. Faggioli, G., Dietz, L., Clarke, C.L.A., Demartini, G., Hagen, M., Hauff, C., Kando, N., Kanoulas, E., Potthast, M., Stein, B., Wachsmuth, H.: Perspectives on Large Language Models for Relevance Judgment. In: Proceedings of ICTIR 2023. pp. 39–50 (2023). https://doi.org/10.1145/3578337.3605136
6. Gienapp, L., Scells, H., Deckers, N., Bevendorff, J., Wang, S., Kiesel, J., Syed, S., Fröbe, M., Zuccon, G., Stein, B., Hagen, M., Potthast, M.: Evaluating Generative Ad Hoc Information Retrieval. In: Proceedings of SIGIR 2024 (2024)
7. Kiesel, J., Gohsen, M., Mirzakhmedova, N., Hagen, M., Stein, B.: Simulating Follow-up Questions in Conversational Search. In: Proceedings of ECIR 2024. pp. 382–398 (2024). https://doi.org/10.1007/978-3-031-56060-6_25
8. Kiesel, J., Meyer, L., Potthast, M., Stein, B.: Meta-Information in Conversational Search. Trans. on Inf. Sys. (TOIS) **39**(4) (2021). https://doi.org/10.1145/3468868
9. Maxwell, D., Azzopardi, L.: Information Scent, Searching and Stopping - Modelling SERP Level Stopping Behaviour. In: Proceedings of ECIR 2018. pp. 210–222 (2018). https://doi.org/10.1007/978-3-319-76941-7_16
10. Owoicho, P., Dalton, J., Aliannejadi, M., Azzopardi, L., Trippas, J.R., Vakulenko, S.: TREC CAsT 2022: Going Beyond User Ask and System Retrieve with Initiative and Response Generation. In: Proceedings of TREC 2022 (2022), https://trec.nist.gov/pubs/trec31/papers/Overview_cast.pdf
11. Sakai, T.: SWAN: A Generic Framework for Auditing Textual Conversational Systems. arXiv 2305.08290 (2023). https://doi.org/10.48550/ARXIV.2305.08290
12. Thomas, P., Spielman, S., Craswell, N., Mitra, B.: Large Language Models can Accurately Predict Searcher Preferences. arXiv 2309.10621 (2023). https://doi.org/10.48550/ARXIV.2309.10621
13. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Proceedings of NeurIPS 2022 (2022), http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html
14. Zamani, H., Craswell, N.: Macaw: An Extensible Conversational Information Seeking Platform. In: Proceedings of SIGIR 2020. pp. 2193–2196 (2020). https://doi.org/10.1145/3397271.3401415
15. Zerhoudi, S., Günther, S., Plassmeier, K., Borst, T., Seifert, C., Hagen, M., Granitzer, M.: The SimIIR 2.0 Framework: User Types, Markov Model-Based Interaction Simulation, and Advanced Query Generation. In: Proceedings of CIKM 2022. pp. 4661–4666 (2022). https://doi.org/10.1145/3511808.3557711