Personalizing Chatbot Communication with Associative Memory

Kirill Soloshenko, Alexandra Shatalina, Elisabeth Kornilova,
Marina Sevostyanova, Konstantin Zaytsev
HSE University, Russia
kasoloshenko@edu.hse.ru, afshatalina@edu.hse.ru,
elankornilova@edu.hse.ru, mksevostianova@edu.hse.ru,
kzaytsev@hse.ru

Abstract

Despite the significant progress made by large language models (LLMs) over the past few years, they are still limited in context and struggle to retain user-specific information over extended interactions, which significantly affects their quality. While current research is focused on expanding the contextual window, our approach is aimed at effectively expanding the context through integrating a database of associative memory into the natural language processing (NLP) pipeline. In order to improve long-term memory and personalization we have utilized methods close to Retrieval-Augmented Generation (RAG).

We implement a multi-agent consecutive pipeline in order to improve the quality of personalization as measured in accuracy, which contains: (1) a cold-start agent to handle sparse initial interaction; (2) a fact extraction agent to detect and extract user inputs from the dialogue; (3) an associative memory agent to store and retrieve contextual data; and (4) a generation agent.

Evaluation results demonstrate promising performance: our pipeline increases the accuracy of the base Gemma3 model by 41%, from 16% to 57%. Hence, with our approach, we demonstrate that personalized chatbots can bypass LLM memory limitations while increasing information reliability under the conditions of limited context and memory.

1 Introduction

Although large language models (LLMs) have spurred considerable progress in natural language processing (NLP), inherent limitations still exist. A well-documented constraint is the difficulty LLMs encounter when generalizing across extended contextual lengths. This presents challenges in applications such as personalized chatbots, where maintaining consistent user-specific information over a long period of different sparse interactions is crucial, and LLMs frequently exhibit a tendency to "forget" previously established details. While existing research, for example, (Jin et al., 2024) and (Ding et al., 2024), explores methods for expanding the context window, and some models are pre-trained with large context windows (Yang et al., 2025), our approach contrastively focuses on achieving extended context through the integration of an associative memory database within the NLP pipeline.

The hypothesis is that, while the immediate inclusion of Retrieval-Augmented Generation (RAG) user-related data may introduce short-term complexity for the LLM, this strategy will ultimately enhance long-term user-specific memory and coherence within the personalized chatbot interaction.

Our pipeline includes four agents that work with the associative memory database to improve the personalization quality. The agents deal with the following tasks: fact extraction, associative memory, generation and the “cold start” issue resolution.

2 Related Works

Our research focuses on personalized communication with a chatbot, the key to which we consider the associative memory.

Chen et al. (2024) in their work, provide an overview of different approaches and datasets in personalized dialogue generation. To start with, the datasets used for training can vary, and while some contain descriptive sentences (Zhang et al., 2018),

Proceedings of the Student Research Workshop associated with RANLP-2025,
pages 62–69, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_008

others have simple key-value attributes like age, gender, location, etc. (Qian et al., 2018).

The article by Zhang et al. (2024) describes common issues that can be encountered during chatbot development. It proposes a more theoretical overview of some of the methods we have utilized during development. The metrics described are similar to those we have used for evaluation of the performance of our pipeline and agents: accuracy, precision, recall, F1-score and top-K.

A relevant issue that is also described within the article is the cold start problem. It is mainly encountered in recommendation systems and can be divided into "user cold start" and "item cold start" (Yuan and Hernandez, 2023). When the system encounters a new user or item it has not seen before and therefore has no information about them, their connection to each other, it still has to offer the user accurate recommendations. This problem is also encountered during chatbot development where, like in a real conversation, there must be topics that are both interesting to the user and relevant to the situation, even when we have little to no information about them in the database.

Zhang et al. (2024) highlight that many studies (Salemi at al., 2023), (Rajput et al., 2023), (Xi et al., 2023) choose to remove users with minimal interaction history during the preprocessing stages. This exclusion potentially undermines the robustness of the systems by disregarding the subtleties and potential insights offered by these underrepresented user interactions. Therefore, by resolving the cold start issue we do not encounter such drawbacks and improve the performance of our pipeline.

There are studies that utilize relevant facts for the personalized response generation like DuLeMon (Xu et al., 2022), which uses a classifier to determine whether a clause in an utterance contains personal information. In contrast, our associative memory implementation relies on the facts contained in the database in the form of triplets: subject, predicate, object, embedded using an arbitrary encoder and ranked by cosine similarity when each new user query is being received.

When the personas were not explicitly given in DuLeMon, they were extracted from dialogue histories. The seminal paper by (Zhang et al., 2018) emphasized that the agent specifically targets conversational data where personal attributes and relationships are often implied through complex linguistic patterns. Wu et al. (2020) and Wang et al. (2022) both underlined the value of implicit user modeling based on linguistic cues, strengthening our rationale for integrating linguistic tools like syntactic trees and coreference resolution. The cited works demonstrated that effective persona extraction requires handling three critical challenges: (1) resolving referential ambiguity, (2) capturing implicit relationships, and (3) maintaining consistency across multi-turn interactions - all of which directly informed the agent's architecture.

The generation agent is the most important part of any chatbot as it is crucial to efficiently generate responses to user's queries. There are many approaches to response generation with LLM. For example, it is possible to finetune the LLM with PEFT as Zhang et al. (2025) do in their work "Personalized LLM Response Generation with Parameterized User Memory Injection". They propose a parameterized Memory-injected approach and combine it with Bayesian Optimization searching strategy and LoRA in order to achieve LLM Personalization. We focused on prompt engineering as we find it one of the most effective ways to generate personalized responses to user's messages. A prompt is an input to a generative model, which is used to guide its output. Prompts make models more flexible and convenient to interact with. There are a number of papers where prompt-engineering approaches are described, for example, in the work of Sander Schulhoff et al. (2024).

The datasets we used for training models and testing agents' performance were Synthetic Persona Chat [1] (Jandaghi et al., 2024) and MultiSession Chat [2] (Xu et al., 2022) as they provided the most accurate data used in personalized dialogues.

[1] https://huggingface.co/datasets/google/Synthetic-Persona-Chat

[2] https://huggingface.co/datasets/nayohan/multi_session_chat

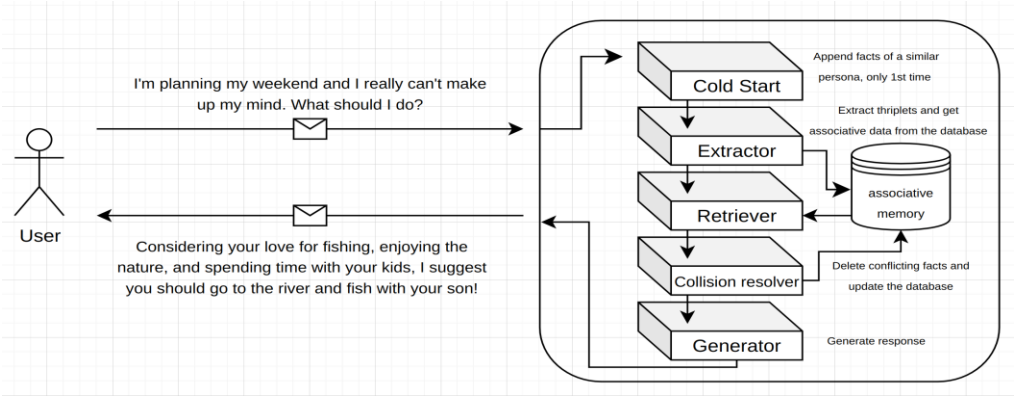

Figure 1: Chatbot pipeline schema, illustrating the key stages from user input processing to response generation: optional cold start, triplet extraction, fact retrieving and collision resolution, response generation.

3 Approach

The scheme of the chatbot pipeline is shown in Figure 1. When a new user starts chatting with our bot, basic facts about them, for instance age, gender and personal interests, go into the cold start agent to get potential dialogue options based on facts about other similar users which are stored in the database in the form of triplets. These topics then go to the associative memory agent for collision resolution. After that the triplets enter our generation agent where they are mixed in with the user's query to produce a response.

If there is a history of communication with the chatbot and the associative memory database contains user information, the pipeline slightly differs. The query first goes through fact extraction, where important information about the user is retrieved from their message in the form of triplets. The associative memory agent then searches the database for information relevant to the query and resolves collisions of triplets extracted in the previous step with the existing data about the user. The filtered facts then get mixed in the user's query.

3.1 The Cold Start Agent

The "cold start" agent exists within the pipeline to deal with new users that have little to no information about them. It is important for conversations with our chatbot to be active and interesting even with unknown users, which is the goal that this agent pursues (Table 1).

Our solution to the "cold start" problem is based on a retrained Sentence Transformers (Reimers and Gurevych, 2019) model to encode persona embeddings and find similar personas based on their cosine similarity. The training dataset was derived from Synthetic Persona Chat. First, embeddings of unique facts were encoded with the encoder (BAAI/bge-small-en) [3] and compared using cosine similarity, connecting the personas they. The model was fine-tuned on a new dataset, which was made from positive and negative pairs of personas obtained previously.

User's persona	Similar persona
I love horses	I love animals, I love dancing, I am a vegan, I love country music, I have a farm with pigs, horses and hens, I would like to go to school to become a veterinarian, I am currently on a diet, I love going to the gym, I have three pets, I love animals and I want to help them

Table 1: Example of cold start agent performance.

3.2 The Fact Extraction Agent

The "fact extraction" agent is designed to identify and structure personal information from dialogue in the form of triplets (subject, predicate, object). The metadata fields such as timestamps are stored alongside the triplets in the database and used, for

[3] https://huggingface.co/BAAI/bge-small-en

instance, during collision resolution. This agent aims to build dynamic user profiles and adapt responses based on user-specific information. We extract facts from dialogues using a rule-based method built on top of a syntactic dependency parser (spaCy) (Honnibal et al., 2020), enhanced with coreference resolution via en_core_web_trf [4] transformer-based model with the coreferee [5] plugin.

The extraction process identifies subject-predicate-object triplets by analyzing the syntactic structure of each utterance including support for complex grammatical constructions. The triplets are passed to the next agent and stored in its database as JSON structures.

Unlike end-to-end neural approaches that treat fact extraction as a sequence-labeling task, our approach explicitly models the syntactic and referential hierarchies inherent in conversational data. The core idea is to traverse the syntactic structure of each sentence to detect subject–verb–object patterns and their variants, including passive constructions, gerunds, embedded clauses, and comparative expressions. To enhance the agent's understanding of discourse-level references, we incorporated a tool for coreference resolution. This was essential for accurately interpreting anaphoric expressions such as pronouns, which frequently occur in dialogues.

Coreference resolution is applied as a preprocessing step. Utilizing coreference resolution we rewrite dialogue text by substituting pronouns with their most salient antecedents based on the coreference chain. This preprocessing improves the accuracy of later syntactic parsing by ensuring that each clause contains fully explicit noun phrases, thereby reducing ambiguity in triplet generation.

The syntactic parsing module analyzes each sentence by identifying the ROOT verb and its dependents to form canonical subject–predicate–object triplets. While basic SVO structures are straightforward to extract, natural language often involves more complex grammatical patterns that obscure the core meaning. To ensure accurate fact extraction, we focused on a targeted set of syntactic constructions that are both frequent in dialogue and crucial for preserving semantic relationships. These include passive voice, dative constructions, control and open clausal complements, nested complement clauses, comparatives, full noun phrase reconstruction, and negation propagation.

To illustrate how this system operates on real-world inputs, Table 2 presents an excerpt from a dialogue and the extracted triplets.

Dialogue	Extracted triplets
- I also like football, I don't watch as often as I would like to though.	(I, like, football) (I, do not watch often, football)

Table 2: Extracted subject–predicate–object triplets from a sample dialogue.

3.3 The Associative Memory Agent

The core idea behind the associative memory agent is to treat the user input as a search engine query. This approach reframes the agent's task as a document ranking problem. While extensive research exists on information retrieval techniques (Kureichik and Gerasimenko, 2024) and (Huang et al., 2024), conventional methodologies seem to be unsuitable for our specific task. The crucial incompatibility arises from the fundamental difference in target data: traditional information retrieval methods typically operate on large-scale documents, while the Associative Memory Agent's task is to process triplets. Consequently, techniques such as inverted indexing, term-based search and tree search, optimized for larger text bodies, lack performance in this context.

The proposed solution leverages an embedding-based similarity search to retrieve relevant information. For each triplet extracted from user input (or the entire input string if no triplets are present) a vector embedding is generated using an arbitrary encoder. The cosine similarity is then computed between the query/triplet embedding and all existing embeddings within the database. The five most similar (by cosine similarity) facts are selected from the database and incorporated into a prompt for the LLM. Finally, the extracted triplets are appended to the database.

3.4 The Generation Agent

For our generation agent we used the Transformers library by HuggingFace [6] in order to make a generation pipeline. We chose Gemma3-1B-Instruct[7] as the model that generates the answer. Gemma 3 models follow the general decoder-only

[4] https://huggingface.co/spacy/en_core_web_trf
[5] https://spacy.io/universe/project/coreferee
[6] https://github.com/huggingface/transformers
[7] https://huggingface.co/google/gemma-3-1b-it

transformer architecture (Team G et al., 2025). The reason we chose it is because this model is light-weighted (only 1 billion parameters), therefore it is allowed to use it in real time with low resources.
In our generation agent we use two prompts: the query prompt (Table 3) and the system prompt (Table 4).

Your ROLE: assistant Your TASK: considering the FACTS about USER, give ANSWERS to his REPLIC. EXAMPLE: FACTS about USER: I am a surgeon, I am social with others, I got to the gym all the time, I like cats. USER SAYS: Do cats make good workout buddies? Your ANSWER: Cats are usually too lazy to join your workouts, but they're great at relaxing with you after the gym and the surgeries. Perfect for a hardworking doctor! FACTS about USER: {} USER SAYS: {} Your ANSWER:

Table 3: The query prompt; the curly brackets contain facts about the user and user's query.

The query prompt includes the current user's query, an instruction for the model and the facts about a user that were retrieved from previous queries.
System prompt is the main one. With this prompt we give the model the generation task and then specify it by saying about facts, context and the length of the answer that we expect. We instruct the model to generate a short answer (2-3 sentences), because without such a request, the model may not respond correctly and begin to reason.
Prompts are prepended to the message history (truncated to 300 tokens) and are submitted to the LLM with all previous context. If no history exists, the cold-start agent initializes the context. In response to queries, the model generates a response based on the facts extracted from the user's messages.

I need your help in the generation task. I will show you some facts about my persona (user). You are an assistant. Generate an answer only to the last user's message/query. Consider the previous context (messages) and facts. You should respond only in 2-3 sentences.

Table 4: The system prompt.

4 Evaluation

To evaluate persona usage during the conversation, a custom dataset was constructed based on the dialogue dataset MultiSessionChat. Our dataset contains 100 English dialogue sets, specifically selecting only a specific person turns within each dialogue. For each of these 100 sets, we manually extracted one fact and formulated a related question. After that, we employed the evaluation procedure for the Gemma 3 without and with our proposed pipeline. The evaluation process consisted of the following steps:

- An instance of a generation agent (either baseline Gemma 3 or chatbot pipeline) is initialized.
- For our pipeline, each of the 100 dialogue sets is processed by the fact extractor agent. This step fulfills a database for subsequent associative memory usage.
- For both approaches–the baseline Gemma 3 and our pipeline–the question, associated with the given dialogue, is posed to the generation agent by prompting. Before the response generation, our pipeline using retriever and collision resolving agents extracts relevant facts from the database and removes a conflicting information. For the baseline Gemma 3, we simply add a dialogue context and question to the prompt.
- Finally, we manually evaluate extracted answers with the golden answers from our constructed dataset.

The experimental design treats the series of 100 dialogue sets as a single broad conversation. This approach aims to assess the ability of the agents to maintain and utilize contextual information across multiple turns. Specifically, we hypothesize that baseline Gemma 3, operating with a limited or absent memory of past interactions, will exhibit a reduced ability to recall prior events compared to the chatbot, which is designed to retain and retrieve relevant facts from its associative memory database.
A total of eight experiments were conducted to evaluate the performance of a chatbot pipeline against a baseline Gemma 3 model. The experimental design varied two key factors: the

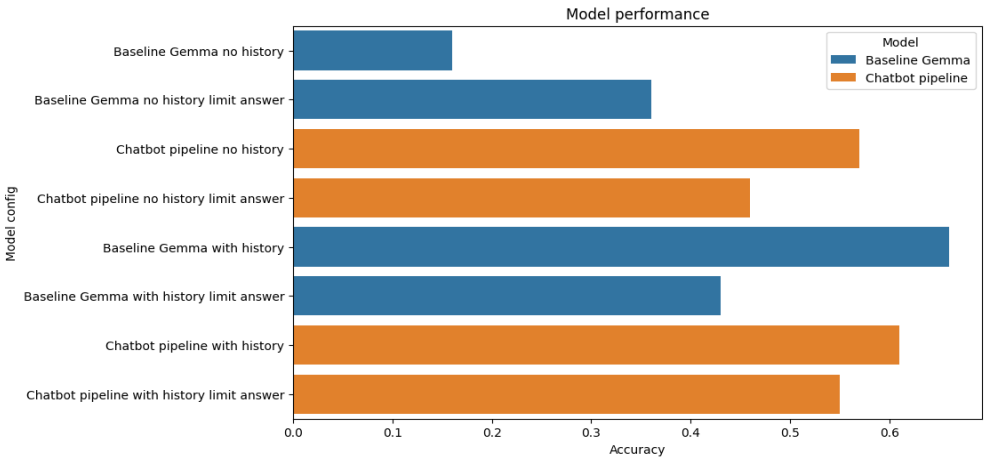

Figure 2: The pipeline performance compared to the baseline Gemma 3.

presence or absence of dialogue history, and the length limitations on the model's response.

5 Results

The results (Figure 2) indicate that the chatbot pipeline outperforms the baseline Gemma 3 model when dialogue history is absent. Specifically, a 41% improvement was observed without response length limitations, and a 16% improvement was observed with response length limitations. Furthermore, the chatbot pipeline outperformed baseline Gemma 3 even with dialogue history enabled in response length limitations conditions (12% margin). However, the chatbot pipeline did not surpass baseline Gemma 3 when both dialogue history and unlimited response lengths were employed. In this configuration, Gemma 3 achieved an accuracy of 66%, while the chatbot pipeline achieved an accuracy of 61%.

One potential reason why our pipeline has lower accuracy than the baseline is that the fact extraction agent extracts noisy information. However, it is worth noting that when using the pipeline without adding conversation history, the accuracy of our approach is almost comparable to using dialogue context. This suggests that our memory-based approach can potentially reduce the memory consumption of response generation in conversational agents.

6 Conclusion

In this study, we presented an approach to personalized chatbot construction by integrating an associative memory framework within a multi-agent pipeline. Through the implementation of the agents (handling cold-start, fact extraction, memory retrieval, and response generation) we demonstrated improvements in several cases in personalization and response accuracy. Thus, our results showed a 41% increase in performance over the baseline Gemma 3 model in memory-constrained settings without access to extended dialogue history.

7 Future Work

Since the fact extraction agent extracts noisy information, further work will be devoted to improving the accuracy of this agent. Since the agent produces false positives quite often, an additional classification model is needed to cope with this problem. The classification model should mark utterances that potentially contain facts. We assume that the combination of a classifier and a parser for fact extraction will reduce the amount of noisy data and, as a result, improve our pipeline.

The next step in our research will be to evaluate the proposed pipeline on other benchmarks. In particular, the LongMemEval (Wu et al., 2025) benchmark aims to evaluate the ability of language models to operate with memory. In this benchmark, there are many dialogues, each of which is divided into long sessions. Our approach to working with memory is close to RAG. Using a fact extraction agent, we can build a database that contains facts and indices of sessions or replicas that contain these facts. This will allow the generation agent to obtain more contextually relevant information for answering a question.

References

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen and Xia Hu. 2024. *LLM Maybe LongLM: Self-Extend LLM Context Window without Tuning.* arXiv preprint arXiv:2401.01325.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang and Mao Yang. 2024. *LongRoPE: Extending LLM Context Window beyond 2 Million Tokens*. arXiv preprint arXiv:2402.13753.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou and Zihan Qiu. 2025. *Qwen3 Technical Report.* 10.48550/arXiv.2505.09388.

Yi-Pei Chen, Noriki Nishida, Hideki Nakayama, and Yuji Matsumoto. 2024. *Recent Trends in Personalized Dialogue Generation: A Review of Datasets, Methodologies, and Evaluations*. 10.48550/arXiv.2405.17974.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. *Generating informative and diverse conversational responses via adversarial information maximization.* Advances in Neural Information Processing Systems, 31.

Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. *Assigning personality/profile to a chatting machine for coherent conversation generation*. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 4279– 4285. International Joint Conferences on Artificial Intelligence Organization.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia *Shao, Diyi Yang, Hamed Zamani, Franck* Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen Ahmed and Yu Wang. 2024. *Personalization of Large Language Models: A Survey*. 10.48550/arXiv.2411.00027.

Hongli Yuan and Alexander Hernandez. 2023. *User Cold Start Problem in Recommendation Systems: A Systematic Review.* IEEE Access. 10.1109/ACCESS.2023.3338705.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. *Lamp: When large language models meet personalization*. arXiv preprint arXiv:2304.11406.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. 2023. *Recommender systems with generative retrieval.* In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 10299–10315. Curran Associates, Inc.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. *Towards open-world recommendation with knowledge augmentation from large language models*. arXiv preprint arXiv:2306.10933.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022. *Long Time No See! Open-Domain Conversation with Long-Term Persona Memory.* In Findings of the Association for Computational Linguistics: ACL 2022, pages 2639–2650, Dublin, Ireland. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. *Personalizing Dialogue Agents: I have a dog, do you have pets too?*. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. *Getting To Know You: User Attribute Extraction from Dialogues.* In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 581–589, Marseille, France. European Language Resources Association.

Zhulin Wang, Xuhui Zhou, Rik Koncel-Kedziorski, Alex Marin and Fei Xia. 2022. *Extracting and Inferring Personal Attributes from Dialogue*. 58-69. 10.18653/v1/2022.nlp4convai-1.6.

Kai Zhang, Yejin Kim and Xiaozhong Liu. 2025. *Personalized LLM Response Generation with Parameterized Memory Injection.* arXiv preprint arXiv:2404.03565.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle and Philip Resnik. 2024. *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*. Preprint at https://arxiv.org/abs/2406.06608.

Pegah Jandaghi, Xianghai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2024. *Faithful Persona-based Conversational Dataset Generation with Large Language Models.* In Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024), pages 114–139, Bangkok, Thailand. Association for Computational Linguistics. American Psychological Association. 1983. Publications Manual. American Psychological Association, Washington, DC.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. *Beyond Goldfish Memory: Long-Term Open-Domain Conversation.* In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Nils Reimers and Irina Gurevych. 2019. *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, 3-7 November 2019, 3982-3992. https://doi.org/10.18653/v1/d19-1410

Matthew Honnibal, Ines Montani, Sofie Van Landeghem and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*. https://doi.org/10.5281/zenodo.1212303

Vladimir Kureichik and Petr Gerasimenko. 2024. *Basic approaches to extracting textual information (overview)*. Izvestiya SfedU. Engineering sciences. 6-14. https://doi.org/10.18522/2311-3103 -2024-4-6

Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang and Yong Yu. 2024. *A Comprehensive Survey on Retrieval Methods in Recommender Systems*. arXiv preprint arXiv:2407.21022.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev and Léonard Hussenot. 2025. *Gemma 3 Technical Report*. 10.48550/arXiv.2503.19786.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, Dong Yu. 2025. *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory.* 10.48550/arXiv.2410.10813.