High-fidelity Human Representation for Large Language Models

Stanford CS224N {Custom} Project

Brian Xu
Department of Computer Science
Stanford University
brianxu@stanford.edu

Henry Weng
Department of Computer Science
Stanford University
hweng@stanford.edu

Abstract

Our primary goal is to understand how to best model a human from textual data in order to retrieve context information about the human relevant for journal-based queries. More concretely, we would like to organize a human's journal entries and devise a retrieval mechanism that would efficiently search the journal for segments of context that would aid a LLM in responding to a user's prompts. By applying an embedding search not only on a user's query itself but also on richer synthetically-generated anecdotes as well as dimensions like recency and salience of memories, we demonstrate a novel retrieval mechanism that meaningfully outperforms Naive RAG for this task.

1 Key Information to include

- Mentor: Olivia Lee
- External Collaborators (if you have any): N/A
- Sharing project: Using some ideas from this project for CS 422 (interactive and embodied learning)

2 Introduction

Mental health is a widespread issue that impacts our society in numerous dimensions. In fact, 1 in 5 U.S. adults struggled with mental illness, 5.5% of adults experience a major depressive episode, and 14.5% of adults concurrently had a substance use disorder and mental health issues in 2021. Serious mental illnesses also cost the US economy an estimated $193.2 billion in lost earnings annually. [1] There are effective solutions for improving mental health and overcoming mental illness, such as Cognitive Behavioral Therapy. However, these solutions often have great barriers to access, such as cost, time, and willingness to get started with treatment for patients.

Our goal is to provide a more accessible point of entry for users to improve their mental wellbeing. By leveraging the reasoning and understanding capabilities of Large Language Models (LLM's), we are working to create an AI agent that can respond to users' wellness-related queries, using a common source of personal memories: journal entries. For instance, given a user's prompt of feeling more anxiety than usual, our agent would ideally select journal entries from the user relating to activities they enjoy or trusted friends they may find solace in. Then, our system would augment a LLM with this context in providing a motivating response to the user.

Our specific task in this paper is to implement a retrieval and storage system for textual journal entry information. Specifically, we will develop a system that takes in a prompt from an user as input, and will return a ranking of the top K relevant text segments from the user's journal. For instance, a user may prompt: "I don't enjoy the things I used to anymore". In response, some ideal

segments the system would retrieve could include: "I have a blast at my swimming practices" or "I felt a bit down today, but Carl helped cheer me up". An example of an entry which should not be prioritized is: "I really don't enjoy eating brussel sprouts". We construct our approach by augmenting elements of Naive RAG in order to better solve the specific task at hand. We compare our results to Naive RAG as a baseline.

3 Related Work

Naive RAG The earliest forms of RAG systems gained popularity after the release of ChatGPT.[2] The Naive RAG process consists of a few key steps: indexing, retrieval, and generation. Indexing describes the process of cleaning documents and splitting them into smaller chunks. This is typically done in a straightforward manner, splitting the document into equal-sized chunks without regard to content. The chunks are then stored in a vector database using an embedding model, allowing for efficient search. The retrieval process uses the same embedding function as indexing and returns the top K chunks by similarity score from the vector database. The generation phase then takes the selected chunks of context and provides it within a prompt to a large language model to formulate a response. Naive RAG retrieval generally faces challenges such as an inability to retrieve all relevant chunks of context.

Social Simulacra Many researchers and creators have attempted to model human behavior through machine representations in the past, whether through simulation sandbox games[3][4] or virtual environments. While recent advances in large language models have led to the resemblance of human behavior, there is a need for architectures that allow for long-term memory and understanding[5], given the complex space of human interactions and emotions.[1] One framework of "generative agents" developed by Stanford researchers proposed several criteria relevant to retrieving relevant memories for believable social agents.[6] These items are: recency (when the memory was last accessed), importance (how mundane or unusual the memory is), and relevance (how related the memory is to the current query). We explore similar criteria in our approach below, incorporating recency and salience as metrics in our Specialized RAG system.

Language Models and Human Behavior In past research, large language models have demonstrated the ability to express complex human behavior from training data. By utilizing chains of prompts[7] and providing personas for each agent, social science studies have been replicated and synthetic data has proven effective at replicating human writing or conversations.[8] The applications of large language models as proxies for human behavior have extended into other fields as well. Models have been used to generate story lines for games as well as help robots plan problem-solving tasks.

4 Approach

Our approach builds upon a Naive RAG implementation, adding in several key components designed to capture relevant information for a more informed search.[9] We build upon some of Naive RAG's limitations. For instance, Naive RAG doesn't take into account the time a memory was made. Additionally, semantic similarity is not the sole metric we want to optimize for. If the user says "I've been feeling anxious recently", it is probably useful to find times in which they are anxious, but it is also very useful to learn about their coping mechanisms and how they have previously dealt with anxiety.

Taking into account these considerations, we propose a modified retrieval method known as Specialized RAG. First, we split a given user's journal into chunks semantically. We then apply a novel approach of generating "sample memories" to conduct a richer search than simple semantic similarity with a query. We independently evaluate the saliency and recency of each memory as well. Finally, we combine all of these measures into a weighted sum score and sort the memories by score, returning the top K. More precisely, the score will be:

$$\text{Score(Memory } i) = a(1-\alpha)^{\text{time}} + \text{salience} + \max_j(\text{L2-similarity(Memory i, Sample Memory j)} \tag{1}$$

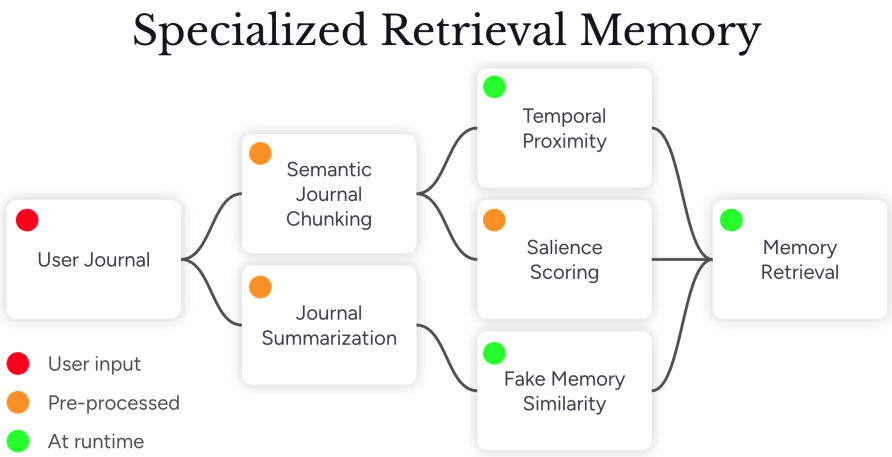

Figure 1: Our Specialized RAG Architecture

We will break down each component of this scoring function below.

Semantic Splitting Naive RAG systems typically perform a naive splitting algorithm, breaking a document into chunks of equal length. However, this approach can break coherent memories into multiple chunks and can lead to a failure to retrieve all relevant context. Thus, we employ a semantic splitting technique. We make use of a LlamaIndex implementation that iterates over naive chunks, computing the embedding similarity between subsequent sentences to decide whether to create a new chunk. Overall, this results in memories that are more self-contained and express complete thoughts.

High-level Summaries We incorporate high-level summaries of each user's journal entry in order to augment further steps in our Specialized RAG system. These summaries will be used to provide additional context in prompts relating to sample memories described in the next section, as well as in our evaluation to allow for a fuller picture of how retrieved context relates to a user.[10] Summaries are typically up to three paragraphs in length.

Sample Memories The generation of sample memories is a novel component core to our Specialized RAG system. Given a user's prompt, we will query a large language model to provide sample memories that a therapist would find helpful in crafting a response. Rather than conducting naive retrieval by performing an embedding search on only the query, we will perform embedding search on these generated sample memories.

The intuition behind this system is that for open-ended or general user queries, we may not be able to capture useful context by simply returning memories that are semantically similar to the query. For instance, if a user says "I've lost interest in a lot of things lately", we would find it more useful to retrieve memories potentially relating to activities the user enjoys or friends they could find solace in, rather than simply a semantically similar memory such as "I lost interest in eating brussel sprouts".

We experimented with several methods of generating these sample memories. In particular, we tried querying a large language model several times for a single sample memory per user and query. We also tried querying the model for a batch of memories all at once per user and query. We found that overall, querying for a batch of memories resulted in a greater diversity of memories which was ideal for providing richer context in a response to the user.

We generated 5 sample memories per user query. Once the sample memories are generated, we iterate through each of the user's memories and computed L2-similarity scores between the user's memory and each of the 5 sample memories. For each of the user's memories, we stored the

maximum L2-similarity score associated with a sample memory. This is used as one factor in our Specialized RAG's weighted sum to determine a memory's relevancy:

$$\text{Sample Memory Score}(\text{Memory } i) = \max_j(\text{L2-similarity}(\text{Memory i, Sample Memory j}) \quad (2)$$

Salience We assessed each memory in a user's journal for a dimension we named salience: how mundane or out of the ordinary a given memory was. In order to quantify this metric, we query a large language model to rate the salience of a memory from a scale of 1 to 10. We provide few-shot examples and ask the model to provide analysis backing up their answer, which improves the accuracy of the output.[11][12] This operation is performed once in preprocessing all memories within a journal and is stored, so it is not an expensive operation in future queries. Salience is another factor in our Specialized RAG's weighted sum score.

Recency We believe that recency is an important metric for consideration as well. Our Specialized RAG incorporates a factor of recency for each memory, represented by an exponential decay term based on the entry's date. We believe this to be especially pertinent for journal entries discussing time intervals or sudden changes in a user's mental state. For instance, if a user queries: "I feel like I can't get out of bed these days," we would like to prioritize memories relevant to recent changes rather than memories from years past, with all other metrics being held equal. We start our recency metric with a score of 1 on the day a memory is entered and apply an exponential decay for the amount of days that have elapsed, with $\alpha = 0.01$ and $a = 0.3$:

$$\text{Recency Score}(\text{Memory } i) = a(1-\alpha)^{\text{time}} \quad (3)$$

5 Experiments

5.1 Data

Because there is no publicly available dataset of people's journal entries, we devised a LLM pipeline to generate a synthetic dataset of realistic journal entries from a diverse set of people.[13] Our dataset consists of the journal entries of 8 people, each of whom has written 30 several-paragraph-long entries over the course of a year.

We generated this dataset by first prompting Claude 3 Opus to generate 8 distinct personas, each of which consists of 20 factors that are typically significant parts of a person's identity. For each persona, we prompted Claude 3 Opus to generate 30 journal entries written over the course of a year based on that persona.

5.2 Evaluation method

During evaluation, we have two parts: LLM-based and human-based evaluations. Overall, we hope to capture the relevancy of retrieved memories in relation to a given user's background and their query. For our LLM-based evaluation, we prompt Claude 3 Sonnet to evaluate the relevancy of each batch of 5 retrieved memories as context for a user's query on a scale of 1 to 10. We provide Claude with a summary of the user's persona in order for it to make an informed decision. We then compute the average scores over 5 different user queries for each persona, comparing the ratings of our Specialized RAG system with ratings generated from a baseline Naive RAG system.

For our human-based evaluation, we make all batches of memories blind and have human participants perform head-to-head evaluations between two batches of memories for each query. One will be the batch retrieved by our Specialized RAG system and the other will be the batch retrieved by Naive RAG. Humans will also see each user's persona for further context when conducting these comparisons.

5.3 Experimental details

When generating our synthetic dataset, we used Claude Opus 3 with a max token length of 1000. For all other large language model queries, including evaluation, we used Claude Sonnet 3 with a max token length of 1000. We provided few-shot examples for evaluation and saliency scoring. For embedding search, we used a Chroma vector database and used the "all-MiniLM-L6-v2" embedding model as our embedding function.

5.4 Results

After running our LLM-based evaluation method over our synthetic datasets, we observed the results in Figure 2. We have included the performance of several subsets of our full Specialized RAG system to infer the relative importance of each component. We see that our fully-featured Specialized RAG performs best on average with an average score per persona of 8.0, compared to a Naive RAG score of 6.6. This was expected, as we incorporated many relevant dimensions of information into our retrieval in this system. Interestingly, the performance of only using our sample memory system is quite similar, with an average score per persona of 7.8. This could suggest that our sample memory system is contributing a significant portion of the improvement of our system over Naive RAG. We see that our system with only salience and recency barely outperformed Naive RAG with a score of 6.7. This suggests that salience and recency add marginal value but may be better expressed in a different form or are uniquely effective for specific queries.

We've also included the latency of our various methods in Figure 3. We can see that our Specialized RAG system provides a significant improvement in memory retrieval but at the cost of much higher latency, with an average of 12.030 seconds. The bulk of our latency is derived from the sample memory system, as we see that this system requires 10.947 seconds on average, which is far higher than the average 0.344 second latency of Naive RAG.

Our human head-to-head evaluation yielded a preference of our full Specialized RAG system's retrieved memories over Naive RAG 80% of the time. This was particularly exciting as human evaluation can be seen as more accurate to the ultimate use case of this project's application, and there was a clear statistically significant preference for our system's retrieved memories.

Persona	**Full Specialized RAG**	**Sample Memories Only**	**Salience and Recency**	**Naive RAG**
Aisha Al-Farsi	8.0	9.1	8.3	7.4
Angela Martinez	5.8	6.2	6.8	6.2
Brian Lee	6.0	6.2	6.4	3.4
Carol Stevens	7.6	5.2	6.0	6.2
Casey Robinson	6.9	6.5	4.8	3.8
David Kim	5.5	5.2	5.0	5.0
Dmitri Ivanov	8.0	8.2	3.8	7.7
Vanessa Johnson	8.7	9.0	6.8	7.3
Average	8.0	7.8	6.7	6.6

Figure 2: Average Relevance Score per Persona for Various RAG Systems

Method	**Average Latency (s)**
Full Specialized RAG	12.030
Sample Memories Only	10.947
Salience and Recency	0.352
Naive RAG	0.344

Figure 3: Average Latency for Various RAG Systems

6 Analysis

Beyond the quantitative results of our Specialized RAG system, we came across several qualitative observations as well. During human evaluation, participants preferred the Specialized RAG's retrieved memories over the Naive RAG memories, as Naive RAG struggled to take into account time of memory. This effect was particularly pronounced for queries with subtle timing cues, such as "I've been feeling down **recently**."

Additionally, our Specialized RAG system tended to perform better on both LLM-based and human-based evaluations for queries that were more open-ended. For instance, when simply given the query "I'm depressed", our Specialized RAG system retrieved memories that were relevant to a persona's past healing experiences and recent memories that may have been tied to recent changes in emotion. However, the Naive RAG system simply searched for moments when the user had used similar wording in phrases of their journal.

7 Conclusion

Through this project, we have presented a novel framework for a RAG system specialized to the task of encoding human memories and aiding a large language model in responding to a user to better their wellbeing. We built our architecture around flaws in Naive RAG, such as inability to account for time of memories as well as limitations of simple semantic search of the query. We find that our system meaningfully outperforms Naive RAG on both LLM-based and human-based evaluations, and we discover that most of this change is driven by our mechanism for generating sample memories as an intermediate step in our retrieval. However, this improved performance comes at a cost of significantly higher latency for the retrieval process.

Our work has a few primary limitations. First of all, we were unable to run our system on actual human journals due to privacy concerns and a lack of public datasets for this task. Using LLM-generated datasets could potentially lead to biases, which should be noted for future work on this problem. It would be greatly fascinating to understand how our system performs on real journals and to implement this system for our own personal use as well. Additionally, we did not have professional therapists available for our human-based evaluations. Given more time and resources, it would be great to gain additional insights on decisionmaking regarding relevant memories.

There are many exciting avenues for future work relating to this project. We considered various additional possible modules in our RAG system, such as re-ranking algorithms and tree-based recursive reflections of previous memories. We also discussed our evaluation methods at length. We considered the effectiveness of sorting all memories by relevance using a LLM for elementary pairwise comparisons, and found that it was strangely inconsistent. There were also interesting properties that arose from ordering text differently within a prompt that can be further studied.

References

[1] National Alliance on Mental Illness NAMI. Mental health by the numbers, 2023.

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[3] Mark O. Riedl. Interactive narrative: a novel application of artificial intelligence for computer games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 2160–2165. AAAI Press, 2012.

[4] Teng Li, Vikram K. Narayana, and Tarek El-Ghazawi. Accelerated high-performance computing through efficient multi-process gpu resource sharing. CF '12, page 269–272, New York, NY, USA, 2012. Association for Computing Machinery.

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi,

Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[6] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.

[7] Tongshuang Wu, Michael Terry, and Carrie J. Cai. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts, 2022.

[8] Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA, 2022. Association for Computing Machinery.

[9] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp, 2023.

[10] Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback, 2021.

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2021.

[12] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3?, 2021.

[13] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. Evaluating large language models in generating synthetic hci research data: a case study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.