

Article

Not peer-reviewed version

LLM- Based Personalized Recommendations in Health

[Boris A. Galitsky](#) *

Posted Date: 29 February 2024

doi: 10.20944/preprints202402.1709.v1

Keywords: Large Language Models; personalization; recommendation; meta-prompts; abductive reasoning



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

LLM- based Personalized Recommendations in Health

Boris Galitsky

Knowledge-Trail Inc, Los Banos CA; bgalitsky@hotmail.com

Abstract: Today, LLMs are not good at personalization providing recommendation. They advise physicians and financial advisors to ask professionals in respective fields for help, even having user information available. Answering questions of software professionals, LLM needs to deliver in-depth answers with codes or algorithms, whereas for professionals in other fields would need definitions and main concepts. The intent of this chapter is to make LLM answer tailored to the needs of users, taking into account available information about them. To do that, we need to generalize available information about a person like her health record, maintaining the privacy of this person. We rely on meta-learning techniques to design a LLM prompt to produce a personalization prompt to obtain a suitable relevant information. Such “meta-prompt” is produced by generalization operation applied to available documents for the user. These documents need to be de-identified so that they are sufficient for personalization on one hand and will maintain user privacy on the other hand. The second neuro-symbolic technique to support personalization is abductive reasoning, acting in parallel to LLM fine-tuning. Traditional recommendation and personalization techniques as well as modern, deep learning – based are presented, and the comparison is drawn to the proposed approach. We also share the evaluation and comparative analyses of these approaches. We consider an example for how to build personalization LLM systems coming from Langchain platform. We will explore how to construct chains to form a personalization profile for a user and apply it to user search and recommendation requests. <https://github.com/bgalitsky/LLM-personalization>

Keywords: Large Language Models; personalization; recommendation; meta-prompts; abductive reasoning

1. Introduction

While prominent Natural Language Processing (NLP) benchmarks like GLUE and SuperGLUE (Wang et al., 2018) have undeniably advanced the NLP landscape, they often adhere to the prevailing “one size fits all” paradigm in modeling and evaluation. Unfortunately, this approach restricts the development of models adaptable to the specific needs of end-users, hindering extensive research on personalization in NLP tasks. In contrast, our paper introduces a comprehensive evaluation framework that incorporates various tasks demanding personalized recommendations.

Language Models (LLMs) are poised to revolutionize the interaction between humans and personalization systems. Unlike conventional recommender systems and search engines, which act as passive mediums for information filtering, LLMs establish the groundwork for active user engagement. This transformative foundation enables proactive exploration of user requests, delivering required information in a natural, interactive, and explainable manner. Additionally, LLMs will significantly broaden the scope of personalization, evolving beyond the sole function of gathering personalized information to a composite function of offering personalized services.

Harnessing large language models as a versatile interface, personalization systems can compile user requests into plans, invoke external tools’ functions (e.g., search engines, calculators, service APIs, etc.) to execute these plans, and integrate the tools’ outputs to accomplish end-to-end personalization tasks. This dynamic approach marks a paradigm shift in how personalization is conceptualized and executed, promising a more interactive and expansive user experience.

Today, LLMs are still being rapidly developed, whereas their personalization capabilities are not significantly addressed. We analyze the challenges in personalization and the opportunities for how personalization can work on top of LLMs.

We discuss development and challenges for the existing personalization system, the newly emerged capabilities of LLMs, and the potential ways of making use of LLMs for personalization.

While personalization has been extensively investigated across various AI application domains, such as information retrieval (IR) and human-computer interaction (HCI), particularly in the context of search engines and recommender systems, its exploration within NLP has been somewhat limited, as highlighted by Xue et al. in 2009. Flek's recent work (2020) has underscored the significance of personalization, especially in tasks involving text classification and generation. This emphasizes the potential of personalization to prioritize users and create systems that are not only accessible but also inclusive. Furthermore, recent studies, including those by Kirk et al. (2023), have illuminated the advantages of personalizing LLMs. Despite these insights and the recognized importance of personalization in numerous real-world scenarios, the development and evaluation of LLMs for generating personalized responses have received limited attention. Consequently, in this chapter, we highlight the pivotal role of personalization in shaping the future of LLM integration systems. This marks the initial step towards the development and evaluation of personalization within the context of LLMs.

1.1. Personalization in healthcare

There is a number of factors making LLMs in healthcare popular:

- (1) The increasing availability of data: LLMs are trained on massive datasets of biological and medical texts. The healthcare industry is generating more data than ever before, which is making it possible to train LLMs that can be used to improve patient care.
- (2) The decreasing cost of computation: The cost of computing power has been decreasing steadily for many years. This has made it possible to train and deploy LLMs at scale.
- (3) The increasing demand for personalized healthcare services: Patients are increasingly demanding personalized healthcare that is tailored to their individual needs. LLMs can be used to generate personalized treatment plans and recommendations, which can help to improve patient outcomes.
- (4) The potential to improve efficiency and accuracy: LLMs can be used to automate a variety of tasks in healthcare, such as scheduling appointments and generating reports. This can free up healthcare professionals to focus on providing care to patients. LLMs can also be used to improve the accuracy of diagnosis and treatment.

In healthcare, the enthusiasm surrounding LLMs stems from their potential to enhance patient care. As LLMs progress in development, we anticipate witnessing further innovative and effective applications of this technology in the future. Nevertheless, it is crucial to acknowledge that LLMs represent a relatively new technology, and their application in healthcare comes with potential risks and limitations. Examples include susceptibility to bias and security attacks. Prior to implementing LLMs in healthcare settings, careful consideration of these risks and limitations is imperative.

In the healthcare sector, LLMs contribute to enhancing patient care through various avenues, including:

- **Diagnosis and Treatment:** LLMs are valuable for analyzing patient data to identify potential health issues, generate treatment plans, and recommend medications.
- **Research:** LLMs play a role in analyzing extensive datasets of medical research papers, aiding researchers in identifying new trends and gaining insights.
- **Education:** LLMs can create personalized learning experiences for healthcare professionals and develop educational resources for patients and their families.
- **Administrative Tasks:** LLMs offer automation capabilities for diverse administrative tasks in healthcare, such as appointment scheduling

By automating healthcare services, improving diagnosis and treatment, and supporting research, LLMs can help to make healthcare more efficient, effective, and accessible.

1.2. Advantages of large language models in healthcare AI

The utilization of LLMs in healthcare offers several advantages over traditional Machine Learning (ML) approaches:

- (1) **Improved Diagnosis and Treatment:** LLMs excel in analyzing patient data to identify potential health issues, generate treatment plans, and recommend medications. This enhances the accuracy and efficiency of healthcare delivery.
- (2) **Support for Research:** LLMs contribute to the analysis of extensive datasets of medical research papers, enabling researchers to identify emerging trends and insights. This, in turn, can lead to the development of new treatments and cures.
- (3) **Personalized Learning Experiences:** LLMs can craft personalized learning experiences for healthcare professionals, aiding them in staying abreast of the latest medical knowledge and best practices.
- (4) **Automated Administrative Tasks:** LLMs are adept at automating various administrative tasks in healthcare, including appointment scheduling and report generation. This automation liberates healthcare professionals to concentrate on delivering care to patients.

In summary, the integration of LLMs in healthcare has the potential to revolutionize the industry, fostering efficiency, effectiveness, and accessibility. As LLMs evolve, we anticipate witnessing even more innovative and effective applications of this technology in the coming years.

A health recommender system (HRS) provides a user with personalized medical information based on the user's health profile.

The most adopted algorithm of recommendation technologies was the knowledge-based approach. Despite existing research progress on HRSs, the health domains, recommended items, and sample size of system evaluation have been limited. In the future, HRS research shall focus on dynamic user modelling, utilizing open-source knowledge bases, and evaluating the efficacy of HRSs using a large sample size.

People are increasingly using the internet to search for health information. Google receives more than 1 billion health questions every day and seven percent of Google's daily searches are health related (Beckers health 2021). Pew Research Center (2021) even showed that eighty percent Internet users have searched for health information (e.g., diet, fitness, drugs, health insurance, treatments, doctors, and hospitals).

Nevertheless, the abundance and intricacy of health-related information have the potential to overwhelm an individual's capacity for information processing, resulting in incomplete or inaccurate consumption of information. Moreover, online information is typically not customized to meet the specific needs of each patient, as highlighted by Carter et al. (2011). Additionally, there is a considerable variation in users' health literacy levels, with some lacking the necessary skills to comprehend medical terminology and vocabulary, evaluate the relevance of extracted data, or verify the validity of information sources. Consequently, the experience for consumers seeking health information on the internet can be notably challenging.

Recommender techniques are traditionally divided into different categories:

- (1) *Collaborative filtering* is the most used and mature technique that compares the actions of multiple users to generate personalized suggestions. An example of this technique can typically be found on e-commerce sites, such as "Customers who bought this item also bought..."
- (2) *Content-based filtering* is another technique that recommends items that are similar to other items preferred by the specific user. They rely on the characteristics of the objects themselves and are likely to be highly relevant to a user's interests (De Croon et al. 2021). This makes content-based filtering especially valuable for application domains with large libraries of a single type of content, such as MedlinePlus' curated consumer health information (Bocanegra et al. 2017).
- (3) *Knowledge-based filtering* is another technique that incorporates knowledge by logic inferences. This type of filtering uses explicit knowledge about an item, user preferences, and other

recommendation criteria. However, knowledge acquisition can also be dynamic and relies on user feedback.

- (4) *LLM-based filtering* is based on prompting and reasoning neuro-symbolic approaches and is the focus of this chapter.

1.3. Contribution

In this chapter, we first explore how to personalize textual data along with an individual item recommendation, given a sequence. Once we optimize what needs to be done with LLM for that, we proceed to the medical domain where an abstract item is instantiated by a clinical treatment and additional health-specific considerations need to come into play. The innovations are as follows:

- (1) The chapter introduces a neuro-symbolic architecture for personalizing Large Language Models (LLMs). In the realm of prompting techniques, we advance towards meta-prompting, illustrating how these meta-prompts construct user personalization profiles, subsequently applying them during search operations to yield personalized outcomes.
- (2) We advocate for the incorporation of abductive reasoning to deduce the most suitable answer for a user based on her personalization profile. Abduction occurs in parallel to fine-tuning to iteratively improve the personalization.
- (3) After scrutinizing existing personalization architectures, we identify those most compatible with LLM integration. Health personalization recommendation methods undergo a comparative analysis, with the selection of the most effective LLM-based approach.
- (4) Our exploration concludes with a practical exercise involving fine-tuning LLM using treatment recommendation data.

2. Personalization scenarios

LLMs have showcased remarkable capabilities in harnessing their internal world knowledge and common-sense reasoning to accurately discern user intent during dialogues. Additionally, LLMs exhibit the ability to engage with users in a natural and fluent manner, providing a seamless and delightful user experience. These strengths render LLMs an enticing option for recommendation systems aimed at improving personalized experiences.

However, notwithstanding the impressive memory capacity of LLMs, they encounter difficulties in memorizing specific knowledge within private and specialized domains when lacking sufficient training. For example, the storage of extensive item databases and user profiles in a recommender system poses a considerable challenge for LLMs. This limitation can result in LLMs generating inaccurate or erroneous responses, making it challenging to govern their behavior within a particular domain. Furthermore, LLMs grapple with the temporal generalization problem as external knowledge continues to evolve and change over time.

To tackle these issues, various tools can be employed to enhance LLMs and bolster their effectiveness as recommendation agents.

- 1) Search engine. Search engines are widely employed to provide external knowledge to LLMs, reducing LLMs' memory burden and alleviating the occurrence of hallucinations in LLMs' responses.
- 2) Recommendation engine. Some works have attempted to alleviate the memory burden of LLMs by equipping them with a recommendation engine as a tool, enabling LLMs to offer recommendations grounded on the item corpus.

The recommendation engine in Chat-REC (Gao et al. 2023) is further divided into two stages: retrieve and reranking, which aligns with typical recommendation system strategies. In the retrieval stage, LLMs utilize traditional recommendation systems as tools to retrieve 20 items from the item corpus as a candidate item set. Subsequently, LLMs employ themselves as tools to rerank the candidate item set. LLMs' commonsense reasoning ability, coupled with the internal world knowledge within them, allow them to provide explanations for the sorting results. The recommendation engine tool used in RecLLM (Friedman et al. 2023) is highly similar to the one in Chat-REC, and it is also divided into retrieval and reranking stages.

Bottom of Form

An overview of our proposed personalized recommendation architecture is shown in Figure 1. The essential recommendation components are:

- (1) *Abductive reasoning*, which requires an LLM to provide an explanation why a certain item is recommended, given the Personalization User Profile;
- (2) *Object-level Prompt*, which is formulated as an exact task of providing an answer for a certain category / specialty of the user;
- (3) *Meta-level Prompt*, which is an abstract request such as finding commonalities in user documents, extracting them from user documents.

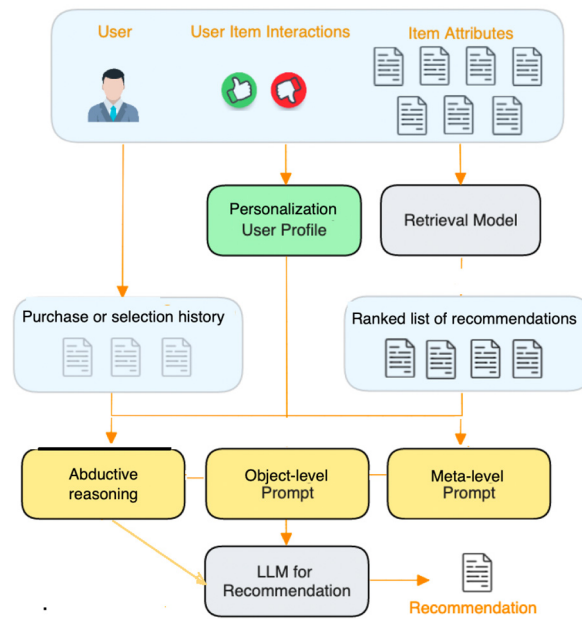


Figure 1. A Hybrid prompts + abduction LLM personalization architecture.

The “Purchase or Selection History” is created by concatenating the items that the user has interacted with. The “Personalization User Profile” is a high-level summarization of the user’s preferences, generated using an LLM based on user-item interactions, item attributes, or even user information if possible. The “Candidates for Recommendation” are the output of an arbitrary retrieval component.

2.1. Problem formulation

We consider every data sample as an independent user u , with a set of user records associated with each sample across all tasks. These user records are instrumental in tailoring language models to individual users using their specific data. Consequently, each data sample can be divided into three distinct components: an input sequence, which serves as the model’s input; a target output, which represents the expected model output; and a profile, which encapsulates any supplementary information that can be used to customize the model in accordance with the user’s unique preferences or needs.

A general recommendation problem can be defined as the task of maximizing a utility function M . M serves as a measure of how useful an item s is to a recommendation context c , represented as $M: C \times S \rightarrow R$, where R is a totally ordered set. Here, the recommendation context c is intended as an information collection composed of the user profile, i.e., the list of her preferred items S_{user} , and those recommended by the system at recommendation time S_{rec} , such that $C = \{S_{user}, S_{rec}\}$. The primary objective is to select an item $s' \in S$ for each recommendation context $c \in C$ that maximizes their utility.

$$\forall c \in C, s'_c = \arg \max_{s \in S} u(c, s).$$

In our case, personalized recommendation is provided via text. For a given textual input t , the goal is to develop a model M that generates personalized output o for user u . This task is formulated as $\arg \max_o p(o|t; u)$. For each user u , the model M can take advantage of user profile $P_u = \{(t_{u1}, o_{u1}), (t_{u2}, o_{u2}), \dots, (t_{um}, o_{um})\}$ where each (t_{ui}, o_{ui}) denotes a pair of input text t and personalized output o of user u (Figure 2).

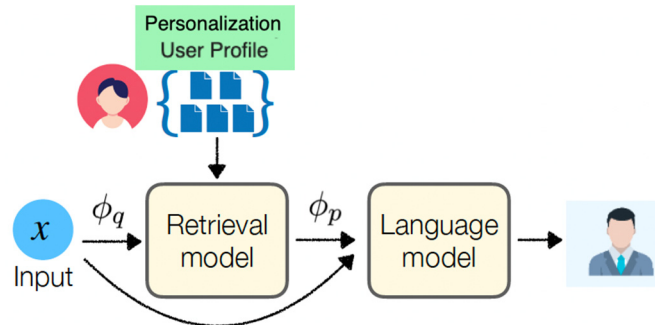


Figure 2. Formulating personalization problem for IR and LLM.

Each personalization user profile comprises an extensive array of data points specific to that user. However, due to the natural limitations on context length in LLMs and considerations of efficiency and cost, it is feasible to include only a subset of these data points as input prompts. Not all elements within a user's personalization profile may be directly applicable to the particular task the user intends to undertake. IR plays a selective role in extracting the relevant data from the user profile that includes the current, unseen test case.

To compute personalization for a given sample in personalization user profile (t_{ui}, o_{ui}) , three steps are used:

- (1) a query construction component that transforms the input t_i into a query q for retrieving from the personalization user's profile;
- (2) a retrieval component $IR(q; P_u)$ that accepts a query q , a user profile P_u and retrieves entries assumed to be most significant from the user profile; and
- (3) a prompt construction component p that assembles a prompt for user u based on input t_i and the obtained entries.

LLM input is obtained using the following: $t_i^{LLM} = \phi_p(t_i, IR(\phi_q(t_i), P_u))$. (t_i^{LLM}, o_i) is used to train and evaluate LLM.

To express the similarity between a recommending system and an LLM, consider the following. A LLM estimates the probability $P(x_1, x_2, \dots, x_T)$ of observing sentence tokens in the given order, namely estimating the entire sequence's joint probability. Applying the chain rule, this probability is expressed as

$$P(x_1, x_2, \dots, x_T) = \prod_{t=1}^T P(x_t | x_1, \dots, x_{t-1})$$

An ideal LLM is expected to autonomously generate text by sequentially selecting individual tokens

$$x_t \sim P(x_t | x_{t-1}, \dots, x_1)$$

Hence both LLM and a recommender learn to predict the most probable next token and next item respectively (Di Palma 2023).

Let us consider an recommendation example. The input is: "I find the purchase history list of user_15466:

4110 -> 4467 -> 4468 -> 4472 I wonder what is the next item to recommend to the user. Can you help me decide?"

The output is: "1581"

(Geng et al. 2022) assigned the user and each item a unique ID. Using a training set with thousands of users (and their purchase histories) and unique items, the LLM is able to learn that certain items are similar to one another and that certain users have inclinations towards certain items (due to the nature of the self-attention mechanism). During the pre-training process across all these purchase sequences, the model essentially goes through a form of collaborative filtering. It sees what users have purchased the same items and what items tend to be purchased together. Combine that with an LLM's ability to produce contextual embeddings, and we suddenly have a very powerful recommendation system.

In this example, although we do not know what item each ID corresponds to, we can infer that item "1581" was selected due to other users purchasing it along with any of the items that "user_15466" already purchased (Aboufoul 2023)

3. Transformer-based personalization

The survey by Yang and Flek (2020) contains an overview of tasks and applications where the main goal is to adapt a model or system to the needs of the user in the area of dialog personalization. There are also works in summarization, style and formality adaptation, and text simplification. The authors of the survey make an observation that personalization techniques vary broadly in the type of condition for personalization. Specifically, the text can be personalized based on factual knowledge of a user, i.e., some meta information, or on stylistic modifiers commonly used by users. It is also mentioned in the survey that a notion of personal writing style is broad and can include emotional stylistic modifiers, sociodemographic traits, and personality specific modifiers.

For example, Fidler and Goldberg (2020) use conditional LLM approach to generate more personalized movie review. They use an RNN with conditional style context: whether a review should be descriptive, professional or personal; the length of the review could also be provided as a condition.

Syed et al. (2020) use a transformer with the additional encoder-decoder framework for fine-tuning towards the style of one of the ten Gutenberg authors. Their results state this approach rewrites the input text with better alignment to the target style. Skopyk et al. (2022) develop a model that trained on users' previously written records will produce a more personalized text for each of them.

Welch et al. (2022) consider approaches to fine-tuning and interpolation that leverage data from similar users to boost personalized LLM performance. The authors consider the case of users with a small number of available tokens and propose ways to find similar users in their corpus and also leverage data from similar users to build a personalized LLM for a new user. trade-offs between the amount of available data from existing users were analyzed, as well as the number of existing users and new users, and how the authors' similarity metrics and methods scale. The study includes the analysis of what types of words our method predicts more accurately and are thus more important.

The diagram by Welch et al. (2022) illustrates the utilization of data, models, and metrics (Figure 3). The diagram comprises two primary sections: a left rectangle demonstrating the computation of three similarity metrics and a right rectangle depicting two methods for leveraging similar user data to create personalized models. Solid lines indicate the flow of golden set user data, while a dashed line represents data from a new user.

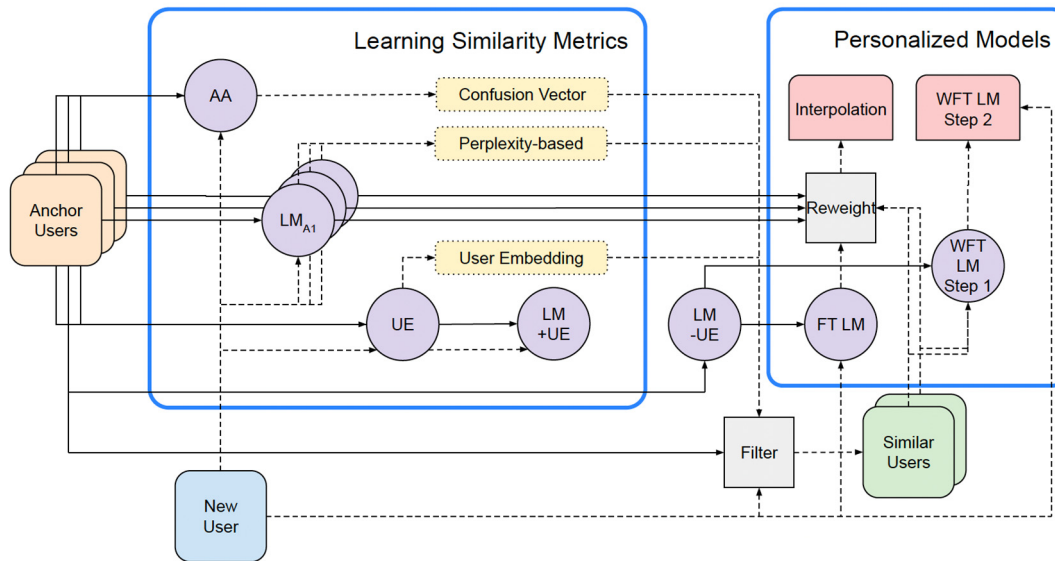


Figure 3. Personalization architecture that leverage data from similar users to boost personalized LLM performance.

Golden set user data is employed to construct the authorship attribution model (AA), individual user LLMs for the perplexity-based metric (depicted as a set with the first as LM_{A1}), and user embeddings (UE). These three metrics are then utilized to filter anchor user data to identify similar users. With this group of users, a baseline LLM is fine-tuned (without UE, denoted $LM-UE$), and subsequently, it undergoes further fine-tuning using new user data for the weighted fine-tuning method (WFT LM). During interpolation, the LLMs of individual golden set users are reweighted and combined with predictions from an LLM fine-tuned on new user data (FT LM).

Given N trained LLMs, one for each user, Welch et al. (2022) use the perplexity of one LLM on another user's data as a measure of distance. We could compare the word-level distributions, though this would be very computationally expensive. In their experiments, Welch et al. (2022) use the probability of the correct words only, or the perplexity of each model on each new user's data. The LLM is trained on all golden set users and fine-tuned for each anchor user. Then the perplexity of each model is measured on the data of each new user. For this matrix of $new \times golden_set$ perplexities, each row, representing a new user, is turned into a similarity vector by computing

$$1 - \frac{c - \min(row)}{\max(row)} \text{ for each cell, } c.$$

3.1. History of recommendation development

The evolution of recommendation technology spans several decades, marked by significant advancements in ML and NLP (Figure 4). Here is a brief history:

- (1) MDL (Minimum Description Length): The concept of recommendation systems can be traced back to the late 20th century, with early models like MDL. These systems focused on simplicity and used algorithms to minimize the description length of the recommendation, laying the foundation for future developments.
- (2) MLP (Multilayer Perceptron): As neural networks gained prominence in the late 20th century, MLPs became a notable recommendation technology. These multilayered artificial neural networks were used for collaborative filtering and content-based recommendations.
- (3) BERT (Bidirectional Encoder Representations from Transformers): In 2018, BERT, a transformer-based model, revolutionized natural language processing. BERT excelled at understanding context and semantics in textual data, enhancing recommendation systems' ability to grasp user preferences and deliver more accurate suggestions.
- (4) GPT (Generative Pre-trained Transformer) Series: GPT models, developed by OpenAI, represent a breakthrough in recommendation technology. Starting with GPT-1 and progressing to GPT-3.5, these models are pre-trained on massive datasets, enabling them to generate human-like text

and understand context, making them applicable to a wide range of natural language processing tasks, including advanced recommendation systems.

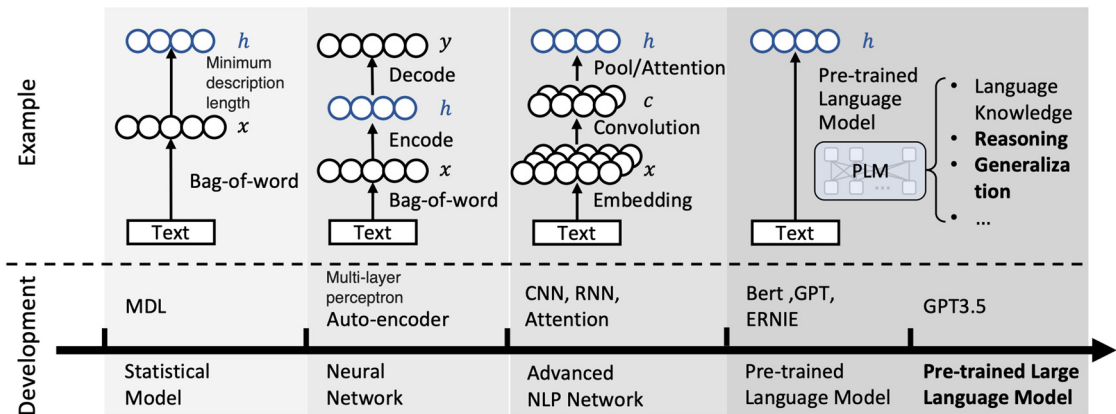


Figure 4. Aligning recommendation architectures with deep learning architectures (Chen et al. 2023).

The GPT-3.5, in particular, is a powerful language model with 175 billion parameters, making it one of the most sophisticated models for NL understanding and generation. Its large-scale pre-training allows it to perform exceptionally well in various recommendation tasks, understanding and generating human-like responses. Overall, from the early days of MDL to the current state-of-the-art GPT-3.5, recommendation technology has undergone a remarkable journey, leveraging advancements in neural networks and language models to provide increasingly accurate and context-aware suggestions to users.

4. Offline and online functionality

At indexing time, upon permission, the personalization system builds a personalization profile (Figure 5). Once it gains the access to personal files on a computer or on a cloud, it classifies the file as useful for personalization or useless for it. Then documents which looks promising for personalization are de-identified. Only text with generalized features like income level or an anonymized disease is retained.

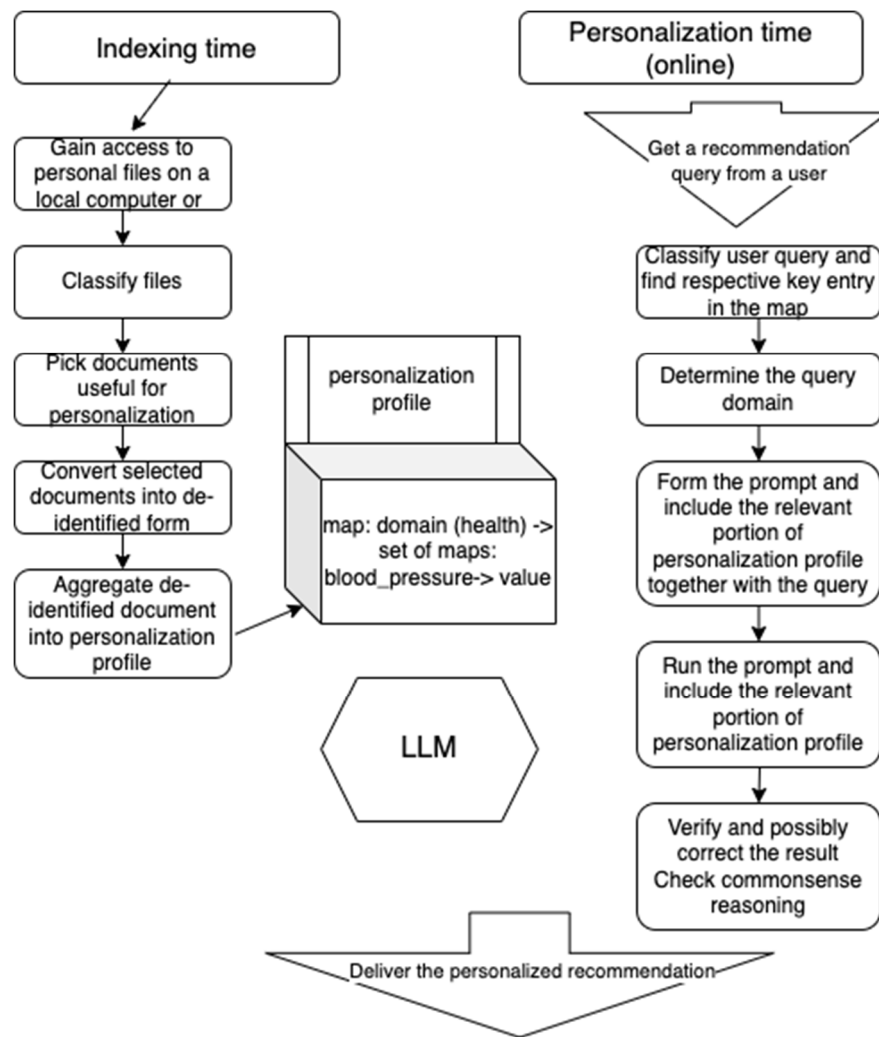


Figure 5. Overall system architecture.

Then we aggregate the totality of de-identified documents: anonymized financial complement other financial documents, and anonymized health records complement other health records. As a result, we include financial needs, health issues, landlord-type recommendations, travel advice etc. obtained from the available document.

A personalization profile includes the attribute value pairs like $\{acceptable_investment_risk: low, \dots, blood_pressure: high\}$. Building of personalization profile involves discovering a meta-prompt, such as 'what are the commonalities between available health documents allowing to identify a certain medical condition X'. The object-level prompt will then be "adjust the treatment recommendation to the identified medical condition X".

5. LLM Personalization via meta-prompting

We follow Lyu et al. 2023, exploring various prompting strategies for enhancing personalized recommendation performance with large language models (LLMs) through input altering via prompts.

We rely on the following prompting strategies:

- (1) basic prompting,
- (2) personalized recommendation-driven prompting,
- (3) user commitment - driven prompting, and
- (4) recommendation-driven + user commitment - driven prompting.

Rather than utilizing LLMs as recommender models, this section investigates the application of prompting strategies to enhance input text with LLMs for personalized content recommendations.

Drawing upon the extensive language datasets on which LLMs have been fine-tuned, our objective is to unlock their potential for generating high-quality and context-aware input text, thereby improving recommendations. We explore various tailored prompting strategies for personalized content recommendation, including basic prompting, recommendation-driven prompting, engagement-guided prompting, and the combination of recommendation-driven and engagement-guided prompting. Through the utilization of these strategies, our aim is to optimize the generation of input text by LLMs, ultimately enhancing the accuracy and relevance of content recommendations.

Recommendation-driven prompting components play a crucial role in enabling LLM to focus on relevant context and align with user preferences (Lyu et al. 2023, Figure 6)

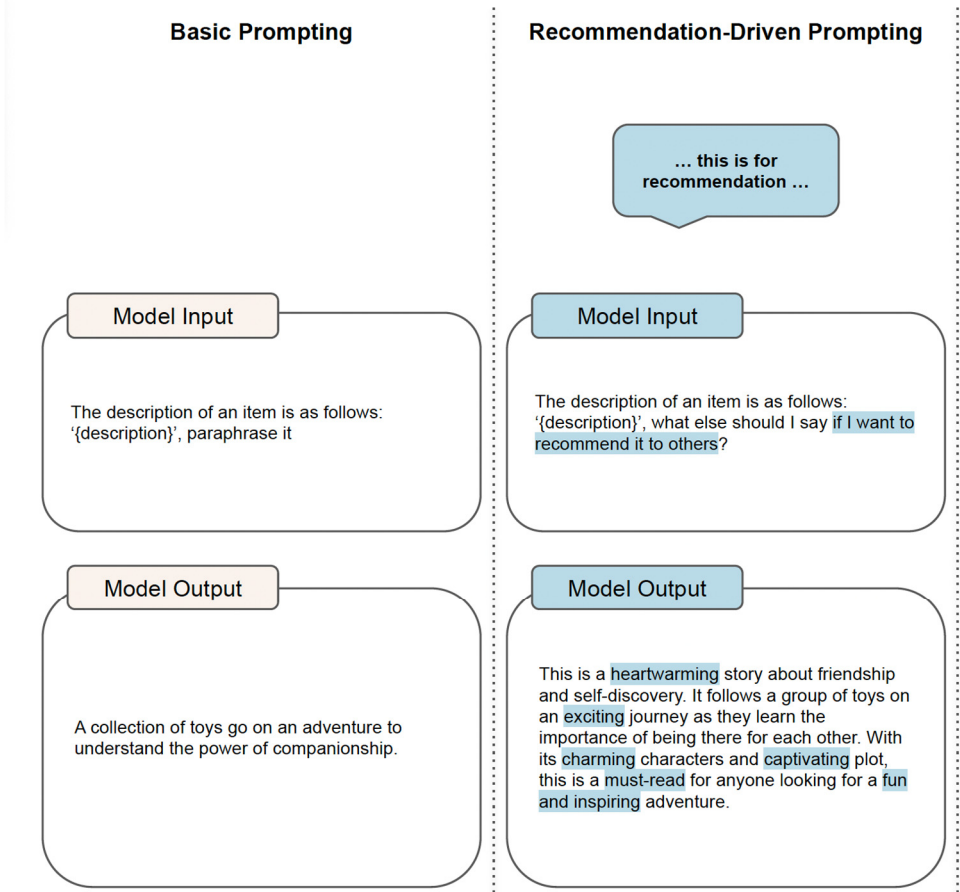


Figure 6. Prompts for recommendation.

The utilization of recommendation-driven prompting possesses several compelling characteristics, rendering it an attractive approach for crafting high-quality content descriptions:

- (1) **Enhanced Context Precision:** Explicitly stating that the generated content description is intended for content recommendation provides models with a clearer understanding of the task. This additional context enables models to align their responses more closely with the purpose of generating content descriptions for recommendation purposes.
- (2) **Guided Generation for Recommendation:** The specific instruction serves as a guiding cue for models, directing their attention towards generating content descriptions better suited for recommendation scenarios. The mention of “content recommendation” likely prompts the Large Language Model (LLM) to focus on key features, relevant details, and aspects of the content that are more instrumental in guiding users toward their preferred choices.
- (3) **Heightened Accuracy of Recommended Entities:** The instruction assists the LLM in producing content descriptions tailored to the requirements of content recommendation. This alignment with the recommendation task results in more relevant and informative descriptions, as the LLM is primed to emphasize aspects crucial for users seeking recommendations.

The user commitment prompting strategy involves utilizing user behavior, specifically user-item commitment, to formulate prompts with the aim of guiding the Large Language Model (LLM) to better encapsulate characteristics within the content description that align with user preferences. The goal is to generate more meaningful descriptions for recommendation tasks using this type of prompt.

To create the commitment-guided prompt, Lyu et al. (2023) combine the content description of the target item, denoted as d_{target} , with the content descriptions of T important neighbor items, represented as d_1, d_2, \dots, d_T . This amalgamation of information serves as the foundation for the prompt, strategically designed to leverage user engagement and preferences in generating more contextually relevant content descriptions.

The commitment-guided prompt makes use of user engagement data to identify significant neighbor items. By incorporating information from these items into the prompt, the LLM has the potential to unveil meaningful connections, similarities, or relevant aspects between the target item and its neighbors. This approach can lead to more accurate, informative, and high-quality content descriptions, ultimately enhancing the overall performance of the content recommendation system.

Personalized recommendation-driven and user commitment-driven prompting components play a crucial role in enabling large language models to focus on relevant context and align with user preferences.

Recommendation-driven and commitment-driven prompting components, together with their corresponding augmented texts are shown highlighted in Figure 7.

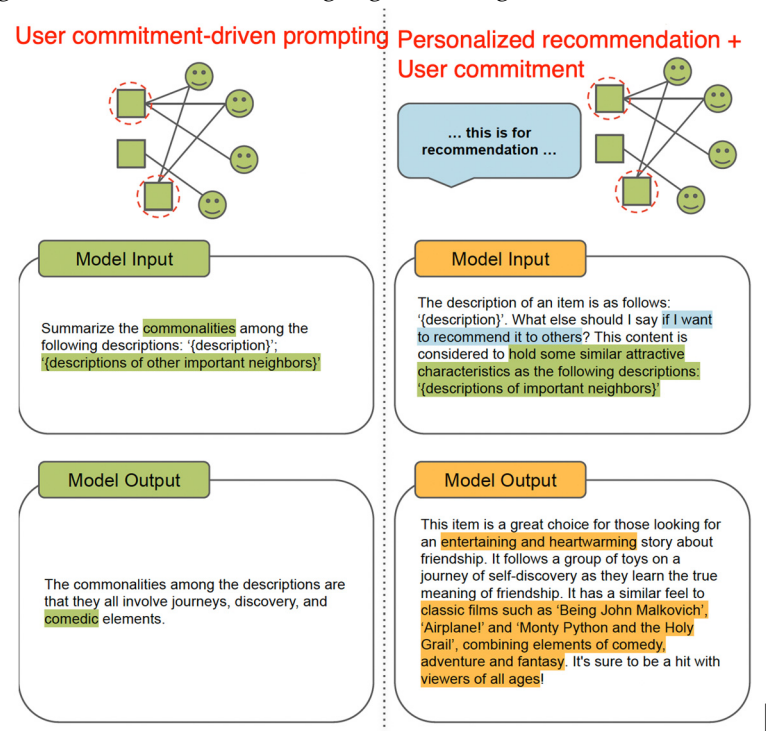


Figure 7. User commitment and personalization prompts.

Finally, both the recommendation-driven and commitment-driven prompts are used: "The description of an item is as follows: ' d_{target} '. What should I say if I want to recommend it to others?". This content is considered to hold some similar attractive characteristics as the following descriptions: ' d_1, d_2, \dots, d_T '."

There are certain limitations when employing LLM for augmenting input text in recommendation systems when constructing prompts. There is a distinction between prompts that solely instruct LLM to modify the content description and those that prompt LLM to infer additional information. In the latter case, where inference beyond the original context is required, the recommendation-driven prompting strategy may not yield the expected improved personalization results. The value of recommendation relevance of the inferred context might not be that high.

There is a need for careful consideration and evaluation of the prompts employed, particularly when instructing LLM to infer information beyond the provided context. While recommendation-driven prompting strategies prove effective for prompts that do not necessitate inference, their effectiveness may be hindered when the prompts require LLM to extrapolate information.

In addition to its superior performance in personalized content recommendation, the incorporation of commitment signals in prompt designs may have broader associated benefits. The commitment-guided prompting strategy instructs the LLM to generate commonalities among different items, resembling the concept of neighborhood aggregation in graph and concept learning (Galitsky and Kovalerchuk 2010).

A Meta-prompt for building user personalized profile

In the realm of personalization, a meta-prompt serves as an abstract request aimed at constructing a user profile from a set of documents. At its abstract level, the meta-prompt seeks to identify commonalities among documents, such as shared phrases that signify entities potentially applicable for personalization. Depending on the nature of the documents and the task at hand, this meta-prompt can be instantiated into a specific task, such as determining values for the user’s personalization profile, including common interests, tools, or locations. For instance, consider multiple software design documents and tutorials. A specific object-level prompt could extract the entity ‘software’, which can then be utilized to personalize recommendations in various domains, including job searches, training, socializing, and other activities.

For example, an instance of a meta-prompt for building a user personalization profile can be “adjust the answer about {topic} to user specialty extracted from {URL} from this profile” (Figure 8).

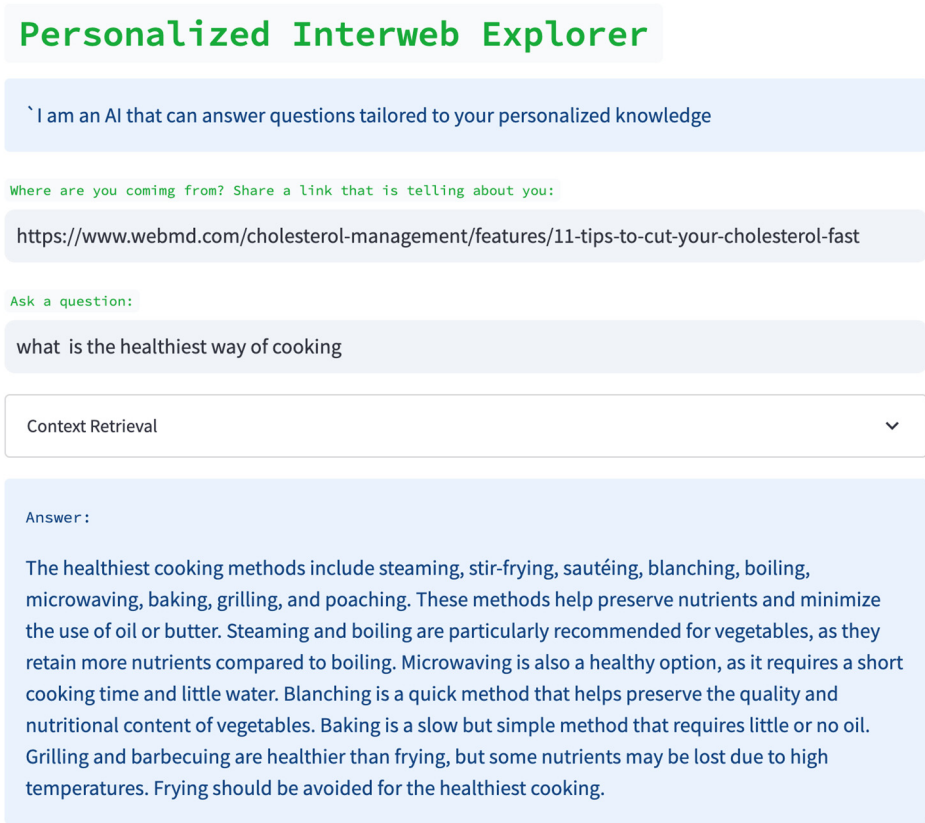


Figure 8. Applying health-related attribute “cutting cholesterol” from the user personalization profile to the answer to the question related to cooking.

We rely on meta-learning techniques to design a LLM prompt to produce a personalization prompt to obtain a suitable relevant information. Such meta-prompt is produced by generalization operation applied to available documents for the user. These documents need to be de-identified so

that they are sufficient for personalization on one hand and will maintain user privacy on the other hand.

We do not know in advance which information from the documents or web pages associated with an individual. So the prompt would sound “extract information from documents which occur in at least two distinct documents”. We assume that occurrence in just one document can be a noisy indication but occurrence in two is a reliable indication. Hence a meta-prompt is needed to produce a prompt which will be applied to a given user question. Such meta-prompt would sound like “Determine which features X in a given document indicate an interest of a user” which will yield the prompt “Adjust search results to the domain of interest X”. The former meta-prompt, once obtained, would yield what is called “object-level” or regular prompt with X instantiated. Whereas the meta-prompt need to be “discovered” or “explored” by a LLM given a collection of documents, the resultant regular prompt is used to obtain the actual recommendation.

5.1. Designing meta-prompts

The pursuit of self-improving systems is driven by the promise of creating artificial intelligence that can learn, adapt, and evolve autonomously, leading to more efficient problem-solving and decision-making. Such systems could revolutionize a wide range of fields, from medicine and climate modeling to space exploration and poetry.

Intrigued by the potential of self-improvement, (Goodman 2023) experimented with a language model chatbot that modifies its own instructions from one episode to the next, by reflecting on a dialogue with a user.

The fundamental concept behind Meta-Prompting is to prompt the agent to self-assess its performance and adjust its instructions accordingly. A viable approach involves employing reflection and introspection. The agent engages in an iterative process that initiates without instructions and progresses through the following steps (Figure 9, Greyling 2023):

- (1) Engage in conversation with a user, who may provide requests, instructions, or feedback.
- (2) At the end of an episode, generate self-criticism and a new instruction using the meta-prompt.

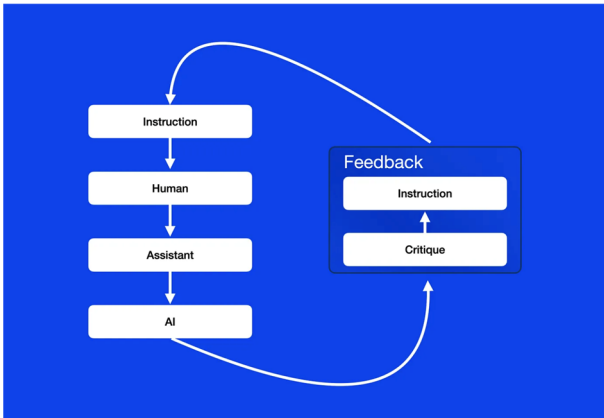


Figure 9. Introspection architecture: meta-prompting for feedback loop.

The meta-prompt for the introspection architecture looks like (Figure 10, Greyling 2023):

```

meta_template = """
Assistant has just had the below interactions with a User.
Assistant followed their "Instructions" closely.
Your job is to critique the Assistant's performance and then
revise the Instructions so that Assistant would quickly and
correctly respond in the future.

####

{chat_history}

####

Please reflect on these interactions.

You should first critique Assistant's performance. What could
Assistant have done better?
What should the Assistant remember about this user? Are there
things this user always wants? Indicate this with "Critique:
...".

```

Figure 10. Meta-prompt for introspection architecture.

5.2. Automated learning of meta-prompts

The technique of prompting represents a significant breakthrough in the realm of few-shot Natural Language Processing (NLP). Recent advancements in the field have transitioned from the use of discrete tokens referred to as "hard prompts" to the adoption of continuous "soft prompts." These soft prompts make use of trainable vectors as pseudo prompt tokens and have demonstrated enhanced performance. While the potential of these soft-prompting methods is quite promising, they are notably reliant on having an appropriate initialization to be effective. Regrettably, achieving an optimal initialization for soft prompts requires a deep comprehension of the inner mechanisms of language models and intricate design. This makes it a challenging and time-consuming neuro-symbolic undertaking that must commence from scratch for each new task.

Prompt-based methods involve the insertion of specific text, referred to as prompts, into input examples. This conversion turns the few-shot task into a (masked) language modeling problem. For example, when analyzing the sentiment of a product review, such as "I will never use it again," a prompt like "It was" can be added to the sentence, resulting in "I will never use it again. It was." This adjustment naturally encourages the LLM to generate a higher probability for "bad" than "good." This transformation effectively closes the gap between pre-training and target tasks, enhancing transferability and diminishing reliance on target task data.

The effectiveness of prompting methods is significantly influenced by prompt design (Gao et al., 2021). Early efforts involved manually crafted prompts or the automatic generation of search prompts. Seeking enhanced performance compared to manually selected prompts, Gao et al. (2021) introduces LM-BFF, a method that searches for both prompt templates and label words. To further maximize the potential of prompts, recent studies have adopted learnable vectors as prompt content, enabling the learning of optimal prompts in a continuous space, termed "soft prompts" (Li and Liang,

2021). As these methods no longer necessitate prompts to consist of real words, they greatly broaden the range of possible prompts, resulting in improved performance.

To tackle this challenge, Hou et al. (2022) introduce a comprehensive soft prompting method known as “MetaPrompting.” MetaPrompting harnesses the widely recognized model-agnostic meta-learning algorithm to autonomously uncover optimal prompt initializations, enabling swift adaptation to new prompting tasks.

The authors depart from task-specific prompt designs, instead concentrating on acquiring a task-general prompt initialization that facilitates more rapid and enhanced adaptation to novel prompting tasks.

Through extensive experimentation, it is demonstrated that MetaPrompting effectively addresses the soft prompt initialization problem, resulting in notable enhancements across three distinct datasets. This approach significantly mitigates the issues associated with soft prompt initialization, demonstrating the acquisition of general metaknowledge to counter the instability of prompt variance.

In Figure 11, x denotes the query sentence, and z is learnable pseudo tokens in soft prompts. ϕ represents all trainable parameters. Meta prompting exploits optimization-based meta-learning to find an initialization ϕ_{meta} that facilitates better and faster adaptation to new tasks.

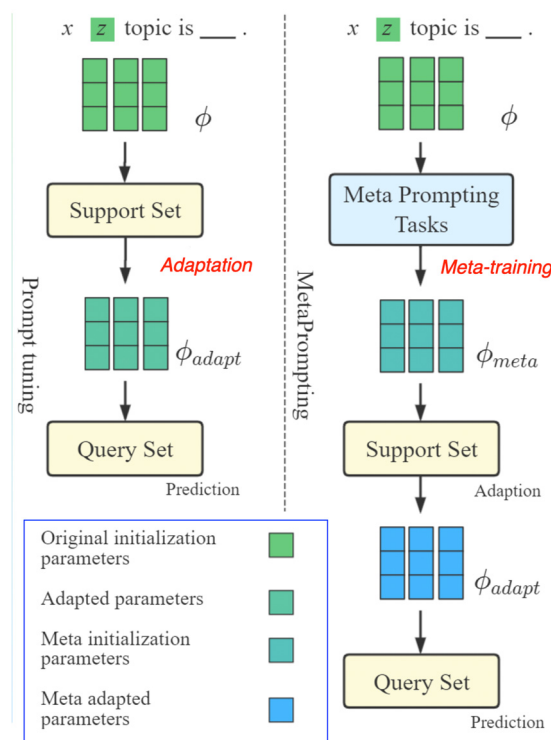


Figure 11. Comparison between conventional soft prompting method and automated learning of meta-prompts. Parameter scales are shown in bottom-left corner.

5.3. Prompting evaluation architecture

Prompting evaluation architecture is depicted in Figure 12. For each generated response, Lyu et al. 2023 first encode it and then concatenate the embeddings with the embeddings of the original content description.

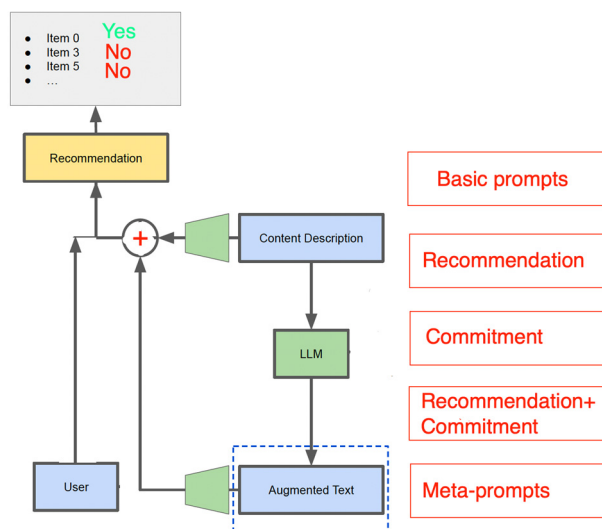


Figure 12. Prompting evaluation architecture.

It turns out that the distinctive words generated with the commitment-guided prompting strategy are fine-grained descriptive words related to movie-goer interests (Figure 13)

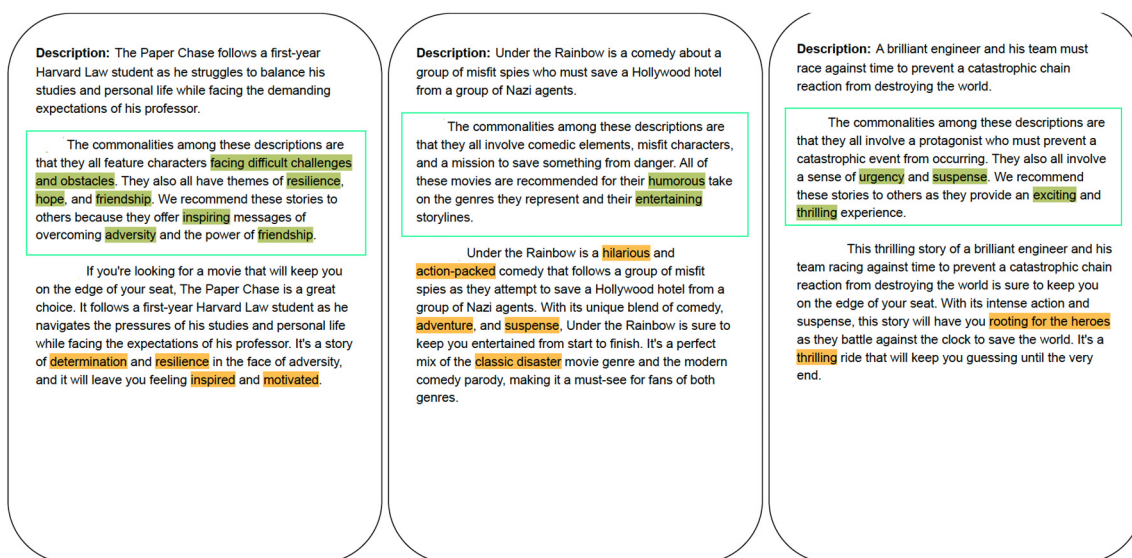


Figure 13. Commitment prompt – based results are in green frame.

6. LLM Personalization via abduction

Logical abduction is the logical formalization of abductive reasoning. Given observed facts and background knowledge expressed as first-order logical clauses, logical abduction can abduce ground hypotheses as possible explanations to the observed facts. This happening in the form of LLM prompts produced by abduction. A declarative framework in Logic Programming that formalizes this process is Abductive Logic Programming (ALP, Denecker, and Kakas 2002).

Abductive learning involves an ML model’s task of translating sub-symbolic data into basic logical facts, while a logical model is employed for reasoning about these interpreted facts using a foundation of first-order logical background knowledge to yield the ultimate output. The main challenge stems from the inherent difficulty of simultaneously training both the sub-symbolic and symbolic models in a cohesive manner. Experimental results of Dai et al. (2019) show that ABL

generalize better than state-of-the-art deep learning models and can leverage learning and reasoning in a mutually beneficial way (Figure 14).

Being more specific about integration of an abstract M and reasoning:

- (1) ML model does not have any ground-truth of the primitive logic facts for training;
- (2) without accurate primitive logic facts, the reasoning model can hardly deduce the correct output or learn the right logical theory.

We consider un-personalized LLM is a sub-symbolic ML without a logical theory. A logical theory focuses on personalization, implemented as re-labeling of data “labelled” within LLM. What LLM outputs can be considered as classification of each sentence or phrase into ‘in’ and ‘out’. What LLM gives is ‘in’ class and what it does not give is ‘out’ class. Default LLM classification has significant inaccuracies: sentence “see the doctor” should be ‘out’ for the doctor user, but it is ‘in’.

When provided with a training sample linked to a final output, logical abduction is capable of making educated inferences regarding the missing information. This might involve deducing logic clauses that complement the existing background knowledge, thereby creating a coherent proof from the sample to its final output. Subsequently, the inferred basic facts and logical clauses are harnessed for training the machine learning model and preserved as symbolic knowledge, respectively. The process involves optimizing consistency to maximize the alignment between the conjectures and the background knowledge. To tackle this intricately complex challenge, we reframe it as a task that involves searching for a function that can make educated guesses regarding potentially erroneous primitive facts.

A baseline ML / LLM is used for learning the perception model LLM. Given an input instance x , LLM can predict the pseudo-labels $LLM(x)$ as groundings of possible primitive concepts in x . These concepts can potentially be used in recommendation. When the pseudo-labels contain mistakes, the LLM needs to be fine-tuned, where the labels are the revised pseudo-labels $r(x)$ returned from logical abduction.

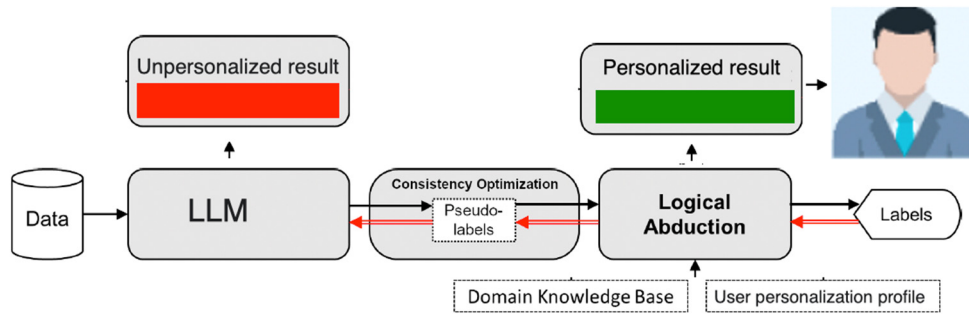


Figure 14. Personalized and Unpersonalized results with LLM and Logical Abduction.

The task of abductive learning can be formalized as follows. The input of abductive learning consists of a set of labelled training data $D = \{ \langle x_i, y_i \rangle; \dots; \langle x_n, y_n \rangle \}$ about a target concept C and a domain knowledge base B , where $x_i \in X$ is the input data, $y_i \in \{0, 1\}$, is the label for x_i of target concept C , and B is a set of first-order logical clauses. The target concept C is defined with unknown relationships amongst a set of primitive concepts symbols $P = \{p_1; \dots; p_r\}$ in the domain, where each p_i is a defined symbol in B . The target of abductive learning is to output a hypothesis model $H = p \cup \Delta_c$, in which:

- (1) $p : X \rightarrow P$ is a mapping from the feature space to primitive symbols, i.e., it is a perception model formulated as a conventional machine learning model;
- (2) Δ_c is a set of first-order logical clauses that define the target concept C with B , which is called knowledge model.

The hypothesis model should satisfy:

$$\forall \langle x; y \rangle \in D (B \cup \Delta_c \cup P(x) \models y) \quad (1)$$

‘ \models ’ is a logical entailment.

Target concept C is viewed as a purpose of the prompt: once it is run, we expect to obtain C .

The perception model LLM is not personalized. LLM model and the knowledge model Δ_c are mutually dependent:

1) To learn Δ_c , the perception results $p(x)$ (the set of groundings of the primitive concepts in x) is required;

2) To obtain p , we need to get the ground truth labels $p(x)$ for training, which can only be logically derived from $B \cup \Delta_c$ and y .

In not personalized LLM, the perceived primitive symbols $p(x)$ is most likely incorrect; therefore, should be called pseudo-groundings or pseudo-labels. As a consequence, the inference of Δ based on Eq. 1 would be inconsistent.

When the personalization knowledge model Δ_c is inaccurate, the logically derived pseudo-labels $LLM(x)$ might also be wrong, which harms the fine-tuning of LLM. In either way, they will interrupt the learning process.

Formally, an abductive logic program can be defined as follows: An abductive logic program is a triplet $\langle B; A; IC \rangle$, where B is background knowledge, A is a set of predicates obtained by abduction, and IC is the integrity constraints. Given some observed facts O , the program outputs a set of ground abduction results of A , such that:

$$\begin{aligned} B \cup \Delta & \models O, \\ B \cup \Delta & \models IC, \\ B \cup \Delta & \text{ is consistent.} \end{aligned}$$

Δ serves as a hypothesis that explains how an observation O could hold according to the background knowledge B and the constraint IC . Δ can be obtained in NL form if we request LLM to derive O from B and the constraint IC via chain of thoughts.

Abductive reasoning takes observed facts about data instances and also takes the combined LLM + reasoning about prompt prediction model $H = p \cup \Delta_c$. Given a fixed Δ_c , the system produces explanation by abduction $p(X)$ according to B and Y . However, when the learning model p has been obtained, the system builds an improved explanation Δ_c according to $B \cup Y \cup P(x)$.

$p(X) = \bigcup_{i=1}^n \{p(x_i)\}$ for coarse labels for all instances $X = \bigcup_{i=1}^n \{x_i\}$ and $Y = \bigcup_{i=1}^n \{y_i\}$ are the improved labels (correctly personalized results) for X .

An LLM can also be requested to perform abduction independently (Figure 15).



A patient is saying to the doctor that he cannot drink sufficient amount of water to properly maintain his kidney. What can prevent a patient from drinking sufficient amount of water?



Several factors can prevent a patient from drinking a sufficient amount of water. It's essential to address these factors, as adequate hydration is crucial for overall health and proper kidney function. Some common reasons for inadequate water intake include:

1. **Medical Conditions:** Some medical conditions, such as kidney disease, diabetes, and certain heart conditions, can affect a person's thirst response or the ability of the kidneys to concentrate urine. This can lead to reduced water intake.
2. **Medications:** Certain medications, including diuretics (water pills), can increase urine



Correct answer is, the patient has problem urinating and therefore avoids drinking extra fluids



Thank you for providing the specific context. If a patient has difficulty urinating, it can indeed be a significant factor preventing them from drinking sufficient fluids. This reluctance to consume additional fluids may be due to discomfort, concerns about frequent urination, or the fear of exacerbating the issue. In such cases, it's important for the patient to discuss their urination problems with a healthcare provider to identify the underlying cause and receive appropriate treatment or management. Addressing the urination issue can then facilitate a more suitable fluid intake plan to maintain proper kidney function and overall health.

Figure 15. Encouraging LLM to perform abduction.

7. Evaluation

We draw the comparison between our personalization algorithm and competitive LLM personalization approach in the shopping domain, and then apply the optimized algorithm in the health domain.

Traditional personalization systems commonly use task-specific metrics like ranking-oriented metrics, NDCG, AUC, and recall for assessing model performance. However, the integration of LLMs into recommender systems brings about significant changes in evaluation tools. Conventional metrics may not adequately capture the novel capabilities and altered recommendation generation processes, necessitating the development of new evaluation tools. In large language model-powered systems, evaluating user preferences is crucial, demanding a user-centric approach. Metrics such as user satisfaction, engagement, and overall experience become pivotal considerations, with user satisfaction surveys and feedback questionnaires serving as valuable options. It is also important to consider factors like diversity, novelty, serendipity, user retention rates, interpretability, and fairness of personalized recommendations. We will do that for a health recommendation domain.

The dataset is randomly split into training, validation, and testing sets using 4:2:1 ratio. Negative training dataset is created using random negative sampling. We follow personalization strategy of Wei et al. (2019) for evaluation: <https://github.com/weiyinwei/MMGCN>. For the validation and testing sets, we pair each observed user-item interaction with 1,000 items that the user has not previously observed. We use the following metrics: *Precision@K*, *Recall@K*, and *NDCG@K*.

We adopt the leave-one-out approach to evaluate the performance of each method, a commonly employed technique in many related studies. In this approach, for each user, we designate the last interacted item as the test data, the item just before the last interaction as validation data, and the remaining items for training. The evaluation of each method is conducted on the complete item set without any form of sampling.

For the validation and testing sets, we pair each observed user-item interaction with 1,000 items that the user has not previously interacted with. It is crucial to emphasize that there is no overlap between the negative samples in the training set and the unobserved user-item pairs in the validation and testing sets, ensuring the independence of the evaluation data.

To assess the performance of top-K recommendations, we utilize widely-used metrics such as *Precision@K*, *Recall@K*, and *NDCG@K*. In our case, we set $K = 10$, indicating that we consider the top 10 recommendations. We present the average scores across five different splits of the testing sets, providing a comprehensive evaluation of the recommendation performance.

We compare Multi-layered perceptron (MLP, Zhou et al. 2022), PALR (Chen et al. 2023), LLM-Rec (Lyu et al. 2023) and FMLP-Rec (Zhou et al. 2022) in the four datasets of sequential item-based recommendations: Movielens, Recipe, Yelp and Sports. Movielens dataset includes 25 million ratings and one million tag applications applied to 62,000 movies by 162,000 users (<https://grouplens.org/datasets/movielens/>). Yelp is a dataset for business recommendation.

The baseline model employs a Multilayered Perceptron (MLP). The augmented text embeddings and the original content description embeddings are combined through concatenation, and the resulting vector is fed into a two-layer MLP. The first MLP layer’s output dimension, along with the input/output dimensions of the second MLP layer, is set to 128. The first MLP layer incorporates a ReLU activation function and a dropout layer. Following this, the dot product of the latent embeddings of the user and the item is computed, and the resulting value is then processed through a Sigmoid function. This Sigmoid function transforms the dot product into a final relevance score, representing the relationship between the user and the item.

TABLE 1

	Movielens	Recipe	Yelp	Sports
--	-----------	--------	------	--------

	Precisio n@10	Recal l@10	NDC G@1 0	Precisio n@10	Recal l@10	NDC G@1 0	HR@ 5	NDC G@1 0	HR @5	NDC G@1 0
MLP	0.291	0.244	0.363	0.032	0.068	0.058				
LLM- Rec	0.315	0.277	0.395	0.039	0.084	0.070				
FMLP- Rec							0.273	0.502	0.380	0.316
PALR		0.072 *	0.044 *							
Ours baseline (LLM finetuni ng)	0.307	0.210	0.320	0.029	0.065	0.043	0.28	0.460	0.35 5	0.261
Ours with abductio n	0.321	0.254	0.390	0.041	0.062	0.075	0.291	0.489	0.37 1	0.332
Ours with meta prompts	0.304	0.231	0.403	0.0361	0.072	0.068	0.273	0.465	0.37 6	0.309

We show three types of performances of our system on a popular datasets (Table 1):

- (1) Ours baseline with LLM finetuning. Its implementation in health domain is available at (Galitsky-github 2024);
- (2) Baseline extended with abduction;
- (3) Baseline extended with meta prompts.

One can observe that our baseline performance is below all the competitive systems. We were not able to match FMLP-Rec on Sports database in all three our architectures, but improved its performance in Yelp domain. The best performance is achieved using abduction, and the second best – using meta prompts, in comparison with LLM-Rec,

FMLP-Rec and PALR. Abduction is the winner in the Movielens dataset, outside of NDCG measure, and also in the Recipe dataset, outside of Recall@10 measure. We observe that abduction and meta prompts gives a substantial boost to personalization performance. Hence these are promising directions to improve overall recommendation accuracy as LLMs are being further developed.

7.1. Sample sessions

Figures 16 and 17 show personalization results in software and health domains respectively.

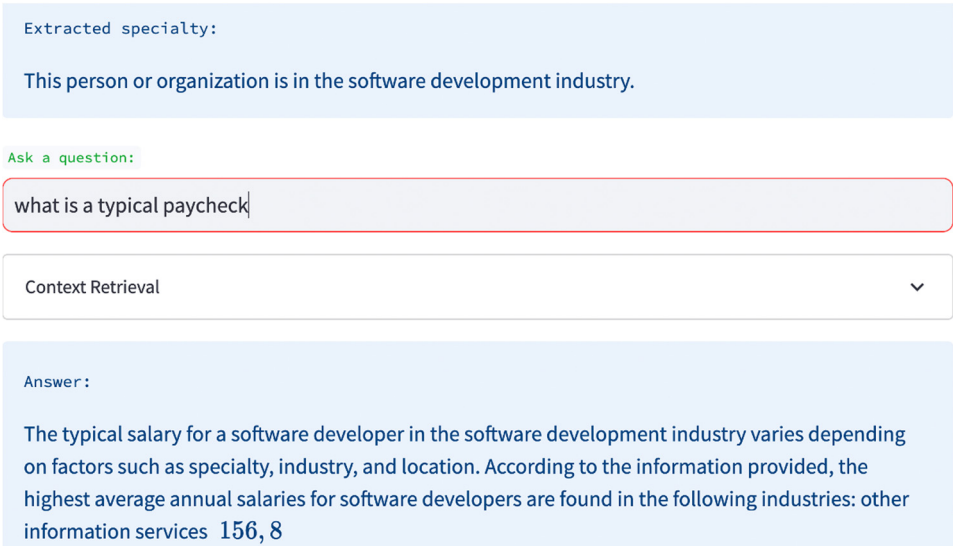


Figure 16. Personalization for github user.

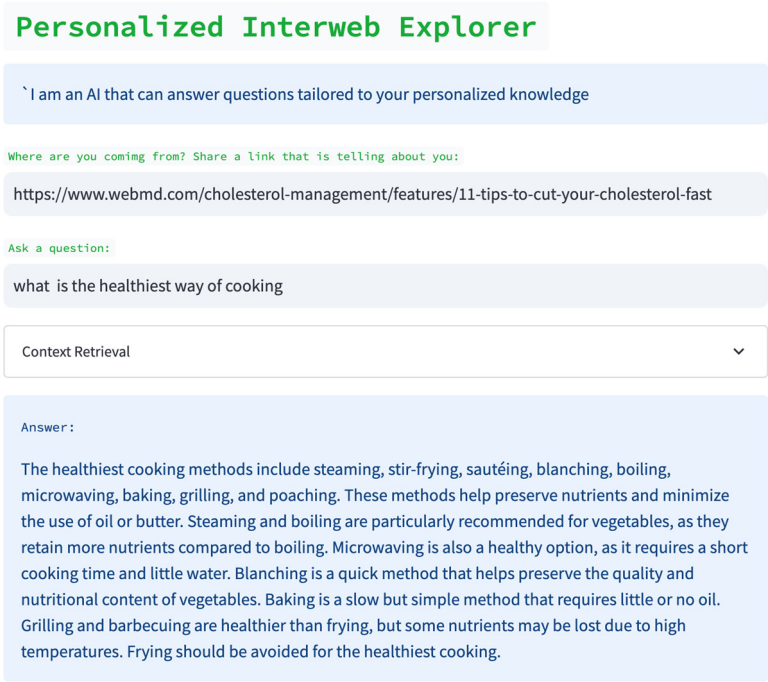


Figure 17. Personalization for cooking method for a person interested in cutting cholesterol level.

8. Personalized Recommendation in Health

In the contemporary digital landscape, healthcare stands as a pivotal domain within the medical field. The demands of the healthcare system involve the analysis of extensive patient data to extract valuable insights and facilitate disease prediction. An intelligent healthcare system is crucial for forecasting health conditions by scrutinizing a patient’s lifestyle, physical health records, and social activities. The emergence of Health Recommender Systems (HRS) underscores their growing significance as a vital platform for healthcare services. Consequently, health intelligent systems have evolved into indispensable tools for guiding decision-making processes within the healthcare sector. Within this context, health intelligent systems have evolved into indispensable tools for decision-making processes in the healthcare sector. Their primary objective is to ensure the timely availability of critical information while upholding standards of information quality, trustworthiness,

authentication, and privacy. As individuals increasingly turn to social networks for insights into their health, the health recommender system becomes instrumental in generating recommendations related to diagnoses, health insurance, treatment methods based on clinical pathways, and alternative medicines, all guided by the patient’s unique health profile.

The recommender system is based on predictive analytics which predicts and recommends appropriate items to the patients. This system can be applied to specific applications. Healthcare analytics is a major area in big data analytics which can be incorporated into the recommender system. An HRS is a decision-making system which recommends proper healthcare information to both health professionals and patients as end patients. By using this system, patients are recommended the proper treatment of disease for avoiding a health risk, and health professionals benefit from the retrieval of valuable information for clinical guidelines along with delivery of high-quality health remedies for patients. This HRS should be trustworthy and reliable so that end patients can use this system to their benefits (Figure 18).

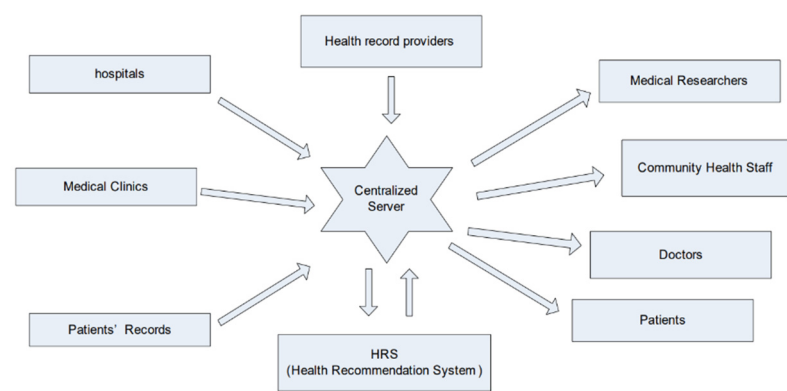


Figure 18. A centralized HRS architecture.

The rapid advancements in data mining and analytics have given rise to the widespread application of big data analytics across various domains. Among these, the healthcare sector has emerged as a particularly promising area, where big data analytics has garnered its own well-deserved recognition and acclaim. In healthcare data, there are three primary distinguishing characteristics associated with big data:

- (1) volume (referring to the sheer amount of data generated by organizations or individuals, originating from both internal and external sources), velocity (indicating the rapid rate at which data is created, captured, and shared),
- (2) variety (encompassing data from diverse sources in varying formats), and
- (3) veracity (concerning the accuracy and consistency of obtained data), all of which play a crucial role in the healthcare domain.

Given the overwhelming volume of unprocessed data and information overload, recommender systems have gained popularity due to their ability to filter and manage large datasets effectively. There arises a pressing need for a new HRS capable of enhancing the healthcare system’s efficiency and handling the healthcare needs of patients with various ailments simultaneously (Sahoo et al. 2019)

The HRS in Figure 19 encompasses several distinct phases involved in the process of recommending specific healthcare items in pre-LLM times. These phases include the training phase, patient profile processing phase, sentiment analysis (Galitsky and McKenna 2017) phase, privacy preservation phase, and the recommender phase. To initiate this process, we begin by gathering a healthcare dataset and subject it to feature selection and classification techniques. A critical component of the HRS is the collection and preparation of a Profile Health Record (PHR) and a patient database. The PHR serves as a vital input for the recommender engine, aiding in the prediction and recommendation of health solutions to patients. Sahoo et al. (2019)’s system extracts relevant information from the patient database, which is linked to the PHR, for the purpose of feature

selection. Subsequently, a classification algorithm is applied to classify and store this knowledge in a repository.

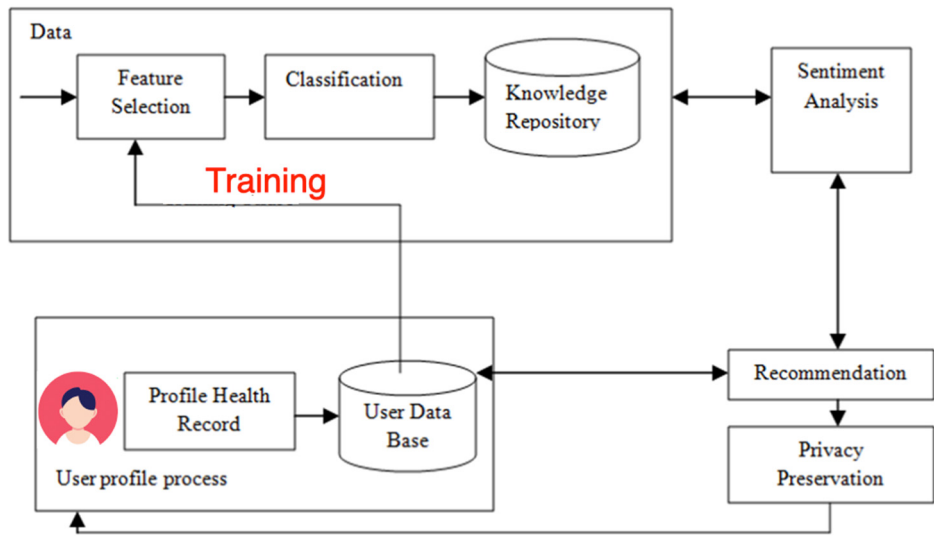


Figure 19. Pre – LLM HRS architecture. Under LLM, feature selection is manual.

To seamlessly integrate into any health-related information system, it is crucial to consider the system context of a an HRS). A profile-based HRS component serves as an extension to an existing PHR system. The data entries within the PHR database constitute the medical history of the PHR owner. With access to medical facts, the HRS calculates a set of potentially relevant items of interest for a target user, such as the PHR owner or an authorized health professional. These items are sourced from reliable health knowledge repositories and may be presented while the user is reviewing the PHR online.

The recommender phase internally consists of three sub-phases: the collection phase, the learning phase, and the recommender phase. Following the application of these phases using the patient database, appropriate treatments are suggested to patients, and health professionals are directed to valuable clinical guidelines and high-quality healthcare treatments. The sentiment analysis phase of the HRS gathers patient opinions to support informed healthcare decisions, offering insights into the perspectives of end users regarding specific themes. The privacy preservation phase ensures the security and confidentiality of the HRS, safeguarding valuable information from unauthorized alterations.

Figure 20 depicts an integration of HRS into the overall health information system. An HRS is expected to facilitate adequate and proper communication. Communication is bidirectional (to the patient and to the recommender). Patients should be able to express in an undisturbed and hassle-free manner to doctors so that doctors can interpret the symptoms of patients and can give recommenders for a particular disease to patients. The visualization of data should address the purpose of the recommender system and able to understand the patients, doctors and their intentions. There should be a proper visualization tool in a recommender that fosters the patient’s willingness to explore options and helps to explain individual recommendations. Since individual differences might play a vital role in the health sector, it is crucial to intensify research in this field

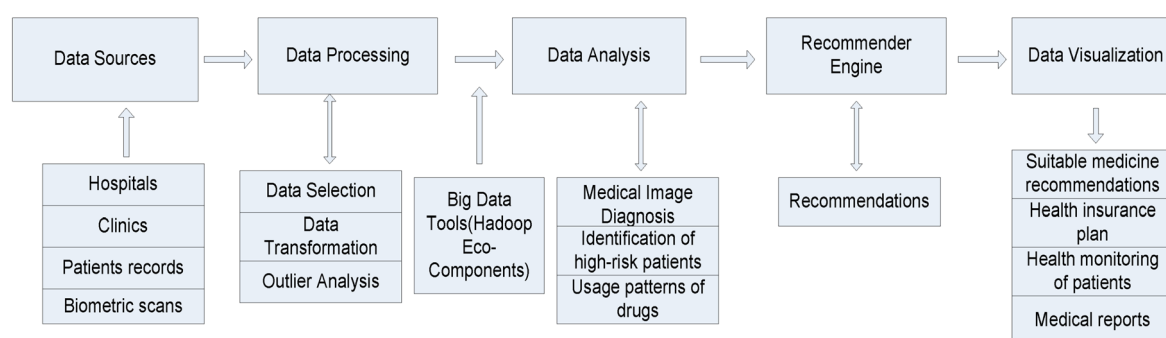


Figure 20. Recommendation environment.

8.1. Recommendation requirements

HRS should be capable of dealing with imprecise terms like ‘Hepatitis’ - ‘chronic viral Hepatitis’, colloquial terms like ‘menses’ – ‘menstruation’, and misspellings. HRS must tackle expert vocabulary primarily used by physicians and other health professionals (e.g., ‘auscultation’, ‘palpation’).

HRS also needs to detect whether clinical conditions mentioned in clinical reports are negated. In particular, medical facts (i.e., terms) occur in conjunction with abbreviations frequently used by physicians. In this context, a negation detection algorithm must cope with various negation patterns, for instance:

‘Patient’s second ECG and subsequent ECGs show no signs of abnormal cardiac activity’ ⇒ exclude negative terms ‘abnormal cardiac activity’ from the recommendation seed. Another example:

‘immune-mediated condition that affects the retina in the absence of retinal damage’ ⇒ exclude ‘retinal damage’.

Ensuring the privacy of user data in PHR systems is paramount and must be maintained under all circumstances. Even administrators of PHR systems should not have any access to or understanding of these data. HRS should possess the capability to seamlessly adapt to a new system localization. This involves allowing users to change their language preferences without necessitating manual adjustments to underlying language resources like text corpora or localized ontologies.

Data entries of medical records are frequently stored as unstructured plain text. This creates further difficulties for IR term matching approaches. A HRS must also recognize expert vocabulary (i.e., common medical abbreviations) and classification system codes primarily used by physicians and other health professionals, such as:

I25.1 - Atherosclerotic heart disease of native coronary artery:

I25.1XX - Atherosclerotic heart disease of native coronary artery with angina pectoris

I25.1YY - Atherosclerotic heart disease of native coronary artery with a documented episode of care

Such difficulties can result in less specific recommendations when integrating classic IR approaches into electronic/personal health record systems

Another HRS imperative emerges from the categorization of health information artifacts concerning their accessibility to laypersons. Depending on a patient’s background knowledge and their proficiency in comprehending expert documents (essentially, the user’s expert level), a Health Recommender System (HRS) should possess the capability to pre-filter medical information artifacts that might prove excessively complex for the intended reader. This pre-filtering not only narrows down the pool of recommendable candidate documents but also enhances computational efficiency by reducing the need for extensive processing.

8.2. Sources and Features of Recommendation

As to the sources of recommendation information, they are as follows:

- (1) Credible and authoritative websites, guidelines, and books: and websites approved by healthcare professional societies such as “American Urology Society” , guidelines published by professional associations such as the European Society of Sports Medicine” guidelines.
- (2) Domain experts: the information source is created by researchers in medical field or clinicians
- (3) Similar patients: recommended items extracted from patient databases (e.g., a database of smoking students who had succeeded in managing smoking), social networks sites by crowdsourcing, or patients with similar health problems.
- (4) Government database such as United States Department of Agriculture food composition guidelines, dietary information from a food nutrition database of the Food and Drug Administration, Nutrition Analysis, and other government health agencies or public health organizations.
- (5) A range of other information resources including online health community, online registered doctors, psychotherapy approaches.

Some recommendation features are enumerated in Table 2 (Cai et al. 2022)

Table 2. User attributes and features employed in personalization.

User Attribute	Features
Demographics	name, gender, age, body mass index, location, average charge, ethnicity, education level, marital status, blood type, weight, height, hip, waist, occupation, address, and income
Health status	symptoms, complications, treatments, health checks, diagnosis, diseases, severity, physiological, capability, depression, sleep, and pregnancy
Laboratory test	blood glucose, total cholesterol, triglyceride, high/low density lipoprotein, uric acid, blood pressure, hypertrophy, carotid artery plaque, carotid femoral pulse wave velocity, ankle brachial index, and heart rate
Disease history	family history, allergy, and smoking or alcohol history
Medication	routine medicine that patient took
Food intake	food types, food size, food names and ingredients, and dietary record
Physical activity	activity level (high, moderate, low), type of activity, duration, past activity, intensity, and frequency
Explicit data (information preference provided by the user)	studies used the direct input preference from patients to generate recommendations OR utilized user ratings
Implicit data	generated from users’ interactions with the system (e.g., navigation or interaction logs)

Generally, user attributes in the included studies were distributed across six categories: demographics, medical condition, food intake, physical activity, explicit data, and implicit data (Cai et al. 2022, Figure 21)

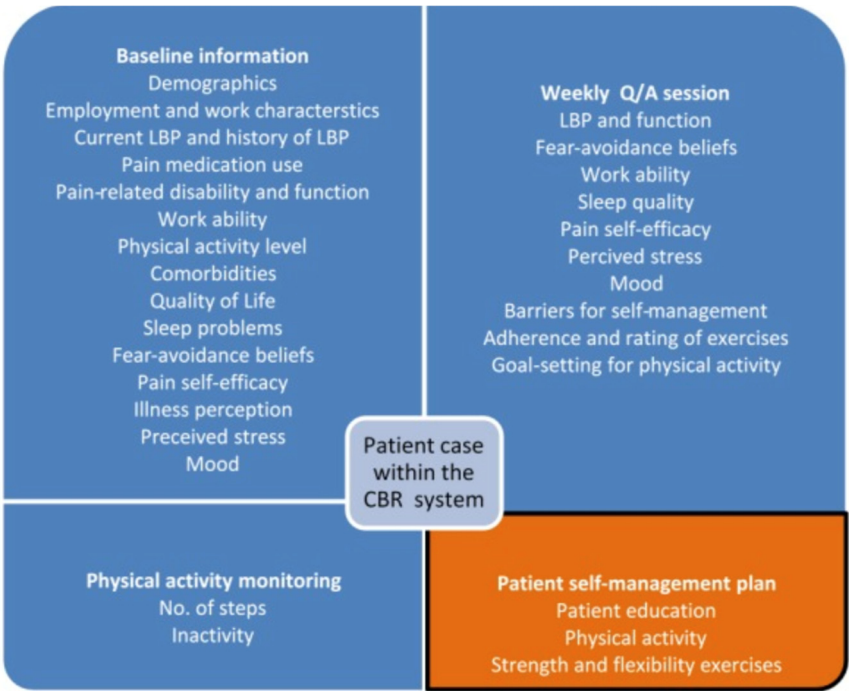


Figure 21. Components of a case description within the HRS system containing the patient characteristics and the compiled advice. Only a relevant sub-set of the weekly questions will be asked in each session. LBP: low back pain; Q/A: question/answer (Mork and Bach 2019).

HRS offers two primary categories of personalized recommendations:

- (1) Personalized Disease-Related Information or Patient Educational Material: This aligns with the Pew Internet and American Life Project’s findings. Supplying patients with disease-specific information is integral to healthcare services, providing insights into symptoms, diagnostic tests, treatment options, side effects, and the skills needed for self-management and informed decision-making.
- (2) Personalized Dietary Information: A healthy diet is crucial for preventing noncommunicable diseases like diabetes, cardiovascular diseases, cancer, and other conditions associated with obesity. HRS can propose food options tailored to individual health goals, aiding users in cultivating healthy eating behaviors through actionable recommendations.

8.3. HRS Architectures

We first look at traditional and then proceed to LLM-based personalized recommendation architectures

A pre-LLM system structure and processing workflow of an IR-based baseline HRS is shown in Figure 22 (Wiesner and Pfeifer 2014). A PHR system feeds in data elements via querying. The process yields a set of recommendable items which are highly relevant to an electronic record.

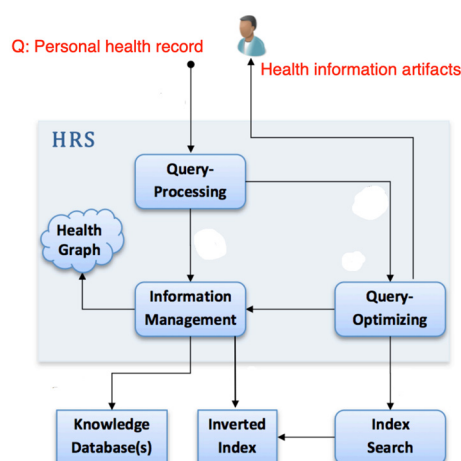


Figure 22. Traditional IR-based HRS.

Lewis et al. (2022) investigate the viability of incorporating collaborative filtering-based algorithms within health recommender systems (HRS) to suggest mental health therapy tasks. These algorithms, known for their effectiveness across diverse domains and often achieving top-tier accuracy, are less commonly discussed in contemporary digital mental health HRS literature. Instead, the focus tends to be on methods such as content-based and contextual bandit paradigms.

Results from offline assessments consistently reveal the superiority of these collaborative filtering algorithms over simpler models, including random or user-item baseline models, particularly in predicting ratings for novel therapy tasks tailored to specific clients. Collaborative filtering algorithms function by leveraging item ratings from similar users to predict rating values for a given user. This finding emphasizes the significance of incorporating therapy task ratings from a collective client group when tailoring recommendations for an individual client. For example, a collaborative filtering recommender system might propose a recommendation like, “Clients with preferences akin to yours have reported significant mood improvement with this previously untried task. Would you like to give it a try now?”

Given their accuracy advantages, along with other strengths compared to alternative HRS paradigms, such as providing more diverse recommendations compared to content-based models, Lewis et al. (2022) advocate for the inclusion of collaborative filtering algorithms in the toolkit for digital mental health recommender systems. These algorithms should be considered alongside alternative HRS approaches that have been assessed in different contexts. Furthermore, it is noteworthy that incorporating contextual information (e.g., time of day and day of the week) and item attributes (e.g., effort level and category) into collaborative filtering algorithms has been found to enhance their accuracy. This suggests that supplementary data can be valuable in assessing whether a client will find a particular task enjoyable at a specific moment.

Top of Form

Bottom of Form

The workflow of the implemented recommendation system requires the diagnose encoded by International Classification of Diseases (ICDs), age, sex to predict the most probable Therapy Keys (TKs) per patient as parameters; and the following steps ensure (Ochoa et al. 2021):

- (1) Synthesize the patient’s information and store the result with the relevant patient parameters: age, gender, patient Identifications (ID), ICDs.
- (2) Cluster ICDs and TKs. This step is required to reduce the dimensionality of both parameters (high number of items) and perform predictions of TKs group number depending on patient parameters, including ICDs groups.
- (3) Train (deep learning model), validate, and export model to medical/hospital documentation and information system.
- (4) Introduce a user interface for the recommender system, using the trained deep learning model based on the medical information system’s data to recommend the treatment keys. As part of

the combination of medical/hospital documentation and information recommender system, this is the process where the physician accepts or discards the recommended TK.

8.4. Neuro-symbolic recommender

Having considered conventional and deep learning-based recommender architectures in health, we now proceed to a hybrid, neuro-symbolic technique. The workflow of the neuro-symbolic recommender is shown in Figure 23.

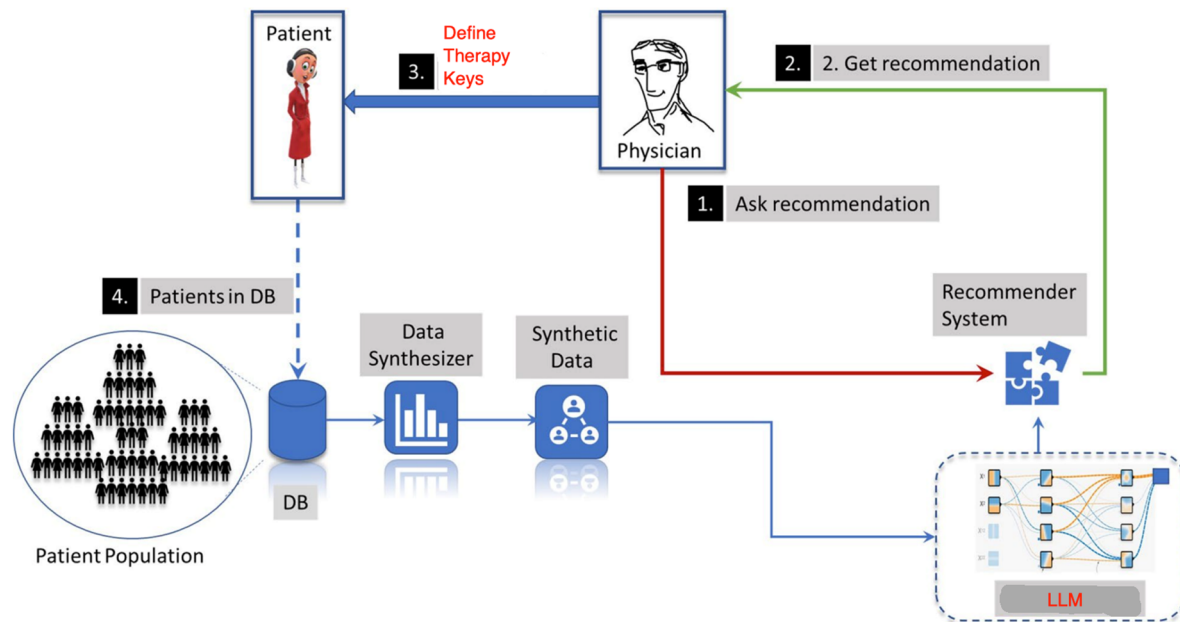


Figure 23. LLM within HRS framework.

Its usage includes the following steps:

- (1) The physician asks for a recommendation of the most frequently used TKs using the recommender system.
- (2) The physician gets a recommendation.
- (3) The physician selects the appropriate recommendations and does the treatment based on his own decision.
- (4) The new updated information about selected TKs is stored in the database (DB).

The HRS is based on the fuzzy classification system, such that the ICD groups, patients' age, and sex are used as input parameters to predict the treatment strategy (Figure 24). A DNN method based on a multilayer neural net composed of multiple hidden layers, where every neuron in layer i is fully connected to every other neuron in layer $i + 1$.

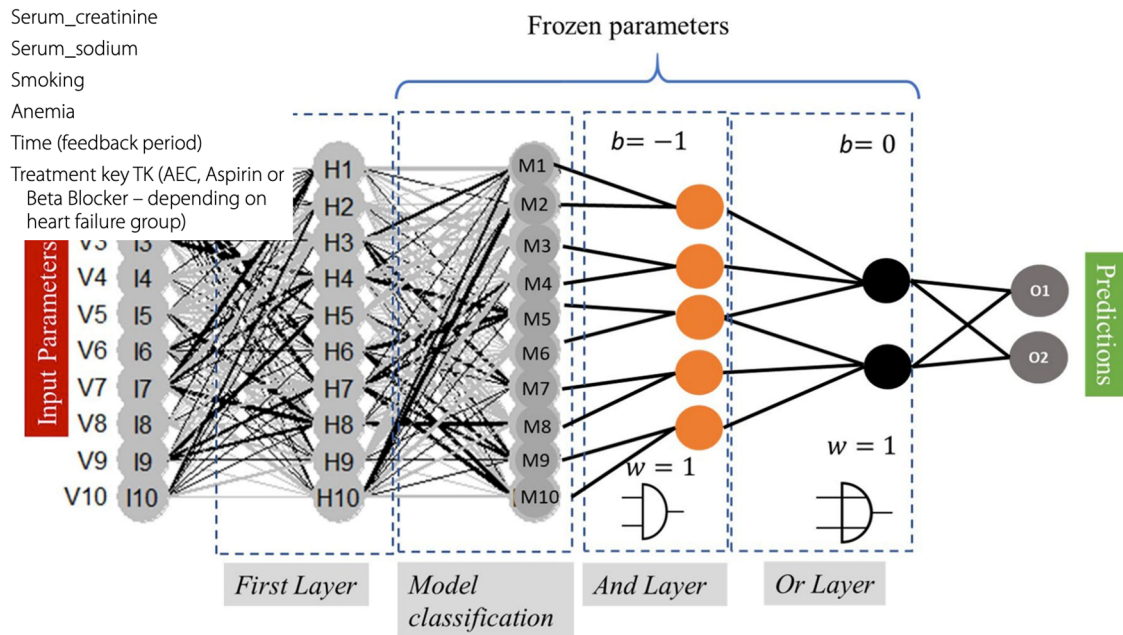


Figure 24. Neuro-symbolic architecture of HRS.

The data contained in electronic health records (EHRs) offer the potential to discover relevant patterns that aim to relate diseases and therapies, and thus discover patterns that could help identify empirical medical guidelines that reflect best practices in the healthcare system. Based on this pattern identification, it is then possible to implement recommendation systems based on the idea that a higher volume of procedures is associated with high-quality models (Ochoa and Mustafa 2021).

Ochoa et al. (2021) implement two AND layers (conjunctions), followed by an OR layer (disjunction), to logically evaluate the nodes M_i , which is a process modeling human reasoning in the decision process. The last layer emulates fuzzy logical operations, based on logical gates, modeled by perceptron with fixed weights and biases, activated by so-called squashing activation functions. These squashing functions approximate the cutting function in the nilpotent logical operators

$$S_{\beta}(x) = \frac{1}{\beta} \ln \left(\frac{1 + e^{\beta \cdot x}}{1 + e^{\beta \cdot (x-1)}} \right),$$

where β is a real nonzero value that needs to be adjusted to let the model be convergence. Thus, the hidden layers can model a threshold-based nilpotent operator [9, 10]: a conjunction, a disjunction, or even an aggregative operator. This means that the weights of the first layer are to be learned, while the hidden layers of the pre-designed neural block, worked as logical operators with frozen weights and biases.

Ochoa et al. (2021) trained two different models for the final model consumption (Figure 25), one with the whole database (Model 1) and another with a database that selects only positive outcomes (Model 2). The goal of this training method is to make predictions based on the positive outcomes and then evaluate the confidence of the prediction.

The training method for the Siamese recommender system is as follows: Two different models are trained based on the whole data set (Model 1) and a dataset consisting only of positive outcomes (Model 2). After that, both models are used to make different predictions. If there is a matching in the predictions, then Prediction 1 is used as the standard with high confidence; otherwise, predictions from Model 2 are provided but have low confidence

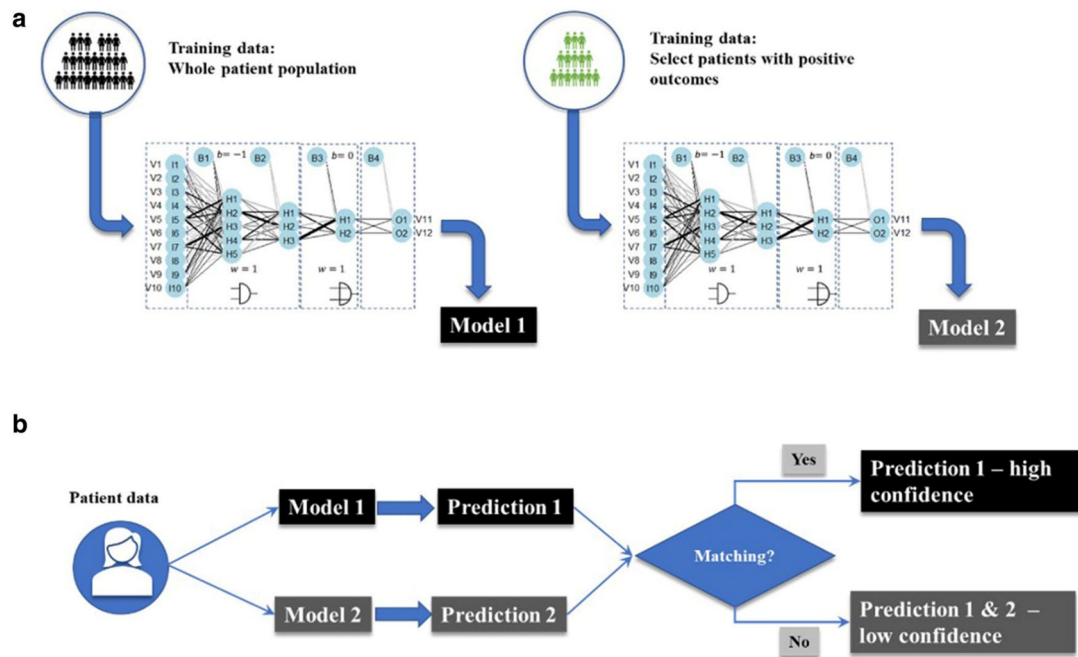


Figure 25. Siamese setting for a treatment recommender system.

8.5. Graph embedding

Patient Safety Indicators (PSIs), created by the federally managed Agency for Healthcare Research and Quality, aim to assess the frequency of potentially avoidable complications or adverse events that patients encounter while hospitalized. The empirical criteria for measurement are derived from widely accepted parameters, encompassing Inpatient Quality Indicators (IQI), Prevention Quality Indicators (PQI), and Patient Safety Indicators (PSI). These parameters are assessed as aggregated metrics, overlooking the inherent patterns and dynamics in the allocation of therapies and medical procedures to patients.

PSIs specifically measure complications and adverse events from:

- (1) medical conditions after admission
- (2) surgical procedures
- (3) obstetric procedures

PSIs have consistently revealed significant variations in complication and adverse event rates among hospitals. Additionally, there is supporting evidence indicating that elevated rates of complications and adverse events may be linked to deficiencies in the quality of care. There is a broad consensus that healthcare providers can mitigate patient complications and adverse events by enhancing the overall quality of care and safety within the healthcare environment.

The integration of information found in Electronic Health Records (EHRs) and their structured modeling is crucial for enhancing patient care, particularly in the development of personalized and adaptable treatment strategies. There is potential in establishing connections between ICD-10 diagnoses and medical procedures. The use of graph-based representations of health attributes can facilitate both detailed and generalized analyses of IQI.

Utilization of a graph-based data representation is anticipated to be highly effective in clustering “similar” patients. With such a model, patients will be interconnected when there are shared or analogous patterns observed among them. This concept leads to the creation of a network-like structure referred to as a patient graph (Ochoa and Mustafa, 2021). This structure can then be analyzed using Graph Neural Networks (GNN) to discern pertinent labels, specifically suitable medical procedures for recommendation. The successful

implementation of this graph framework not only relies on the quality of the underlying diagnostic system but also necessitates a comprehensive understanding of how patients with specific diseases are identified.

Patient clustering is conducted under the assumption that each patient, denoted as k , who shares the same or similar ICD pattern, is connected to other patients with similar ICDs. This means that patients with akin combinations of morbidities and comorbidities are expected to have similar Treatment Keys (TKs). This approach offers an advantage over traditional clustering methods due to its flexibility, allowing for the easy evaluation of different connectivity patterns in new graphs. Unlike traditional clustering methods like k-means or c-means, this graph-based clustering approach doesn't necessitate the use of ad-hoc definitions or parameters, as it establishes a direct correlation between ICDs and TKs.

Although embedding methods have demonstrated success in various applications, they have a fundamental limitation: their ability to capture complex patterns is intrinsically constrained by the dimensionality of the embedding space. (Nickel and Kiela, 2017). (Ochoa and Mustafa 2021) perform a patient embedding in a graph object G , as is shown in Figure 27, such that the corresponding ICD distributions are encoded by an object representing an interlinking between similar patients, an object that can be defined as a similarity score \mathcal{G}_{ij} , which is essentially the adjacency matrix for the linking between the patient i and j .

The objective is to recommend TKs for a patient depending on her diagnosis (ICDs) and additional metadata, like patients' age, gender, and pertinence to a specific health center. The TKs and ICDs are represented as a multi-label matrix $\mathcal{M}_{ICD} \in \mathbb{R}^{I \times M}$, where M is the number of unique ICDs and I is the total number of patients, such that each element in the matrix can be defined as \mathcal{M}_{ICD}^{im} (Figure 26).

	Q60.0	E83.4	N18.2	N18.82	D64.8	N25.8	N11.9	I11.00	N18.5	E83.38	..
26536	41	41	41	41	0	0	0	0	0	0	..
15328	0	82	82	0	0	0	0	0	0	0	..
22045	0	0	82	0	41	41	0	0	0	0	..
3931	0	41	41	41	41	0	0	0	0	0	..
9135	41	0	0	0	0	82	41	0	0	0	..
...
26042	0	0	0	0	0	2	0	0	1	0	..
3622	0	0	0	0	0	1	0	0	1	0	..
22119	0	0	0	0	0	2	0	0	0	0	..
19855	0	0	0	0	0	0	0	0	1	0	..
13913	0	0	0	0	0	1	0	0	1	2	..

Figure 26. Sample ICD multi-label matrix for the patient population \mathcal{M}_{ICD} , where the columns are the ICDs and the rows are the patients.

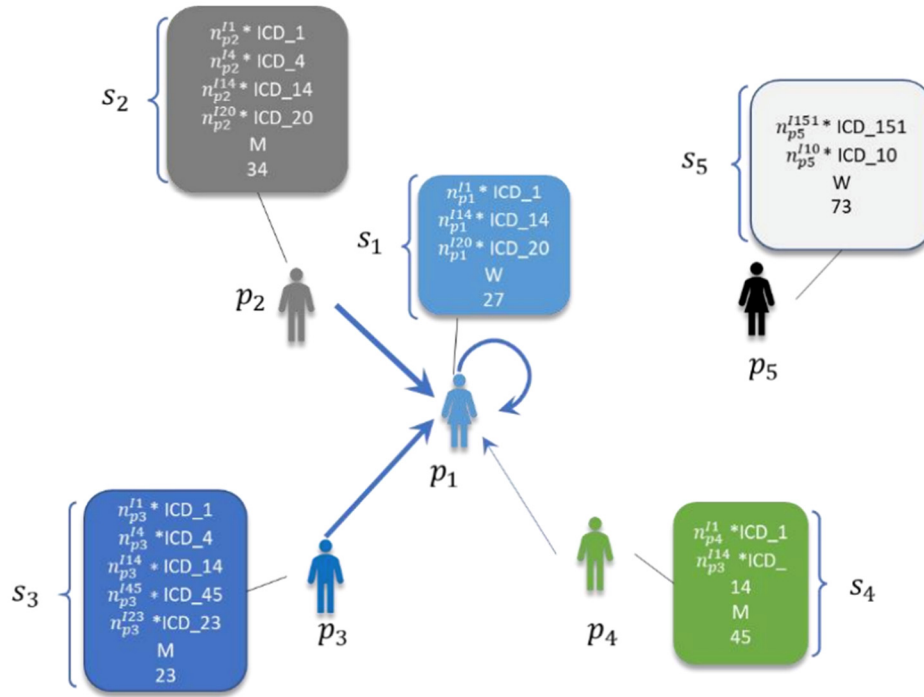


Figure 27. Graph encoding of patient's features.

Since a fix clustering method aggregates information in the defined cluster, with graph structures the information is aggregating in each node informing how strong is the similarity between patients, relying on similarity score of (Rocheteau et al. 2021). This similarity is defined by means of a score \mathcal{G}_{ij} , which is a function that depends on the ICD distribution as well as the gender and age relatedness between patient i and patient j

$$d_{ij} = \sum_{m=1}^M g_{ij}^m s_i^m s_j^m + c[\delta(\mu_i, \mu_j)(1 - |\alpha_i - \alpha_j|)]$$

where g_{ij}^m is essentially a weight for the ICD distribution for patients i and j , m is the number of elements of each of the \vec{s}^i vector, i.e., $\vec{s}^i = \{s_i^1, s_i^2, \dots, s_i^m, \dots\}$, and M is the total number of columns of the reduced-dimension \mathcal{M}'_{ICD} matrix. Additionally, there is a scoring that depends on the patient's meta parameters, where $\delta(\mu_i, \mu_j)$ is the Kronecker's delta (1 if $\mu_i = \mu_j$, 0 otherwise), and c is an importance hyper parameter, so that patients with similar parameters get a high score, adjusting the initial ICD scoring in this manner. The goal of this definition is to bias the graph construction to similar patients based not only on similar diagnoses, but also on similar age and gender.

This score is used to define the strength of the linking between two nodes. The corresponding structure of the similarity score \mathcal{G}_{ij} is a function that depends on this relatedness d_{ij} and the clustering of similar elements in the graph, and which essentially works as the adjacency matrix linking the nodes p_i and p_j . For the definition of this matrix, neighbor elements should be recognized depending on the distance between p_i and p_j . For this computation we estimate the nearest neighbors using a k-Nearest neighbors' method by finding a predefined number of training samples closest in distance to the new point and predict the label from that, such that

$$\hat{\mathcal{G}}_{ij} = \hat{\mathcal{G}}(d_{ij}(a), k, r) = d_{ij} \Leftrightarrow r > |d_{ij_k} - d_{ij}| \rightarrow 0 \text{ for } j \in \{j_1, j_2, \dots, j_k\},$$

where k is the maximal number of nearest neighbors explored by the clustering algorithm, a is an internal calibration parameter of the metric, and r is the radius where the clustering is performed.

The graph encoding, as shown in Figure 27, is then used as input for a convolutional neural graph in order to identify corresponding labels for each patient, located in each node of the network. In our case, the labels are the corresponding TKs assigned to each node (patient).

The whole algorithm is presented in Figure 28. One can see a shaped-charge architecture: the DNN is followed by the kNN.

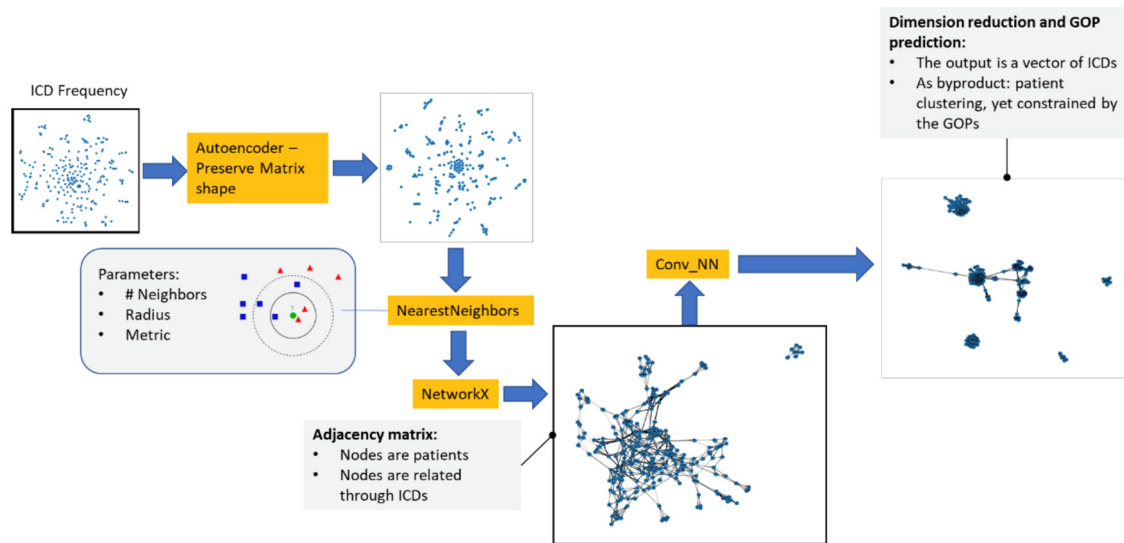


Figure 28. Training setting for patient graph learning.

8.6. Computing similarity between patients

To organize EHRs as a graph and propose a novel deep learning framework, structure-aware Siamese Graph neural networks (SSGNet, Gu et al 2022, Figure 29) perform robust encounter-level patient similarity learning while capturing the intrinsic graph structure and mitigating the influence from missing values. SSGNet initializes a patient graph with the instances in EHRs, then jointly learns the patient encounter representation and pairwise similarity. Two parameter-sharing Pro-GNNs are employed in a Siamese architecture to derive the embedding representation for each patient encounter pair. The low-rank regularized loss, contrastive loss and cross entropy loss are combined to supervise the end-to-end training process and enhance the model capacity against sparse relationships and missing values.

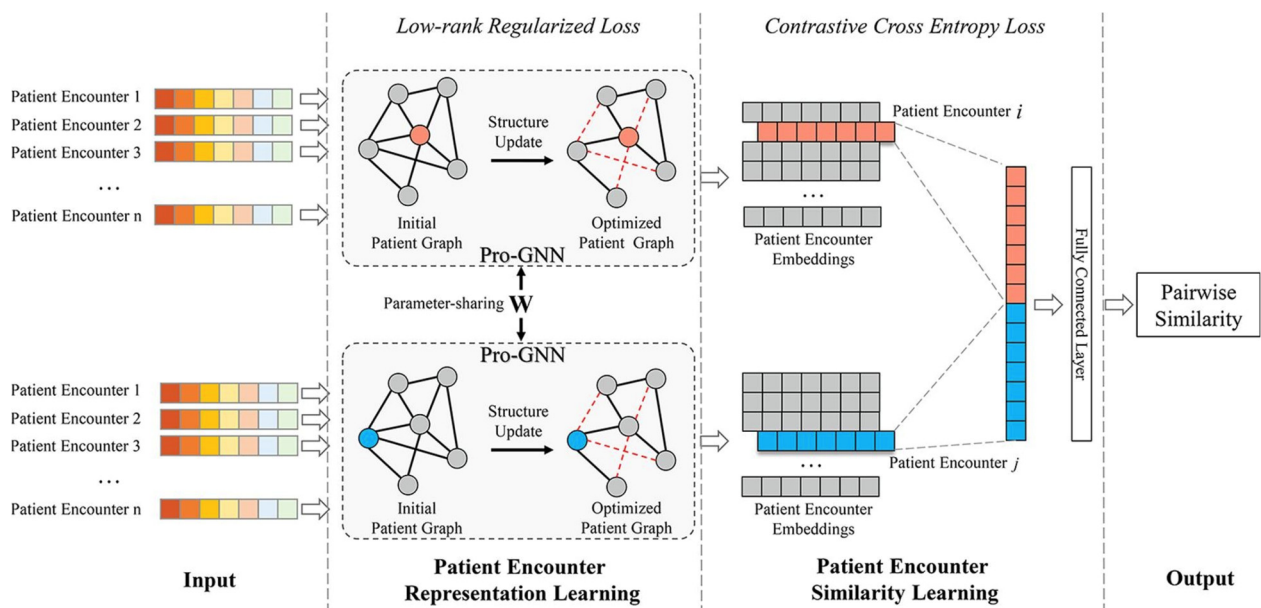


Figure 29. The overview of the proposed SSGNet for encounter-level patient similarity learning.

The suggested SSGNet approach initiates EHRs by configuring them as a patient graph. It combines two Pro-GNNs within a Siamese network framework, facilitating the concurrent learning

of encounter-level patient representations and pairwise similarities. This architecture is adept at capturing the inherent structure of the patient graph and reducing the impact of missing values within EHRs.

Let $X \in R_{N \times M}$ denotes a matrix containing the features of all instances, where the feature vector of each instance v_i is N -dimensional and represented by x_i . The patient graph is represented as $G = (V, E)$. Each instance is considered as a node thus the instance set V serves as the node set of the graph. E is the edge set, and $e_{ij} \in E$ denotes the edge between node v_i and v_j , which reflects the similarity between them.

SSGNet regards X as input and outputs pairwise similarity. In general, the main architecture of SSGNet consists of two modules:

- (1) patient encounter representation learning, and
- (2) patient encounter similarity learning.

The two modules can be trained in an end-to-end manner simultaneously. Specifically, after the patient graph is initialized, we adopt Pro-GNNs as the backbone to derive patient encounter embedding representations. Given a node pair (v_i, v_j) in the patient graph, we employ two parameters having Pro-GNNs in a Siamese architecture to embed individual nodes, respectively. Next, for patient encounter i and patient encounter j , the obtained embedding representations are concatenated and then fed into a fully connected layer to classify whether they are similar (Figure 30).



Figure 30. Similar patients have similar diseases.

9. Deployment of LLM Personalization

9.1. Service-based architecture for personalization

Figure 31 shows an overview of the framework, including three major components:

- (1) Interface,
- (2) Orchestrator, and
- (3) External Sources.

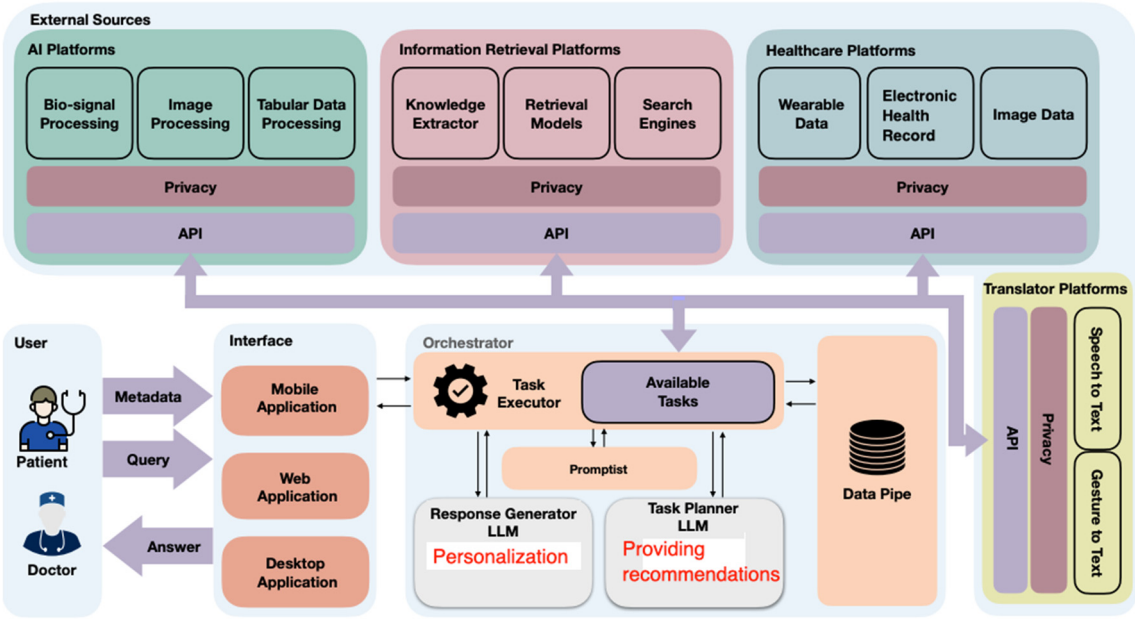


Figure 31. A service-based personalization architecture.

The interface serves as a connection point between users and agents, offering interactive tools accessible via mobile, desktop, or web applications. It incorporates multimodal communication channels, encompassing text and audio. Queries from users are received by the interface and then relayed to the Orchestrator. It is important to note that questions can be presented through various modes of human communication, such as text, speech, or gestures. Within this framework, users have the capability to provide metadata alongside their queries, which may include images, audio, gestures, and more. As an example, a user might capture an image of their meal and inquire about its nutritional values or calorie content, with the image serving as metadata.

The *Orchestrator* assumes the responsibility of problem-solving and decision-making to generate suitable responses based on user queries. It incorporates Neisser’s (1976) Perceptual Cycle Model, enabling the Orchestrator to perceive, transform, and analyze input queries and metadata for response generation. In this process, input data is aggregated, transformed into structured data, and analyzed to plan and execute actions. The Orchestrator engages with external sources to obtain necessary information, conduct data integration and analysis, and extract insights, among other functions. Here, we delineate five key components of the Orchestrator.

The Task Executor, functioning within the Orchestrator, executes actions by adhering to the decision-making process established by the Task Planner and invoking modules upon request. The Task Executor has dual responsibilities. Firstly, it serves as a data converter, translating input multimedia data into prompts. When the Task Executor receives user queries, including text, audio, and more, it calls on relevant external platforms to convert them into text.

Microsoft Azure provides a range of tools and services that can be leveraged for personalization in application development. These services often include machine learning capabilities, data analytics, and other AI-driven tools. Developers can use these resources to build personalized experiences for users, such as tailored content recommendations, targeted marketing, and more (Figure 32).

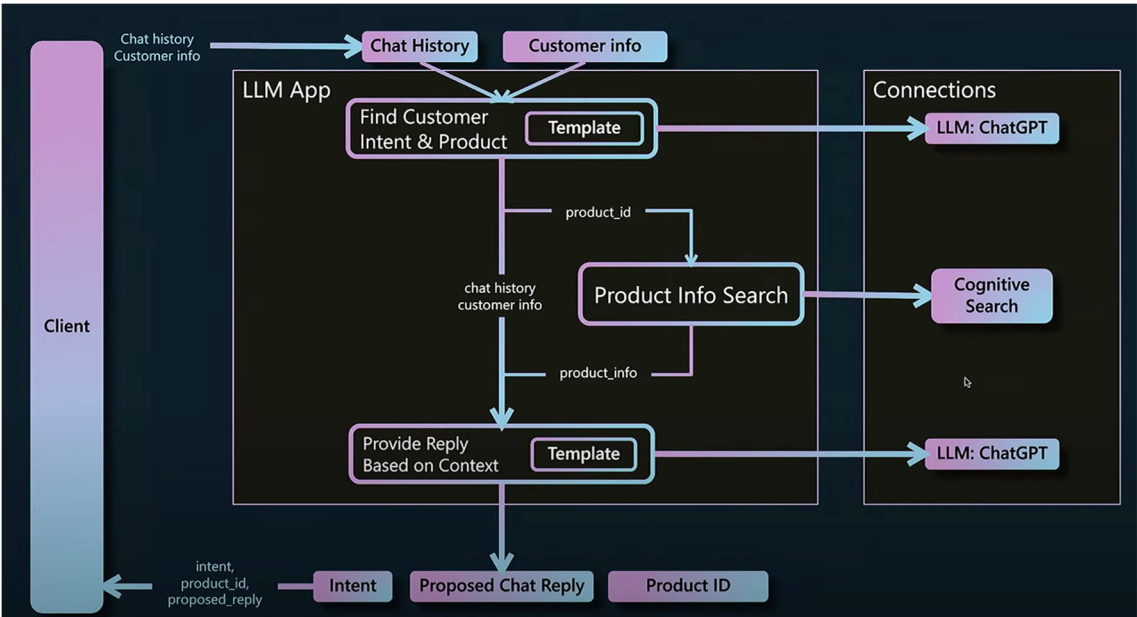


Figure 32. Microsoft Azure LLM development platform ready for personalization.

9.2. Decision Support

The decision support is delivered to patients through a mobile phone application in the form of self-management advice. This app is distributed for both Android and iOS platforms using the React Native framework. The process for creating and customizing self-management plans is outlined in steps 1-6 as depicted in Figure 33.

Before initiating the use of the app, patients are required to complete a web-based questionnaire (1) that gathers information on various personal characteristics. This information is then transmitted to the app server (2) to initiate the first Case-Based Reasoning (CBR) decision support cycle and obtain the initial plan of actions. This plan is subsequently delivered to the patients’ cell phones (5) and can be accessed by the patients (6). All further interactions occur via the app, which collects personalized data from patients every 10 days (4) and also physical activity data of patients from their wearable devices (3).

The user interaction component (4) involves a Question-and-Answer (Q/A) module utilized to adjust the weekly self-management plan. This adjustment is based on the responses to questions covering topics such as lower back pain, functional ability, fear avoidance, work ability, barriers to self-management, pain self-efficacy, sleep, perceived stress, mood, and adherence to the self-management plan. The sole objective is to ask questions that are pertinent to updating the current decision support for individual patients while avoiding unnecessary repetition or irrelevant queries for their follow-up.

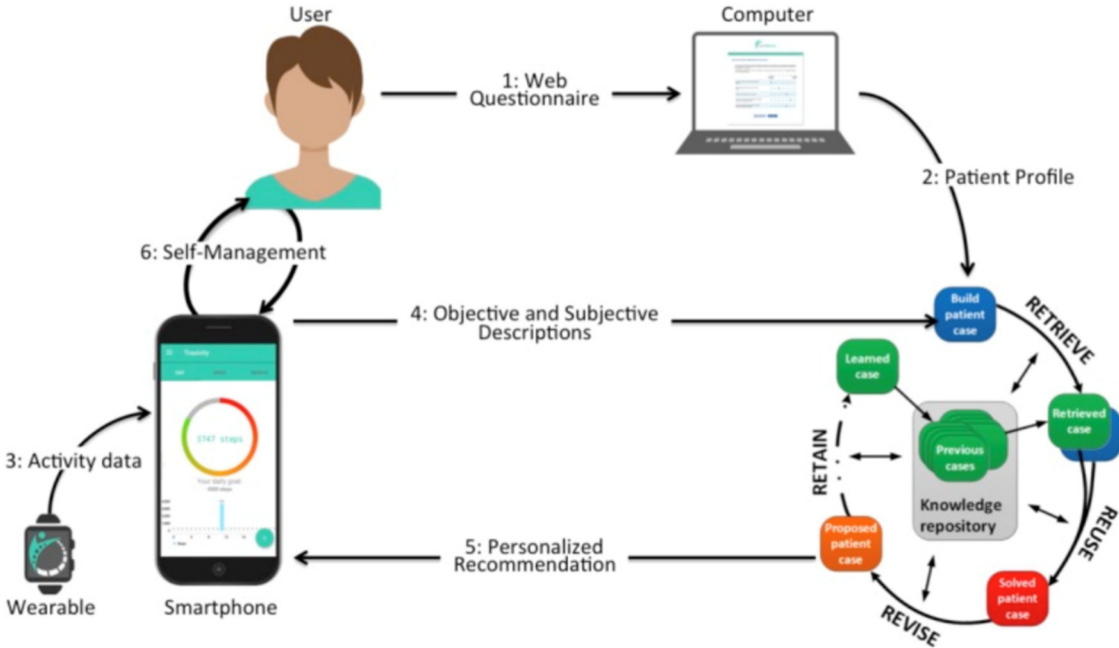


Figure 33. Decision support functionality.

10. Discussion and Conclusions

He et al. (2023) observe that even without fine-tuning, large language models can outperform existing fine-tuned conversational recommendation models in terms of relevance (Figure 34). Surprisingly, LLMs can function as standalone recommendation systems without requiring additional feature engineering or manual processes typically involved in common recommendation systems. This capability stems from the nature of LLMs and how they are pre-trained and operate.

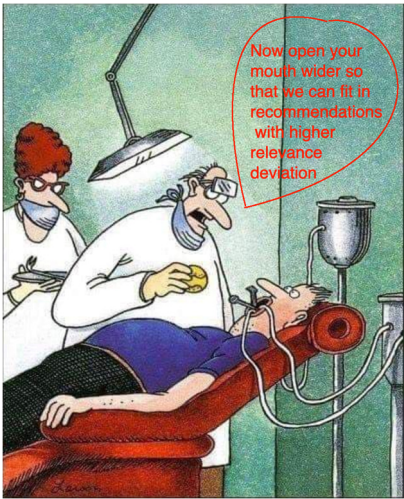


Figure 34. High deviation in recommendation relevance.

- There are the following advantages of LLM-based personalization in comparison with baseline LLM:
- (1) engage with user via interactions
 - (2) present outputs in a verbalized and explainable way;
 - (3) receive feedback from the user;
 - (4) make adjustment on top of the feedback from passively making search;
 - (5) recommendation help to proactively figuring out user’s need and seeking for user’s preferred items.

Also, LLMs may go beyond simply making personalized search and recommendation:

- (1) Take notes of users’ important information within their memory,
- (2) make personalized plans based on memorized information when new demands are raised,
- (3) execute plans by leveraging tools like search engines and recommendation systems

As to the difficulties in LLM personalization (Figure 35), they are as follows:

- (1) Personalization requires an understanding of user preference, which is a domain-specific knowledge rather than the commonsense knowledge learned by LLMs;
- (2) Unclear how to effectively and efficiently adapt of LLMs for personalized services;
- (3) LLMs could memorize user’s confidential information while providing personalized services;
- (4) There is a risk that most of the recommendations are similarity based (i.e., items that are semantically similar to the items purchased so far), though I do think with extensive training data of user purchase history, we can ameliorate this issue via the “collaborative filtering” approach that this method could mimic;
- (5) Also, since there is no weighting assigned to various actions or events a user undertakes, from exploration to the eventual purchase of an item, our reliance is solely on the LLM’s prediction of the most probable next token(s) for recommendations. We are unable to consider factors such as items bookmarked, items observed for a duration, items added to the cart, and similar user interactions.

It is noted that incorporating abduction and meta prompts significantly enhances personalization performance. Consequently, these avenues hold promise for enhancing overall recommendation accuracy as LLMs continue to undergo further development.



Figure 35. Difficulties of using LLM for personalization

HRSs have the capacity to inspire and involve users in modifying their behavior. They can offer individuals enhanced choices and practical insights derived from observed behavior. The overarching goal of HRS is to enable individuals to actively monitor and enhance their health by delivering personalized recommendations with the aid of technology, as outlined by De Croon et al. (2021). According to Sun et al. (2023), both the health care professionals along with the end users should participate in the future design and development of HRSs to optimize their utility and successful implementation.

A personalized agent designed to represent users in decision-making processes involving the consideration of mental attributes of both human and automated agents was introduced in (Galitsky 2002). The proposed framework included the Natural Language Multiagent Mental Simulator (NL_MAMS) as a component for reasoning and user profile specification. This simulator had the capability to predict the entire range of possible mental states a few steps ahead based on the current mental state of the involved agents.

The field of health services complaints stands out as one of the initial areas where personalized approaches to managing complaints are crucial. Galitsky (2005) introduced a personalized conflict

resolution agent that employs reasoning about mental attributes to handle scenarios involving multiagent interactions. This approach was implemented in the context of complaint analysis, necessitating a sophisticated user interface and machine learning to guide customers on crafting a valid complaint and assisting in its structured representation. The author illustrated that exchanging information solely on the mental actions of conflicting agents, even without domain-specific knowledge, is often adequate for assessing the validity of a complaint.

In their study, Zhou et al. (2023) discovered that user behavior data logged often contains noisy interactions. Through empirical analysis, they demonstrated the efficacy of filtering algorithms from the digital signal processing community in mitigating the impact of noise on recommendation models based on large language models (LLMs). Building upon this insight, the authors introduced the FMLP-Rec model, incorporating learnable filters for enhancing sequential recommendation tasks.

Galitsky (2016) introduced a framework aimed at extracting user interests from social network profiles, such as those on Facebook, in order to tailor recommendations for products and services. Traditional personalization systems, relying on keyword matching of user interests and product attributes, demonstrated low recall. This limitation necessitated a shift toward a more generalized, category-based framework. The challenge lies in the need for substantial reasoning about products and their categories, aligning the taxonomy of a user's expressed public profile with that of products and services. To address inconsistencies between these taxonomies, Galitsky proposed a mapping expressed as a Defeasible Logic program. In this program, a potential mapping can be overridden by others if additional relevant information becomes available. The study introduces a Facebook app and a mobile app with similar functionality, designed for recommending activities. The accuracy and usability of these apps are evaluated. Additionally, a content management system supporting personalized recommendations is outlined as part of the proposed framework.

Galitsky (2021) proposed an innovative approach to conversational recommendation that deviates from traditional methods of questioning users about their preferences. Instead, our recommender system monitors the user's interactions with others, including customer support agents (CSAs), and intervenes in the conversation only when there is a pertinent recommendation to make, and the timing is appropriate. The Recommender Joining Conversation (RJC) model utilizes information extraction, discourse and argumentation analyses, as well as dialogue management techniques to generate recommendations for products and services based on the user's needs inferred from the ongoing conversation. We specifically address scenarios where customers discuss their issues with CSAs, aiming to resolve problems and receive recommendations for products tailored to address these issues. Evaluations conducted in various human-human and human-chatbot dialogues demonstrate that RJC is an effective and unintrusive method, providing highly relevant and persuasive recommendations.

Recent research has demonstrated the successful application of personalized word embeddings in addressing ambiguity in various downstream tasks (Welch et al., 2023). Garimella et al. (2017) explored the impact of location and gender on associations with words such as "health" and other stimuli like "stack" — does it evoke thoughts of books or pancakes? Welch et al. (2020) delved into different associations, such as the interpretation of "embodying" an idea, which may vary depending on religious or economic beliefs. Similarly, the word "wicked" may denote "evil" or serve as an intensifier, contingent on geographical location (Bamman et al., 2014). These instances illustrate how personalized representations can enhance distinctions in meaning. However, static representations have their limitations. A diversity of meaning for various combinations of horses, men and heads is shown in Figure 36.

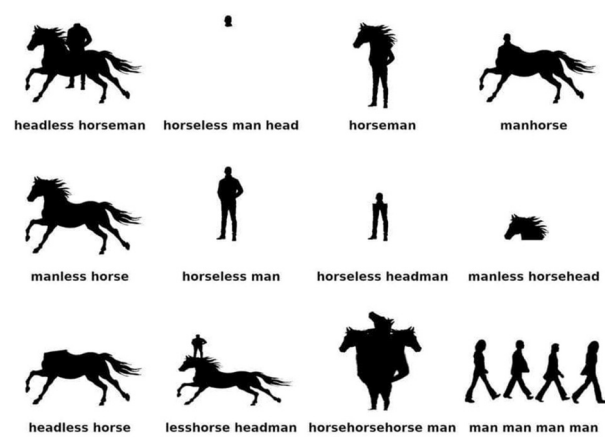


Figure 36. Visualization of superpositions of horse-man-head.

We concur with Wiesner and Pfeifer (2014) in asserting that collaborative filtering is unsuitable for a Health Recommender System. Collaborative filtering involves examining user profiles across users, posing a significant security risk, especially considering the stringent confidentiality requirements of Personal Health Record systems. While it might be argued that the similarity measuring across user profiles is conducted by the recommendation engine (a machine) rather than humans, assuring users that their profiles remain private proves challenging. In practice, explaining this concept to an average user, concerned about data confidentiality, becomes complex. Additionally, any technical implementation must be meticulously designed to prevent security vulnerabilities. Processing data from different users in the same computational units simultaneously (i.e., server-side sessions) could potentially expose security holes, making it easier for hackers to exploit and access another user’s profile data.

We now demonstrate that LLMs are good at resolving ambiguities; this property is naturally leveraged by recommenders. Let us now look at comparison of ‘testing’ and ‘sanitation’. Observations reveal that in certain contexts, ‘testing’ may refer to assessing the functionality of a device, and ‘sanitation’ may relate to a pest control issue. In another context, both terms may signify conditions related to the COVID-19 pandemic. Personalized LLMs, designed to better predict individual expressions, could effectively address these nuances, as they learn dynamic encodings of words. In another example, a meaning of “personalization elixir” is depicted in Figure 37.



Figure 37. Personalization elixir.

Towards the conclusion of this section, we present specific examples illustrating the application of LLMs in healthcare:

- *Babylon Health*: A UK-based company, Babylon Health utilizes LLMs to deliver virtual healthcare services through its app. Patients can engage in conversations with doctors or nurses assisted by LLMs. The LLM aids in diagnosing conditions, recommending treatments, and providing answers to inquiries.
- *InteliHealth*: Operating in the United States, InteliHealth employs LLMs to craft personalized health plans. Through the InteliHealth website, patients respond to questions about their health and lifestyle, and the LLM generates a tailored plan encompassing recommendations for diet, exercise, and medication.
- *Google Health*: Google Health, functioning as a personal health record (PHR), integrates LLMs to assist patients in managing their health. Users can track their medical history, medications, and symptoms using Google Health. The LLM then generates reports and insights to enhance patients’ understanding of their health.

These examples provide a glimpse into the diverse applications of LLMs in healthcare. As LLMs continue to evolve, we anticipate witnessing even more innovative and effective uses of this technology in the coming years.

11. Hands-on

The reader is recommended to run the notebook https://colab.research.google.com/github/bgalitsky/LLM-personalization/blob/main/Treatment_Recommendation_Personalization_System.ipynb. This recommendation system is based on LLM using a T5 architecture. LLMs embed text allowing for similar words/terms to have similar embeddings, and the recommendation system leverage these embeddings to recommend the next treatment for a patient, based on her treatment history. The P5 team uses a number of input-output prompts, essentially giving the model a list of treatments a patient has already received, and asking which treatment should follow (from a list of candidates). Embedding the treatment history as a series of tokens allows the model to determine what types of treatment the patient is expected to receive, solely based on the words and their embeddings/”meanings”. In turn, the selection of the next treatment becomes similarity based (Figure 38).

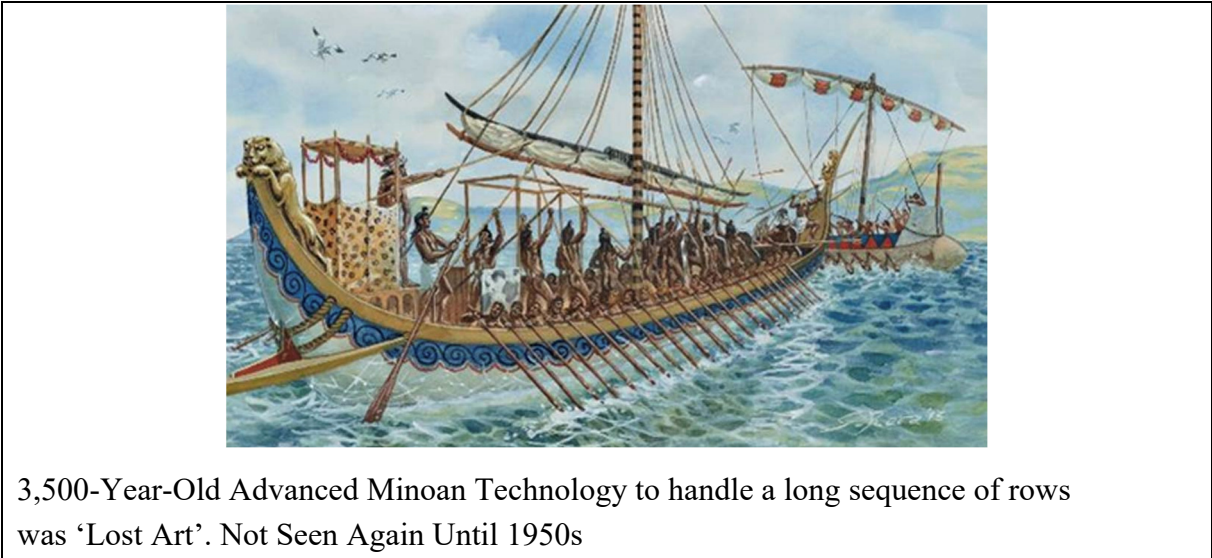


Figure 38. Learning from a long list of similar items.

References

1. Abbasian M, Iman Azimi, Amir M. Rahmani, Ramesh Jain (2023) Conversational Health Agents: A Personalized LLM-Powered Agent Framework. <https://aps.arxiv.org/abs/2310.02374>
2. Aboufoul M (2023) Using Large Language Models as Recommendation Systems. A review of recent research and a custom implementation. <https://towardsdatascience.com/using-large-language-models-as-recommendation-systems-49e8aeef29b>
3. Bamman D, Chris Dyer, and Noah A. Smith. (2014) Distributed representations of geographically situated language. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).
4. Beckers health (2023) <https://www.beckershospitalreview.com/healthcare-information-technology/google-receives-more-than-1-billion-health-questions-every-day.html>, accessed on 6 February 2021).
5. Bocanegra CL, Sevillano JL, Rizo C, Civit A, Fernandez-Luque L. HealthRecSys: a semantic content-based recommender system to complement health videos. BMC Med Inform Decis Mak. 2017 May 15;17(1):63. doi: 10.1186/s12911-017-0431-7
6. Cai Y, Yu F, Kumar M, Gladney R, Mostafa J. Health Recommender Systems Development, Usage, and Evaluation from 2010 to 2022: A Scoping Review. Int J Environ Res Public Health. 2022 Nov 16;19(22):15115.
7. Carter, E.L.; Nunlee-Bland, G.; Callender, C. 2011. A Patient-Centric, Provider-Assisted Diabetes Telehealth Self-Management Intervention for Urban Minorities. Perspect. Health Inf. Manag.
8. Chen, J., Liu, Z., Huang, X., Wu, C., Liu, Q., Jiang, G., Pu, Y., Lei, Y., Chen, X., Wang, X., Lian, D., & Chen, E. (2023). When Large Language Models Meet Personalization: Perspectives of Challenges and Opportunities. ArXiv, abs/2307.16376.
9. Colab 2023
https://colab.research.google.com/github/Mohammadhia/t5_p5_recommendation_system/blob/main/T5P5_Recommendation_System.ipynb
10. Dai, W., Xu, Q., Yu, Y., & Zhou, Z. (2019). Bridging Machine Learning and Logical Reasoning by Abductive Learning. Neural Information Processing Systems.
11. De Croon R, Van Houdt L, Htun NN, Štiglic G, Vanden Abeele V, Verbert K. Health Recommender Systems: Systematic Review. J Med Internet Res. 2021 Jun 29;23(6):e18035.
12. Denecker, M. and Kakas, A.C. (2002). "Abduction in Logic Programming". In Kakas, A.C.; Sadri, F. (eds.). Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski. Lecture Notes in Computer Science. Vol. 2407. Springer. pp. 402–437.
13. Di Palma D, Giovanni Maria Biancofiore, Vito Walter Anelli, Fedelucio Narducci, Tommaso Di Noia, Eugenio Di Sciascio (2023) Evaluating ChatGPT as a Recommender System: A Rigorous Approach. <https://arxiv.org/abs/2309.03613>
14. Fidler, Jessica, and Yoav Goldberg. "Controlling linguistic style aspects in neural language generation." arXiv preprint arXiv:1707.02633 (2017).
15. Flek L 2020. Returning the N to NLP: Towards contextually personalized classification models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7828–7838
16. Friedman L, S. Ahuja, D. Allen, T. Tan, H. Sidahmed, C. Long, J. Xie, G. Schubiner, A. Patel, H. Lara et al., "Leveraging large language models in conversational recommender systems," arXiv preprint arXiv:2305.07961, 2023.
17. Galitsky B -github (2024) Treatment Recommendation Personalization System. https://github.com/bgalitsky/LLM-personalization/blob/main/Treatment_Recommendation_Personalization_System.ipynb
18. Galitsky B (2002) Designing the personalized agent for the virtual mental world. AAAI FSS-2002 Symposium on Personalized agents, 21-29
19. Galitsky B (2005) A Personalized Assistant for Customer Complaints Management Systems. AAAI Spring Symposium: Persistent Assistants: Living and Working with AI, 84-89
20. Galitsky B (2016) Providing personalized recommendation for attending events based on individual interest profiles. Artif. Intell. Res. 5 (1), 1-13
21. Galitsky B (2017) Content inversion for user searches and product recommendations systems and methods US Patent 9,336,297
22. Galitsky B (2021) Recommendation by Joining a Human Conversation. In book: Artificial Intelligence for Customer Relationship Management: Solving Customer Problems, Springer, Cham

23. Galitsky B, EW McKenna (2017) Sentiment extraction from consumer reviews for providing product recommendations. US Patent 9,646,078
24. Galitsky B, Kovalerchuk B (2014) Improving web search relevance with learning structure of domain concepts. In: Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday. 341-376. Springer New York
25. Galitsky B(2023) Generating recommendations by using communicative discourse trees of conversations. US Patent 11,599,731
26. Gao T, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In Proc. of ACL-IJCNLP, pages 3816–3830.
27. Gao Y, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-rec: Towards interactive and explainable LLMs-augmented recommender system," arXiv preprint arXiv:2303.14524, 2023.
28. Garimella A, Carmen Banea, and Rada Mihalcea. 2017. Demographic-aware word associations. In nProceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.
29. Geng S, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22). Association for Computing Machinery, New York, NY, USA, 299–315. <https://doi.org/10.1145/3523227.3546767>
30. Goodman N (2023) Meta-Prompt: A Simple Self-Improving Language Agent. <https://noahgoodman.substack.com/p/meta-prompt-a-simple-self-improving>
31. Greylink C (2023) LLMs and meta-prompting. <https://cobusgreyling.medium.com/meta-prompt-60d4925b4347>. Last downloaded Nov 22, 2023
32. Gu Y, Xuebing Yang, Lei Tian, Hongyu Yang, Jicheng Lv, Chao Yang, Jinwei Wang, Jianing Xi, Guilan Kong, Wensheng Zhang. 2022 Structure-aware siamese graph neural networks for encounter-level patient similarity learning, Journal of Biomedical Informatics, Volume 127,
33. He, Z., Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large Language Models as Zero-Shot Conversational Recommenders. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom
34. Hou Y (2023) Awesome LLM-powered agent. <https://github.com/hyp1231/awesome-llm-powered-agent>
35. Hou Y, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. 2022. MetaPrompting: Learning to Learn Better Prompts. In Proceedings of the 29th International Conference on Computational Linguistics, pages 3251–3262
36. Kakas AC, R. A. Kowalski, and F. Toni. Abductive logic programming. Journal of Logic Computation, 2(6):719–770, 1992.
37. Khaleel I., Wimmer B.C., Peterson G.M., Zaidi S.T.R., Roehrer E., Cummings E., Lee K. Health Information Overload among Health Consumers: A Scoping Review. Patient Educ. Couns. 2020;103:15–32.
38. Kirk HR, Bertie Vidgen, Paul Röttger, and Scott A Hale. 2023. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. arXiv preprint arXiv:2303.05453
39. Lewis, Robert, Ferguson, Craig, Wilks, Chelsey, Jones, Noah and Picard, Rosalind. 2022. Can a Recommender System Support Treatment Personalisation in Digital Mental Health Therapy? A Quantitative Feasibility Assessment Using Data from a Behavioural Activation Therapy App.
40. Li, Xiang Lisa and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In Proc. of ACL-IJCNLP, pages 4582–4597.
41. Lyu H and Song Jiang and Hanqing Zeng and Qifan Wang and Si Zhang and Ren Chen and Chris Leung and Jiajie Tang and Yinglong Xia and Jiebo Luo (2023) LLM-Rec: Personalized Recommendation via Prompting Large Language Models, arXiv2307.15780
42. Mork PJ, Bach K; selfBACK Consortium. A Decision Support System to Enhance Self-Management of Low Back Pain: Protocol for the selfBACK Project. JMIR Res Protoc. 2018 Jul 20;7(7):e167.
43. Neisser, U. (1976). Cognition and reality. San Francisco, CA: W. H. Freeman and Company.
44. Nickel, M., and Kiela, D. (2017). Poincaré Embeddings for Learning Hierarchical Representations. ArXiv170508039 Cs Stat.
45. Ochoa JGD, Mustafa FE. Graph neural network modelling as a potentially effective method for predicting and analyzing procedures based on patients' diagnoses. Artif Intell Med. 2022 Sep;131:102359.

46. Ochoa, J.G.D., Csiszár, O. & Schimper, T. Medical recommender systems based on continuous-valued logic and multi-criteria decision operators, using interpretable neural networks. *BMC Med Inform Decis Mak* 21, 186 (2021)
47. Pew Research Center 2021 <https://www.pewresearch.org/internet/2005/05/17/health-information-online/>, accessed on 6 February 2021)
48. Rocheteau, E., Tong, C., Veličković, P., Lane, N., and Liò, P. (2021). Predicting Patient Outcomes with Graph Representation Learning. *ArXiv210103940 Cs*.
49. Sahoo, A.K.; Pradhan, C.; Barik, R.K.; Dubey, H. DeepReco: Deep Learning Based Health Recommender System Using Collaborative Filtering. *Computation* 2019, 7, 25.
50. Salemi A, S Mysore, M Bendersky, H Zamani (2023) LaMP: When Large Language Models Meet Personalization. *arXiv:2304.11406*
51. Skopyk Khrystyna, Artem Chernodub and Vipul Raheja (2022) Personalizing Large Language Models
52. Sun Y, Zhou J, Ji M, Pei L, Wang Z. Development and Evaluation of Health Recommender Systems: Systematic Scoping Review and Evidence Mapping. *J Med Internet Res*. 2023
53. Syed, Bakhtiyar, et al. Adapting language models for non-parallel author-stylized rewriting." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34.
54. No. 05. 2020."
55. Wang A, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
56. Wei Y, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua(2019). MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *ACM MM'19, NICE, France, Oct. 21-25, 2019*
57. Welch C, Jonathan K. Kummerfeld, Veronica Perez-Rosas, and Rada Mihalcea. 2020. Compositional demographic word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*
58. Welch, Charles & Gu, Chenxi & Kummerfeld, Jonathan & Pérez-Rosas, Verónica & Mihalcea, Rada. (2022). Leveraging Similar Users for Personalized Language Modeling with Limited Data. *ACL*, 1742-1752.
59. Wiesner M, Pfeifer D. Health recommender systems: concepts, requirements, technical basics and challenges. *Int J Environ Res Public Health*. 2014 Mar 3;11(3):2580-607.
60. Xue, Gui-Rong & Han, Jie & Yu, Yong & Yang, Qiang (2009) User language model for collaborative personalized search. *ACM Trans. Inf. Syst.*. 27. 10.1145/1462198.1462203.
61. Yang F, Zheng Chen, Ziyang Jiang, Eunah Cho, Xiaojiang Huang, Yanbin Lu (2023) Palr: Personalization aware LLMs for recommendation," *arXiv preprint arXiv:2305.07622*,
62. Yang, Diyi, and Lucie Flek. (2021) Towards User-Centric Text-to-Text Generation: A Survey." *International Conference on Text, Speech, and Dialogue*. Springer, Cham, 2021.
63. Zhou K, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2388–2399

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.