EU-FarmBook

Search and recommender system v1

Deliverable 2.4

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Deliverable summary

Call	HORIZON-CL6-2021-GOVERNANCE-01
Topic	HORIZON-CL6-2021-GOVERNANCE-01-24
Project	EU-FarmBook: supporting knowledge exchange between all AKIS actors in the European Union
Acronym	EU-FarmBook
Project No.	101060382
Management	Universiteit Gent
Duration	84 Months
Start date	01/08/2022
End date	31/07/2029

Deliverable	D2.4 Search and recommender system v1
Type	R (Document/Report)
Dissemination level	PU (Public)
Due Date	31/07/2023
Submission Date	30/09/2023
Work Package No.	WP2
Lead Beneficiary	TNO
Authors	Daan Vos (TNO) | Daan Di Scala (TNO) | Liv Ziegfled (TNO) | Mike Wilmer (TNO) | Joachim de Greeff (TNO)
Contributors	Stephan Raaijmakers (TNO) | Montserrat Cuadros Oller (Vicomtech)
Version	Version 1

History of Changes

Version 1	31/09/2023	TNO	First version

Index

Figure index

List of Abbreviations

KO	Knowledge Object
LLM	Large Language Model
LoRA	Low-rank adaptation of large language models
MVP	Minimum Viable Product
RAG	Retrieval Augmented Generation
TD-IDF	Term Frequency-Inverse Document Frequency

Executive Summary

This deliverable describes in detail the functioning of the FarmBook Search & Recommender system V1. The goal of the system is to provide a user with the best matching knowledge from the FarmBook database. To do so, a user's query is processed through the search and recommender system, while taking the user's preferences into account. The overall system architecture is discussed, explaining how the user interaction flow takes place through the conversational module, the user's preference elicitation process is explained, which is used to make the experience more personal, and the various ways are described in which the search module operates and the Recommender system functions.

1. Introduction

Within this deliverable we describe the architecture, inner workings and functionalities of the FarmBook Search & Recommender system V1. The goal of the FarmBook system is to provide a user with relevant information from the FarmBook database. In a typical interaction with the system, a user engages in a conversation with the system through a chatbot interface (Section 2). Through this conversation, the system determines which information the user wants by using two methods: search (Section 3) and recommendation (Section 4). Both methods try to find relevant knowledge in the FarmBook database and serve these to the user. See Figure 1 for an overview of this interaction.

In order to provide the user with the best possible knowledge and recommendations, we personalize the user-system interaction by aligning the systems output with the user preference. As a prerequisite, we engage in user preference elicitation (Section 5).

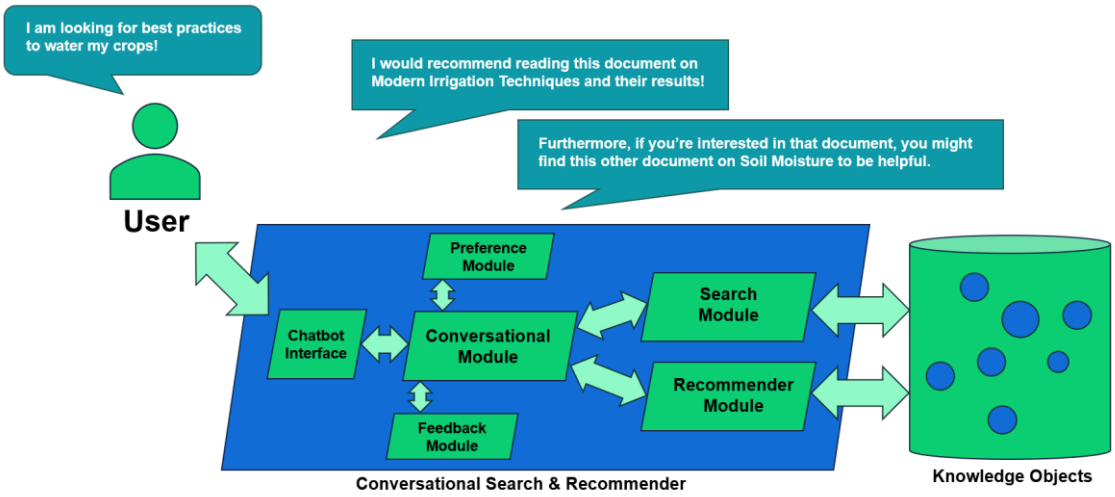


Figure 1: Technical architecture conversational search and recommendation

The architecture of the FarmBook Search & Recommender system follows a common conversational recommender system structure (Jannach et al., 2021). The goal is to give relevant answers to the questions asked by the user, and prompt them with related knowledge. Note that not only exact matches to the questions are found, but rather synonymous recommendations. To do so, the system uses information stored in the FarmBook documents known as Knowledge Objects (KOs) and their metadata, such as title, keywords, link to the documents, etc.

The internal workings of the conversational recommender system are split into different front-end and back-end modules. First, the front-end user interface allows users to interact with the system. The back-end modules consist of a conversational module handling the dialog with the user; a search module handling the linking and ranking the data into a personalized search space and a recommender module handling the retrieval and ranking of information based on personal preferences. We will now discuss these back-end modules in turn.

2. Conversational Module

To facilitate a natural and fruitful conversation of users with the FarmBook Search & Recommender system, the user is guided in their search for information through a dialog flow, see Figure 2.

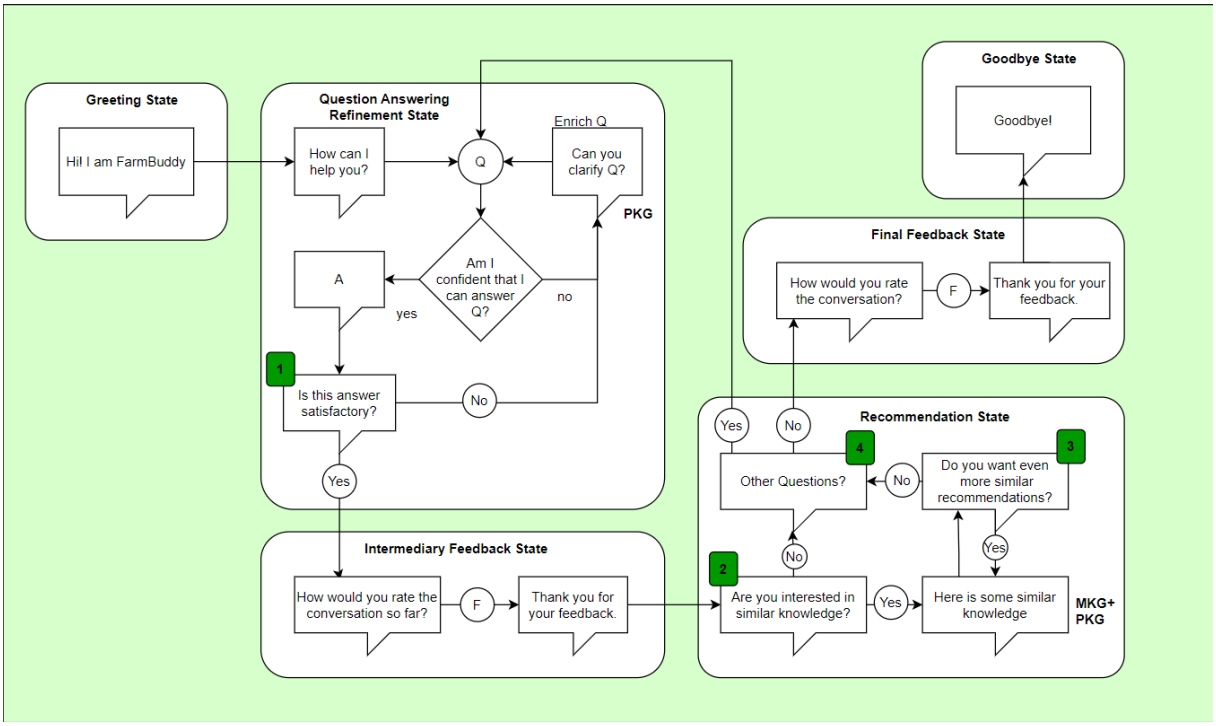


Figure 2: Dialog flow

To guide the user, the system makes use of a dialogue flow model (Figure 2) guiding the conversation through different states. In the Greeting state the chatbot greets the user, and if desired, the system starts with asking a couple of initial preference elicitation questions to gather information about the user's preferences, with two main goals: Building up a user profile, to tailor the search & recommendation modules, as well as to decrease the search space, to mitigate the cold start problem (Bobadilla et al., 2012).

After the initial greeting, the system enters the Question Answering Refinement state. Here, the user is prompted for their question and the chatbot tries to answer as best as possible, by retrieving KOs and estimating its confidence in being able to answer. If it is not confident in answering the question, it will prompt the user with clarification questions. After doing so, the chatbot generates an answer based on the extracted information.

Further down in the dialog flow, the system will reach the recommendation state. Here, the chatbot will generate recommendations based on the earlier answered questions. If the user is interested in more information, the conversation will return to the question answering state. Otherwise, the conversation ends by the system telling the user goodbye.

We collect user feedback during and at the end of the conversation regarding the experience of the conversation and the quality of recommendation. This feedback is used for future training of the system, specifically for mitigation of the *cold start problem*, i.e. not knowing anything about what the user wants (elaborated upon in section 5).

Our work on the search and recommender approaches and preference elicitation is discussed further in-depth in the following sections.

3. Search Module

In this section we describe the functioning of the search module in more detail. To provide an answer to a user generated query, the following four high level steps are taken:

1. Store chunks of the FarmBook KOs in a vector database (section 3.1)
2. Measure the distance in semantic meaning between the user query and the KOs in the vector database (section 3.1)
3. Estimate how close the query and the most similar KOs are in semantic meaning (section 3.2)
4. Return the most similar KOs or merge multiple KOs together to output a final answer (section 3.3)

The FarmBook KOs are well annotated which allows for search on the meta data level and on the content level. Search on the meta data level takes into account features such as the language of the knowledge object, modality and the agricultural domain (i.e. forestry or crop farming). Ultimately, a combination of the two levels of search is expected to achieve the best performing retrieval performance.

3.1. Vector database

Vector databases transform input (i.e. text, images) into high dimensional vectors. This transformation can be achieved in numerous ways such as a bag-of-words, word embeddings or sentence embeddings. A common practice is to use chunks (small snippets) of input to increase search efficiency and to provide finer detailed search results. The meta data level search uses the ElasticSearch vector database while the content level search uses the Meta FAISS vector database. Both sections are explained in more detail below.

Meta data level search
Search on the meta data level is achieved by ElasticSearch. Elasticsearch is more than just a search engine. It transcends its basic classification as a search engine by offering a range of advanced features that significantly enhance data interaction and retrieval capabilities.

Its robust suite of functionalities includes full-text search, fuzzy search, autocomplete, MoreLikeThis, and so on. The full-text search capability facilitates efficient and contextually rich data retrieval, setting it apart from conventional search algorithms. It offers refined search results by understanding the deeper semantic meaning within the text corpus. In cases where search queries contain minor errors or types, the fuzzy search feature becomes invaluable. This function employs the Levenshtein distance algorithm to identify and return results that are approximate matches to the query term, which improves the flexibility and user-friendliness of the search process. Autocomplete serves as another essential component by accelerating the user's search experience. As the user starts to type, this feature offers predictive suggestions, streamlining the search process and enhancing usability.

The MoreLikeThis feature, which employs the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm, adds another layer of sophistication. This is particularly useful for identifying content similar to what the user has searched, similar to how Google shows related questions or topics.

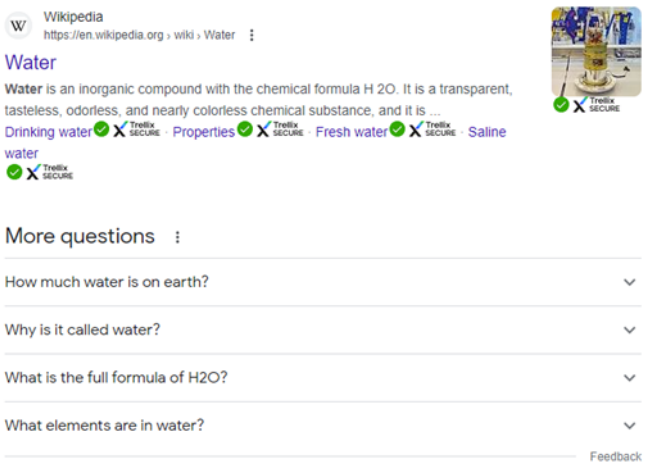


Figure 3: Example of the MoreLikeThis feature

Elasticsearch is not a static system. As the data grows and search requirements change, it enables greater adaptability and flexibility. We plan to include exploring geo-based search and multi-lingual support. In summary, Elasticsearch is more than just a search engine; it is a multifaceted tool equipped with advanced features designed to raise the standard of modern search tasks.

Content level search
Search on the content level is achieved by META's FAISS vector database (Johson, Douze, Jégou, 2019). A beneficial property of FAISS is that it offers paralelization support on multiple GPUs with high performance as a result. Another useful feature of FAISS is to estimate the distance between a query and the results, which will be further introduced in the next section. The FAISS vector data base has to be initialized with embeddings.

For the FarmBook project we have opted to use INSTRUCTOR XL (Kasai et al, 2022) sentence embeddings since they achieve rank 1 on the embeddings leaderboard benchmark[1]. After transforming the input to vectors, a user query needs to be transformed too. Then, using a distance metric, the similarity between the query and the vectorized documents can be calculated and the best matching documents can be retrieved.

3.2. Confidence estimation

The search system will always provide some answer to a user question, even if the user query and retrieved documents are very far away in semantic meaning. For this reason, it is wise to work with a "confidence estimation" that acts as a threshold. The flow of the conversation will differ depending on whether the user query is below or above the threshold (see the "Am I confident that I can answer Q" task in the "Question Refinement state in Figure 2: Dialog flow).

3.3. Prompting/final answer generation

Thus far, the search system is able to output one or multiple knowledge object chunks. However, these chunks do not immediately answer the user query. For this reason, three different approaches have been developed to output a final answer, namely:

1. Summarization
2. Retrieval Augmented Generation
3. Finetuning using LoRA

For the *summarization* approach, one or more retrieved chunks are summarized into a couple sentences. The employed summarization model is a Llama 2 13B[2] (Touvron et al., 2023) Large Language Model (LLM) finetuned on the Samsum corpus (Gliwa, Mochol, Biesek, & Wawer, 2019).

[1] Link to the Huggingface embedding leaderboard
[2] Link to the Llama 2 13B model

The *Retrieval Augmented Generation (RAG)* approach also employs a Large Language Model but in a different way as the previous approach. In this method, the LLM receives as input the user query and the retrieved chunk and is prompted to give a final answer. The LLM that is used for this method is a quantized Llama 2 70B model[3].

Finally, the *Finetuning using LoRA* approach takes a base LLM but finetunes the model for a specific task. The specific task that the model is finetuned on is similar to the RAG approach. The distinction between the two methods is that instead of specifying this task in the prompt, the task is being taught to the model itself. The finetuning method that is being used for this is named Low-Rank Adaptation of Large Language Models (Shen et al, 2021). While LLMs are extremely large in size and for this reason resource intensive to train or finetune, the LoRA method only finetunes a small subset of the layers which improves the speed drastically while keeping the required resources low.

In our first experiments the *summarization* and *retrieval augmented generation* approaches perform better than the *Finetuning using LoRA* approach.

4. Recommendation Module

On top of the search mechanisms, we aim to provide recommendation of similar KOs to the user through this system. Many existing recommender systems utilize machine learning (or deep learning) approaches, which is a double-edged sword in AI approaches. Machine learning approaches perform well because of high accuracy and low executing time. However, a challenge is their explainability and correctness, and they are often considered as a kind of "black box" approach. To ensure correctness of the recommendations, we utilize the FarmBook knowledge graph for recommending KOs to the user.

To provide the recommendation, we retrieve KOs from the database that are similar to the earlier Knowledge Object given as answer to the user's search. Based on this, we build a weighted knowledge graph. We then rank the most similar and relevant KOs, to be returned as recommendation. We generate a recommendation based on the top ranked similar KOs. The recommendation is generated with a similar approach as mentioned in Section 3.3 to ensure a coherent summarized answer in natural language.

Multiple approaches (denoted as λ) for the ranking of similarity between KOs are considered:

- Direct keyword-keyword matching (λ1); Matching KOs that have a high overlap of the same keywords.
- Direct keyword-description and keyword-title matching (λ2); Matching KOs that have their keywords appear in other KOs' description or titles.

[3] Link to the used model

- Direct language-language, category-category, and modality-modality matching (λ3); Similar to keyword-keyword matching, only that this type of matching is used to only result an exact match between languages, category or modality of the compared KOs. If KOs fall in another category, users with strong preferences for their own domain might not be interested. Furthermore, this type of matching can also be used to ensure that only KOs that are written in the same language are recommended back to the user.
- Semantic weighted similarity matching (λ4); In order to determine the similarity between two KOs based on semantic similarity, we calculate their weighted similarity score (WS), see Figure 4. The WS score is determined by vectorizing the different attributed in the metadata and comparing their similarity based on calculating their cosine similarity. This allows for ranking besides exact matching, but finding phrases or information that are semantically related (e.g. cows and calves). On top of the similarity calculations, we take weights into account to differentiate the importance of the different metadata attributes (e.g. a KO's category is more important than its modality, so the category gets weighted higher).

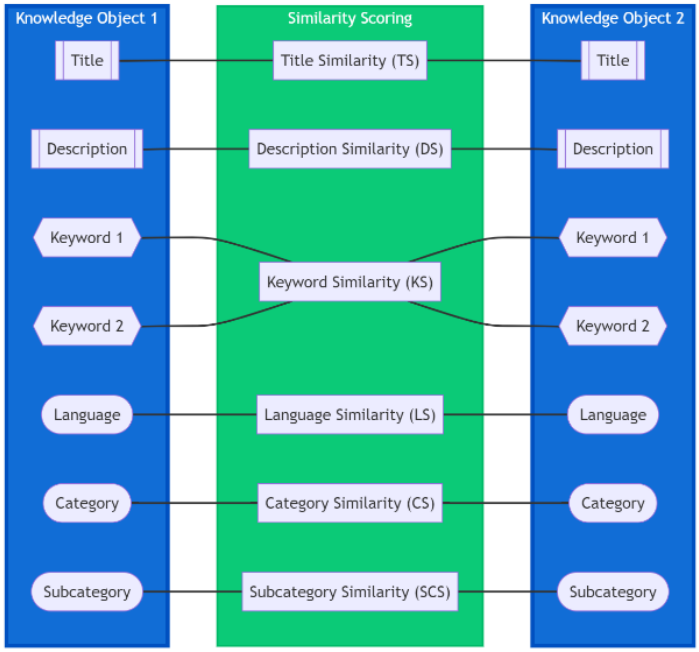


Figure 4: Similarity Scoring between KOs

Each of the above rankings are implemented in the system with the possibility to either *select* one approach (e.g. only λ4), *aggregate* approaches (combine the best results from (λ1, λ2, λ3, λ4), or *combine* the resulting scores of each of the approaches (λ1 + λ2 +λ3 + λ4). It is also possible to *mix* approaches (sometimes λ1, sometimes λ3, etc.) or to have these approaches act *supportive* of each other (initially λ1, if unsatisfying results switch to λ2, etc.).

By utilizing these different approaches of recommendation, we can take some of the user preferences into account. The top ranked similar knowledge is combined and provided

back to the user in a similar fashion as how we generate answers to the user's questions. This is the final part of the dialog flow, as seen in Figure 2: Dialog flow, before ending the conversation.

5. Preference elicitation

To provide personalized recommendations to the user, the chatbot must first learn what the user's needs are and what they are looking for. This can be difficult when not much is known about the user's characteristics, for instance in the user's initial interaction with the system, which is also referred to as the *cold-start problem*. To ensure a good user experience and to prevent losing the users in the first interaction with the chatbot, it is vital that the chatbot elicits the user's search preferences in a natural and user-friendly way. However, there is a lack in research on how people naturally describe their preferences (Radlinksi et al., 2019) and the design choices of conversational recommender systems are rarely discussed (Jannach et al., 2021).

To develop insight into how to design the initial interaction with the chatbot more effectively, we have set-up an experiment that investigates the effect of three different user preference elicitation methods on the user experience. The three methods vary in the amount of guidance from the chatbot, ranging from a more closed system where users are fully guided through the characteristics of the to-be-recommended items, to a very open system in which users can describe their preferences with minimal guidance. All conditions have hypothesized advantages and disadvantages (e.g. systems with a lot of guidance may be more straightforward but feel less natural; systems with less guidance give the user more freedom to express their preferences in their own words, but this is also harder for the chatbot to interpret correctly). Hence, this experiment will shed more light on the trade-offs of these different elicitation methods and will therefore help us to make decisions of when to apply which strategies in the dialog. This experiment is planned to take place in October 2023. The FarmBook ambassadors will be asked to participate in this online experiment taking approximately 15 minutes, and other individuals working in the agriculture and forestry domain from related Horizon Europe projects will be also asked to participate.

5.1. Personas

To better understand the potential end-users of the chatbot, the 18 personas developed by the EUREKA team were analyzed. This gives deeper insight into the motivations of different user types to use the FarmBook chatbot and also sparked ideas on how to personalize the dialogue and the interaction with the chatbot based on different characteristics (e.g. practical relevance of chatbot's responses based on user's role, language, assistance with chatbot based on user's digital literacy). These opportunities for personalization will be further investigated in the future, for instance by using a knowledge graph based on personas and matching new users to one of the sample personas for a better estimation of what they might be interested in when having little information available about themselves.

5.2. Ambassador meetings

To collect requirements for the chatbot, we started organizing online ambassador focus groups. The first focus group was aimed at getting more insight into the ambassador's past experiences with chatbots and to identify the first requirements for the FarmBook chatbot. We wanted to find out whether there are some recurring themes amongst the experiences of the ambassadors that could point towards functionalities we should (not) incorporate into the FarmBook chatbot. A summary of these experiences can be found in Figure 5 and the main requirements ambassadors brought up are displayed in Figure 6.

From the ambassador focus group it also became apparent that different user types may prefer to use the chatbot in different ways. Users who are expected to frequently use the FarmBook platform may prefer using a desktop version of the chatbot on the platform, while ambassadors mentioned that practitioners may prefer a simpler chatbot integrated in messaging apps such as WhatsApp or Facebook. These relations between the user types and specific use cases and preferred usage modes are an interesting point for further research, for which new personas might have to be created with an explicit focus on chatbot usage.

Furthermore, the attending ambassadors were asked to hand in some sample queries from their own work domain that they might want to ask the FarmBook chatbot into the future. This exercise shed light on some of the different types of questions potential users of the chatbot might want to pose. For instance, requests for receiving entire contributions were identified, but also requests for specific answers for best practices. Additionally, some "meta" questions were asked about the platform and the contributions, including questions about the purpose of the FarmBook platform and the creators of the contributions. Some queries were also related to receiving (technical) help with the chatbots, such as how to start a conversation with the chatbot. Further analysis of such sample queries will be conducted in the future to highlight which types of queries the chatbot should be able to handle.

In the future, more regular focus groups will be organized so that additional requirements can be identified and to ensure that the ambassadors are actively involved in the design and development of the chatbot.

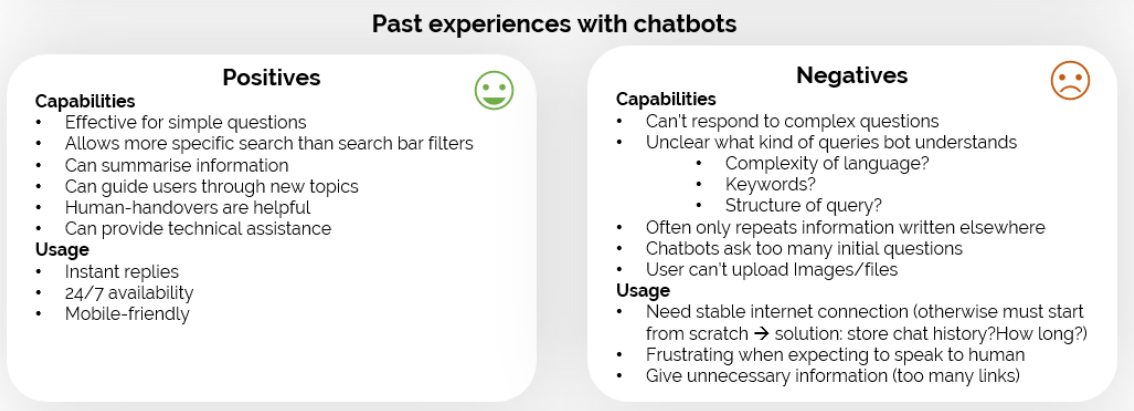


Figure 5: Summary of ambassador's experiences with existing chatbots

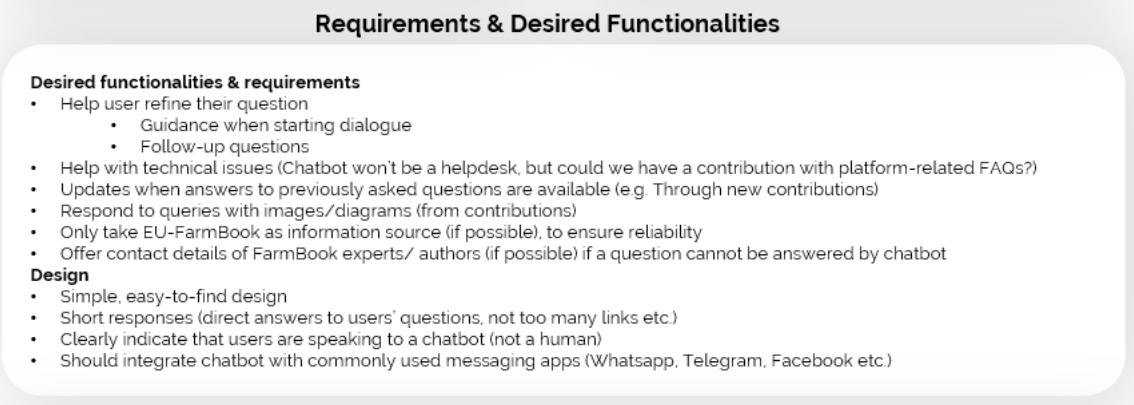


Figure 6: Summary of ambassador's requirements for FarmBook chatbot from focus group 1

6. Conclusion

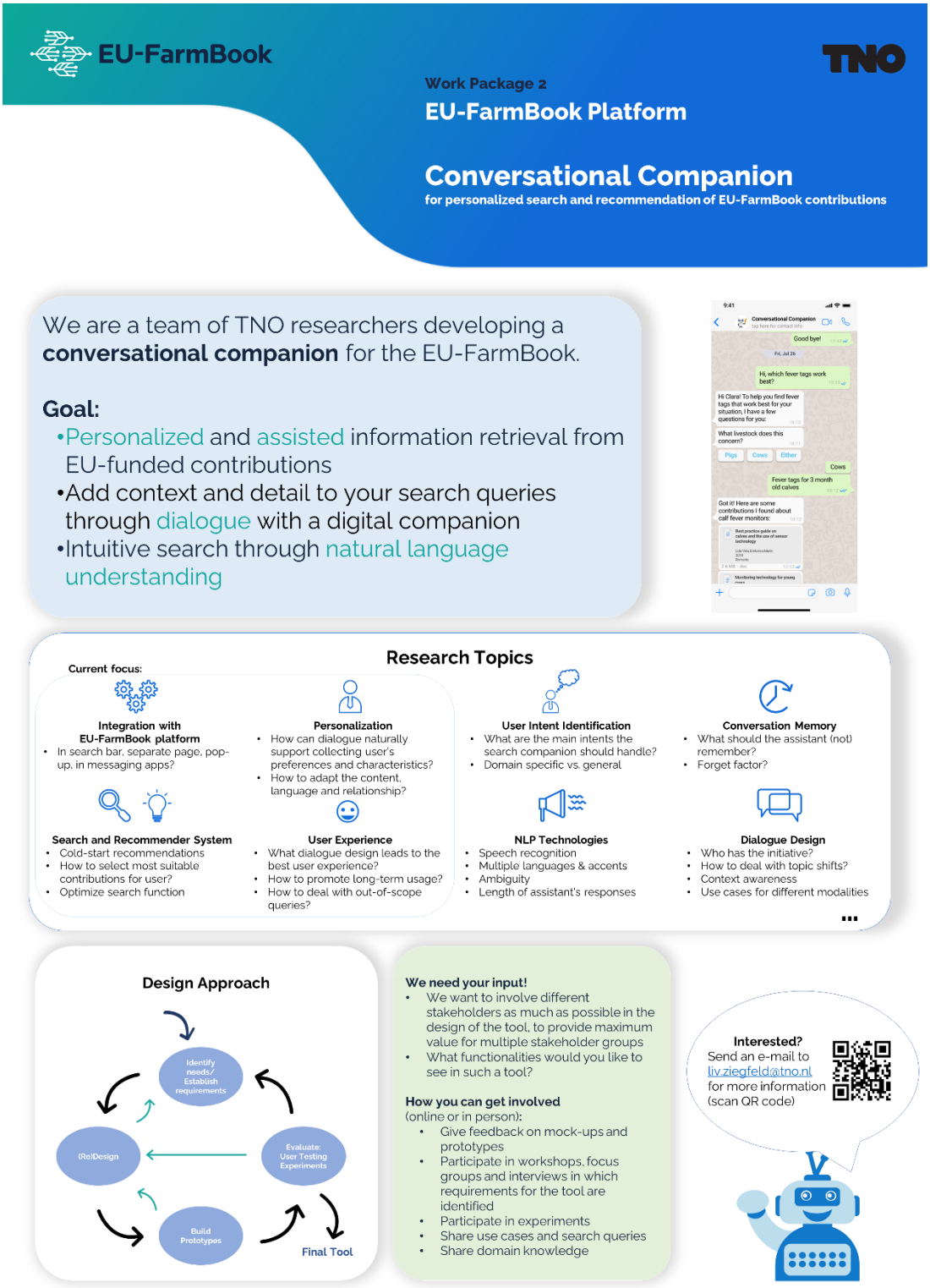
In this document we have described in detail the functioning of the FarmBook Search & Recommender system V1. Most notably, our presentation of the overall system architecture describes how the user interaction flow takes place through the conversational module. We explained how user preferences are elicited in order to make the user experience more personal, as well as the various ways in which the Search module operates our current Recommendation system.

This constitutes V1 of the system which will be implemented in the FarmBook MVP scheduled for September 2023. This will be the basis of extensive testing and refinements, upon which V2 of the system will be based.

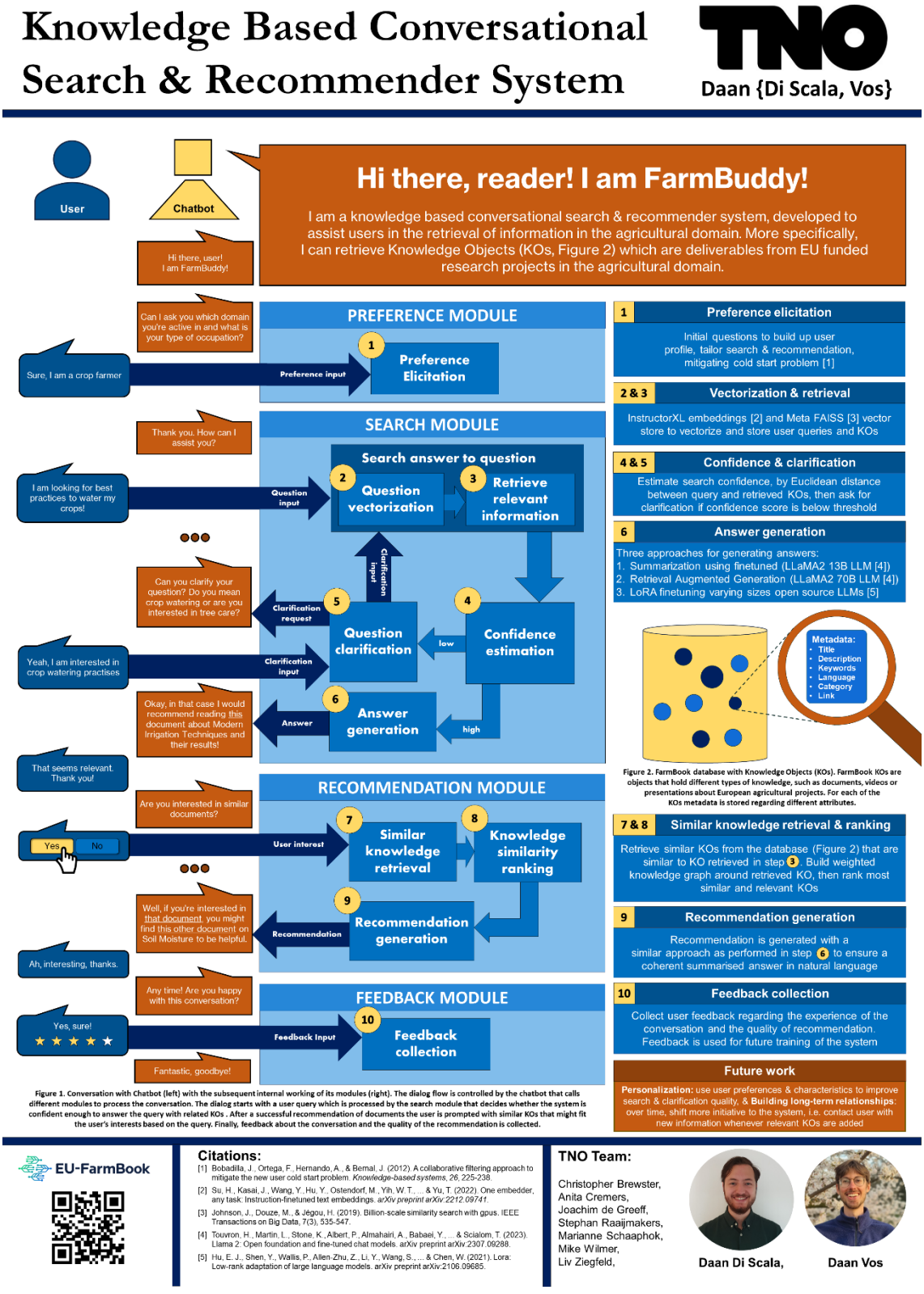
7. References

- Gliwa, B., Mochol, I., Biesek, M., & Wawer, A. (2019). SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization (pp. 70–79). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5409
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.
- Jannach, D., Manzoor, A., Cai, W., & Chen, L. (2021). A survey on conversational recommender systems. ACM Computing Surveys (CSUR), 54(5), 1-36.
- Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. IEEE Transactions on Big Data, 7(3), 535-547.
- Radlinski, F., Balog, K., Byrne, B., & Krishnamoorthi, K. (2019). Coached conversational preference elicitation: A case study in understanding movie preferences.
- Su, H., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., Yih, W. T., ... & Yu, T. (2022). One embedder, any task: Instruction-finetuned text embeddings. arXiv preprint arXiv:2212.09741
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-based systems, 26, 225-238.

8. Annex



- Poster presented at Tallinn Farmbook Consortium meeting June 2023 illustrating the functioning of the FarmBook Conversational Companion (using the system described in this Deliverable).



- Poster presented at CLIN September 2023 in Antwerpen illustrating the design of the Knowledge Based Conversational Search & Recommender System (this Deliverable).