



HAL
open science

Food KBQA Recommender: A knowledge base question answering system powered meal plan recommender

Felix Bölz, Diana Nurbakova, Audrey Serna, Sylvie Calabretto, Lionel Brunie,
Harald Kosch

► To cite this version:

Felix Bölz, Diana Nurbakova, Audrey Serna, Sylvie Calabretto, Lionel Brunie, et al.. Food KBQA Recommender: A knowledge base question answering system powered meal plan recommender. WI-IAT'24: The 23'rd IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology, Bangkok, Thailand, 9-12 December 2024, Dec 2024, Bangkok, Thailand. hal-04957127

HAL Id: hal-04957127

<https://hal.science/hal-04957127v1>

Submitted on 19 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Food KBQA Recommender: A knowledge base question answering system powered meal plan recommender

Felix Bölz

CNRS, UCBL, LIRIS, UMR5205 & DIMIS
Univ Lyon, INSA Lyon & University of Passau
Villeurbanne, France & Passau, Germany
felix.boelz@uni-passau.de

Diana Nurbakova

CNRS, UCBL, LIRIS, UMR5205
Univ Lyon, INSA Lyon
Villeurbanne, France
diana.nurbakova@insa-lyon.fr

Audrey Serna

CNRS, UCBL, LIRIS, UMR5205
Univ Lyon, INSA Lyon
Villeurbanne, France
audrey.serna@insa-lyon.fr

Sylvie Calabretto

CNRS, UCBL, LIRIS, UMR5205
Univ Lyon, INSA Lyon
Villeurbanne, France
sylvie.calabretto@insa-lyon.fr

Lionel Brunie

CNRS, UCBL, LIRIS, UMR5205
Univ Lyon, INSA Lyon
Villeurbanne, France
lionel.brunie@insa-lyon.fr

Harald Kosch

DIMIS
University of Passau
Passau, Germany
harald.kosch@uni-passau.de

Abstract—This study investigates the efficacy of a hybrid recommendation model for personalized meal plans, integrating Knowledge Base Question Answering (KBQA), Information Retrieval (IR), and Recommendation techniques. It utilizes the hybrid model to consider both different dietary preferences and nutritional requirements. In addition, it tries to bridge the gap between the recommender itself and its effectiveness in the real world by offering interfaces for integrating persuasion via explanation and gamification.

The findings contribute to extending knowledge about the development of food recommendation systems in constrained contexts. The system can address health awareness by considering user-defined constraints. However, in big use cases, it has issues with its scalability. Future work involves refining the data generation processes and exploring non-KBQA models for broader scalability and adaptability.

Index Terms—KBQA Recommendation, Information Retrieval, Health-aware Recommendation, Persuasive Systems, Food Recommendation

I. INTRODUCTION

There exists a rise in diet-related health issues, ranging from obesity and cardiovascular disease to nutrient deficiencies and eating disorders [1]. To tackle the problem of overweight and obesity, there are different solutions ranging from more education and behavior changes (like increased physical activity or changed nutrition) to surgeries for the most critical cases of obesity [2], [3]. The argumentation is that traditional dietary guidance methods, which offer generic advice and one-size-fits-all solutions, are insufficient for addressing the complex needs of individuals with diverse backgrounds, preferences, and health concerns. Recently, a bunch of healthy food recommender system based applications and approaches has been proposed in academia and industry (e.g. [4]–[10]). However, most food-related recommenders overlook crucial factors such as individual dietary preferences, nutritional requirements, and

psychological considerations which can make a big difference in effectiveness in the real world [6], [11], [12].

The use of recommender-based applications does not replace medical assistance. Thus, it is usually assumed, that the users have an accompanying nutritionist who can bring expertise to the proposed approach by tweaking its user-specific configuration. Otherwise, the user should still be able to interact with the system, but it falls back on general health guidelines suitable for an average user. It is within this context, that the proposed health-aware meal plan recommender tries to bridge the gap by combining traditional food recommendation with information retrieval and a Knowledge-Base Question-Answering system (KBQA) to offer a detailed and adaptable approach that fits the specific needs and preferences of each individual. To make use of rich food knowledge, we use a knowledge graph. To reason over it a KBQA system seems to be a prominent option. Our approach is a comprehensive extension of Chen’s [10] proposed KBQA system.

A. Requirements motivating the proposed approach

Traditional recommendation systems often struggle to deal with the complex aspects of dietary advice, missing important factors like personalized nutritional requirements and health concerns (e.g. allergies, diseases like diabetes). Also, as the goal is to trigger a *healthy* behavior change which is a gradual process, the recommendation of *just* well-fitting items might not help the users to change their behavior but more likely to nudge them into behaving like previously. Thus, the focus should rather be on the health-related aspects than on the recommender’s accuracy. Moreover, in the context of the target audience of overweight patients having an accompanying nutritionist, it is important to include external user-specific

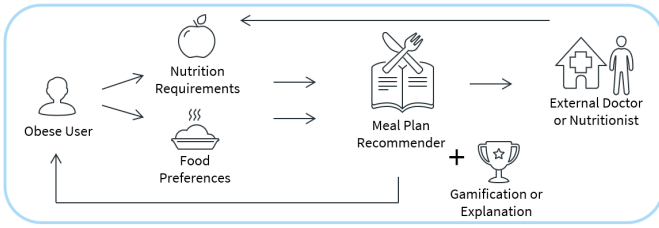


Fig. 1. Use case of an obese or overweight user using the proposed KBQA Recommender System. The user has two distinct restrictions, namely his *Nutrition Requirements* and *Food preferences*. An accompanying *Nutritionist* may also influence the nutritional requirements. The recommended item is a *Meal Plan* consisting of recipes for each meal type. To increase its effectiveness, persuasion techniques like *Gamification* or *Explanations* might be added to the recommendation.

medical and nutritional expertise. Those arguments motivate the following requirements, also shown in Fig. 1.

- *User Preferences*: Incorporating users’ general food preferences, ingredient likes and dislikes, as well as users’ nutritional preferences spanning low, medium, and high levels. The motivation behind those levels is to provide a simplified option to interact with nutrition criteria for users with low nutrition-related education. The levels are set before training the KBQA system and are defined per dish per 100g.
- *Nutritional Guidelines*: Following predefined guidelines including specific nutrient ranges established via specific diets or in collaboration with the users’ accompanying nutritionist. Guidelines refer to the nutritional values of one serving of a recipe.
- *Meal Plans*: Transitioning from singular recipe recommendations to holistic daily meal plans, fulfilling all dietary requirements and accommodating prescribed calorie distributions per meal type. The distributions, as guidelines, can either be user-defined or follow general recommendations from related works. [13]
- *Real-world effectiveness*: Improving likeliness of behavior change via explanation and gamification.

B. Research Questions

To guide the exploration and evaluation, two research questions are formulated as follows. (1) How does the KBQA recommender perform compared to a traditional recommender on the same data set? and (2) How well does the proposed hybrid recommendation model meet the different dietary preferences and nutritional needs of users?

C. Contributions

By addressing both questions the research offers the following contributions.

- *Integration of a Hybrid Model*: Combining knowledge base question answering (KBQA), information retrieval (IR), and recommendation techniques to create a personalizable and adaptable recommendation system.
- *Health-awareness*: Enabling personalized nutrition and expert knowledge integration by offering configuration

settings like different nutrition levels or more fine-grained options.

- *Mitigation of Data Sparsity*: Employing threshold discarding and user clustering to enhance the relevance and accuracy of recipe recommendations.
- *Meal Plan Recommendation*: Recommending fitting recipes for meal types (*Breakfast, Lunch, Dinner, Snacks*) while still respecting nutritional guidelines and requirements.
- *Enhancing Effectiveness through Persuasion*: Introducing conceptual persuasive explanation and gamification elements to augment the efficacy and user engagement of the meal plan recommendations.

The remainder of the paper is structured as follows. In Section II, a brief overview of related works is provided. Section III introduces the proposed approach. Section IV showcases the performed experiments and the evaluation, while Section V concludes the paper and gives a brief outlook on future works. All resources are available through our GitLab¹ repository.

II. RELATED WORK

Existing research on healthy food recommendation [4]–[9] has primarily focused on personalized recipes or diet suggestions. However, many of these systems overlook crucial health factors like allergies and nutritional needs and often lack consideration of comprehensive food knowledge. Also, context information originating from smart devices or food- or mood-related diaries could be taken into consideration [6]. Neglecting these aspects can not only lead to a loss of trust in the system but also pose potential risks to users’ physical and mental well-being [10], [14].

The definition of *healthy food* varies across studies. Some emphasize calorie intake [8], while others consider factors like cholesterol levels [9] or a combination of nutrients such as protein, sodium, cholesterol, and saturated fats [15]. Recent approaches also incorporate guidelines from health organizations like the World Health Organization (WHO) [16] or national agencies to classify recipes based on their overall healthiness measured by nutrition scores [17]. However, these general scores alone lack personalization and fail to address individual health requirements effectively. Additionally, there have been attempts to focus on particular groups with specific health concerns, like those with thyroid problems, diabetes, or high blood pressure [18]–[20].

Several challenges persist in healthy food recommendation systems. The cold-start problem and data sparsity issues, where insufficient user data prevents (good) personalized recommendations, remain an issue. Solutions include collecting additional food diary data or using content-based knowledge [8], [10]. Evaluation metrics are another challenge, with existing approaches often focusing solely on recommendation accuracy rather than healthiness [21]. Additionally, promoting behavior change towards healthier eating habits requires more

¹<https://gitlab.com/felix134/food-kbqa-recommender>. The README.md file provides additional technical information regarding the data and steps to follow to reproduce the shown results, figures, and tables.

than just recommending recipes and demands the implementation of persuasion strategies and long-term user studies for an effective evaluation [15], [22]–[24].

Chen et al. have proposed *PFoodReq* [10], a *Knowledge Base Question Answer System*, allowing a user to ask for recipe recommendations with additional constraints like nutritional requirements and returning the best-fitting recipes. *PFoodReq* uses BAMnet [25] as the underlying KBQA system but the approach itself is claimed to be independent from a QA model. Chen et al. consider explicit user requirements incorporating health factors like allergies and nutritional needs. Though the proposed approach seems to be quite prominent in terms of dealing with a number of constraints, it faces several limitations and shortcomings. First is the lack of diverse nutrition requirements, as they only support the definition of *fat*, *protein*, and *carbohydrate* requirements. It is unclear, whether their approach is extendable to support a higher variety and multiple concurrent nutritional requirements. Also, their approach is a one-shot recommendation, meaning that only the best n -fitting recipes are recommended to the user. Lastly, the persuasion aspect is not in the scope of their work as a general behaviour change process is not the aim.

III. THE FOOD KBQA RECOMMENDER

The proposed *Food Knowledge-Base Question-Answer Recommender* is based on Chen’s *PFoodReq* [10]. This heavily modified and extended approach is used as a basis for the proposed meal-plan recommendation approach as described in the following.

To start with, a **running example** is established:

Bob, an overweight user, has a prescribed diet given by his doctor, that limits the daily calorie intake to 2000 kcal and the maximal allowed amount of sugar per meal to 10 g. Also, he should avoid *unhealthy* food. Moreover, due to an allergy, peanuts must never be part of any meal. In addition, Bob wants to limit his sodium intake to a *low* level. Those mentioned requirements are valid for all recommendations. For the current day, the user already had his breakfast and needs a meal plan for lunch, dinner, and potentially snacks. Also, that day, he felt like having dinner with something baked and bread.

A. Integration of a health-aware Hybrid Model

The architecture of the *Food KBQA Recommender* is depicted in Fig. 2, also referring to the already set-up running example. To recommend such a restrained and personalized meal plan, the question-answer dataset generation and the learning process of Chen et al. [10] is extended as follows.

The training set consists of multiple question-answers pairs, where a question asks for a certain constrained recipe, and the answers are a set of valid recipes. The KBQA model is however not trained for recommending meal plans directly (those are constructed later on), but for recommending recipes.

1) *QA Dataset Preliminaries*: Thus to construct the question-answer dataset, the system expects a set of *Templates*, *Training Constraints* limiting the possible answers, and a *Knowledge Base* containing recipes including nutrition and ingredient information as well as *Tags*.

a) *Templates*: Templates suggested in [10] are inspired by social media platforms like *Reddit*. We have modified them in such a way, that multiple nutrition requirements in parallel can be generated.

b) *Training Constraints*: The *Training Constraints* are described in more detail in Fig. 3. The goal is to populate the training set with the same types of requirements expected later in the automated recommendation step. So referring to the running example, when it is expected that multiple nutrient types might be restricted together (low levels of sodium, certain sugar, calories, and NutriScore ranges), such behavior must be trained for and thus be present in the dataset. Intuitively, the added concurrent restrictions the system must handle and be trained for might either explode the size of the to-be-generated dataset or decrease performance/accuracy. The question is, whether this increased complexity is already noticeable in the use case described in this paper.

To get an overview, the constraints are divided into the following categories: 1) nutrition constraints, 2) nutrition scores, 3) guideline constraints, 4) ingredients, 5) tags.

Nutrition constraints refer to certain limits for each recipe per 100g per serving. The actual limits are defined before the training step and are labeled as either *low*, *medium*, or *high*. In the given example *low sodium* is of the current category. The intuition behind this is, that every user, regardless of his/her nutritional education, can easily configure the recommender nutrition-wise. For the training step, none or multiple nutrients and levels are chosen to limit each question-answers pair.

Nutrition Scores are utilized to approximate the recipes’ healthiness into one score. Among multiple problems such a generalization approach brings, the goal is to coarsely differentiate between very healthy, very unhealthy, and in-between recipes. In the example *NutriScore of minimal C²* is set to avoid *unhealthy* recipes altogether. An additional idea is, that at some point the system just slightly increases the nutrition score, such that the user automatically receives slightly healthier recommendations to nudge him/her into the desired direction. Available scores are the *NutriScore*, the *WHO Score*, and the *FSA Score* as given by the underlying Knowledge Graph [27].

Guideline constraints refer to the nutritional values of one whole serving. This option is meant for users who know their nutritional needs (e.g. a diet or instructions from their nutritionist). In the running example, those are given by the ranges of sugar and calories.

The remaining constraints are *Ingredients* and *Tags*. For example, disliked Ingredients (e.g. peanuts) are used to limit for allergies.

²The NutriScore [17], [26] is a 5-colour nutrition label ranging from A (the best) to E (the worst) and rating the overall nutritional value of food products.

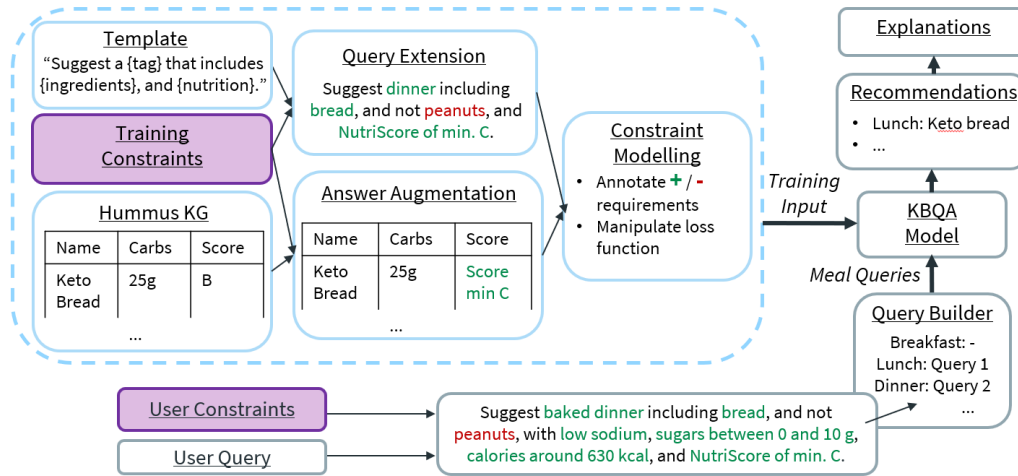


Fig. 2. KBQA Recommender Architecture: The components in the blue dashed box show the training process, while outside components represent the recommendation process. Components with a filled purple background refer to constraints used for model training and recommendation and are described in more detail in Fig. 3 and Fig. 4.

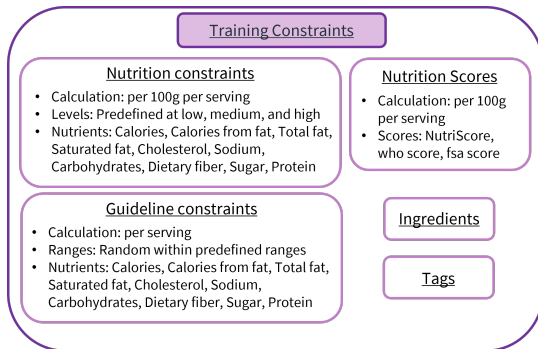


Fig. 3. Constraints shown in this figure might be present in each question-answers pair used for training. The actual constraints and their number of occurrences are chosen randomly respecting a configuration set before training. Among Nutritional motivated constraints (*Nutrition constraints*, *Guideline constraints*, *Nutrition Scores*) the training data also includes the recipes' *Ingredients* and *Tags*.)

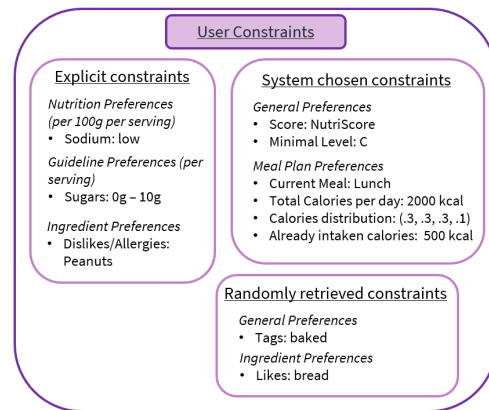


Fig. 4. This figure shows constraints used for formulating the user queries for the recommendations. *Explicit constraints* vary from user to user and focus on the health context. They can directly be set by the user. The *System chosen constraints* are indirectly influenced by the user and depend on the context the user is in. *Random retrieved constraints* originate from the user's previous interactions and serve to ask for recipes the user might be interested in.

c) *Knowledge Base:* The architecture (Fig. 2) expects a *Knowledge Base* containing recipes with their nutritional information, nutrition scores, ingredients, and tags as input for the training step. For our approach, we use *HUMMUS KG* [27] as it contains the needed data.

2) *QA Generation:* The QA generation is done similarly to *PFoodReq* [10] system. For each question-answers pair, a random *Template* is populated with random *Training Constraints* at the *Query Extension* step, serving as a question. The set of possible answers is generated by querying the *Knowledge Base* with the *Training Constraints* in addition to the *Answer Augmentation*. This process modifies the answer graph (which contains the possible recipes as answers) by generalizing nutrition-related requirements to the given constraints.

For example, suppose the requirement is to have a recipe with a minimal NutriScore of *C* and a candidate answer is of score *B*. In that case, this answer in the scope of the question is generalized to *minimal NutriScore C*.

3) *Constraint Modelling:* Both the query and the answers are internally annotated in the *Constraint Modelling* step to modify the *KBQA Models'* loss function, especially for negative constraints like allergies or disliked ingredients. This *loss function* is taken as-is from Chen [10]. The result is a training (respectively testing or validation) set with the mentioned annotations to interact with the *KBQA Model*.

4) *Query Builder:* The recommender now assumes a trained *KBQA Model* and *User Constraints* to generate queries for meal plan recommendations. In Fig. 4, the distinction is made depending on how the constraints are chosen for the recommendation task. The *Explicit constraints* include all nutritional constraints that a user might set explicitly by himself/herself to personalize his/her recommendations. They include the coarse nutrition levels, the more specific

TABLE I
DISTRIBUTION OF CALORIES ACROSS EATING OCCASIONS FOR ADULTS (19-59 Y.O.) IN TARGET CALORIE LEVEL (KCAL) AND PERCENTAGE OF TOTAL CALORIES (%) [28].

Total kcal	Breakfast kcal	Breakfast %	Lunch kcal	Lunch %	Dinner kcal	Dinner %	Snacks kcal	Snacks %
2000	440	22	640	32	640	32	280	14

guidelines, and ingredient dislikes or allergies. Exceptions are nutrition scores, and calorie guidelines, which are part of the *System chosen constraints*. Those parameters are set by the system depending on context information like the users' current meal type, and a current calorie budget.

The calculation of the budget can be configured for each user and depends on a calorie distribution in regards to different meal types (refer to Table I), already intaken calories, and the daily calorie limit estimated by the user's personal information like BMI and activity level. Lastly, *Randomly retrieved constraints* e.g. tags and liked ingredients are used to find well-fitting recipes regarding the users' flavor preferences. The list of possible candidates depends on the user's previous interactions and highly influences the accuracy of the final recommendations.

Depending on the actual *User Constraints* the *Query Builder* generates queries for all not-already consumed meal types of the day. Alternatively, the user directly formulates a query, skipping the *Query Builder* entirely.

5) *KBQA Model*: All queries are processed with the already trained *KBQA Model* and result in multiple sets of ranked recommendations.

6) *Recommendation*: Each meal type has an independent set of recommendations that is valid within the corresponding calorie limit for the given meal type.

7) *Explanation*: For each recommendation set *Explanations* can be generated using the query contents directly, or the KG connections between a current user, its user-recipe interactions, the recommended recipes, and the query contents.

B. Mitigation of Data Sparsity

The underlying *HUMMUS KG* [27] is relatively sparse increasing the difficulty to deploy a model with an acceptable accuracy. In combination with threshold discarding strategies (which remove users and recipes with a too small number of interactions), *SKlearn's MiniBatchKMeans*³ rating-based clustering is added. Thus, the overall data set density is increased by resetting all user-recipe interactions to the clusters (in a way *super-users*). This issue has not been tackled in [10].

C. Addressing the Cold Start Problem

One well-known issue for recommender systems is the cold start problem, meaning that when new users appear in the system history and context data are missing on which the system relies. A standard solution is to recommend new users popular items until enough additional data is gathered such

³<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html>

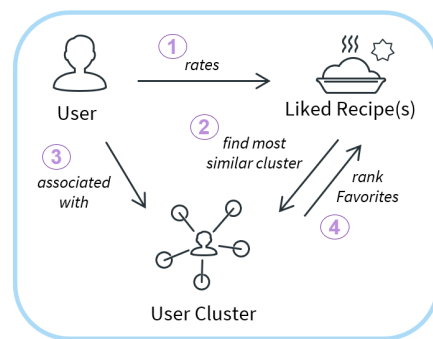


Fig. 5. Heuristic to find similar user clusters for a new user.

that the systems' knowledge of a user is sufficient. In this scope, the data includes the users' weight, height, and physical activity profile to calculate the *Body Mass Index* (BMI) and to approximate the users' daily intake of calories, used for the recommended meal plans later on. Also, interaction data is mandatory in the design. Thus, each new user must initially like or dislike at least 10 recipes. The collected interaction data is used to link the new user with a cluster (or superuser) from the data set used for training.

To efficiently find a similar cluster, the user is not requested to rate arbitrary recipes, a heuristic shown in Fig 5 is applied. First, the most popular and frequent recipe must be rated; this step remains the same for each new user as long as the underlying data set and thus the trained model does not change. For each consecutive step, the cosine similarity of the current user's and all clusters' recipe interactions is calculated. The next to-be-rated recipe is among the favorite recipes of the most similar cluster. In the end, the user will be linked with the overall most similar cluster. Until such a user is appended to the training set or re-linked to a better fitting cluster in the future, all following recommendations are based on the linked cluster and the user's data e.g. nutrition requirements.

D. Enhancing Effectiveness through Persuasion

To increase the effectiveness, multiple strategies to persuade the user to follow the recommendations are implemented.

Persuasion by explanation assumes that a more educated user might be more open and likely to follow the given advice. As the recommendation process includes an information retrieval part and a usage of a KBQA System model, a lot of context information is already involved. Explicitly used information for the query generation like calorie limits, used tags, ingredients, and all additional constraints can directly be used for the explanation. Also, the original KG can be utilized by finding paths between the current users' clusters, their interactions, and the information used by the query. The presentation of those explanations is only given in textual form by using templates. Referring to the running example, Table II shows an excerpt of possible explanations.

As shown in [29], the visualization of explanations affects the effectiveness and could be extended further.

TABLE II
EXAMPLES OF TEXT EXPLANATIONS THAT ARE GENERATED FROM THE USER-SPECIFIED REQUIREMENTS.

Type	Explanation
Rating based	We chose this recipe for you because similar users like you gave this recipe an average rating of 4.7.
Ingredient based	We chose this recipe for you because it contains bread which you asked for.
Nutrition based	We chose this recipe for you because it has a sugar amount of 7g.
Tag based	We chose this recipe for you because it is a dinner recipe.
Hybrid/Combined	We chose this recipe for you because it does not contain peanuts, has a calorie amount of 400kcal, and is a baked recipe.

Persuasion by gamification refers to elements that try to awaken a user’s desire to play with fun parts of the system, increasing motivation and likeliness to follow the system’s advice. Most gamification elements are independent of the underlying system like the persuasion method *Progress Tracking* (of e.g. weight) or *Achievements*. However, other elements can be directly integrated with the presented approach. *Challenges* can be generated similarly to the user’s other recommendations. For example, eating a meal without sugar would require the system to add the guideline constraint of 0g *sugar* on query construction.

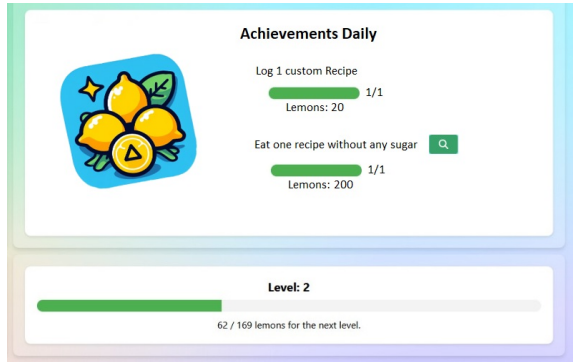


Fig. 6. Exemplified gamification page combining *Rewards (Lemons)*, *Progress Tracking*, *Achievements*, and *Challenges*.

Rewards can be granted to the user by some kind of currency, like for example shown in Fig. 6. The reward can be used to change the system behavior by, for example, being less restrictive in terms of required nutrition scores or adding ingredients or tags that are associated with rewards like chocolate to the query.

As a last example, also refer to Fig. 7, the persuasive effect of giving or spreading *Likes* can be directly integrated into the initial task of recipe rating. For example, the rating process could be similar to the swiping mechanic which is used by *Tinder*.

IV. EXPERIMENTS

Experiments to explore the model’s accuracy, capability of learning the nutrition feature, and scalability are conducted.



Fig. 7. Exemplified *swipe* mechanic for rating recipes.

Table III shows the proposed approach in comparison to *PFoodReq* baseline [10] (replicated by using their original approach but with the *HUMMUS KG* instead of the *FoodKG* as data base). All approaches share the same hyper-parameters with the baseline but differ in the previously described modifications and data set configurations. To get an understanding of the scalability limits, three models of different training sizes (*Number of Questions*) are compared. The *Number of Tags* is similar to the baseline because both data sets share the majority of the scalability limits, three models of different training sizes (*Number of Questions*) are compared. The *Number of unique Recipes* ranges from approximately 100 to 10 000 which is a direct result of the dataset sizes. It refers to all unique recipes in the answer set of the training data, not the actual number of recipes. However, when comparing the relations between the number of questions and recipes between all approaches, it is apparent that the approaches have way fewer unique recipes in the training set. The reason is the *Average Requirements per Question*, meaning the average number of keywords that influenced the answer set of the training data. As those requirements, caused by the extensions and modifications, are double as high as for the baseline, it heavily limits the number of valid answers (recipes) for each question. Regarding the evaluation results, *MAP* seems to be relatively independent of the model’s size and is similar to the baseline. However, especially the *Average Precision* (and thus the other measures) decreases when changing the models’ dataset size.

As mentioned previously, the extension’s goal is to have a higher degree of influence on the final recipe recommendations. This includes the possibility of asking for specific nutrient ranges, coarser nutrition levels, ingredients, tags, and nutrition scores. To give an incentive for one of those, Fig. 8 and Table IV show the health improvement of recommendations when asked for a better (in this case higher) nutrition score. Unfortunately, the improvement is marginal and the underlying KBQA model seems to have trouble learning the hereby used *NutriScore* levels *low* and *high*.

TABLE III

RESULTS OF THE EXPERIMENTS IN COMPARISON TO THE REPLICATED BASELINE [10]. IT SHOWS THE TRAINING DATASET’S PROPERTIES *Number of unique Recipes (#Rec.)*, *Number of Questions (#Ques.)*, *Number of Tags (#Tags)*, *Average Requirements per Question (Avg. Req.)* AND EVALUATION METRICS *Mean Average Precision (MAP@10)*, *Mean Average Recall (MAR@10)*, *Average Precision per question (Avg. P.)*, *Average Recall per question (Avg. R.)*, and *Average F1-score per question (Avg. F1)* AS IMPLEMENTED IN THE *recmetrics* library.

Description	#Ques.	#Tags	#Rec.	Avg. Req.	MAP@10	MAR@10	Avg. P.	Avg. R.	Avg. F1
<i>PFoodReq</i>	4.4k	215	9.5k	2.25	45.46%	45.06%	42.33%	56.50%	42.78%
Food KBQA (S)	1.3k	207	0.1k	4	53.13%	41.43%	53.13%	41.43%	42.10%
Food KBQA (M)	8.2k	317	1k	3.9	43.16%	26.15%	18.13%	31.71%	15.00%
Food KBQA (L)	18.9k	317	10k	4.4	49.94%	26.83%	7.85%	30.53%	7.18%

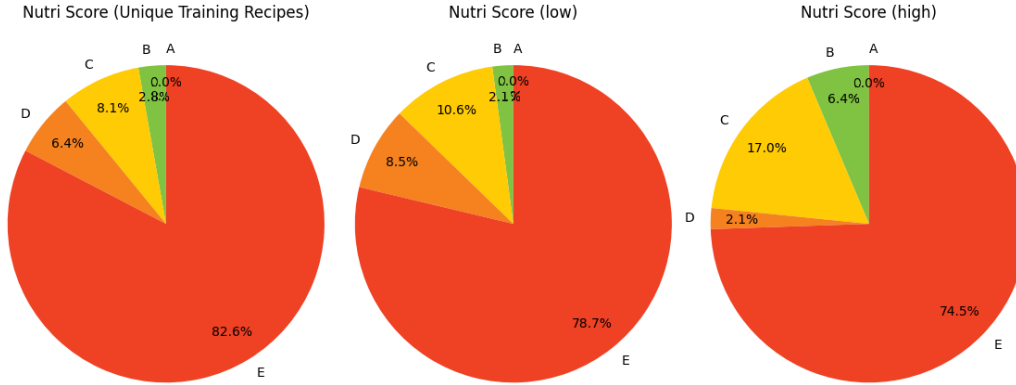


Fig. 8. *NutriScore* distributions for the KBQA (L) model: The first chart shows the distribution of the scores in the training set (unique recipes in the answers). The Second one shows the distribution of the scores for the recommendation with no constraints while the last with the constraint of a higher score.

TABLE IV

AVERAGE *NutriScore* IN THE TRAINING DATA SET (*Training*), WHEN USED FOR RECOMMENDATION WITH A LOW *NutriScore* (*Rec.*), AND FOR CONSTRAINED RECOMMENDATION WITH A HIGH *NutriScore* (*Const. Rec.*). A LOW *NutriScore* IS DEFINED BY ALLOWING ALL CATEGORIES FROM A TO E (BECAUSE IT IS ASSUMED THAT IT IS NOT EXPLICITLY ASKED FOR AN UNHEALTHY RECIPE BUT INSTEAD FOR ANY RECIPE), WHILE THE HIGH *NutriScore* ONLY INCLUDES CATEGORIES A AND B. THE PERCENTAGE IMPROVEMENT BETWEEN THE LOW AND HIGH CATEGORIES IS GIVEN IN THE LAST COLUMN. THE SCORE IS NORMALIZED BETWEEN 1 AND 0 WHILE THE LATTER IS THE *unhealthiest* POSSIBLE SCORE.

Description	Training	Rec.	Cons. Rec.	Improvement
Food KBQA (S)	5.6%	3.3%	3.8%	12.8%
Food KBQA (M)	8.2%	7.5%	8.8%	16.7%
Food KBQA (L)	8.5%	5.3%	8.1%	52.9%

A. Discussion

In terms of *MAP* the proposed model performs similarly to *PFoodReq* [10] baseline. However, there are inherent problems that limit the usability of the proposed approach. First, the increased complexity in terms of additional requirements decreases the accuracy (average *Precision*, *F1*) in comparison to the baseline. Second, the training size also negatively influences the accuracy. Those two issues influence the overall scalability of the approach by limiting either the size or the number of requirements the model supports. Assuming having small organizations like a local nutrition clinic that only provides or requires small data sets, the proposed approach still seems to be promising. Third, the model can learn the proposed additional constraints, but the effect on the nutrition score is far smaller than expected. The reason is, that all

requirements function as soft constraints. In addition, the way of the question-answer set generation, especially the unbalanced recommendation answers in terms of nutrition score distribution, produces an unbalanced data set. Fourth and lastly, the KBQA model also requires a set of constraints (or keywords) to recommend meal plans. For a manual recommendation, the user can define his or her query freely, but it is unclear how to optimally extract and choose user preferences like ingredients or tags to get a good recommendation.

V. CONCLUSION AND FUTURE WORK

The presented KBQA system offers additional configuration options useful for some use cases, the automatic query generation, the meal-plan generation, support for persuasion, and the support of nutrition scores. It is assumed that a user with or without the help of an accompanying nutritionist can formulate health-aware constraints. The system incorporates a set of additional constraints like coarse nutrition levels (*high*, *medium*, *low*), specific nutrition ranges, and nutrition scores. The approach is feasible for smaller use cases but cannot be scaled to larger datasets without reducing the supported additional constraints. Also, especially problematic in the food domain, the system only works with soft constraints which might be desirable for some categories like for example *liked ingredients* but not for others like *disliked ingredients* or *allergies*. The last issue is the unexpected minor improvement in nutrition score when explicitly asking the model for it.

In cases where a KBQA system is not necessarily required, it might be more promising to directly use the presented ideas

(especially the additional constraints) on a standard recommender by influencing its rewards for each user. Moreover, motivated by the lack of scalability, using Large Language Models (LLMs) like *Retrieval Augmented Generation* (RAG) are a promising direction. Another point is, that especially aspects like persuasion including explanations and gamification, and its possible change in the recommendations' effectiveness in the real world need to be evaluated by real users in the future. Lastly, the by the system already envisioned inclusion of medical experts like doctors or nutritionists should be evaluated as well in an online setting.

REFERENCES

- [1] World Health Organization. (2023) Obesity info page of the world health organization. [Online]. Available: <https://www.who.int/health-topics/obesity>
- [2] ——. (2003) Global Strategy on Diet, Physical Activity and Health. [Online]. Available: <https://www.who.int/publications/i/item/9241592222>
- [3] J. L. Colquitt, K. Pickett, E. Loveman, and G. K. Frampton, "Surgery for weight loss in adults," *Cochrane Database of Systematic Reviews*, vol. 2014, no. 10, aug 2014.
- [4] N. Jia, J. Chen, and R. Wang, "An attention-based convolutional neural network for recipe recommendation," *Expert Systems with Applications*, vol. 201, p. 116979, Sep. 2022. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0957417422004043>
- [5] X. Gao, F. Feng, H. Huang, X.-L. Mao, T. Lan, and Z. Chi, "Food recommendation with graph convolutional network," *Information Sciences*, vol. 584, pp. 170–183, Jan. 2022. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0020025521010549>
- [6] W. Min, S. Jiang, and R. Jain, "Food Recommendation: Framework, Existing Solutions, and Challenges," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2659–2671, oct 2020.
- [7] M. Rostami, M. Oussalah, and V. Farrahi, "A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering," *IEEE Access*, vol. 10, pp. 52 508–52 524, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9775081/>
- [8] J. M. Rehg, S. A. Murphy, and S. Kumar, *Towards Health Recommendation Systems: An Approach for Providing Automated Personalized Health Feedback from Mobile Data*. Cham: Springer International Publishing, 2017. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-51394-2>
- [9] A. Starke, C. Trattner, H. Bakken, M. Johannessen, and V. Solberg, "The cholesterol factor: Balancing accuracy and health in recipe recommendation through a nutrient-specific metric," *CEUR Workshop Proceedings*, vol. 2959, pp. 1–11, 2021.
- [10] Y. Chen, A. Subburathinam, C.-H. Chen, and M. J. Zaki, "Personalized Food Recommendation as Constrained Question Answering over a Large-scale Food Knowledge Graph," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, mar 2021, pp. 544–552.
- [11] A. Kumar, P. Tanwar, and S. Nigam, "Survey and evaluation of food recommendation systems and techniques," in *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*. India: Bharati Vidyapeeth, New Delhi as the Organizer of INDIACom - 2016, 2016, pp. 3592–3596. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7724932>
- [12] D. Elsweiler, C. Trattner, and M. Harvey, "Exploiting Food Choice Biases for Healthier Recipe Recommendation," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, aug 2017, pp. 575–584.
- [13] S. P. Murphy, A. L. Yaktine, and C. W. Sutor, *Child and Adult Care Food Program*. Washington, D.C.: National Academies Press, may 2011.
- [14] W. Yue, Z. Wang, J. Zhang, and X. Liu, "An Overview of Recommendation Techniques and Their Applications in Healthcare," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 701–717, apr 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9358461/>
- [15] R. Yera Toledo, A. A. Alzahrani, and L. Martinez, "A Food Recommender System Considering Nutritional Information and User Preferences," *IEEE Access*, vol. 7, pp. 96 695–96 711, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8765311/>
- [16] World Health Organization, "Healthy diet," Apr. 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/healthy-diet>
- [17] Agence Nationale de Santé Publique, "Nutri-Score," Jan. 2023. [Online]. Available: <https://www.santepubliquefrance.fr/determinants-de-sante/nutrition-et-activite-physique/articles/nutri-score>
- [18] V. S. Vairale and S. Shukla, "Recommendation of food items for thyroid patients using content-based knn method," *Lecture Notes in Networks and Systems*, vol. 132, pp. 71–77, 2021.
- [19] D. Mogaveera, V. Mathur, and S. Waghela, "e-health monitoring system with diet and fitness recommendation using machine learning," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021, pp. 694–700.
- [20] V. M. Deshmukh and S. Shukla, "Content-Restricted Boltzmann Machines for Diet Recommendation," in *Data Science and Security*, S. Shukla, A. Unal, J. V. Kureethara, D. K. Mishra, and D. S. Han, Eds. Singapore: Springer Singapore, 2021, vol. 290, pp. 114–121. [Online]. Available: https://link.springer.com/10.1007/978-981-16-4486-3_12
- [21] D. Li, M. J. Zaki, and C.-h. Chen, "Health-guided recipe recommendation over knowledge graphs," *Journal of Web Semantics*, vol. 75, p. 100743, Jan. 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1570826822000282>
- [22] A. El Majjodi, A. D. Starke, and C. Trattner, "Nudging Towards Health? Examining the Merits of Nutrition Labels and Personalization in a Recipe Recommender System," in *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. Barcelona Spain: ACM, Jul. 2022, pp. 48–56. [Online]. Available: <https://dl.acm.org/doi/10.1145/3503252.3531312>
- [23] R. Orji, L. E. Nacke, and C. Di Marco, "Towards Personality-driven Persuasive Health Games and Gamified Systems," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Denver Colorado USA: ACM, May 2017, pp. 1015–1027. [Online]. Available: <https://dl.acm.org/doi/10.1145/3025453.3025577>
- [24] J. Marshall, P. Jimenez-Pazmino, R. Metoyer, and N. V. Chawla, "A Survey on Healthy Food Decision Influences Through Technological Innovations," *ACM Transactions on Computing for Healthcare*, vol. 3, no. 2, pp. 1–27, Apr. 2022. [Online]. Available: <https://dl.acm.org/doi/10.1145/3494580>
- [25] Y. Chen, L. Wu, and M. J. Zaki, "Bidirectional Attentive Memory Networks for Question Answering over Knowledge Bases," in *Proceedings of the 2019 Conference of the North*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 2913–2923.
- [26] J. Chantal and S. Hercberg, "Development of a new front-of-pack nutrition label in France: the five-colour Nutri-Score," *Public health panorama*, vol. 03, no. 04, pp. 712–725, 2017.
- [27] F. Bölz, D. Nurbakova, S. Calabretto, A. Gerl, L. Brunie, and H. Kosch, "HUMMUS: A Linked, Healthiness-Aware, User-centered and Argument-Enabling Recipe Data Set for Recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023*, vol. 1, no. 1. New York, NY, USA: ACM, sep 2023, pp. 1–11.
- [28] S. Murphy, A. Yaktine, C. West Sutor, and S. Moats, "Institute of medicine (us) committee to review child and adult care food program meal requirements," 2011.
- [29] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor, "Personalized explanations for hybrid recommender systems," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, vol. Part F1476. New York, NY, USA: ACM, mar 2019, pp. 379–390.

APPENDIX A ONLINE RESOURCES

All resources are available through our GitLab⁴ repository.

⁴<https://gitlab.com/felix134/food-kbqa-recommender>. The README.md file provides additional technical information regarding the data and steps to follow to reproduce the shown results, figures, and tables.