Personalized Food Recommendation as Constrained Question Answering over a Large-scale Food Knowledge Graph

Yu Chen[1], Ananya Subburathinam[1], Ching-Hua Chen[2], Mohammed J. Zaki[1]
hugochan2013@gmail.com,subbua@rpi.edu,chinghua@us.ibm.com,zaki@cs.rpi.edu
[1] Rensselaer Polytechnic Institute, Troy, NY [2]IBM Research, Yorktown Heights, NY

ABSTRACT

Food recommendation has become an important means to help guide users to adopt healthy dietary habits. Previous works on food recommendation either i) fail to consider users' explicit requirements, ii) ignore crucial health factors (e.g., allergies and nutrition needs), or iii) do not utilize the rich food knowledge for recommending healthy recipes. To address these limitations, we propose a novel problem formulation for food recommendation, modeling this task as constrained question answering over a large-scale food knowledge base/graph (KBQA). Besides the requirements from the user query, personalized requirements from the user's dietary preferences and health guidelines are handled in a unified way as additional constraints to the QA system. To validate this idea, we create a QA style dataset for personalized food recommendation based on a large-scale food knowledge graph and health guidelines. Furthermore, we propose a KBQA-based personalized food recommendation framework which is equipped with novel techniques for handling negations and numerical comparisons in the queries. Experimental results on the benchmark show that our approach significantly outperforms non-personalized counterparts (average 59.7% absolute improvement across various evaluation metrics), and is able to recommend more relevant and healthier recipes.

CCS CONCEPTS

• **Information systems → Question answering**; **Recommender systems**; **Personalization**; **Query reformulation**.

KEYWORDS

food recommendation, personal health, healthy diet, constrained question answering, knowledge graphs

ACM Reference Format:
Yu Chen[1], Ananya Subburathinam[1], Ching-Hua Chen[2], Mohammed J. Zaki[1]. 2021. Personalized Food Recommendation as Constrained Question Answering over a Large-scale Food Knowledge Graph. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441816

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WSDM '21, March 8–12, 2021, Virtual Event, Israel
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441816

1 INTRODUCTION

Many chronic diseases (e.g., type 2 diabetes) are related to poor eating habits [14]. Diet management is hence becoming indispensable in our daily lives. However, how to translate personalized health conditions and health knowledge into day-to-day dietary practices remains a challenging problem. Food recommendation has become an important means to help guide users to adopt healthy dietary habits [21, 23, 41, 45–47, 53]. Most existing methods recommend recipes based on recipe content (e.g., ingredients) or user behavior history (e.g., eating history). Following the classical paradigm of recommendation systems, they typically treat the food recommendation problem as a matrix completion or classification problem.

However, previous methods have three main drawbacks. First, they mostly do not consider users' explicit requirements (e.g., "breakfast dishes with bread") when doing food recommendation. Second, they mainly focus on recommending recipes which comply well with users' eating habits without considering crucial health factors such as allergies and nutrition needs. This might have a potential risk of encouraging users to eat similar but unhealthy recipes, especially if users already have poor eating habits. Third, most of them fail to utilize the rich food knowledge for recommending healthy recipes. There are many available recipe datasets providing ingredient lists and other recipe information that can be utilized for the purpose of food recommendation, such as [44], Yummly (www.yummly.com), and Allrecipes (www.allrecipes.com). Recently, Haussmann et al. [28] released a large-scale and unified food knowledge graph (KG), namely FoodKG, which brings together food ontologies, recipes, ingredients and nutritional data. Resources like this can be used for more effective food recommendation.

In this work, we propose to treat the task of personalized food recommendation as constrained question answering over a food KG, which manages to address the above limitations in a unified way. Specifically, recommendation is conducted in a Knowledge Base Question Answering (KBQA) [8] setting where a KBQA system takes as input a user query (e.g., "what is a good breakfast that contains bread?") and retrieves all recipes from the KG that satisfy the query. This allows user queries to be posed more naturally. Our approach further offers recommendations that comply well with "personalized" requirements based on a user's persona, which includes the user's unique health conditions (e.g., allergies), applicable health guidelines (nutrition needs), and even food logs. We propose a personalized KBQA approach, pFoodReQ for **P***ersonalized* **Food Re***commendation via* **Q***uestion answering* (pronounced as 'FoodRec'), which regards such personalized requirements as additional constraints. In order to adapt regular KBQA approaches [11] to handle the personalized KBQA setting, novel techniques are proposed to append personalized requirements to the raw user query via *query expansion*, handle numerical comparisons via *KG augmentation*, and negations via *constraint modeling*. Lastly, to validate our approach, we create a QA style food recommendation benchmark based on FoodKG [28] and American Diabetes Association (ADA) lifestyle guidelines [2]. Our main contributions are as follows:

- We propose a KBQA-based personalized food recommendation framework that regards personalized requirements from dietary preferences and health guidelines as additional constraints to QA. To the best of our knowledge, we are the first to propose the QA-based strategy for food recommendation.
- Novel techniques including *KG augmentation* and *Constraint modeling* are proposed to effectively handle numerical comparisons and negations in the queries.
- We create a QA style benchmark for personalized food recommendation based on a large-scale food knowledge graph and health guidelines.
- Experimental results show that our approach significantly outperform non-personalized counterparts (average 59.7% absolute improvement across various evaluation metrics) and is able to recommend more relevant and healthier recipes.

2 RELATED WORK

Recommendations systems abstract away the myriad of choices and decisions that have to be made before arriving at the choice that caters to the needs and preferences of an individual. To this end, they have been deployed successfully in many diverse domains from recommending movies [15, 36] and music [12, 32] to e-commerce [48] and healthcare [30, 50].

2.1 Food Recommendation

Most existing food recommendation approaches recommend recipes based on recipe content (e.g., ingredients) [10, 18, 21, 45], user behavior history (e.g., eating history) [21, 47] or dietary preferences [47]. Based on their survey, [23] established that most users would be willing to adapt their taste in favour of healthier recipe options, and developed a health-aware recommender algorithm that takes personal profile information and stated preferences into account when making recipe recommendations. Additionally, recipe recommendation websites and apps also use some combination of user-selected ingredients, user history, or certain food tags (such as "Italian", or "kid-friendly") to retrieve and recommend recipes. ChooseMyPlate.gov (https://www.choosemyplate.gov/myplatekitchen/recipes) includes nutrition focus criteria, cost, and cooking equipment as additional filters to their recipes. Unlike these services, *our work improves personalization by automatically applying persona focused nutritional guidelines and ingredient constraints to every query, in addition to allowing recipe queries to be posed as natural language questions.*

Following the classical paradigm of recommendation systems, most existing methods treat the food recommendation problem as a matrix completion or classification problem. To evaluate a recommendation system, they take recipe ratings, or whether a user selects the recommended recipe or not, as ground-truth. Our method is more user-friendly and has the potential to be used for interactive or conversational food recommendation, for example, in a personalized health recommendation app. Furthermore, various requirements (e.g., dietary preferences) can be addressed in a unified way as constraints to a QA system. The most similar work to ours is [27]. Whereas they presented a demo QA application that answers fact based simple, comparison and constraint-based questions, their work does not tackle personalization. In contrast, we explicitly focus on personalized food recommendation.

2.2 Knowledge Base Question Answering

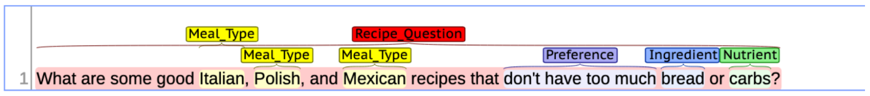
The task of knowledge base question answering (KBQA) is to automatically find answers from an underlying KG given a natural language question. KBQA approaches fall into two broad categories: semantic parsing (SP) based [1, 9, 31, 34, 51, 54] or information retrieval (IR) based [6–8, 11, 16, 25, 52]. Semantic approaches typically transform questions into intermediate logic forms, which can be posed as queries to a KG. However, they require a pre-defined set of lexical triggers or rules, which limits their domain of applicability and also their scalability. Retrieval based methods directly retrieve answers from a KG based on the information from the questions. They usually do not require hand-made rules and can therefore scale better to large and complex knowledge graphs. These methods span many embedding-based methods, that focus on mapping answers and questions into a common embedding space, where one can query the KG independent of its schema, without requiring any grammar or lexicon. We extend IR based KBQA methods to the personalized setting, where personal and contextual information must be utilized so as to find correct answers from a KG.

3 APPROACH

3.1 Personalized KBQA Benchmark

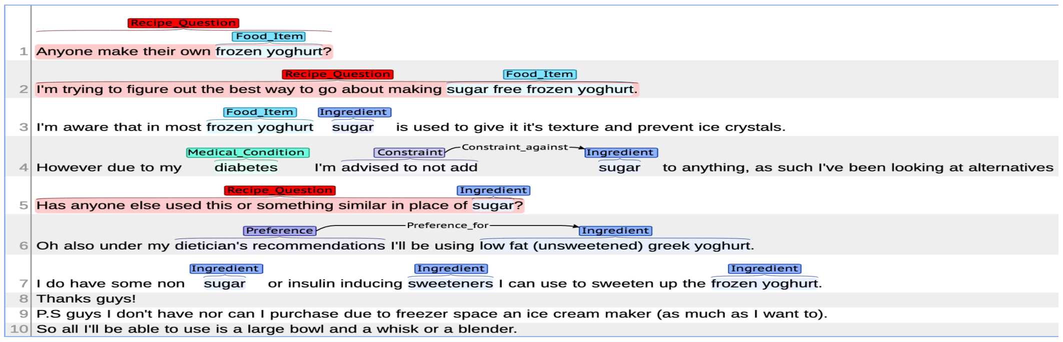
We create a benchmark QA dataset based on the extensive FoodKG [28] knowledge graph that contains over 1 million recipes, along with their ingredients and nutrients, and using ADA lifestyle guidelines [2]. We refer the interested readers to [28] for details about the FoodKG. To the best of our knowledge, this is the first dataset that supports QA style personalized food recommendations related to ingredients, nutrients and recipes. Each example in the dataset contains a *user query*, *dietary preferences*, *health guidelines* associated with the user, and the *ground-truth answers* (i.e., recipe recommendations). Note that the ground-truth answers in our benchmark are those recipes from FoodKG that satisfy both explicit requirements (mentioned in the user query) and personalized requirements (mentioned in the dietary preferences or health guidelines). The details on how we create the benchmark are given next.

3.1.1 Template-based Natural Language Question Generation. We create natural language questions that ask for food recommendation with explicit requirements. To generate realistic benchmark questions that reflect real word requirements and constraints, we examined over 200 submissions related to recipe and diabetes made on the sub-reddits "AskCulinary", "Cooking", "1200isplenty", "Baking", "AskDocs", "AskScience", and "AskReddit" on the social media forum, Reddit (http://www.reddit.com/). Figure 1 shows some examples of real user queries on reddit, which we have annotated for mentions of ingredients, nutrients, food items, as well as stated preferences and constraints for or against ingredients or nutrients. From the reddits, we identified a total of 156 posts that ask for recipe suggestions. Out of these, Table 1 shows the breakdown of how many



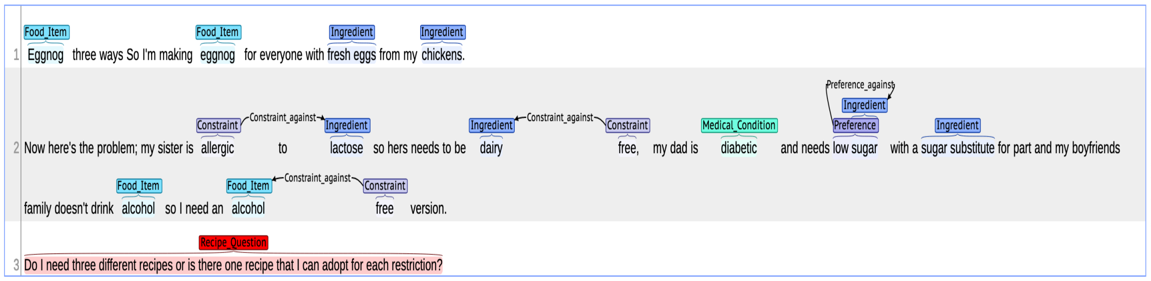
What are some good Italian, Polish, and Mexican recipes that don't have too much bread or carbs?

(a) Simple Query



Anyone make their own frozen yoghurt?
I'm trying to figure out the best way to go about making sugar free frozen yoghurt.
I'm aware that in most frozen yoghurt sugar is used to give it it's texture and prevent ice crystals.
However due to my diabetes I'm advised to not add sugar to anything, as such I've been looking at alternatives
Has anyone else used this or something similar in place of sugar?
Oh also under my dietician's recommendations I'll be using low fat (unsweetened) greek yoghurt.
I do have some non sugar or insulin inducing sweeteners I can use to sweeten up the frozen yoghurt.
Thanks guys!
P.S guys I don't have nor can I purchase due to freezer space an ice cream maker (as much as I want to).
So all I'll be able to use is a large bowl and a whisk or a blender.

(b) Ingredient Preference Query



Eggnog three ways So I'm making eggnog for everyone with fresh eggs from my chickens.
Now here's the problem; my sister is allergic to lactose so hers needs to be dairy free, my dad is diabetic and needs low sugar with a sugar substitute for part and my boyfriends family doesn't drink alcohol so I need an alcohol free version.
Do I need three different recipes or is there one recipe that I can adopt for each restriction?

(c) Ingredient Exclusion Query

Figure 1: Examples of real user questions on Reddit annotated with food items, ingredients, nutrients, allergy/preference constraints and other tags. (a) Simple query about cuisine type and ingredient/nurtient constraints. (b) Query illustrating various preferences and dietician-guided recommendation. (c) Query indicating various constraints related to allergy, medical condition and diet restriction.

posts mention specific ingredients, specific nutrients ('fat' , 'carbs.'), food items, meal type, or preferences/constraints for/against specific ingredients/nutrients. Inspired by these real user queries, in our benchmark we generate questions that include the following four common types of constraints found in the reddit submissions: i) positive ingredient constraints stating what ingredient(s) can be included in the recipe, ii) negated ingredient constraints stating what ingredient(s) cannot be included, iii) nutrient based constraints such as "low carb" or "high protein", and iv) cuisine based constraints such as "Indian", or "Mediterranean".

Table 1: Statistics from 156 recipe requests from Reddit.

Mention Type	Number of Posts
Ingredient	112
Nutrient	48
Food Item	109
Meal Type	51
Preference/Constraint	104

Nutrient based limits in the templates are randomly generated from a combination of [low/medium/high] [fat/protein/carbohydrates]. The thresholds defining low, medium, and high are set for each of these nutrients based on established online sources [13, 26, 55]. We also accommodate queries that feature a combination of these constraint types. Example: "What {limit} {nutrient} {tag} dishes can I make that do not contain {in_list}?" We have 56 different template questions in total that we use to generate the benchmark questions, that span the main types of questions gleaned from the reddit submissions. Table 2 shows some example templates, and the questions generated from them.

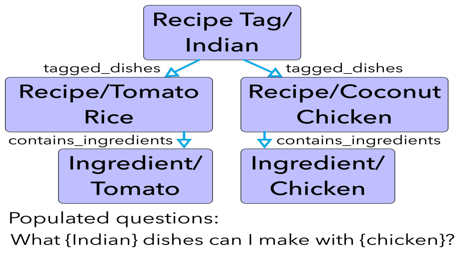


Figure 2: Template-based natural question generation.

As shown in Figure 2, in order to generate a question such as "what Indian dishes can I make with chicken?", we first randomly sample a subgraph from FoodKG which is defined as a *h*-hop neighborhood surrounding a food tag entity (e.g., "Indian"). We then randomly sample a question template (e.g., "what [tag] dishes can

Table 2: Examples of questions templates and generated questions (guided by real user questions on social media).

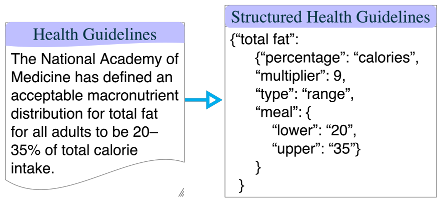
Template	Generated Question
What are {tag} recipes that contain {in_list}?	What are jellies recipes that contain orange?
What {tag} recipes can I cook without {in_list}?	What turkish or dinner-party recipes can I cook without canned milk?
Recommend {limit} {nutrient} {tag} recipes which have {in_list}?	Recommend low protein russian recipes which have onions?
Suggest {limit} {nutrient} {tag} dishes which don't contain {in_list}?	Suggest low fat egyptian dishes which don't contain lean ground beef?

Table 3: Data statistics of the personalized KBQA benchmark. The test set includes in-domain and out-of-domain instances.

	Train	Dev	Test (in-domain)	Test (out-of-domain)	Test
Size	4,621	1,540	1542	727	2,269
Avg. raw query len.	11.6	11.6	11.5	11.5	11.5
Avg. expanded query len.	35.3	35.1	35.1	35.1	35.1
Avg. # of answers	3.2	2.9	2.6	3.0	2.7
Avg. # of constraints	5.0	5.0	5.0	5.0	5.0

I make with [ingredients]?") from the template pool, and fill the slots with KG entities (e.g., "chicken") sampled from the subgraph.

3.1.2 Generating Dietary Preferences. A good food recommendation system should respect a user's personalized requirements even though the user does not explicitly mention them in a query. Keeping this in mind, to make our benchmark more realistic, associated to each user query, we randomly generate dietary preferences on ingredient likes and allergies, where ingredients are randomly sampled from an ingredient pool that includes all the ingredients contained in the recipes belonging to the food tag mentioned in the query. The average/min/max sizes of ingredient pools are 717, 58 and 2,093, respectively.



The National Academy of Medicine has defined an acceptable macronutrient distribution for total fat for all adults to be 20–35% of total calorie intake.

```
{"total fat":
    {"percentage": "calories",
    "multiplier": 9,
    "type": "range",
    "meal": {
        "lower": "20",
        "upper": "35"}
    }
}
```

Figure 3: A guideline example from the ADA.

3.1.3 Compiling Health Guidelines. A characteristic that distinguishes our proposed method from most existing food recommendation methods is that we want our system to recommend "healthy" recipes that comply with health guidelines. Therefore, we carefully select some food-related guidelines from ADA lifestyle guidelines [2], that pertain to nutrient and micronutrient budgets, and treat them as additional requirements for food recommendation. Since those guidelines are in natural language, we convert them into structured representations (e.g., a hash table storing key-value pairs). For instance, as shown in Figure 3, the guideline suggests that a good percentage of calories from total fat should be between 20% and 35%. This kind of macronutrient is calculated by dividing the amount of calories from total fat by the amount of total calories in a recipe. Note that the multiplier value in the example (right hand panel) indicates that "food nutrient as fat (lipids) contains 9 kilocalories per gram (kcal/g)". Each dietary preference in the dataset is randomly assigned actual health guidelines from the ADA. For more realistic setting, we allow multiple guidelines per user; we set the number of guidelines per user to be 1, 2 or 3 with probabilities 0.85, 0.1 and 0.05, respectively, to avoid conflicting or over-restricting guidelines when creating the dataset. As such, our framework is generic enough to handle several health guidelines.

3.1.4 KBQA Benchmark Dataset. We split the personalized KBQA benchmark dataset into training (4,621 examples), development (1,540 examples) and test sets (2,269 examples) where 727 examples in the test set are out-of-domain examples. Out-of-domain examples were created based on 23 recipe tags (out of totally 238 recipe tags) which were never seen in the training and development sets. The data statistics are provided in Table 3.

3.2 pFoodReQ: Personalized Food Recommendation Framework

We treat the task of personalized food recommendation as constrained question answering over a food KG with additional personal constraints from dietary preferences and health guidelines. Formally, we formulate the following problem setting: a personalized food recommendation system is supposed to take as input a natural language question, dietary preferences as well as health guidelines, and retrieve all recipes from a food KG that satisfy the requirements contained in the input. The overall architecture of our proposed pFoodReQ framework is shown in Figure 4, which consists of the *Query Expansion (QE)*, *KG Augmentation (KA)*, *Constraint Modeling (CM)*, and *KBQA* modules.

3.2.1 KBQA Module. The KBQA module is responsible for retrieving recipes from the KG that satisfy the constraints mentioned in the expanded query (will be discussed later). Our approach to personalization is independent of the underlying KBQA method. We can use any KBQA method that accepts a query and a KG subgraph, and returns the most relevant answers from the KG subgraph. As in most KBQA methods, a KG subgraph is extracted by first detecting a topic entity (i.e., the main entity in a query), and then fetching a h-hop neighborhood (which contains entities and relations) surrounding the topic entity. We will test our framework with various KBQA models in our experiments.

3.2.2 Query Expansion. An effective food recommendation system should respect personalized requirements from the dietary preferences and health guidelines. As shown in Figure 4, considering a

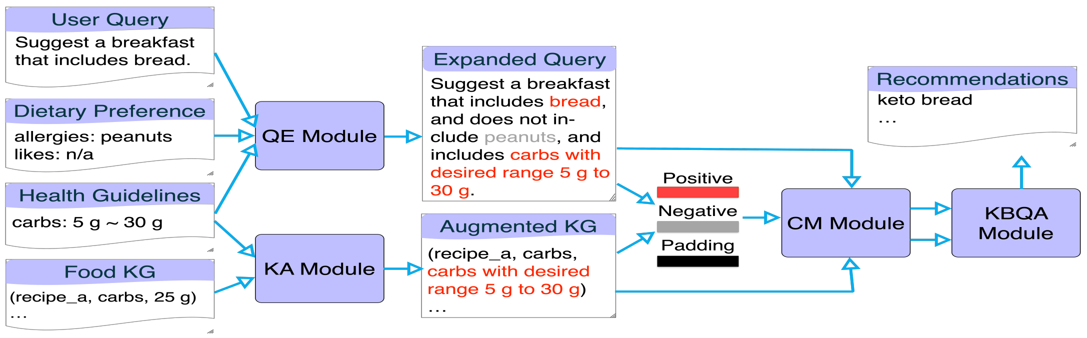


Figure 4: pFoodReQ architecture: QE, KA and CM stand for Query Expansion, KG Augmentation and Constraint Modeling, respectively. Positive constraints are in red, negative constraints in gray, and non-constraints (or padding) in black.

user query "suggest a breakfast that contains bread", along with the raw user query, the dietary preferences indicate that the user is allergic to "peanuts", and the associated health guidelines suggest that the user consumes "carbohydrates with desired range 5g to 30g". In order to make our system aware of the personalized requirements, we transform the raw user query into one that additionally incorporates the personalized requirements. Formally, given a number of personal constraints, say $\{c_1, c_2, ..., c_m\}$, from dietary preferences or applicable health guidelines, and the raw user query q in natural language, we append each of the constraints c_i to q, and finally obtain the expanded query $\tilde{q}$. In the above example, the expanded query becomes "suggest a good breakfast that contains bread, and does not have peanuts, and contains carbohydrates with desired range 5g to 30g". Thus, given the same query q, our system is able to provide different users with tailored food suggestions that suit their needs based on the expanded query $\tilde{q}$.

3.2.3 *KG Augmentation.*

In order to answer a query with a numerical constraint like "contains carbohydrates with desired range 5g to 30g" from health guidelines, a KBQA system needs to handle numerical comparison. For example, in Figure 4, the example *recipe_a* has 25g of carbohydrates, and therefore to check whether the recipe meets the guideline, the system would have to determine that"25 is larger than 5" and "25 is less than 30". Unfortunately, the inability to handle numerical comparison is a known limitation of neural networks. Even though one can embed a number into a continuous vector, regardless of the out-of-vocabulary issue, it is extremely difficult to train a neural network to do reliable numerical comparisons in some high-dimensional embedding space.

We propose an effective technique that dynamically augments a KG subgraph, i.e., we create or *materialize* a new view of the KG subgraph, based on the results of symbolic number comparison before feeding it to a neural network-based KBQA system. Let G be the set of numerical constraints expressed in natural language (e.g., g = "carbohydrates with desired range 5g to 30g", where $g \in G$). Let v be the node in the KG subgraph that denotes the nutrition amount (e.g., the number of grams of carbohydrates) of a recipe. Then v is replaced with g if v is within the desired numerical range (e.g., $[5, 30]$). More formally, for each constraint $g \in G$, we model it as an indicator function $I_g(v) = 1$ if $v \in \text{range}(g)$; otherwise, $I_g(v) = 0$. If $I_g(v) = 1$, then v is replaced with g; otherwise, we set it to empty. In this way, no changes need to be made to the KBQA system to handle numerical comparisons. Of course, when $I_g(v) = 0$, we can even completely remove the entire recipe node from the candidate set since it does not satisfy the constraint. However, we choose to keep the recipe node because i) we want to train our KBQA system to retrieve the right answers on its own, which we think is more generic and ii) this opens a door for modeling soft constraints, which can can enhance the robustness and usefulness of our system, allowing it not to necessitate adherence to all of the constraints.

3.2.4 *Constraint Modeling.*

When dealing with queries involving ingredient allergies, a KBQA system should be able to handle negative constraints. However, it is well-known that neural networks are not good at handling negation [5, 19, 22]. A neural network-based KBQA system can have trouble distinguishing a negative constraint like "does not have peanuts" from a positive constraint like "has peanuts". We propose an embedding-based method for constraint modeling. The idea is to explicitly indicate the existence of a negative constraint to a KBQA system so that it can learn how to appropriately process the negative constraint through training. Formally, when encoding an expanded user query $\tilde{q}$, we add an additional constraint type markup $m_i \in M$ to each query word q_i to indicate which type of constraint it belongs to. M is the set of constraint types that include *positive, negative and padding* where *padding* means the word does not belong to any constraint. As shown in the expanded query of Figure 4, words in black belong to the *padding* type as they are not part of the constraint set. When encoding a candidate answer node from a KG subgraph, we also add such constraint type markups to words occurring in its context node v in order to explicitly indicate the constraint type to a KBQA system. Note that all constraint type markups are linearly mapped into a continuous embedding space via a $|M| \times d_m$ weight matrix where $|M|$ is the size of the constraint set M and d_m is the dimension of the constraint embeddings. The resultant constraint embeddings are then concatenated with the word embeddings before being further processed by a KBQA system.

3.3 Training the Personalized KBQA system

As outlined in Figure 4, our personalized KBQA system takes as input an expanded natural language question $\tilde{q} = \{q_i\}_{i=1}^{|\tilde{q}|}$ which is represented as a sequence of words (q_i), and all the candidate answers extracted from the h-hop augmented KG subgraph surrounding a topic entity, which we denote as $A = \{a_i\}_{i=1}^{|A|}$. Our pFoodReQ model maps them into a joint embedding space, which can capture their latent semantic meanings. Given the representation (or embedding) of the expanded question $\tilde{q}$, denoted $\tilde{\mathbf{q}}$, and the representations of candidate answers, denoted $\mathbf{a}_i$ for answer a_i, we compute the matching score $S_{\text{qa}}(\tilde{q}, a_i)$ between every question and potential answer pair via their dot-product $S_{\text{qa}}(\tilde{q}, a_i) = \tilde{\mathbf{q}} \bullet \mathbf{a}_i$. The candidate answers are then ranked by their scores.

In the training phase, we force positive candidate answers to have higher scores than negative candidate answers by using a triplet-based loss function:

$$L = \sum_{a^+ \in A^+, a^- \in A^-} \ell\big(S_{\text{qa}}(\tilde{q}, a^+), S_{\text{qa}}(\tilde{q}, a^-)\big)$$

where $\ell(y, \hat{y}) = \max(0, 1 + \hat{y} - y)$ is a hinge loss function, and A^+ and A^- denote the positive and negative answer sets, respectively. Positive answers are the ground-truth answers, while negative answers are randomly sampled from the candidate answer set. At testing time, considering there can be multiple answers to a given question, we keep those candidates whose scores are close to the highest score, within a certain margin θ, as good answers. Note that θ is a hyper-parameter which controls the degree of tolerance.

3.4 Recommendation with Food Logs

Food recommendation systems often utilize a user's eating history [21, 47] to produce recommendations more aligned to a user's tastes. In addition to personalization through query expansion with profile based likes/dislikes and guidelines, we also support the concept of personalization based on a user food-log by leveraging recipe embeddings [35].

3.4.1 Food Log Generation. Food logs are intended to reflect a user's eating habits. Since we do not have access to extensive food logs that match the recipes in our KG, we create simulated food logs, and utilize them for further personalizing our system's responses. We focus on five diets to simulate a varied and distinct eating pattern in the food logs: Mediterranean [40], DASH (Dietary Approaches to Stop Hypertension)[3, 49], Vegan [20], Indian [43], and Keto [17, 39]. To create the simulated food logs, we first construct a list of search terms relevant to recipes and ingredients found in foods associated with each of these diets. Through this approach we can search through all the recipe names in our KG for recipes that highlight these food items or ingredients. We iteratively generate six logs for each diet type by sampling the search results of ten random search terms for that diet. Via this process, we obtain 30 simulated food logs spanning the 5 diet types, with 26 recipes on average.

3.4.2 Recipe Similarity. Determining the most similar recipes to the user's food log requires a representation of all recipes. We utilize recipe embeddings [35] to obtain a distributional representation of all recipes. Next, for each recipe, we construct a 10-nearest neighbor graph using cosine similarity. In addition to the food log, we utilize the above 10-nearest neighbor graphs to identify the top k most similar recipes to the recipes in the user's log. More specifically, we construct this list by taking the k recipes with the most similarity to the food log, where the similarity between a recipe and the food log (denoted as $S_{\text{sim}}(g, a_i)$) is computed as the largest cosine similarity between the recipe and all the recipes in the food log.

3.4.3 KG Subgraph Expansion. In order to consider the aforementioned top k most similar recipes as additional candidate answers for food recommendation, we proceed to expand the original KG subgraph (i.e., surrounding the food tag entity) by incorporating those recipes and their associated ingredient and nutrient information. Notably, we can directly apply pFoodReQ to the expanded KG subgraph without any modification, and pFoodReQ is able to compute a confidence score $S_{\text{qa}}(\tilde{q}, a_i)$ for each pair of expanded question $\tilde{q}$ and recipe a_i.

3.4.4 Answer Ranking. When ranking candidate recipes for recommendation, in principle, we prefer those recipes that not only satisfy various constraints contained in the expanded question (as before), but also are similar to the user's food log. To this end, we proceed to compute an overall score $S(\tilde{q}, f, a_i)$ combining both the KBQA confidence score and the recipe similarity score:

$$S(\tilde{q}, f, a_i) = (1 - \lambda) \cdot \widetilde{S}_{\text{qa}}(\tilde{q}, a_i) + \lambda \cdot \widetilde{S}_{\text{sim}}(f, a_i) \tag{1}$$

where λ is a hyperparameter for controlling the importance of recipe similarity. $\widetilde{S}_{\text{qa}}(\tilde{q}, a_i) = (S_{\text{qa}}(\tilde{q}, a_i) - S_{\min})/(S_{\max} - S_{\min})$ is the normalized KBQA score, and $S_{\text{qa}}(\tilde{q}, a_i)$ is the unnormalized KBQA score defined in Section 3.3. $\widetilde{S}_{\text{sim}}(f, a_i) = (S_{\text{sim}}(f, a_i) + 1)/2$ is the normalized cosine similarity between food log f and recipe a_i, and $S_{\text{sim}}(f, a_i)$ is the cosine similarity defined in Section 3.4.2. Note that $S_{\max}$ and $S_{\min}$ are the maximal and minimal KBQA scores among KBQA scores of all candidate recipes for question $\tilde{q}$. During the inference stage, we use a threshold θ for selecting good answers.

4 EXPERIMENTS

We evaluate our proposed pFoodReQ model against competitive baseline methods on the personalized food recommendation benchmark. The dataset as well as the implementation of our model are publicly available at https://github.com/hugochan/PFoodReq.

4.1 Baseline Methods

Our main baseline is BAMnet [11], a state-of-the-art IR-based KBQA method that directly retrieves answers from a KG subgraph by jointly embedding the question and KG subgraph. We also include a sophisticated embedding-based KBQA method called MatchNN which was motivated by [6]. To demonstrate that this is a non-trival task, we further include a Bag-of-Word (BOW) [37] vectors based KBQA baseline. For both MatchNN and BOW, the candidate answer is encoded as the sum of the corresponding answer type, answer path and answer context embeddings. When encoding a question (or KG aspect such as answer type), MatchNN employs a bidirectional LSTM [29] for encoding the sequence of tokens, and BOW simply takes the average of their pretrained word embeddings [42]. Note that none of the above baselines explicitly handle personalized QA, and have access to personalized requirements such as dietary preferences and health guidelines. They are included as non-personalized baselines in our experiments.

To examine whether our proposed techniques are generic and can be applied to extend regular KBQA methods for handling personalized food recommendation, we develop personalized versions of the aforementioned three KBQA baselines by leveraging our proposed techniques including query expansion, KG augmentation and constraint modeling. Therefore, we have three personalized methods that include P-BOW, P-MatchNN and pFoodReQ (i.e., personalized BAMnet) where pFoodReQ is our main model.

4.2 Data and Metrics

FoodKG [28] is a large-scale KG that integrates recipes, food ontologies and nutritional data. It comprises about one million recipes (sourced from [38]), 7.7 thousand nutrient records (sourced from the USDA: www.usda.gov), and 7.3 thousand types of food (sourced from [24]). Overall, it has over 67 million triples. In this work, we use FoodKG as the underlying food KG. ADA Lifestyle Management [2] includes ADA's current clinical practice recommendations and is intended to provide the components of diabetes care, general treatment goals and guidelines, and tools to evaluate quality of care. We carefully select food-related guidelines on nutrient and micronutrient needs for "healthier" food recommendation. Following the steps described in Section 3.1, we create a food recommendation benchmark based on FoodKG and ADA guidelines.

Following previous work on KBQA [4, 11], macro F1 scores (i.e., the average of F1 scores over all questions) are reported on the test set. Besides F1, we also report Mean Average Precision (MAP) and Mean Average Recall (MAR) scores that are common metrics for evaluating recommender systems [37]. Note that for all metrics, we consider all the answers returned by a system.

4.3 Model Hyperparameter Settings

When constructing the vocabularies of words (from questions and KG), entity types (from KG) and relation types (from KG), we only consider those questions and their corresponding KG subgraphs appearing in the training set. The vocabulary sizes of words, entity types and relation types are 7009, 7 and 11, respectively. Given a topic entity, we extract its 2-hop subgraph (i.e., $h = 2$) to collect candidate answers, which is sufficient for our benchmark. We initialize word embeddings with 300-dimensional pre-trained GloVe vectors [42]. The batch size is set as 32. The answer module threshold θ is set to 0.9 and 0.2 for pFoodReQ and pFoodReQ+RecipeSim, respectively. The value of θ^G is set to 0.2 (see Section 4.7). The value of λ is set to 0.3. In the training process, we use the Adam optimizer [33]. We use $d_m = 40$ for constraint modeling. All hyperparameters are tuned on the development set; for BAMnet we use the default values mentioned in Chen et al. [11].

Table 4: Experimental results (in %) on the test set.

Method	MAP	MAR	F1
BOW	2.1	2.0	2.3
MatchNN	2.7	2.7	3.0
BAMnet	3.1	3.0	3.1
P-BOW	4.5	4.4	4.2
P-MatchNN	45.5	45.1	41.2
pFoodReQ	**62.7**	**61.8**	**63.7**

4.4 Experimental Results

Table 4 shows the evaluation results comparing our method against other baselines. We observe that the personalized systems significantly outperform their non-personalized counterparts, which verifies the effectiveness of our proposed techniques for utilizing the personal information (i.e., dietary preferences and health guidelines) to provide more accurate and personalized food recommendations (regardless of the baseline system used). Our pFoodReQ model consistently outperforms all models. It overshadows BAMnet by a significant margin (59.6% absolute improvement in MAP, 58.8% absolute improvement in MAR, and 60.6% absolute improvement in F1) in all evaluation metrics. It also outperforms P-MatchNN, with an average improvement of 18.8% absolute improvement over all metrics. Moreover, we can observe that there is still scope for improvement on this benchmark, which requires a certain level of reasoning to understand the different constraints in a query and to satisfy them based on the corresponding KG subgraph. While pFoodReQ performs the best, the results indicate that our benchmark is quite challenging and can spur further research.

4.5 Human Evaluation

We performed human evaluation for food recommendations by presenting a set of eight evaluators with fifty questions from a randomized test, along with the user personas spanning ingredient preferences (likes and dislikes) and applicable nutrition guidelines. For each question, we provide the answers from BAMnet, P-BOW, P-MatchNN and pFoodReQ models in a randomized order.

Table 5: Human evaluation results (± standard deviation) on the test set. The rating scale is from 1 to 10 (higher scores indicate better results).

Method	Score
BAMnet	3.82 (0.70)
P-BOW	4.69 (0.47)
P-MatchNN	7.05 (0.44)
pFoodReQ	**7.76** (0.50)

Each answer presented to the user contains the top three recipes (if more than three recipes were retrieved), the ingredient list, and the nutritional values. The evaluators were instructed to rank the answers in the order of their (subjective) preference on a scale from 1 − 4, with 1 being the best, indicating how well they think the recommended answer set satisfies the various constraints mentioned in the input, without knowing which of the four systems generated the answers. We adopt a simple rank aggregation strategy to compute the overall score for each system based on the ranks provided by human evaluators. Specifically, we map the ranks 1, 2, 3 and 4 to the corresponding scores 10, 7, 4 and 1, so that systems with better ranks have higher scores. Evaluation scores from all 8 evaluators are collected and averaged as final scores. As a result, the highest score a system can get is 10 while the lowest score is 1. As shown in Table 5, pFoodReQ achieves the best result. Furthermore, all of the personalized baselined fared better than the non-personized BAMnet approach. These results conclusively demonstrate the general applicability and effectiveness of our personalization techniques (independently of the method).

4.6 Model Analysis

4.6.1 Ablation Study. We perform an ablation study to systematically investigate the impact of different model components for our pFoodReQ method. Here we briefly describe the ablated systems: –

Table 6: Ablation study (in %) on the test set.

Method	MAP	MAR	F1
pFoodReQ	**62.7**	**61.8**	**63.7**
− KA	58.5	57.8	58.6
− CM	29.7	29.3	25.9
− QE	4.5	4.5	4.2

CM removes the Constraint Modeling module, − KA removes the KG Augmentation module, and − QE removes the Query Expansion module. Note that for the ablated systems, we remove one module each time. Table 6 shows the experimental results of the ablated systems as well as the full model. It confirms our finding that all of the three techniques introduced for handling the personalized recommendation setting greatly contribute to the overall model performance. Among them, the Query Expansion module contributes most, which demonstrates the importance of incorporating rich personal information in order to provide accurate recommendation. By removing the QE module, we observe an average 58.3% performance drop across all evaluation metrics. The CM module is also important as we see an average 34.4% performance drop across metrics if it is removed. This shows the importance of handling negations for personalized food recommendation. Lastly, the KA module also contributes to the overall model performance.

Table 7: Predicted answers by various methods.

Test Query
Query: Could you recommend eggplant dishes which do not contain ketchup? **Preference:** Likes: ["white wine vinegar"], Dislikes: ["sesame", "red peppers"] **Guideline:** Protein: 5.0 g to 30.0 g **Gold: ["Moussaka (Vegan or Vegetarian, You Choose.)"]**
Baseline Methods
BOW: ["Yunnan Style Grilled Veggies With Dipping Sauce"] **MatchNN:** ["Sicilian Eggplant Parmagiana With Spinach", "Savannah Dip", "The Real Melitzanosalata", "Eggplant, Tomato and Mozzarella Stacks, Hot or Cold", "The Ultimate Veggie Burger Thai Style", "Rigatoni With Roasted Eggplant Puree", ...] **BAMnet:** ["Sicilian Eggplant Parmagiana With Spinach", "Savannah Dip", "The Real Melitzanosalata", "Eggplant, Tomato and Mozzarella Stacks, Hot or Cold", "The Ultimate Veggie Burger Thai Style", "Rigatoni With Roasted Eggplant Puree", ...] **P-BOW:** ["Black Cherry Tomatoes and Eggplant Pasta Sauce", "Italian Sausage and Eggplant Soup", "Claricaś Melitzanosalata", "Caponata", "Greek Eggplant Dip", ...] **P-MatchNN:** ["White Bean Ratatouille"]
Ablated Methods
pFoodReQ w/o QE: ["Sicilian Eggplant Parmagiana With Spinach", "Savannah Dip", "The Real Melitzanosalata", "Eggplant, Tomato and Mozzarella Stacks, Hot or Cold", "The Ultimate Veggie Burger Thai Style", "Rigatoni With Roasted Eggplant Puree", ...] **pFoodReQ w/o KA:** ["White Bean Ratatouille"] **pFoodReQ w/o CM:** ["Moussaka (Vegan or Vegetarian, You Choose.)", "White Summer Lasagna", "Pasta With Eggplant and Pork"]
pFoodReQ Method
pFoodReQ: ["Moussaka (Vegan or Vegetarian, You Choose.)"]

4.6.2 *Case Study.* Table 7 shows the predicted answers from pFoodReQ and its ablated variants, and other baselines on a testing example. In this example, the user is asking for eggplant dishes that do not contain ketchup. The user's persona includes both likes (e.g., white wine vinegar) and dislikes (e.g., red peppers), and a health/diet guideline for protein intake. As we can see, pFoodReQ retrieves the ground-truth response from the KG while other systems make many imprecise recommendations. For example, pFoodReQ w/o CM recommends dishes that contain "red peppers" such as "Pasta With Eggplant and Pork" when the user dislikes "red peppers". The same mistake is also made by other systems like pFoodReQ w/o QE, MatchNN, and BAMnet that recommend "The Real Melitzanosalata" (which also contains red peppers). Without effectively utilizing the health guideline, many systems also recommend dishes that have protein values outside of the stipulated range.

Table 8: Experimental results (in %) on the test set with and without using recipe similarity.

Method	MAP	MAR	F1
pFoodReQ	27.3	26.4	32.6
pFoodReQ+RecipeSim	**34.5**	**33.0**	**36.6**

4.7 Recommendation considering Food Logs

We demonstrate the ability of our food recommendation system in utilizing food logs. We reuse the pretrained pFoodReQ model evaluated in Section 4.4. In order to adapt it to utilize the food log information, we only need to enhance it with the *KG subgraph expansion* and *answer ranking* techniques introduced in Section 3.4 during the inference phase. We call the resulting system pFoodReQ+RecipeSim. For empirical comparison we generate new ground-truth answers that respect both the constraints (as before) and the recipe similarities, using the following strategy. For each example in the test set, we combine the ground-truth KBQA answer set R_{qa} and the top k similar recipes R_{sim} to obtain a potential ground-truth answer set R_p. Then for each recipe in R_p, we compute an overall score by combining its KBQA score (i.e., 1 if all constraints are satisfied, and 0.5 if partial constraints are satisfied; otherwise, it is 0) and its recipe similarity score using Eq. (1). Finally, we select a subset of recipes (denoted as R) from R_p based on their scores using a margin threshold θ^G following the same strategy mentioned in Section 3.3.

As shown in Table 8, pFoodReQ+RecipeSim significantly outperforms pFoodReQ across all evaluation metrics. This is because pFoodReQ+RecipeSim additionally considers the similarity between the candidate recipe and the user's food log, and can recommend recipes that are more similar to the user's eating history while still satisfying various constraints on ingredients and nutrients.

5 CONCLUSIONS AND FUTURE WORK

In this work, we treated the task of personalized food recommendation as constrained QA over a food KG, which considers personalized requirements from dietary preferences, health guidelines, and food logs, as additional constraints, utilizing a user's persona. Novel techniques were proposed to effectively handle numerical comparisons and negations in QA. A benchmark for personalized food recommendation was also created for evaluating food recommendation systems. Experimental results on the benchmark verified the effectiveness of our proposed method. Future work could consider automatic ways of mining personal preferences and applicable health guidelines from a user's historical behavioral data.

ACKNOWLEDGMENTS

This work is supported by IBM Research AI through the AI Horizons Network (under the RPI-IBM HEALS Project).

REFERENCES

[1] Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *WWW*. 1191–1200.

[2] American Diabetes Association. 2019. 5. Lifestyle management: standards of medical care in diabetes–2019. *Diabetes care* 42, Supplement 1 (2019), S46–S60.

[3] Katie Bandurski. (accessed August 16, 2020). 7-Day DASH Diet Meal Plan. https://www.tasteofhome.com/collection/dash-diet-meal-plan/

[4] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*. 1533–1544.

[5] Eduardo Blanco and Dan Moldovan. 2011. Semantic representation of negation using focus detection. In *ACM-HLT Conference*. 581–589.

[6] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *EMNLP*. 615—-620.

[7] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075* (2015).

[8] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *ECML-PKDD*. 165–180.

[9] Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL Conference*. 423–433.

[10] Meng Chen, Xiaoyi Jia, Elizabeth Gorbonos, Chinh T Hoang, Xiaohui Yu, and Yang Liu. 2020. Eating healthier: Exploring nutrition information for healthier recipe recommendation. *Information Processing & Management* 57, 6 (2020).

[11] Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Bidirectional Attentive Memory Networks for Question Answering over Knowledge Bases. In *NAACL*.

[12] Zhiyong Cheng, Jialie Shen, Lei Zhu, Mohan S Kankanhalli, and Liqiang Nie. 2017. Exploiting Music Play Sequence for Music Recommendation.. In *IJCAI*, Vol. 17. 3654–3660.

[13] ClevelandClinic. 2020 (accessed May 31, 2020). Healthy Fat Intake. https://my.clevelandclinic.org/health/articles/11208-fat-what-you-need-to-know

[14] K.B. DeSalvo, R. Olson, and K.O. Casavale. 2016. Dietary guidelines for americans. *JAMA* 315, 5 (2016), 457–458. https://doi.org/10.1001/jama.2015.18396

[15] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *SIGKDD Conference*. 193–202.

[16] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *ACL-IJCNLP*, Vol. 1. 260–269.

[17] Andreas Eenfeldt and Bret Scher. (accessed August 16, 2020). 14-Day Keto Meal Plan with Recipes & Shopping Lists. https://www.dietdoctor.com/low-carb/keto/diet-plan

[18] MA El-Dosuky, MZ Rashad, TT Hamza, and AH El-Bassiouny. 2012. Food recommendation using ontology and heuristics. In *International conference on advanced machine learning technologies and applications*. Springer, 423–429.

[19] Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *ACL Conference*. 495–504.

[20] People for the Ethical Treatment of Animals. 2020. Try this irresistible two-week vegan meal plan. https://www.peta.org/living/food/sample-two-week-vegan-meal-plan/

[21] Peter Forbes and Mu Zhu. 2011. Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. In *RecSys Conference*.

[22] Dipesh Gautam, Nabin Maharjan, Rajendra Banjade, Lasang Jimba Tamang, and Vasile Rus. 2018. Long Short Term Memory Based Models for Negation Handling in Tutorial Dialogues. In *The Thirty-First International Flairs Conference*.

[23] Mouzhi Ge, Francesco Ricci, and David Massimo. 2015. Health-aware food recommender system. In *RecSys Conference*. 333–334.

[24] Emma J Griffiths, Damion M Dooley, Pier Luigi Buttigieg, Robert Hoehndorf, Fiona SL Brinkman, and William WL Hsiao. 2016. FoodON: A Global Farm-to-Fork Food Ontology.. In *ICBO/BioCreative*.

[25] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *ACL*, Vol. 1. 221–231.

[26] Anne Harding. 2010 (accessed May 31, 2020). How to Count Carbs in 10 Common Foods. https://www.health.com/condition/type-2-diabetes/how-to-count-carbs-in-10-common-foods

[27] Steven Haussmann, Yu Chen, Oshani Seneviratne, Nidhi Rastogi, James Codella, Ching-Hua Chen, Deborah L McGuinness, and Mohammed J Zaki. 2019. FoodKG Enabled Q&A Application. In *International Semantic Web Conference Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas)*.

[28] Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L McGuinness, and Mohammed J Zaki. 2019. FoodKG: A Semantics-Driven Knowledge Graph for Food Recommendation. In *International Semantic Web Conference*. Springer, 146–162.

[29] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[30] Hengyi Hu, Adam Elkus, and Larry Kerschberg. 2016. A Personal Health Recommender System incorporating personal health records, modular ontologies, and crowd-sourced data. In *ASONAM Conference*. 1027–1033.

[31] Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, and Dongyan Zhao. 2018. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. *TKDE* 30, 5 (2018), 824–837.

[32] Yucheng Jin, Nyi Nyi Htun, Nava Tintarev, and Katrien Verbert. 2019. ContextPlay: Evaluating User Control for Context-Aware Music Recommendation. In *ACM Conference on User Modeling, Adaptation and Personalization*. 294–302.

[33] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[34] Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *EMNLP-CoNLL*. 754–765.

[35] Diya Li and Mohammed J Zaki. 2020. RECIPTOR: An Effective Pretrained Model for RecipeRepresentation Learning. In *SIGKDD Conference*.

[36] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, and Guojun Wang. 2019. HAES: A New Hybrid Approach for Movie Recommendation with Elastic Serendipity. In *CIKM Conference*. 1503–1512.

[37] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

[38] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2018. Recipe1M: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *arXiv preprint arXiv:1810.06553* (2018).

[39] Jessica Migala. (accessed August 16, 2020). The Keto Diet: 7-Day Menu and Comprehensive Food List: Everyday Health. https://www.everydayhealth.com/diet-nutrition/ketogenic-diet/comprehensive-ketogenic-diet-food-list-follow/

[40] Jessica Migala. (accessed August 16, 2020). What Is the Mediterranean Diet? Food List, Meal Plan, Benefits, More: Everyday Health: Everyday Health. http://www.everydayhealth.com/mediterranean-diet/guide/.

[41] Weiqing Min, Shuqiang Jiang, and Ramesh Jain. 2019. Food Recommendation: Framework, Existing Solutions and Challenges. *arXiv preprint arXiv:1905.06269* (2019).

[42] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP Conference*. 1532–1543.

[43] Priyanka Sadhukhan. (accessed August 16, 2020). Diabetes Diet Chart for Indians-What To Eat And Avoid. https://www.stylecraze.com/articles/diabetes-diet-plan-indians/

[44] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. In *CVPR Conference*.

[45] Chun-Yuen Teng, Yu-Ru Lin, and Lada A. Adamic. 2012. Recipe Recommendation Using Ingredient Networks. In *ACM Web Science Conference* (Evanston, Illinois). 298–307.

[46] Christoph Trattner and David Elsweiler. 2017. Food Recommender Systems: Important Contributions, Challenges and Future Research Directions. *CoRR* abs/1711.02760 (2017). arXiv:1711.02760 http://arxiv.org/abs/1711.02760

[47] Mayumi Ueda, Syungo Asanuma, Yusuke Miyawaki, and Shinsuke Nakajima. 2014. Recipe recommendation method by considering the users preference and ingredient quantity of target recipe. In *International MultiConference of Engineers and Computer Scientists*, Vol. 1. 12–14.

[48] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *SIGKDD Conference*. 839–848.

[49] Helen West. (accessed August 16, 2020). The DASH Diet: A Complete Overview and Meal Plan. https://www.healthline.com/nutrition/dash-diet#sample-menu

[50] Martin Wiesner and Daniel Pfeifer. 2010. Adapting recommender systems to the requirements of personal health record systems. In *ACM International Health Informatics Symposium*. 410–414.

[51] Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*. 960–967.

[52] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957* (2016).

[53] Longqi Yang, Cheng-Kang Hsieh, Hongjian Yang, John P Pollak, Nicola Dell, Serge Belongie, Curtis Cole, and Deborah Estrin. 2017. Yum-me: a personalized nutrient-based meal recommender system. *ACM Transactions on Information Systems (TOIS)* 36, 1 (2017), 7.

[54] Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *ACL*. 1321–1331.

[55] Kathleen M. Zelman. 2018. Daily Protein Requirements: Are You Getting Enough? https://www.webmd.com/food-recipes/protein