IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 25, 2021, accepted February 6, 2021, date of publication February 10, 2021, date of current version February 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058559

A Word Embedding-Based Method for Unsupervised Adaptation of Cooking Recipes

ANDREA MORALES-GARZÓN, JUAN GÓMEZ-ROMERO, AND MARIA J. MARTIN-BAUTISTA, (Member, IEEE)
Department of Computer Science and Artificial Intelligence, University of Granada, 18010 Granada, Spain

Corresponding author: Andrea Morales-Garzón (andreamgmg@correo.ugr.es)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation program under Grant 816303 (Stance4Health) and in part by the University of Granada through a program of research initiation for master's degree students.

ABSTRACT Studying food recipes is indispensable to understand the science of cooking. An essential problem in food computing is the adaptation of recipes to user needs and preferences. The main difficulty when adapting recipes is in determining ingredients relations, which are compound and hard to interpret. Word embedding models can catch the semantics of food items in a recipe, helping to understand how ingredients are combined and substituted. In this work, we propose an unsupervised method for adapting ingredient recipes to user preferences. To learn food representations and relations, we create and apply a specific-domain word embedding model. In contrast to previous works, we not only use the list of ingredients to train the model but also the cooking instructions. We enrich the ingredient data by mapping them to a nutrition database to guide the adaptation and find ingredient substitutes. We performed three different kinds of recipe adaptation based on nutrition preferences, adapting to similar ingredients, and vegetarian and vegan diet restrictions. With a 95% of confidence, our method can obtain quality adapted recipes without a previous knowledge extraction on the recipe adaptation domain. Our results confirm the potential of using a specific-domain semantic model to tackle the recipe adaptation task.

INDEX TERMS Data mapping, food computing, natural language processing, recipe adaptation, word embedding.

I. INTRODUCTION

Food is essential to the human being. Our dietary habits have a huge impact on health and, thus, in quality of life. According to the World Health Organization,[1] a healthy diet prevents malnutrition and protects against various diseases such as cancer, stroke, and diabetes. Furthermore, an unhealthy diet and lack of physical activity can lead to obesity, posing a health risk. New technologies and a growing interest in healthy eating have resulted in a significant increase of the amount of nutritional data available on the internet. This has led to the advent of many websites and communities whose main scope is recipe sharing (e.g. AllRecipes or Yummly). Not only do they provide access to largest sources for food data collections, but they also provide user interactions, ratings, reviews, food relations, and culinary procedures. This volume of data allows food data treatment to face problems of interest to the population such as the already mentioned diseases [1].

Culinary data go far beyond the need for feeding. Food is also closely linked to personal experience and identity. Diets reflect our personal preferences and cultural context [2]. Thus, food science research has become very relevant and extends to many other areas. This especially applies to the joint use of food data to address economical, ecological, and social challenges [3]. Of particular importance here are those computing algorithms which make use of data from users' interaction in social media or online cooking communities to get a better understanding of societal patterns of behavior [4]. This generalized interest, together with the ubiquity of smartphones, has encouraged the use of machine learning

The associate editor coordinating the review of this manuscript and approving it for publication was Biju Issac.

[1] https://www.who.int/news-room/fact-sheets/detail/healthy-diet

techniques for automatizing challenging tasks, such as understanding flavor networks or food culture correlations [5]. This has given rise to the concept of food computing, which refers to the use of food data to improve the quality of life and understanding of human behavior [6].

Food computing has favored the automatization of some complex tasks such as diet recommendation and analyzing this data more accurately [7]. Likewise, it has led to the development of food systems that allow access to recipe collections with a better user experience, where their data is structured, and users can look up recipes, tags, and data of their interest. However, the mentioned systems do not provide adaptations or modifications based on user preferences [8]. Currently, one main research line of food computing is focused on recipe adaptation to allow users to customize recipes and menus to their own needs. These needs can take several forms: a user could change a recipe because of a lack of an ingredient in their pantry, allergy restrictions, intolerance to certain foods, medical prescriptions, dietary limitations, or just because they prefer not to include a specific food in their diet.

To address the adaptation task, food computing systems in the literature have mainly focused on detecting ingredient pairings from cook books and specialized websites. Classical approaches represented substitute relations using ontologies [9]. Most recent works take advantage of the textual representation using data mining and predictive techniques [10], [11]. In our approach, we follow the lines of the latter. Particularly, we focus on extracting the culinary knowledge from the semantics contained in recipe cooking instructions. General domain language models can learn a good representation of vocabulary of general purpose when trained in large datasets. However, the food computing vocabulary is highly specialized with abundant nutrition and cuisine terminology. Given the better results obtained with domain-specific models when dealing with specialized vocabulary [12], we decided to train a domain-specific word embedding model, able to represent specialized food information. We learn the food representations capturing their meaning with a domain-specific word embedding model.

In contrast to related works, which focus on ingredient lists, we consider the cooking instructions in the recipes for learning the cooking structure of food. Thus, we incorporate features like cooking methods and food combinations to find the best substitute in a food database. Furthermore, we also propose a fuzzy metric between documents for finding the best alternative ingredients providing more flexibility than the standard euclidean distance.

The main contributions of the work are the following: (1) a specific domain word embedding representation for food items, (2) an automatized method for matching food and nutrition databases with the ingredients used in a recipe, (3) an unsupervised recipe adaptation algorithm based on user preferences and restrictions. By unsupervised adaptation, we mean that the algorithm does not need human-curated knowledge to detect similarities, since the model is trained with a corpus of unlabeled test to estimate word representations [13]. In this way, the algorithm automatically measures ingredient similarity for the substitutions by using the language model, and then it completes their information with a nutrition database. We evaluated the performance of the proposed method with an online survey, where users checked the adaptation of existing recipes obtained with our approach. Experimental results show the effectiveness of the proposed method for recipe adaptation task.[2]

The rest of the article is organized as follows. Section II reviews the related literature for the recipe adaptation task. Section III explains the methodology followed in this work. Section IV describes the datasets used in the work for both experimentation and validation steps. Section V shows the empirical performance of the proposed approach with some examples to illustrate its behavior. Section VI describes the results obtained in the validation of the method. Section VII analyzes and discusses the implications found in the experimental section. Finally, Section VIII highlights the major conclusions of this work and the future lines of research.

II. RELATED WORK

This section reviews the most relevant approaches followed for recipe adaptation. We place special emphasis on the use of natural language processing techniques for learning relations and patterns between food items. We also describe the word embedding models utilized for food computing tasks.

Recipe analysis studies have been widely performed for a better understanding of cuisine patterns. Flavor networks [14], food and culture relations [15], [16], and multi-model recipe analysis [17] have been applied to several purposes, mostly related to recipe recommendation [18]. A related strand of research has been focused on the study of dietary adaptations involving user preferences, nutritional intake and healthy recipes [6]. To that end, many works have centered on studying recipes and how to adjust them according to those restrictions [19]. In both cases, the literature highlights the need for understanding of ingredient relations in a recipe for its posterior modification [11]. Based on this idea, two main avenues have been explored: using a food-based ontology to model the possible relations between ingredients and using data mining techniques to extract them from recipe texts directly.

Ontology-based approaches are the classical way of addressing this problem. These methods obtain the most similar entities in terms of their food properties and use in recipes. In [20], a simple adaptation process used an ontology to substitute banned ingredients with a random selection of the most similar ingredients subset in the ontology. Instead of considering each ingredient separately, a more straightforward solution was proposed in [21], where the authors developed a recipe retrieval framework able to adapt recipes to specific preferences and dietetic restrictions. An ontology was designed for storing the concept of ingredient with their

[2]The code and materials to reproduce the results are available at: https://github.com/andreamorgar/recipe-adaptation

main properties. These properties included type of ingredient (e.g., pork is an animal meat), composition (e.g., mayonnaise is made of eggs and vegetable oil), and substitution relations (e.g, zucchini is a substitute for broccoli).

Additionally, other classical approaches have also been applied to the adaptation task. In [20], knowledge discovery and case-based reasoning were applied to modify recipes not only including new ingredients but also removing others, so the coherence of the recipe is maintained. The TAAABLE project, a textual case-based cooking system to answer cooking queries [9], deserves a special mention. Among other challenges addressed in the project (e.g., recipe retrieval), the recipe adaptation problem was seen as a response to users' questions provided in natural language. For example, let us assume that the user prefers to use bananas instead of apples in the recipe ''baked apple pie''. The system adapts the recipe using a case-based reasoning engine with a knowledge adaptation database. Adaptation strategies were included to substitute an ingredient for another to satisfy a restriction. These strategies considered adaptation knowledge as specific rules.

The mentioned frameworks have to deal with a major problem. Ingredients substitutes can hardly be obtained directly due to the complexity of food relations [21]. The same ingredient can be used in many diverse recipes (e.g., flour for fried fish and also for making a cake). Ingredients can also appear with different names (e.g., eggplant versus aubergine or prawns versus shrimp). The cooking procedure can also modify the suitability of the ingredients for a determined recipe (e.g., raw tomatoes and lettuce are a good pairing because they often appear together in salads, but this is not the case for fried tomato and lettuce). Consequently, ingredients from the same group of food do not necessarily follow any specific taxonomy, adding complexity to this task.

To overcome this problem, authors have to use rules obtained from a food thesaurus or make this process manually using recipe books. These approaches have three main issues: (1) studying the different combinations available for a recipe is a very tedious task, (2) there is a great difficulty in obtaining a specific-domain ontology validated by experts which stores ingredient substitutions, (3) the scope of these models is limited because it only works with the ingredients already stored in the system. Furthermore, this information has to be included for every new ingredient added to the system [21].

As an alternative, data-based approaches can tackle the complexity of finding suitable substitutes for ingredients in recipes [22]. The outbreak of recipe sharing websites has resulted in a deep food analysis involving cuisines, ingredients, and, their relations. Data mining techniques are used to detect food pairings and ingredient alternatives in culinary texts. In [23], authors considered that the potential substitute ingredients have to be cooked in the same way. Data mining techniques were employed to detect which ingredients had a similar cooking procedure. This information was also considered to improve the adaptation task. In [10], the authors built an ingredient substitute network using user reviews from recipe websites. These sources provided useful data for making recipe adaptations, for example, by giving constructive insights for further recipe improvement. Expressions of interest such as ''replace *a* with *b*'' or ''*a* instead of *b*'' were parsed for this aim. The resulted network can be applied to predict which ingredient combinations are preferable to others.

More recent machine learning methods have been mainly used for the analysis of cuisines and the main relations among ingredients [6]. From a wide perspective, these relations have been studied by using the textual description of foods and flavor networks [24]–[26]. Food items have also been studied by using predictive language models based on embeddings (see Section V-A). Food2vec uses a word embedding model trained with lists of ingredients to understand relations between ingredients and cuisines of the world [27]. Another model also trained in the field is Recipe2vec, more centered on comparing and retrieving recipes, but not publicly available [28]. Both works used predictive algorithms to detect culinary patterns and intrinsic relationships between ingredients, which can be applied to identify food pairings. In [11], the authors used an embedded ingredient representation for the recipe generation framework NutRec. The algorithm inputs were user preferences as an ingredient list. These ingredients were projected into a latent space where the ones that usually appear in a recipe are close to each other. The latent space allowed to identify suitable ingredients for adding them to the initial ingredient set in the recipe completion task. Finally, using an estimation of the needed amount of the ingredients, the model generated a pseudo-recipe and returned the most similar recipe in the database to this one.

There are a few differences between our approach and these related works.

- A significant advantage of our approach is that the adaptation is unsupervised and fully automated, instead of conventional semi-automatic procedures that use a combination of regex patterns, n-grams, and syntactic distances to detect food substitutes.
- In contrast to ontology-based recipe adaptation procedures, we propose to use a word embedding representation to model recipe ingredients and detect food alternatives.
- We go further than previous food-based recipe adaptation procedures such as NutRec, and learn relations between ingredients with whole recipe preparations instead of ingredient lists. Notice that the word embedding model is very appropriate for this purpose and learns the representations directly from these preparations. Thus, we consider the main context of each ingredient in the recipes to obtain a quality representation for foods (i.e., ingredient relations, cooking procedures, dressings, and other intrinsic information such as flavors).

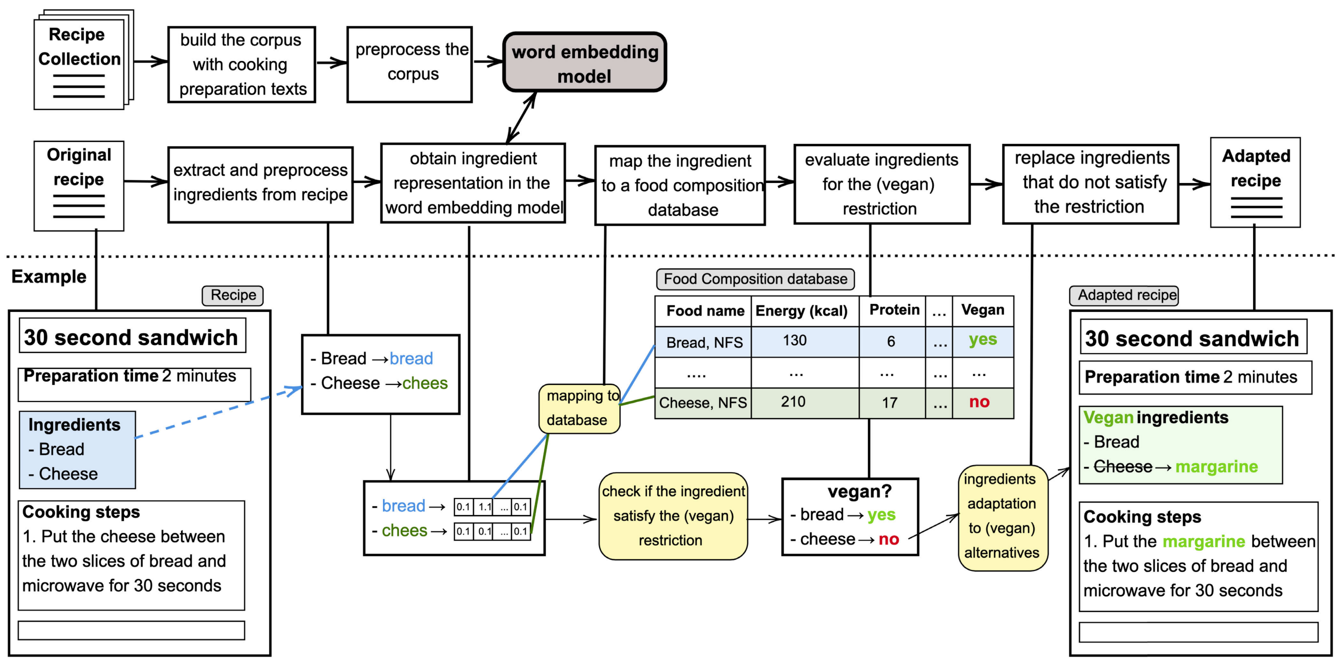

FIGURE 1. A general description of the whole process described in this article. It shows the steps to adapt a recipe given a dietary description. To complete the workflow, we give an example of a vegan adaptation of a recipe whose ingredients do not satisfy the restriction.

III. METHODOLOGY

A. OVERVIEW

Our approach addresses the recipe adaptation problem from the ingredient level, i.e., we adapt a recipe by substituting ingredients for suitable alternatives. Figure 1 details the workflow of the proposed adaptation task. The figure also shows an example of an specific type of recipe adaptation, in this case, to vegan restrictions.

First, we propose to use a domain-specific word embedding model for representing the food items. We used a textual corpus with a collection of recipe cooking preparations for training the model. Using recipe steps instead of the ingredient list, the model takes direct advantage of the context and food descriptions, relating similar ingredients with similar cooking procedures. This fact allowed us to detect both equivalent and substitute relations between foods.

Second, to adapt a recipe according to the user preferences, we took into account the nutritional characteristics of the ingredients. Given that online recipes do not detail nutritional information at this level, it is required their joint use with an external database of foods. We used a fuzzy distance measure to map an ingredient with its corresponding item in the food database. Notice that we used the word embedding to detect equivalence relations between food items, so we could access the complete nutrition values of an ingredient in a food database. Working simultaneously with both sources (i.e., the recipe and the nutritional information), we were able to detect quality ingredients substitutions. The adaptation task also used the embedded food representations obtained with the language model. We exploited the semantic features captured in the vectors with the nutrition information of foods to build recipe variants. In this way, we considered the semantics of the recipe to preserve the essence of the recipe after the modification. The variants can appear as simple alternatives (e.g., a substitute for tomato in a salad), dietary-allowed substitutions (e.g., a vegan version of a meat taco), or favoring user's preferences (e.g., a low-carb version of a dish) among others.

B. WORD EMBEDDING

We used a word embedding model to obtain numerical representations of food terms. This kind of models have proven useful for determining semantic similarity between short texts in nature language processing problems [29], as it is the case of ingredient descriptions. A word embedding is an unsupervised deep learning model that provides numeric vector representations for capturing word semantics [30]. The multidimensional space obtained with the model represents meaningful distances and relations between words. For this, the model uses the context around each word in the vocabulary—assuming that words in similar contexts have similar meanings.

The most widely-known word embedding model is word2vec, a neural network that projects words into a latent space and then reconstruct them based on the context [31]. Word2vec has two main implementations: continuous bag-of-words (CBOW) and Skip-gram [32]. In CBOW, given a window of context words, the model projects them into a latent semantic space and tries to predict the current word. In Skip-gram, the goal is not to predict the word itself but the context surrounding each word. In practice, CBOW outperforms Skip-gram when it comes to frequent vocabulary, while Skip-gram works well with non-frequent words from the corpus.

Another commonly-used word embedding model is fasttext [33]. It is similar to word2vec, but it learns representations for character n-grams instead of for single words. This approach allows taking into account word morphology by processing subword information, which helps to solve the problems posed by the representation of unfrequent words in morphologically rich languages. As a main drawback, fasttext is computationally expensive since it requires considerable memory to create the subword embeddings. A different approach is used in GloVe [34], which is not based on local word contexts but on global co-occurrences of words. While GloVe aims at improving word2vec by avoiding the dependency on short local contexts, in practice they show a similar performance. As it occurrs with fasttext, GloVe also demands considerable memory space because it is trained on the completed word co-occurrence matrix.

Selecting the best word embedding model for a task is usually not straightforward. The analysis of word2vec, fasttext and Glove developed in [35] concluded that there is no word embedding model which outperforms the other ones in every natural language processing task, and that they usually behave similarly when measuring distance between texts. Accordingly, we carried out a similar experiment to compare the outputs and the performance of these three models for the computation of food item similarity. Our study yielded the same conclusion: they behaved similarly in text similarity tasks (see Section V for details). Therefore, we used word2vec due to its simplicity.

It is worth to mention that in the last years, many advanced neural language models have been proposed, such as BERT, Elmo or GPT. Although they are more focused on text-generation and query-answering tasks [36], they could be adapted to be used in our framework. This remains as future work.

Typically, word embedding models are pre-trained with massive general-purpose data obtained from sources such as Google News [37], showing good performance in multiple tasks. Nevertheless, this approach is not suitable for specialized domains, which require fine-grained representation of words. Not surprisingly, word embedding models trained with domain-specific corpora (even of reduced size) have shown better performance in specific tasks than those trained with (larger) general-purpose corpora [12]. Hence, we proceeded to train our word2vec model with a food-related corpus.

In conclusion, we trained the word2vec model with a recipe corpus including cooking instructions which have substantially more information than only using ingredient lists. This way, the word embedding model captures ingredient relations while considering context information such as the cooking method. The model was configured as a shallow two-layer neural network encoding the bag-of-words (CBOW) architecture.

C. INGREDIENT MAPPING

The numerical representation obtained from the word embedding model can be used to detect potential synonyms among the data. Food descriptions scarcely are composed of one word, and in this case, a simple distance between embedded vectors is not sufficient. Thus, the measure has to consider the overall textual description. To deal with language ambiguity, we considered a fuzzy implementation of the distance metric. This metric provides more flexibility and robustness for the mappings. Particularly, we used a fuzzy distance metric which studies semantic similarity between short documents [38]. For this, the metric considers the distance between the vectors obtained with the word embedding model.

Equation 1 shows the similarity metric used in this work. Let S_1 and S_2 be two documents (i.e., two food descriptions). We define T_1 and T_2 as their corresponding token sets (see Section V for further explanation of the token set obtention). The function $\tilde{D}$ calculates the similarity between the token sets. Instead of focusing on word-to-word approaches, textual descriptions are considered as whole entities. We use the Euclidean distance between word vectors to calculate the membership degree of each token t_i to the minim set (see Formula 5). Note that if the membership degree between a token t_i of T_1 to T_2 is 1, there is a coincident term in both descriptions. Thus, we can extend this calculation to obtain the overlap set of T_1 and T_2. Also note that by doing this, we are considering the membership function of the semantic features in the text.

$$\tilde{D}(S_1, S_2) = \frac{\phi_{12}}{\phi_1 + \phi_2 - \phi_{12}} \tag{1}$$

$$\phi_1 = \sum_{x \epsilon T_1} (\mu T_1)(x) \tag{2}$$

$$\phi_2 = \sum_{x \epsilon T_2} (\mu T_2)(x) \tag{3}$$

$$\phi_{12} = \sum_{x \epsilon T_1 \cup T_2} \min(\mu T_1)x \times \min(\mu T_2)x \tag{4}$$

$$\mu_{T_i}(x) = \begin{cases} 1, & d_E(t_i, x) = 0 \\ 0, & d_E(t_i, x) = \infty \\ h\left(\dfrac{1}{d_E(t_i, x)}\right), & 0 < d_E(t_i, x) < \infty \end{cases} \tag{5}$$

where $d_E(t_i, x)$ is the Euclidean distance between t_i y x and h is the sigmoid function.

D. RECIPE ADAPTATION

The steps described above can be applied to the search for similar ingredients for the adaptation task. Figure 1 shows a complete example of the adaptation procedure. First, we extracted the ingredients from the recipe. The recipe objects are structured, and there is a specific field for the list of ingredients used in the cooking procedure. Then, we obtained their semantic representation, and we used it to map them to the food composition database. With this step, we accessed the ingredient nutritional information for later adaptations. Note that for the mapping task, we represented the items in the database with the embedding model as well. We used these embedded representations for finding alternatives to the ingredients, thus obtaining an adapted recipe. We considered three main ways to adapt a recipe:

1) Similarity-based adaptation: we provided alternative ingredients based on the similarity relations obtained with the domain-specific word embedding model. Given that the model contains intrinsic information of the ingredients relations, we created successful ingredient modifications.
2) Preference-based adaptation: we provided alternative ingredients to the ones given in the recipe, considering the preferences of the user. For example, if the user wants to lose weight, they may require alternative low-carbs ingredients to adapt a meat recipe. Another example could be a professional athlete who could adapt a recipe looking for high protein alternatives for the ingredients of the recipe.
3) Adaptation based on food restrictions: in this case, we refer to those users that have food constraints. For example, a vegetarian or a vegan user would search for an adaptation of a hamburger recipe. Another example is a user with specific allergies or food intolerances. In this case, we adapted the recipe by changing the not allowed ingredients for suitable alternatives that could easily suit that recipe. We evaluated each ingredient in a recipe to determine if they comply with a specific food restriction. For those that did not fulfill the requirements, we replaced them with an alternative option that satisfies the constraint. Figure 1 shows an adaptation of this type (vegan restrictions).

Notice that, as a novelty, we employed a word embedding model to adapt a recipe. Thus, the process is fully automatic since it does not require an additional knowledge extraction step for building a knowledge adapting database.

E. VALIDATION

The process of evaluating a recipe is subjective and depends on many factors like culture, flavors, and personal taste. This fact makes it difficult to decide if an adapted recipe is correct or not. For that, we developed an online survey where users validate and also evaluate the adequacy of a collection of adapted recipes. To avoid subjectivity and exclude the influence of external factors, e.g., personal preferences or cultural influence, the users were instructed to evaluate the correctness and coherence of the recipe adaptations. Therefore, surveyed users did not assess the recipe flavors according to their personal taste, but to general cuisine conventions. To do so, they were restricted to only decide if the substitutes were coherent and feasible. We also addressed the noise derived from recipe comprehension. For this, we added a section in the validation interface to indicate if the user understood the recipe. In this way, we can discard unreliable reviews.

We considered four types of adaptations: based on similar ingredients (the method returns suggestions for each ingredient on the recipe), preference-based (the method returns low-calorie alternatives for each ingredient), and vegetarian and vegan alternatives, where the method returned alternatives for those ingredients not vegetarian or veggies. In the two latter cases, we also shown food alternatives for those ingredients of the recipe that are allowed in these diets.

IV. DATA

In total, we employed three food datasets. Particularly, we used two different recipe corpus for training the word embedding model and validating the recipe adaptation task, respectively. Using recipes of different origin allowed us to assess the generalization capability of the model. The third dataset corresponds to a database of foods, which we used for obtaining the nutrition values of the ingredients involved in the adaptation.

1) TEXTUAL CORPUS FOR TRAINING THE WORD EMBEDDING

To train the food word embedding model, we built a textual corpus using a recipe dataset published on the archive.org.[3] In this collection, there are 267,071 recipes scraped from the following recipe sites:

- AllRecipes[4]: a social network to share recipes, cooking tips, and media, to inspire users to create new recipes.
- BBC Food Recipe[5]: a collection of recipes from the chefs and programs of the BBC. It contains recipes classified by season, festivities, and ingredients, among others.
- CookStr[6]: recipe website which organizes cookbooks and recipes to make them universally accessible.
- Epicurious[7]: digital brand focused on food and culinary art. It has over 300,000 recipes, videos and tips for daily cooking.

The archive.org recipe collection is organized in four JSON files, each one of them corresponding to one of the food sources enumerated above. Table 1 shows the structure of one of the recipes contained in the archive.org dataset. We used the "instructions" field of each recipe in the dataset to build the training corpus.

2) RECIPE CORPUS FOR THE ADAPTATION TASK

We used the Kaggle dataset called Food.com Recipes and Interactions[8] for doing the experimentation with the adaptation task. This dataset contains recipes obtained from

[3]https://archive.org/download/recipes-en-201706
[4]AllRecipes website: https://www.allrecipes.com/
[5]BBC Recipes website: https://www.bbc.co.uk/food/recipes
[6]CookStr website: https://www.cookstr.com/
[7]Epicurious website: https://www.epicurious.com/
[8]www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions

TABLE 1. A Recipe From Archive.org Dataset. It Contains the Whole Structure for That Recipe.

Field	Content
author	"Stephanie"
cook_time_minutes	25
description	"I just started adding my favorite things to basic cornbread and I came up with something great!"
error	false
footnotes	
ingredients	"1/2 cup unsalted butter, chilled and cubed", "1 cup chopped onion", "1 3/4 cups cornmeal", "1 1/4 cups all-purpose flour", "1/4 cup white sugar", "1 tablespoon baking powder", "1 1/2 teaspoons salt", "1/2 teaspoon baking soda", "1 1/2 cups buttermilk", "3 eggs", "1 1/2 cups shredded pepperjack cheese", "2 ounces roasted marinated red bell peppers, drained and chopped", "1/2 cup chopped fresh basil"
instructions	"Preheat oven to 400 degrees F (205 degrees C). Butter a 9x9x2 inch baking pan.", "Melt 1 tablespoon butter in medium nonstick skillet over medium-low heat. add onion and saute until tender, about 10 minutes. Cool.", "Mix cornmeal with the flour, baking powder, sugar, salt and baking soda in a large bowl. Add 7 tablespoons butter and rub with fingertips.", "Whisk buttermilk and eggs in a medium bowl to blend. Add buttermilk mixture to dry ingredients and stir until blended.", "Bake cornbread until golden and tester inserted comes out clean, about 45 minutes. Cool 20 minutes in pan. Cut cornbread into squares."
photo_url	"http://images.media-allrecipes.com/userphotos/500x315/582853.jpg"
prep_time_minutes	55
rating_stars	4.32
review_count	46
time_scraped	1498204021
title	"Basil, Roasted Peppers and Monterrey Jack Cornbread"
total_time_minutes	100
url	"http://allrecipes.com/Recipe/6664/"

Food.com,[9] and it is oriented to the study of recipes and their interactions among users. Thus, it consists of two collections (i.e., the recipe, and the user interactions collections). We used the file corresponding to the raw recipes for experimentation, as the recipe set for adapting recipes. Besides, we employed different recipe sources for making adaptations and training the word embedding model. In this way, we ensured that our results with this module were not compromised. The recipes in the collection are described in English and includes all the recipe attributes required for adapting (i.e., each recipe includes ingredients and preparation steps). Table 2 describes the structure of the data. It shows the attributes stored for each recipe in the dataset. Those in bold are the ones we used for the adapting task. Table 3 shows some examples of the recipe data we used for the adaptation task.

[9]www.food.com

TABLE 2. Structure of Data in the Food.com Recipe Dataset. Fields in Bold are the Ones We Use in This Article. We Used This Dataset for Validating our Method. We Used the Recipes in it to Obtain New Versions of Them.

Field	Content
name	recipe name
id	recipe ID
minutes	minutes to prepare the recipe
contributor_id	user ID who submitted this recipe
submitted	data recipe was submitted
tags	tags for recipe
nutrition	nutrition information (calories (#), total fat (PDV), sugar (PDV) , sodium (PDV) , protein (PDV) , saturated fat
n_steps	number of steps in recipe
steps	recipe steps
n_ingredients	number of ingredients in recipe
ingredients	recipe ingredients
description	user-provided description

TABLE 3. Examples of Recipes From Food.com Dataset. The Columns Correspond to the Columns Used in the Adaptation Process.

Name	Steps	Ingredients
better cake mix	['add these 3 ingredients to boxed caked mix', 'mix as usual']	['butter', 'flour', 'baking powder']
30 second sandwich	['put the cheese between the two slices of bread and microwave for 30 seconds']	['bread', 'cheese']
bacon and tomato spaghetti	['brown bacon till crispy , you can drain some fat if you want but it gives it flavor , add tomato soup', 'cook noodles and add to pan , heat threw']	['condensed tomato soup', 'bacon', 'spaghetti noodles']

3) NUTRITION DATABASE OF FOODS

We used a third dataset consisting of a food composition database. This dataset allowed us to access the nutritional information of the ingredients. For this work, we used the Composition of foods integrated dataset (CoFID). This dataset is maintained by the Public Health England (PHE) agency from the Department of Health and Social Care in England, with the purpose of bringing together all the available data as a single, consolidated dataset [39]. We chose this database because it is a well-known open-access food database of reference with multiple food variety and representations. The coFID dataset consists of fifteen tables that contain nutritional information of 2913 foods of different kinds: eggs, vegetables, fruit, nuts and seeds, and many others. In this work, we have used the "proximates" table, which allows us to access nutritional tags of commonly consumed foods. In Table 4, we can see a simplified structure of this table.

V. EXPERIMENTS

A. WORD EMBEDDING MODEL

We used the archive.org corpus presented in Section IV to train the word embedding model. Thus, we ensured a suitable domain-specific representation of the items. As introduced, we built a corpus with the cooking instructions of each recipe by using the "instruction" field. As shown in Table 1, this data was not in a directly manipulable format for training a

TABLE 4. Some Food Items Stored in coFID Database. Rest Items Correspond to Nutritional Features Such as Energy (Kcal), Vitamins, and so on. We Use This Dataset for Consulting the Nutritional Tags of the Ingredients in the Adaptation Task.

Food Code	Food Name	Description	Group	Energy (kcal)	...
13-157	Asparagus, raw	Tough base of stems removed	DG	25	...
18-005	Beef, fat, average, cooked	Average of 8 different cuts	MAC	533	...
19-551	Beef Stroganoff, homemade	Recipe	MR	171	...

language model. Therefore, it was necessary to apply data preprocessing techniques to restructure the textual data to use them as an input in the word embedding training step. For this, a file was generated for every recipe, storing their instruction steps as plain text. Before the training step, we read all files to assemble the corpus properly. Figure 2 shows this procedure.

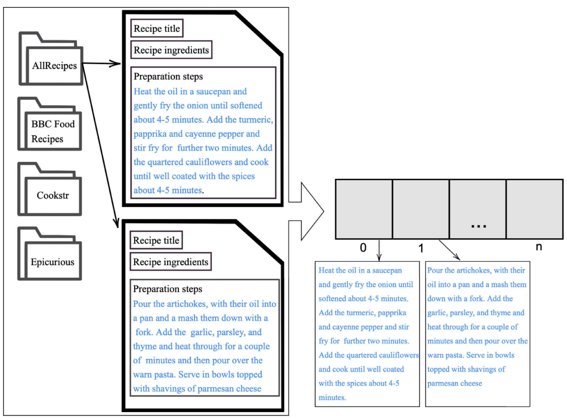

FIGURE 2. Procedure to build the corpus from the data source.

Table 1 also shows that the recipe preparation steps are raw text. To achieve quality results in our experiments, we need to clean the cooking instructions text. For this, we used the topic modeling features included in the Gensim python library.[10] Figure 3 shows the cleaning process we apply to the cooking instructions. The last step deserves a special mention, where the bigram model is capable of detecting compound words such as "tortilla chips". Table 5 shows other compound words found in the corpus. Notice that using this specialized corpus, we can identify common expressions that often appear in the recipes, such as "small bowl". With this, we can better handle the food descriptions that usually appear in cooking recipes.

1) Remove punctuation, digits, and symbols: we removed all non-alphabetic characters from the text. In this step, we also converted the remaining words to lowercase.
2) Tokenization: we obtained the preparation steps as a list of tokens when each of them represents one word of the raw text.

[10]https://radimrehurek.com/gensim/

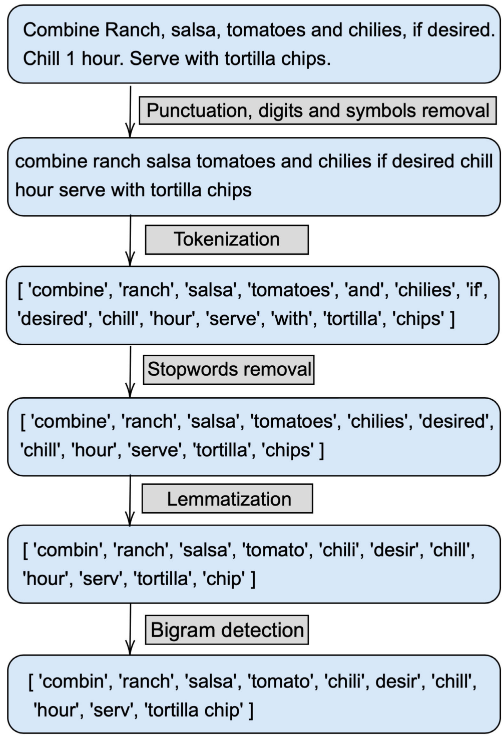

FIGURE 3. A step-by-step example of the cleaning process with the cooking instructions of a recipe from the corpus. The first square corresponds to the original cooking instructions text and the last one to the token list used for training the word embedding.

TABLE 5. Some Examples of Bigrams Detected in the Word Embedding Model. The Model Can Identify Both Compound Words (e.g., Brown Sugar) and Expressions That Often Appear in the Culinary Language (e.g., Cook Time).

Type of bigram	Food bigrams	
words that commonly appear together	room temperature	preheat oven
	mixing bowl	before serving
	medium saucepan	cook time
	stirring frequently	medium heat
compound words	brown sugar	vanilla extract
	chili powder	food processor
	vegetable oil	black pepper
	apple cider	green beans

3) Stopwords removal: stopwords are usual tokens that do not provide useful information in the vocabulary, and therefore it is advisable to dispense with them (e.g., "the", "a" or "your") [40]. For this, we used the English stopwords list provided by the Gensim library, removing them from the recipe corpus.
4) Lemmatization: we applied lemmatization to reduce all terms to their roots. The different morphological word variations are transformed into a unique common form (e.g., "cooked" and "cooks" are both treated as "cook").
5) Bigram detection: in natural language, we commonly find words belong together. Culinary language is not an exception, and some examples are "black pepper", "red wine" or "brown sugar". We trained a bigram model in the corpus to detect pairs of words in the data.

In this way, if the bigram model recognizes that ''brown sugar'' are two words that tend to appear together, the word embedding model will consider it too.

We learned the embedded representations of the words in the corpus during 30 epochs, the window size was fixed to 5 and the vector size of the vectors to 300. These hyper-parameter values were chosen after experimentation. Also, the model ignored the terms that appear less than three times in the learning phase. This step is performed internally in the Gensim implementation of the algorithm. Finally, the model vocabulary resulted in 11,288 words.

Table 6 shows the most similar items for five given words in the vocabulary. In Figure 4, we replicated this same process and obtained a visualization of them in the semantic space. Due to the high dimensionality of the learned vectors, we generated a two-dimensional visualization using the t-distributed stochastic neighbor embedding algorithm [41]. With the visualization, we can evaluate the adequacy of the embedded representations. In this case, the most similar items for each word are in the same Voronoi partition. Each Voronoi partition contains one group of different foods. This partition shows a previously selected food (in bold) and the five more similar food items obtained with the embedding model. Notice that food items definitions are preprocessed with the cleaning task explained above.

TABLE 6. Experiments With the Word Vectors: We Use the Vectors to Detect the Most Similar Items.

Token	Most similar tokens in vocabulary
garlic	garlic gloves, onion, shallot, garlic powder, turmer, cumin
potato	sweet potato, cauliflow, potato parsnip, turnip, parsnip, squash
lemon	lime, orang, cilantro_lime, tahini lemon, lemon zest, lime_zest
pepper	black pepper, parslei, paprika, red_pepper, oregano, tarragon
mozzarella	parmesan, provolon, pepperoni, cheddar, fontina, swiss

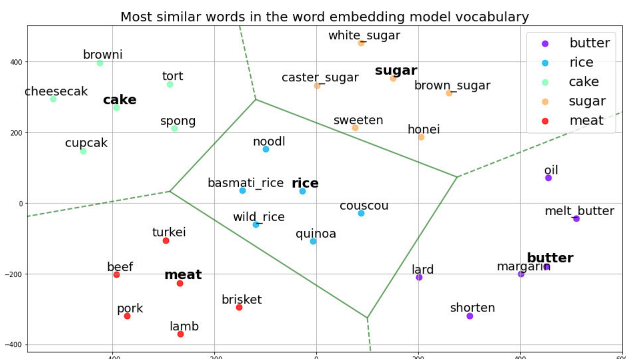

FIGURE 4. A visualization of the model. Voronoi partitions allow us to identify semantic regions in the space. It shows the most similar words in vocabulary given five: butter, rice, cake, sugar, meat (in bold). Note the relations between the rice partition and the rest.

B. WORD EMBEDDING COMPARISON

We performed a comparison of the behavior of different word embedding models to decide which one was the most suitable for our problem. For this, we decided to train different word embeddings setting the same recipe corpus and hyper-parameters (i.e., 30 epochs, window size fixed to 5 and vector size of 300) for the training step. Specifically, we performed the experimentation with the word embedding models word2vec (CBOW implementation), fasttext, and GloVe.

We studied the mappings obtained with each one of them to test if there were substantial differences among the resulting mappings. For this, we extracted the 1000 most habitual ingredients from the Food.com recipe dataset, and we obtained the mapping of each of them to the coFID database. Figure 6 shows a pairwise comparison of the number of coincident mappings (y axis) of both models after applying different distance thresholds (x axis). Each line corresponds to the number of coincident mappings of *a vs b* considering only the very same result obtained by *a* and *b* (blue), a coincidence in the mapping by *a* with any of the first two mappings by *b* —and vice versa— (black), and a coincidence in the mapping by *a* and any of the first three mappings of *b* —and vice versa— (orange).

The results show that the three models have very similar performance, as a large percentage of the mappings obtained with the models' match. Specifically, the best match is almost always contained in the best three of the others. Also, when the absolute distance value obtained by a model is low, i.e. the model is totally confident in the mapping, the three models are coincident. Finally, we measured the inter-rater agreement of the three models, simultaneously. We calculated the Fleiss' Kappa coefficient to see how equivalent are the answers given by the models [42]. This coefficient is an extension of the original Kappa coefficient. In this case, the relative observed agreement among raters and the expected probability of agreement are computed for each rated pair and then averaged. We obtained a value of Fleiss' Kappa of 0.6248, which means that there is a substantial agreement between the models [43].

Our results are in line with the conclusions presented in [35]. In this work, the authors developed an exhaustive comparison of word embedding model performance in text similarity tasks over 13 datasets. As already discussed in Section III, they concluded that the three models obtained similar results, being more similar the ones reached with word2vec and fasttext. Given the high similarity among the models' performances, we decided to continue the work with word2vec. This decision was based on the simplicity of the model and the sound performance obtained in previous food computing works [27].

Results show that there is a high degree of agreement between the algorithms.

C. INGREDIENT MAPPING

As introduced in Section III, we mapped the ingredients of the recipe to a food database to obtain their nutrition tags. We used for this purpose the food database coFID. As seen in Table 4, this database contains raw foods, cooked foods, and also full recipes of dishes. We only used food ingredients, so we preprocessed the database to work only with foods.

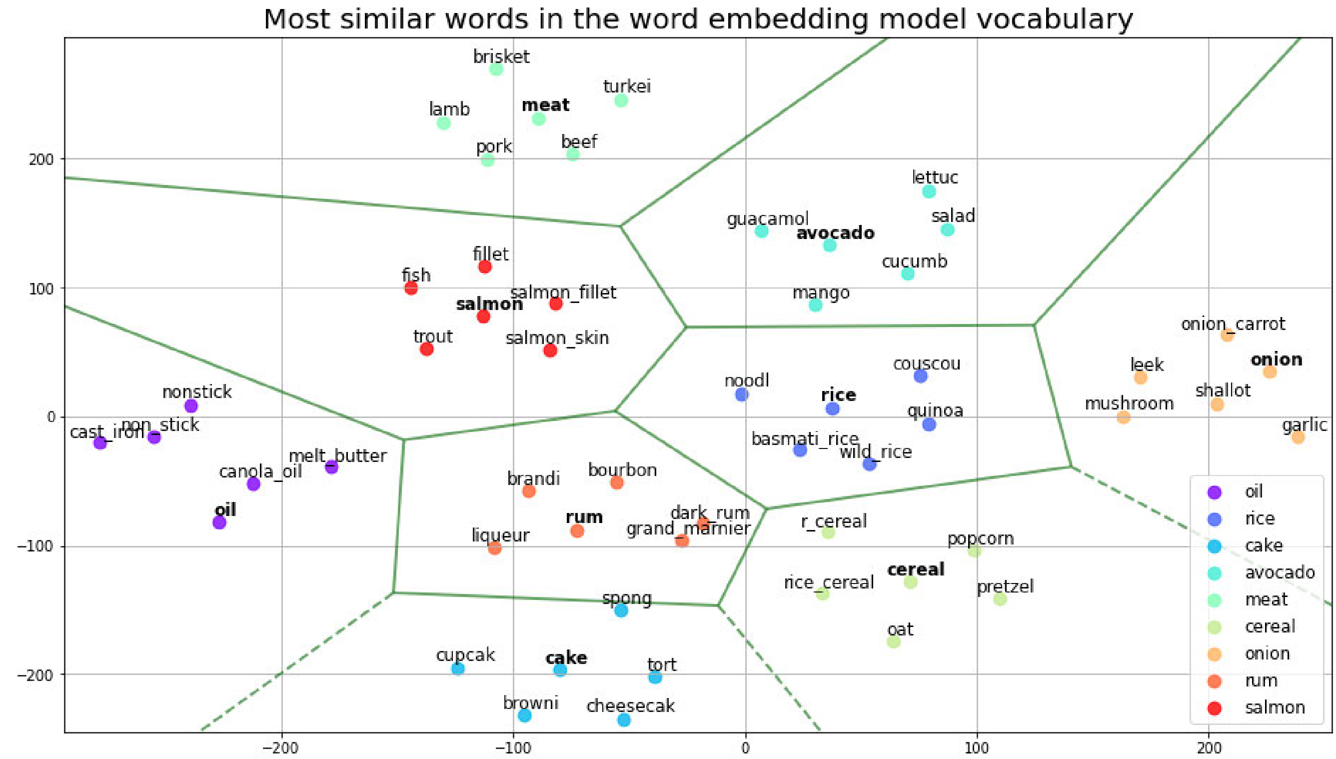

FIGURE 5. A visualization of the model. Voronoi partitions allow us to identify semantic regions in the space. It shows the most similar words in vocabulary given nine: oil, rice, cake, avocado, meat, cereal, onion, rum, salmon (in bold).

Thus, we removed the items whose food group correspond to dishes and cooked recipes (e.g., the group ''fish products and dishes''). We also removed the human milks, to avoid suggesting them in recipes.

Given a recipe from the Food.com dataset, we extracted their ingredients and we identified them in the coFID food composition database. For this, we first obtained the embedded representation of the ingredients. Thus, we preprocessed each one of them, cleaning the text with the same procedure used for the training corpus in the word embedding model. Notice here that an ingredient could be represented for more than a word. We obtained the embedded representation for each one of the terms involved in the food description. We applied the same process to the terms in the food composition database. Finally, we used the document mapping function described in Equation 1 to find the best match for each ingredient. Table 7 shows some representative food mappings reached with this approach. It is divided in four blocks ordered by the accuracy of the mapping. First block shows examples of quality mappings, and the fourth shows cases where the mapping function is not able to detect a good equivalent.

D. RECIPE ADAPTATION

Once we have linked the recipe ingredients with a food composition database, we have enough nutritional knowledge to adapt recipes to food preferences and restrictions.

We distinguished three different kinds of making recipe adaptations. The first possibility is to modify the ingredients of the recipe for other foods also suitable to the original ones. In this case, there is no preference included that the final recipe must satisfy. For this, we made use of the mapping function to find the most similar items stored in the food database. Notice that by doing this, we are using the semantic features captured in the food vector representations. Secondly, for the preference-based adaptation, we included an order relation to the mapping function. With this, we prioritized a given preference when obtaining the most suitable alternative for each ingredient. In this work, we illustrated this kind of adaptation by considering the calorie content of the recipe ingredients. We adapted an existing recipe by giving low-calorie ingredient alternatives. Finally, for adapting recipes to food restrictions, we focused on adapting recipes to vegan and vegetarian dietary restrictions. For this purpose, we added two new columns to the food composition database to register if an food item is vegan or vegetarian friendly. We used the food group attribute to determine this value (e.g., ingredients tagged as ''egg'' are vegetarian but not vegan). We also reviewed the mappings in the search for wrongly classified items. Table 8 shows how the data would result after this modification. Finally, Table 9 presents examples of adapted recipes with the proposed method. It shows an example of each kind of adaptation covered in this work. The recipe adaptations correspond to

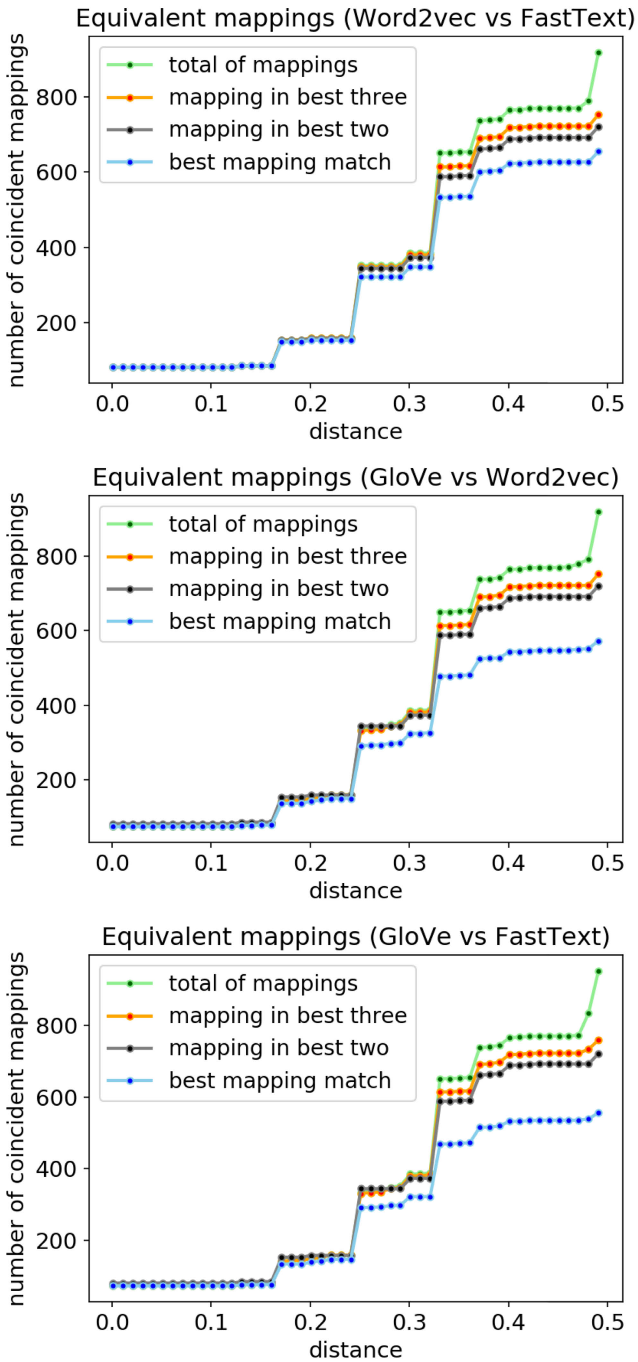

FIGURE 6. Plot visualization of the word embedding algorithms' comparison for mapping food items. Figures show the coincident mappings obtained with the Word2vec, fasttext, and GloVe models compared two by two. The green line (labeled as "total of mappings") represents the total number of mappings obtained after taking *x* as the threshold distance to consider a mapping valid. The blue line (labeled as "best mapping match") represents the number of coincident mappings. The black line (labeled as "mapping in best two") represents the number of semi-coincident mappings in which the best value for one model is within the two best values of the other model (and vice versa). The orange line (labeled as "mapping in best three") represents the same as the orange line but considering the best three mappings instead of two.

similarity-based, preference-based, and adaptation to vegetarian and vegan restrictions, respectively.

TABLE 7. Some Mappings Obtained With the Document Distance Metric. The Table is Divided in Three Main Blocks, Each One of Them Representing Different Kind of Results. First Block Correspond to Cases Where the Mapping was Exact. Second Block Correspond to Cases Where the Mapping was Good, but There was a Different Level of Detail in the Mapped Foods. Third Block Correspond to Mappings Where the Main Food was Correct. Fourth Block Shows Foods with an Inadequate Mapping.

	Ingredient from recipe	Resulted mapping in database
(1)	Lemon juice	Lemon juice, fresh
(2)	Sweet pickle relish	Pickle, sweet
(3)	Cheddar cheese	Cheese, Cheddar, English
(4)	Olive oil	Oil, olive
(5)	Onion	Onions, raw
(6)	Mustard	Mustard, wholegrain
(7)	Sea salt	Salt
(8)	Ground cinnamon	Cinnamon, ground
(9)	Water chestnut	Water chestnuts, raw
(10)	Unsalted butter	Butter, unsalted
(11)	Brown rice	Rice, brown, wholegrain, raw
(12)	Spring onion	Spring onions, bulbs only, raw
(13)	Ground beef	Beef, rump steak, from steakhouse, lean only
(14)	Long grain and wild rice blend	Rice, white, long grain, raw
(15)	Frozen mixed vegetables	Vegetables, mixed, frozen, boiled in unsalted water
(16)	Cheese	Cheese, Caerphilly
(17)	Soy sauce	Soy sauce, light and dark varieties
(18)	Yellow curry paste	Curry paste
(19)	Light coconut milk	Coconut milk
(20)	Vegetarian refried beans	Beans, papri, raw
(21)	Almonds	Almonds, toasted
(22)	Granulated sugar	Sugar, Demerara
(23)	Eggs	Scotch eggs, retail
(24)	Flour	Flour, gari (cassava flour)
(25)	Grape leaves	Grapes, average
(26)	Espresso	Coffee, cappuccino, latte
(27)	Potatoes	Potato rings
(28)	Cream	Liqueurs, cream
(29)	Carrots	Carrot juice
(30)	Seedless grapes	Grapes, average
(31)	Butter	Butter, unsalted
(32)	Milk	Milk, whole, UHT
(33)	Fish sauce	Chilli sauce
(34)	Truffle oil	Oil, wheatgerm
(35)	Chili peppers	Pepper, white
(36)	Ground cloves	Allspice, ground
(37)	Egg yolks	Lager, alcohol-free
(38)	Baking soda	Cod, flesh only, baked
(39)	Mini chocolate chip	Peanuts, raisins and chocolate chips
(40)	Frozen pie crust	Pie, apple, pastry, double crust, retail

VI. RESULTS

We chose a random set of recipes from the Food.com dataset. This dataset is plenty of desserts, sauces, and potato-based dishes, many of them using very similar ingredients. We selected 20 recipes of each kind of adaptation that represented a considerable variety of recipes like desserts, pasta, drinks, recipes with vegetables or breakfast and brunches recipes. Then, we developed a website to allow users to evaluate the recipe. For this, we shown the original and the adapted version of a recipe so the user could see the changes and suggestions obtained with our approach.

TABLE 8. Some Food Items Stored in coFID Database After Adding the Attributes "Vegan" and "Vegetarian". Rest Items Correspond to Nutritional Features Such as Energy (Kcal), Vitamins, and so on. We Use the New Attributes to Consult the Suitability of Using a Specific Food in a Vegetarian or Vegan Recipe.

Food Code	Food Name	Vegan	Vegetarian	Energy (kcal)	...
13-157	Asparagus, raw	Yes	Yes	25	...
18-005	Beef, fat, average, cooked	No	No	533	...
12-354	Cheese, Danish blue	No	No	342	...

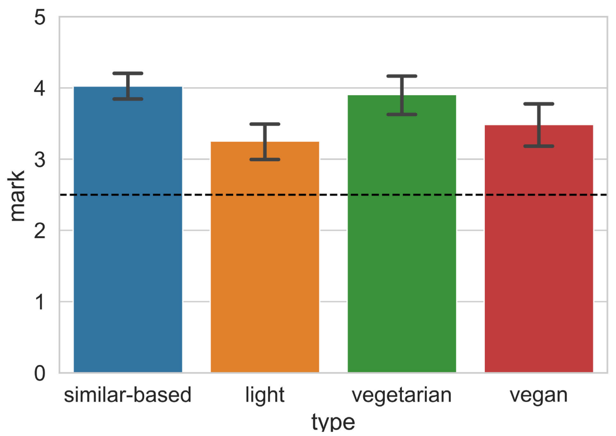

FIGURE 7. Plot visualization of the survey results based on the adaptation type. The figure shows the confidence interval for future recipe evaluation with a confidence level of 95%.

We tested the adapted recipes by conducting a survey into a group of 40 citizens with a general knowledge of cooking. Each user validated a random selection of five recipes of each adaptation type. Also, we developed questions to capture the opinion of the adapted recipe. The users evaluated each recipe with a satisfaction level between 0 and 5 (where 0 is the worst, 2.5 is acceptable, and 5 is the best mark). It also included an overall evaluation and additional suggestions for better adaptation. Additionally, we included a question about their understanding of the recipe to determine the validity of their evaluation.

We obtained a total of 590 reviews from 40 users. Among them, 8 reviews indicated a lack of understanding of the recipe. Thus, we excluded them from the analysis. Figure VI shows the average mark obtained in each group of recipes. Regardless of the kind of adaptation, the adapted recipes outperform the 2.5 satisfaction degree with a confidence level of 95%. Notice that the results obtained with the similar-based type corroborate the sound performance of the word embedding model. Figure 8 shows the histogram of the average recipe mark obtained with the survey. It shows that the mode of the evaluations is superior to 2.5. Also, for each type of adaptation, the distribution is left-skewed and thus, recipe marks tend to concentrate in the good ratings. We can see again from Figure 8 how similar-based adaptation achieves the best performance.

VII. DISCUSSION

A. WORD EMBEDDING

The success of predictive language models to deal with domain-specific problems has been widely explored in the literature. In this article, we have seen that word embedding models can be also applied to detect lower level relationships: the ingredients. When working with ingredients, there are far more relations than the ingredients that participate in the same recipe or specific cuisines of the world. With our approach, the adapting process is fully automatized. Using the semantic information captured with word embedding models, we can obtain successful substitutions of foods. Using cooking instructions for training, the model considered intrinsic knowledge in a recipe. Thus, we were able to achieve quality food substitutions that in another way would be far more complex to obtain.

Figure 4 shows the effects of using recipes for modeling our data. We can see how the ingredients contained in the Voronoi partitions have a relation of similarity. For example: "rice", "wild rice", "basmati rice" or "salmon", "salmon skin" and "salmon fillet" share partitions. Also, we can see these semantic relations within the partitions. For example, the cereal partition shows another interesting relation: rice cereal is in a middle way between rice and cereal, whereas popcorn is farther from rice. Also, Figure 4 shows an interesting relation between partitions. We can see how the partition corresponding to "rice" is in the middle of the other four. This is particularly interesting because rice shares a special relationship with the elements of the other partitions. For example, rice is used in desserts (see "cake" and "sugar" partitions) and also in lunch meals (see "meat" and "butter" partitions). Also, notice that sweet foods appear on the top of the graph, while the saltier ones appear on the bottom part. Additionally, "butter" and "cake" partitions have strong semantic relations (they tend to appear near each other). This case might occur because butter is a common ingredient in desserts.

Figure 5 shows another representation of the model, in this case with a larger number of partitions. We can also appreciate how the sweetest food partitions are closer to each other. Besides, other relations are represented, like the relation between salmon and meat partitions (the distance between "salmon" and "meat" partitions is short, and the same occurs for "rice" and "cereal"). Thus, the semantic space visualizations show that the word embedding model works remarkably well with this kind of data. This fact encourages the usage of these learned features to feed the mappings. Also, we can use this visualization to get a better understanding of the data. However, ingredients partitions are rare to found, and even sometimes can give to food sets with no intuitive relations. This is due to the fact that an ingredient hardly is combined with the same type of ingredients. For example, flour, is used in desserts, but also for frying fishes. Cultural influence also could cause that the relations between ingredients do not appear clear.

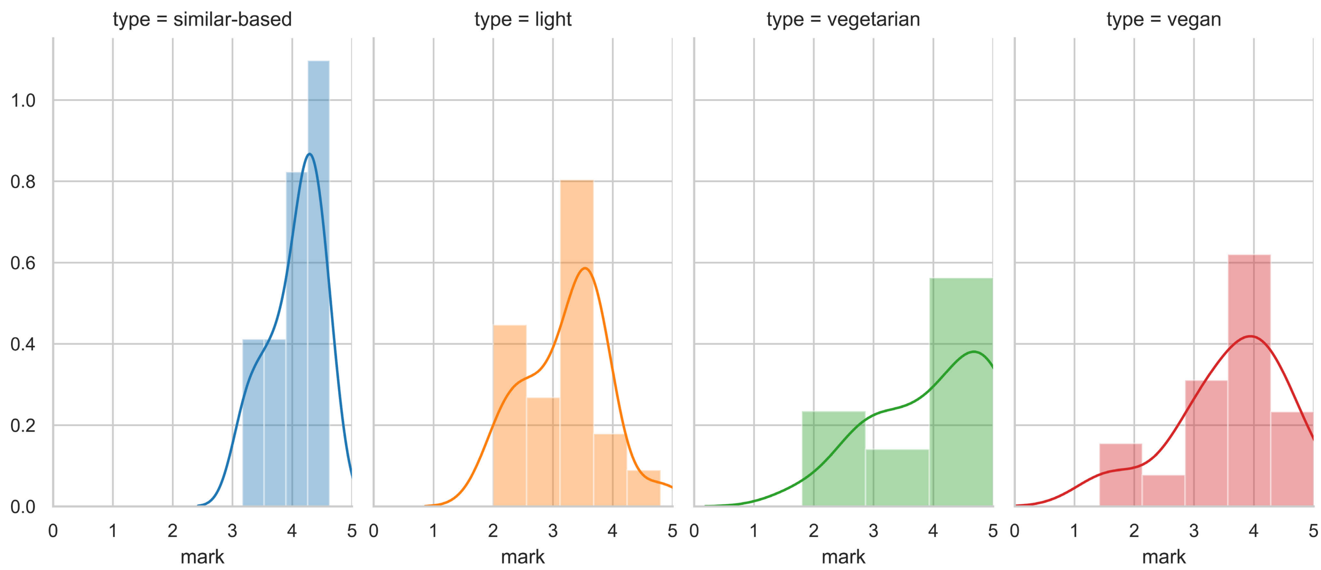

FIGURE 8. **Histograms of average recipe score based on the adaptation type. We also plot the density curves.**

B. INGREDIENT MAPPING

Experiments have highlighted the effectiveness of the approach to extend the information of the ingredients. Table 7 shows some mappings obtained with the fuzzy document metric. First and second blocks corresponds to some of the best mappings achieved with the metric. We can see how we are able to detect ingredients in the database, even when a more detailed description is given (see rows (11) and (12) in the table). The mappings show the good behavior of the mapping function to detect similar food items. For achieving coherent results, it is essential the role that plays the semantic space representation given by the word embedding model. Also, notice that we have used a metric that works well with short description texts. This fact facilitates dealing with food descriptions and ingredients.

Recipe datasets contain not validated recipes uploaded from users from all over the world. This fact causes that they have incorrectly spelled words, not detailed descriptions, and colloquialisms. Besides, users could include foods hard to find an equivalent in a food database. Combining the semantic information of the word embedding with a fuzzy metric based on sets allows us to find quality mappings. The second block of Table 7 shows mappings of this nature, where the metric achieves good correspondences in spite of the differences in the level of detail of the foods. For example, this occurs in row (23). It shows an specific type of eggs, because it is not able to find a less detailed egg item in the database.

Besides, these problems also lead to inaccurate mappings. Third block of Table 7 show some examples where the best mapping is not achieved, but the model is capable of detect the main ingredient. Finally, the four block shows some incorrect mappings. Notice that the obtained results depend on the database scope. The database must contain a representative and actual collection of foods to achieve quality mappings. In our problem, it is essential that the database contains the ingredients (or equivalents) as well as good food alternatives to them. Given the magnitude of the recipe data, we can not ensure that all the ingredients that appears in a recipe are contained in a food dataset. Thus, the mapping for these situations could be inappropriate. In our case, the database contains the most common ingredients in the kitchen, so we have no problems finding great ingredients and substitutes there.

C. RECIPE ADAPTATION

Finally, both the word embedding method and the ingredient mapping are combined to obtain recipe adaptations. Our approach relates food information from heterogeneous sources, enriching the nutrition knowledge of the recipe nutrients. It opens many possibilities to adapt recipes to specific requirements. For example, by using nutritional knowledge of food databases with the mapping functions, we could have access to many nutrient values such as sugar, starch, or salt. Thus, we can enrich the adaptation method to be able to provide powerful adaptations for assisting medical recommendations, food allergies and intolerance, and nutritional advice.

Table 9 shows an adapted recipe for each one of the types covered in this work. For the first recipe (i.e., ‘‘bacon cheddar spread’’), the method returns ingredients alternatives similar to the ones appearing in the recipe. Our method can detect different kinds of cheese (for better suggestions), dressing, and foods. Also, when an ingredient is well-known, food databases usually store it in many cooked forms. The method detects them as the better candidates to become food substitutes due to their similarity with the original

TABLE 9. Examples of Each Kind of Recipe Adaptation Covered in This Work. Similarity-Based (it Suggests Food Alternatives That May Fit in the Recipe). Preference-Based (it Suggest Some Low-Calorie Food Alternatives). Vegetarian and Vegetarian Restrictions for Proposing Ingredients Allowed in These Diets. Bold Ingredients are Recommendations That Must be Considered for Satisfying the Vegetarian/Vegan Restriction.

Recipe	Adaptation type	Ingredient	Adapted ingredient
(1) bacon cheddar spread	similarity based	bacon	chorizo • ham • piccalilli
		cheddar cheese	cheese, derby • cheese, caerphilly • cheese, cheddar type, 30% less fat
		onion	onions, baked • onions, dried, raw • onions, fried in lard
		salad dressing	Dressing, French • Salad cream • Tripe, dressed, raw
(2) eggs rothko	preference-based (low-calorie foods)	butter	butter, spreadable (75-80% fat) • butter, salted • cocoa butter
		brioche bread	bread, wholemeal, average • bread, seeded • Bread, naan, retail
		eggs	eggs, quail, whole, raw • eggs, duck, whole, raw • eggs, turkey, whole, raw
		cheddar cheese	cheese, cheddar type, half fat • cheese, cheddar type, 30% less fat • cheese, caerphilly
(3) easy lime basil roast chicken	vegetarian restriction	whole chickens	**eggs, chicken, whole, raw** • eggs, chicken, whole, dried • eggs, chicken, whole, poached
		lime	lime juice, fresh• lime juice cordial, diluted • lime juice cordial, undiluted
		dried basil	mint, dried • parsley, dried • chervil, dried
(3) easy artichoke dip	vegan restriction	mayonnaise	**mustard, wholegrain** • dill, fresh • dill, dried
		parmesan cheese	**chives, fresh** • bran, wheat • basil, fresh
		artichokes	artichoke, globe, raw • artichoke, Jerusalem, boiled in unsalted water, flesh only • artichoke, globe, base of leaves and heart, boiled in unsalted water

ingredient (e.g., "onion"). The second recipe (i.e., "egg Rothko") shows some low-calorie recommendations among the most accurate alternatives for each original ingredient. In this case, we want to highlight how the method can detect low-calorie options for "cheese" by looking at their nutritional content. This achievement is due to the combination of ingredient recipes with a food database. Notice that this approach could be applied to achieving other nutrition goals (e.g., low-sugar and high-protein diets). The third and fourth recipes in Table 9 show food suggestions considering the vegetarian/vegan restriction. Special mention to the bold items. The method can detect the need for adaptation and suggests some allowed foods. In this case, we want to highlight that our method can find good options despite the restrictive condition that must fulfill the candidates. Foods in bold represent the most accurate alternative that the method returns for turning the recipe into vegetarian/vegan.

It is worth highlighting that we use a non-supervised predictive method to adapt the recipe. However, the results demonstrate the potential of our approach. It can deal with a wide range of recipes and give powerful alternatives for most of its ingredients. We want to remark the difficulty of the subjective factor when adapting recipes in an automatized way. Ingredients tend to appear in very different contexts, sometimes with not usual food combinations. People's taste when creating a recipe is also essential. These factors make it harder to model a recipe and, thus, an adaptation. The survey results show that the huge majority of reviews achieve a great satisfaction level. It works remarkably well regardless of the adaptation type. This fact corroborates the performance of the proposed type of adaptation.

VIII. CONCLUSION AND FUTURE WORK

This work proposes a new unsupervised method for recipe adaptation. We combined a word embedding model with a fuzzy-based document distance to find the most similar ingredients in the adaptation task. For this, we used cooking instructions text to extract intrinsic relations between ingredients when cooking. This approach enables us to find the most suitable ingredients maintaining the essence of the recipe.

We have shown that word embedding models obtain good food representations. They can help to understand how ingredients are combined and substituted. To adapt a recipe, we need to relate them and find the most similar food element. So, these representations are useful when combined with a suitable metric. This work also consider a fuzzy-based document distance for identify food description in a nutrition database. This metric is more flexible than standard ones. We have shown that this metric performs very well in mapping food elements between databases. Combining both (i.e., the word embedding representation and the fuzzy metric), we were able to adapt recipes successfully. We presented four cases: non-restricted adaption, when the user wants possible suggestions to modify the recipe; light adaptation, when the user wants a recipe version with low in calories; and vegan and vegetarian restrictions, when the user wants a veggie version of the recipe due to the diet they follow.

We validated the adaptation method with an online web survey where 40 users participated. There were 80 recipes, that is, 20 of each case. The results show the adequacy of this method, and that the semantic content of foods makes sense and can also be applied directly to solve a real-world problem.

For future work, we plan to extend this work considering different cuisine types in the adaptation process. The method will find the most suitable ingredients in each recipe depending on the style. Besides, we also plan to feed the system with users' interactions to recipes and also include expert knowledge to guide the ingredient substitutions. The user reviews will allow us to deal with tastier ingredient combinations, while expert knowledge will provide nutritional advice and coherent food pairings.

We are also going to apply this methodology using more advanced language models, such as BERT, to capture the role of the recipe ingredient in the adaptation task, and fine-tuning of general-purpose embeddings, to leverage existing pre-trained models. We plan to study in depth the interaction of the word embeddings with the mapping function to achieve better results.

REFERENCES

[1] D. Fried, M. Surdeanu, S. Kobourov, M. Hingle, and D. Bell, "Analyzing the language of food on social media," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2014, pp. 778–783.

[2] S. K. Khanna, "Food and culture: A reader," in *Ecology Food Nutrition*, vol. 48, no. 2, 2nd ed. C. Counihan and P. van Esterik, Eds. New York, NY, USA: Routledge, Mar. 2009, pp. 157–159, doi: 10.1080/03670240802577408.

[3] I. D. Brouwer, J. McDermott, and R. Ruben, "Food systems everywhere: Improving relevance in practice," *Global Food Secur.*, vol. 26, Sep. 2020, Art. no. 100398.

[4] S. Sajadmanesh, S. Jafarzadeh, S. A. Ossia, H. R. Rabiee, H. Haddadi, Y. Mejova, M. Musolesi, E. D. Cristofaro, and G. Stringhini, "Kissing cuisines: Exploring worldwide culinary habits on the Web," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW Companion)*, 2017, pp. 1013–1021.

[5] K.-J. Kim and C.-H. Chung, "Tell me what you eat, and i will tell you where you come from: A data science approach for global recipe data on the Web," *IEEE Access*, vol. 4, pp. 8199–8211, 2016.

[6] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *ACM Comput. Surveys*, vol. 52, no. 5, pp. 1–36, Oct. 2019.

[7] A. Nadamoto, S. Hanai, and H. Nanba, "Clustering for similar recipes in user-generated recipe sites based on main ingredients and main seasoning," in *Proc. 19th Int. Conf. Netw. Inf. Syst. (NBiS)*, Sep. 2016, pp. 336–341.

[8] M. Kazama, M. Sugimoto, C. Hosokawa, K. Matsushima, L. R. Varshney, and Y. Ishikawa, "A neural network system for transformation of regional cuisine style," *Frontiers ICT*, vol. 5, p. 14, Jul. 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fict.2018.00014

[9] A. Cordier, V. Dufour-Lussier, J. Lieber, E. Nauer, F. Badra, J. Cojan, E. Gaillard, L. Infante-Blanco, P. Molli, A. Napoli, and H. Skaf-Molli, "TAAABLE: A case-based system for personalized cooking," in *Successful Case-based Reasoning Applications-2*. Berlin, Germany: Springer, 2014, pp. 121–162.

[10] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," in *Proc. 4rd Annu. ACM Web Sci. Conf. (WebSci)*, 2012, pp. 298–307.

[11] M. Chen, X. Jia, E. Gorbonos, C. T. Hoang, X. Yu, and Y. Liu, "Eating healthier: Exploring nutrition information for healthier recipe recommendation," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102051.

[12] F. Nooralahzadeh, L. Øvrelid, and J. T. Lønning, "Evaluation of domain-specific word embeddings using knowledge resources," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 1–8.

[13] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu, "BioWordVec, improving biomedical word embeddings with subword information and MeSH," *Sci. Data*, vol. 6, no. 1, pp. 1–9, Dec. 2019.

[14] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Sci. Rep.*, vol. 1, no. 1, p. 196, Dec. 2011.

[15] S. Abbar, Y. Mejova, and I. Weber, "You tweet what you eat: Studying food consumption through Twitter," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 3197–3206.

[16] W. Min, B.-K. Bao, S. Mei, Y. Zhu, Y. Rui, and S. Jiang, "You are what you eat: Exploring rich recipe information for cross-region food analysis," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 950–964, Apr. 2018.

[17] H. Lim, X. Huang, S. Miller, J. Edelmann, T. Euken, and S. Voida, "Smart cook: Making cooking easier with multimodal learning," in *Proc. Adjunct Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, Sep. 2019, pp. 129–132.

[18] W. Min, S. Jiang, S. Wang, J. Sang, and S. Mei, "A delicious recipe analysis framework for exploring multi-modal recipes with various attributes," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 402–410.

[19] R. Yamanishi, N. Shino, Y. Nishihara, J. Fukumoto, and A. Kaizaki, "Alternative-ingredient recommendation based on co-occurrence relation on recipe database," *Procedia Comput. Sci.*, vol. 60, pp. 986–993, Jan. 2015.

[20] S. G. Mota and B. D. Agudo, "Acook: Recipe adaptation using ontologies, case-based reasoning systems and knowledge discovery," in *Proc. Cooking Comput. Workshop*, 2012, pp. 41–45.

[21] J. DeMiguel, L. Plaza, and B. Díaz-Agudo, "ColibriCook: A CBR system for ontology-based recipe retrieval and adaptation," in *Proc. ECCBR Workshops*, 2008, pp. 199–208.

[22] M. De Clercq, M. Stock, B. De Baets, and W. Waegeman, "Data-driven recipe completion using machine learning methods," *Trends Food Sci. Technol.*, vol. 49, pp. 1–13, Mar. 2016.

[23] Y. Shidochi, T. Takahashi, I. Ide, and H. Murase, "Finding replaceable materials in cooking recipe texts considering characteristic cooking actions," in *Proc. ACM Multimedia Workshop Multimedia Cooking Eating Activities (CEA)*, 2009, pp. 9–14.

[24] J. Takahashi, T. Ueda, C. Nishikawa, T. Ito, and A. Nagai, *Implementation of Automatic Nutrient Calculation System for Cooking Recipes Based on Text Analysis* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2012, pp. 789–794.

[25] Y. Chen, "A statistical machine learning approach to generating graph structures from food recipes," Ph.D. dissertation, Graduate School Arts Sci., Brandeis Univ., Waltham, MA, USA, 2017.

[26] M. Chang, L. V. Guillain, H. Jung, V. M. Hare, J. Kim, and M. Agrawala, "RecipeScape: An interactive tool for analyzing cooking instructions at scale," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, p. 451.

[27] J. Altosaar. (2017). *food2vec—Augmented Cooking With Machine Intelligence*. [Online]. Available: https://jaan.io/food2vec-augmented-cooking-machine-intelligence/

[28] BuzzFeed and Tasty. *Recipe2Vec: How Word2Vec Helped us Discover Related Tasty Recipes*. Accessed: 2017. [Online]. Available: https://pydata.org/nyc2017/schedule/presentation/65/

[29] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 1411–1420.

[30] J. Camacho-Collados and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning," *J. Artif. Intell. Res.*, vol. 63, pp. 743–788, Dec. 2018.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[32] X. Rong, "Word2vec parameter learning explained," 2014, *arXiv:1411.2738*. [Online]. Available: http://arxiv.org/abs/1411.2738

[33] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[34] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[35] G. Berardi, A. Esuli, and D. Marcheggiani, "Word embeddings go to Italy: A comparison of models and training datasets," in *Proc. IIR*, 2015, pp. 1–8.

[36] K. Ethayarajh, "How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings," 2019, *arXiv:1909.00512*. [Online]. Available: http://arxiv.org/abs/1909.00512

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[38] A. Morales-Garzón, J. Gómez-Romero, and M. J. Martin-Bautista, "A word embedding model for mapping food composition databases using fuzzy logic," in *Proc. Int. Conf. Inf. Process. Manage. Uncertainty Knowl.-Based Syst.* Cham, Switzerland: Springer, 2020, pp. 635–647.

[39] P. H. England. *Composition of Foods Integrated Dataset (COFID)*. Accessed: Mar. 2019. [Online]. Available: https://www.gov.uk/government/publications/composition-of-foods-integrated-dataset-cofid

[40] H. Saif, M. Fernandez, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of Twitter," in *Proc. 9th Int. Conf. Lang. Resour. Eval. (LREC)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 810–817. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/292_Paper.pdf

[41] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[42] N. Gisev, J. S. Bell, and T. F. Chen, "Interrater agreement and interrater reliability: Key concepts, approaches, and applications," *Res. Social Administ. Pharmacy*, vol. 9, no. 3, pp. 330–338, May 2013.

[43] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: The kappa statistic," *Family Med.*, vol. 37, no. 5, pp. 360–363, 2005.

ANDREA MORALES-GARZÓN received the degree in computer science and the master's degree in computer engineering from the University of Granada, Spain, in 2018 and 2020, respectively. She is currently pursuing the Ph.D. degree in natural language processing and the master's degree in data science. She is also a Researcher with the Intelligent Databases and Information Systems IDBIS Research Group in the Stance4Health project on the topics of intelligent systems in food computing, applied to dietary assessment, and smart personalized nutrition.

JUAN GÓMEZ-ROMERO received the degree in computer science and the Ph.D. degree in intelligent systems from the University of Granada (UGR), in 2004 and 2008, respectively. He was a Lecturer with the Applied Artificial Intelligence Group, Universidad Carlos III de Madrid, and also a Visiting Researcher with the Data Science Institute, Imperial College London. He is currently an Associate Professor with the Department of Computer Science and Artificial Intelligence, UGR. He has participated in more than 20 projects in artificial intelligence applied to security, ambient intelligence, and energy efficiency problems. His current research interests include the practical use of deep learning and deep reinforcement learning for engineering and social challenges.

MARIA J. MARTIN-BAUTISTA (Member, IEEE) received the degree and the Ph.D. degree in computer science from the University of Granada, Granada, Spain, in 1996 and 2000, respectively. Since 1997, she has been a Full Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. She is currently a Member of the Intelligent Data Bases and Information Systems (IDBIS) Research Group. She has supervised several Ph.D. thesis and authored or coauthored more than 100 articles in high impact international journals and conferences. She has participated in more than 20 R+D projects and has supervised several research technology transfers with companies. Her current research interests include big data analytics in data, text and web mining, intelligent information systems, knowledge representation, and uncertainty. She was a program committee member of several international conferences.