mathematics

Article

Domain Heuristic Fusion of Multi-Word Embeddings for Nutrient Value Prediction

Gordana Ispirova [1,2,*], Tome Eftimov [1] and Barbara Koroušić Seljak [1]

[1] Computer Systems Department, Jožef Stefan Institute, 1000 Ljubljana, Slovenia; tome.eftimov@ijs.si (T.E.); barbara.korousic@ijs.si (B.K.S.)
[2] Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia
[*] Correspondence: gordana.ispirova@ijs.si; Tel.: +386-1477-3519

Abstract: Being both a poison and a cure for many lifestyle and non-communicable diseases, food is inscribing itself into the prime focus of precise medicine. The monitoring of few groups of nutrients is crucial for some patients, and methods for easing their calculations are emerging. Our proposed machine learning pipeline deals with nutrient prediction based on learned vector representations on short text–recipe names. In this study, we explored how the prediction results change when, instead of using the vector representations of the recipe description, we use the embeddings of the list of ingredients. The nutrient content of one food depends on its ingredients; therefore, the text of the ingredients contains more relevant information. We define a domain-specific heuristic for merging the embeddings of the ingredients, which combines the quantities of each ingredient in order to use them as features in machine learning models for nutrient prediction. The results from the experiments indicate that the prediction results improve when using the domain-specific heuristic. The prediction models for protein prediction were highly effective, with accuracies up to 97.98%. Implementing a domain-specific heuristic for combining multi-word embeddings yields better results than using conventional merging heuristics, with up to 60% more accuracy in some cases.

Keywords: domain-specific embeddings; domain knowledge; machine learning; data mining; macronutrient prediction; representation learning; word embeddings; paragraph embeddings

Citation: Ispirova, G.; Eftimov, T.; Koroušić Seljak, B. Domain Heuristic Fusion of Multi-Word Embeddings for Nutrient Value Prediction. *Mathematics* **2021**, *9*, 1941. https://doi.org/10.3390/math9161941

Academic Editor: Jan Rauch

Received: 7 July 2021
Accepted: 12 August 2021
Published: 14 August 2021

1. Introduction

Nutrition, although indispensable throughout human history, has seen the "light of the day" only in the past few decades with the development of modern science.

In the early and middle years of the last century, modern nutrition science was focused on the discovery and synthesis of essential micronutrients and their effect on deficiency diseases. With the rapid spike in the food supply, the demand for nutritional and other food-related components has constantly increased. Nowadays, like in fashion, there are trends in nutrition, i.e., the so/called "diet culture". There is constantly a new "type" of diet: gluten-free, vegan, keto, carb-free, paleo, carnivore diet, and the list goes on. Even though all these diets are geared towards people suffering from a certain disease, or intolerance, they are accepted by many that do not have these.

The global epidemic of obesity, diabetes, and inactivity is very real, and the single strand of connection between them is poor dietary habits. Cardiovascular disease, high blood pressure, diabetes, some cancers, and other chronic diseases [1], as well as bone-health diseases, are related to bad dietary habits [2]. Dietary assessment is crucial for patients suffering from different diseases, of course, the central focus being on diet and nutrition-related ones, and it is also very important for professional athletes, and as a consequence of the accessibility of mobile applications and gadgets communicating with a smartphone (the so-called "online food diaries"), it is slowly becoming an everyday habit for many individuals, for either health or fitness reasons, as well as for both. Developed western countries are fighting obesity, which is increasing by the minute, and this contributes to raised public health concerns about certain macronutrient subcategories, specifically about saturated fats, and added or free sugar. Micronutrients, such as sodium, which should be slowly monitored and tracked in individual suffering from diseases like osteoporosis, stomach cancer, and kidney disease, and fiber, critical for patients suffering from irritable bowel syndrome (IBS), are a matter of concern for nutritional epidemiologists.

In the Food and Nutrition domain, the focus in recent years has been heavily put on data collection, and now, facing this data flood, there is a need for methods dealing with it. Calculating nutrient content is a very demanding and important task, and even though two foods can have roughly the same ingredients, their nutrient content can vary significantly, which makes the tracking and calculating of nutrients very complicated and challenging. We recently proposed an approach, called P-NUT (Predicting NUTrient content from short text descriptions) [3], for macronutrient value prediction of a food item from learned vector representations of text describing the food item (its name).

The nutrient content calculation is usually a process of estimating and calculating the nutrient quantities from measurements and exact ingredients [4–6], and before P-NUT, it had not been viewed as a prediction task. The instructions for calculating nutrient content from measurements and ingredients are demanding; there a few steps to the procedure for nutrient content calculation of a multi-ingredient food: selecting the appropriate recipe, collecting data for the nutrient content of the ingredients, ingredient nutrient level corrections for the weight of edible portions, adjustment of the content of each ingredient for effects of preparation, summation of ingredient composition, adjustment of final weight or volume, and determination of the yield and final volumes. These steps are used when all the ingredients and measurements are available, and when there is no data available for the ingredients, then the data for the uncooked ingredients, in combination with the appropriate yield factors to adjust the weight changes and retention factors for nutrient losses or gains during cooking, are used [7].

While in P-NUT, we used the vector representation of the short text descriptions of the food products as input features to the ML algorithms, in this study, we are working with recipe data, and we propose using the embeddings of the lists of ingredients for each recipe as the input features. Each list of ingredients is a list of simple or complex foods, which are multi-word strings and not sentences; for example, the recipe name is "No-bake oatmeal cookies", and the list of ingredients is as follows: "sugars, granulated; cocoa, dry powder, unsweetened; milk, fluid, 1% fat, without added vitamin a and vitamin d; butter, without salt; vanilla extract; peanut butter, smooth style, without salt; oats". This means their embeddings are uncontextualized, and to merge the vector representations of the separate ingredients, we propose a new domain-specific heuristic that combines the quantities of the ingredients as well. Using this heuristic, the results from this study show how domain knowledge can lead to better results when considering a prediction task in the Food and Nutrition domain.

The rest of the paper is structured as follows: the section Methods begins with the related work and an explanation of the P-NUT ML pipeline, then in Domain-Specific Embeddings, the merging heuristic and the process of obtaining the domain-specific embeddings are described. In Data, a structure and description of the data used in the experiments is provided, and in Experimental Setup, an explanation of how the experiments are conducted in detail is presented. The experimental results and the methodology evaluation are presented in Evaluation Outcomes, as well as the benefits of such an approach and its novelty. In the end, in Discussion, the possible obstacles for the implementation are pointed out and what can be done with future work to overcome them, and at last, in Conclusions, the importance of the methodology is summarized.

2. Materials and Methods

2.1. Related Work

As far as we are aware, P-NUT [3] is the first ML pipeline for macronutrient prediction of foods/recipes using only short text descriptions. Prior work, involving ML, in this direction has been based on image recognition, i.e., utilizing deep learning either for identifying and classifying foods from food images [8], calorie calculation from food images [9], or different types of dietary assessment through food images [10].

These studies relied strongly on textual data retrieved from the Web and were mainly with the goal of predicting total calories. For tracking macronutrient intake, there are many mobile and web applications [11,12]; these systems offer user assistance in achieving dietary goals (losing or gaining weight, allergy management, or simply starting and maintaining a healthy diet). However, in order to provide these services, these systems require manual input of the food details with the portion sizes. This a very time-consuming process and often results in this being very tedious and time-consuming, resulting in users quitting the usage of these applications. Besides that, ordinary users rely on self-reports for calorie intake, which are most often misleading.

Other work including ML in this direction are in the agriculture sector, related to predicting nutrient content in soil. In [13], the authors predicted nutrients in soil using six commonly used techniques: random forest, decision tree, naïve bayes, support vector machine, least-square support vector machine, and artificial neural network, and they showed that the most common and complicated method does not always achieve the best prediction accuracy. Furthermore, in [14], the authors discussd distinct machine learning classifiers and their associated work in soil nutrients.

2.2. P-NUT

The pipeline in P-NUT consists of three parts:

1. Representation learning: Introduced by Mikolov et al. in 2013 [15] and Pennington et al. in 2014 [16], word embeddings have become indispensable for natural language processing (NLP) tasks in the past couple of years, and they have enabled various ML models that rely on vector representation as an input to benefit from these high-quality representations of text input. This kind of representation preserves more semantic and syntactic information of words, which leads to their status as being state-of-the-art in NLP.
2. Unsupervised machine learning: Nutrient content exhibits notable variations between different types of foods. In a big dataset, including raw/simple and composite/recipe foods from various food groups, the content of macronutrients can have values from 0–100 g per 100 g. Needless to say, and as proven in [3,17], better models for macronutrient content prediction are built when grouping–clustering the instances according to domain-specific criteria [17].
3. Supervised machine learning part: The final part of the P-NUT methodology is supervised ML, where separate predictive models are trained for the nutrients that we want to predict. The nutrient values are continuous data; therefore, the models are trained with single-target regression algorithms, in which, as input, we have the learned embeddings of the short text descriptions, clustered based on the chosen/available domain-specific criteria. Selecting the right ML algorithm for the purpose is challenging; the default accepted approach is selecting a few ML algorithms, setting the ranges for the hyper-parameters, hyper-parameter tuning, utilizing cross-fold validation to evaluate the estimators' performances (with the same data in each iteration), and at the end, benchmarking the algorithms and selecting the best one(s). The most commonly used baselines for regression algorithms are the central tendency measures, i.e., mean and median of the train part of the dataset for all the predictions.

Given the historical success of text embeddings in NLP, in P-NUT, we used the most well-known word/paragraph embedding algorithms (Word2Vec [18], GloVe [16], and Doc2Vec [19]). All of these are uncontextualized or context-independent embedding algorithms, i.e., there is just one vector (numeric) representation for each word (i.e., Word2vec and GloVe) or each chunk of words/paragraph (i.e., Doc2vec). Otherwise, to put this in other words, different senses of the word, if there are any, are combined into one single vector. Uncontextualized word embeddings, due to their ability to be used in resource and memory capacity-limited settings, are used in many NLP tasks today. In recent years, especially in the Biomedical domain, context-dependent embedding algorithms [20–22] have emerged as superior to the aforementioned uncontextualized ones. These algorithms generate multiple vector representations for the same word, based on the context in which the word is used. In the two evaluations of P-NUT [3,17], we dealt with short text descriptions of recipes (multi-word strings, that do not represent a complete sentence), since looking at them from a semantic point of view, they do not have the complete necessary structure to form a sentence, i.e., a subject, a verb, and an object. In this study, we are dealing, again, with multi-word strings or, in many cases, even one-word strings; such chunks of text cannot be treated as sentences when generating vector representations; therefore, using context-dependent embedding algorithms is not of use here, so we opted for uncontextualized embeddings.

2.3. Domain-Specific Embeddings

In Figure 1, a flowchart of the methodology is presented. The obtaining of the domain-specific embeddings for the list of recipe ingredients is a two-step process:

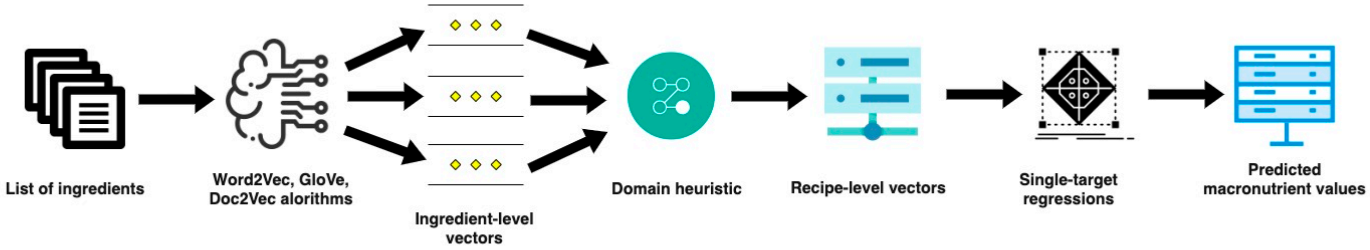
List of ingredients
Word2Vec, GloVe, Doc2Vec alorithms
Ingredient-level vectors
Domain heuristic
Recipe-level vectors
Single-target regressions
Predicted macronutrient values

Figure 1. Flowchart of the presented approach.

Single-ingredient embeddings: obtaining multi-word embeddings. If we have:

$$recipe_k \in \{recipe_1, \ldots, recipe_l\}, \tag{1}$$

as the recipe, we are generating a domain-specific embedding, where $k \in \{1, l\}$, and l is the number of recipes in the dataset, then:

$$ingredients_k = \{I_1, \ldots, I_m\}, \tag{2}$$

as the list of ingredients for $recipe_k$. Furthermore, we have I_i as the ith ingredient of the list of ingredients:

$$I_i = \{word_1, word_a, \ldots, word_n\}, \tag{3}$$

Then, the single ingredient embedding is obtained in two ways:

Utilizing the sum and average as heuristics for merging the vector representations of each word in the ingredient obtained with word embedding algorithms.

$$E[word_a] = [x_{a1}, x_{a2}, \ \ldots, \ x_{ad}], \tag{4}$$

where $E[word]$ is the vector representation (embedding) of a separate word, $a \in \{1, \ldots, n\}$, and d is the dimension of the word vectors. Then, the vector representation of the ingredient will be obtained with:

$$E[I_i] = \left[\frac{x_{11} + \ldots + x_{n1}}{n}, \frac{x_{12} + \ldots + x_{n2}}{n}, \ldots, \frac{x_{1d} + \ldots + x_{nd}}{n}\right], \tag{5}$$

averaging the vector representation of all the words from which it consists of, or:

$$E[I_i] = [x_{11} + \ldots + x_{n1}, x_{12} + \ldots + x_{n2}, \ldots, x_{1d} + \ldots + x_{nd}], \tag{6}$$

by summing the vector representations of the words it consists of.

This involves utilizing the vector representations for the full multi-word strings considered as paragraphs, obtained with a paragraph embedding algorithm. If the same applies as Equations (1)–(3), then:

$$E[I_i] = [x_1, x_2, \ldots, x_d], \tag{7}$$

$E[I_i]$ is the multi-word string vector representation for the ingredient I_i, where d is the predefined dimension of the vectors.

Embeddings on the recipe level: Here, we define a domain-specific heuristic for merging uncontextualized multi-word embeddings. If the same applies as Equations (1) and (2), and:

$$weights_k = [w_1, w_2, \ldots, w_m], \tag{8}$$

then, this is the list of the weights expressed in grams for each ingredient for $recipe_k$, calculated on the whole recipe. On the grounds that the nutrient values of a food are, by rule, given per 100 g and the fact that we are making the predictions for the nutrients using 100 g, the weights for each ingredient in 100 g of the recipe are calculated:

$$weights_{100g_k} = \left[w_{100\mathrm{g}_1}, w_{100\mathrm{g}_2}, \ldots, w_{100g_m}\right], \tag{9}$$

Then, to obtain the domain-specific embedding for the whole recipe, we implement the following heuristic for combining the single ingredient embeddings:

$$E_{recipe_k} = w_{100g_1} \times E[I_1] + \ldots + w_{100g_m} \times E[I_m] \tag{10}$$

2.4. Data

In the experiments for evaluation of the P-NUT methodology in [3], we used food consumption data containing nutritional information about food items. For exploring the bias of the domain knowledge over the prediction task, in [17], we evaluated the extended P-NUT methodology on the Recipe1M dataset, which is publicly available [23], and it is a large-scale structured corpus, which contains over a million cooking recipes, as well as 13 million food images. For the evaluation in this study, we used the Recipe1M dataset, as well, and out of those million recipes for this study, we utilized the ones that contained nutrient content, i.e., a total of 51,235 recipes. For each of the 51,235 recipes, the following information was available:

- recipe title: a short textual description of the recipe;
- recipe instruction: description of instructions for preparing the recipe;
- nutrient content: quantity of fat, protein, salt, saturates, and sugars per 100 g for the whole recipe, expressed in grams;
- ingredients: list of the ingredients needed;
- nutrient content of ingredients: for each ingredient, quantity in grams of fat, protein, saturates, sodium, and sugar per 100 g of the ingredient;
- quantity of each ingredient;
- units of measurement per each ingredient, by the household measurement system: cup, tablespoon, teaspoon, etc.;
- weight in grams per ingredient for the whole recipe.

In the following table (Table 1), we give an example data instance from the dataset.

Table 1. Example instance from the Recipe1M dataset.

Recipe Title	**No-Bake Oatmeal Cookies**					
Ingredients	"sugars, granulated", "cocoa, dry powder, unsweetened", "milk, fluid, 1% fat, without added vitamin a and vitamin d", "butter, without salt", "vanilla extract", "peanut butter, smooth style, without salt", "oats"					
Nutrients per 100 g	Energy	Fat	Protein	Salt	Saturates	Sugars
	378.64	35.40	3.81	0.06	21.01	8.59

3. Results

3.1. Data Preparation

In this study, from the Recipe1M dataset, the data of interest are the list of ingredients, the weights of each ingredient, and the quantities per 100 g of the recipe for the five nutrients of concern. Before generating the single ingredient embeddings, for each ingredient, the text undergoes some basic NLP pre-processing: tokenization, normalization, noise removal, and lemmatization.

3.2. Experimental Setup

The experimental setup after the data pre-processing was as follows:

1. Generate embeddings on single ingredient level (multi-word non-contextualized embeddings) using the Word2Vec, GLoVe, and Doc2Vec algorithms [15,16,19,24]. For the Word2Vec and GloVe embeddings, we took into consideration different values for the dimension of the vectors and the sliding window size. For the vector dimensions, we chose [50, 100, 200]. For the Word2Vec embeddings, the two types of feature extraction available, CBOW and SG, were considered. For the chosen dimensions, we assigned different values, namely [2, 3, 5, 10], to the parameter called the 'sliding' window, which indicates the distance within a sentence between the current word and the word being predicted. With Word2Vec, combining the above-mentioned parameter values and the two heuristics, a total of 48 models were trained, while with GloVe, a total of 24 models were trained. When training the paragraph embeddings with Doc2Vec, we considered the same dimension sizes [50, 100, 200] and sliding window sizes [2, 3, 5, 10], as well as the two types of architectures, PV-DM and PV-DBOW, and we used the non-concatenative mode, meaning training separate models for the sum option and average option. Therefore, there were 48 Doc2Vec models trained in total.
2. Generate embeddings on recipe level: use the above-defined domain heuristic (Equation (10)) to merge the embedding on single ingredient level.
3. Apply single-target regression algorithms: building models for predicting the five given macronutrients: fat, protein, salt, saturates, and sugars. This part consisted of several steps:

    I. Selecting the regression algorithms: Linear regression, Ridge regression, Lasso regression, and ElasticNet regression (using the Scikit-learn library in Python [25]).

    II. Selecting ranges for the parameters for each algorithm and performing hyper-parameter tuning: A priori assignment of the ranges and values for all the parameters for all the regression algorithms. With GridSearchCV (from the Scikit-learn library in Python [25]), the best parameters for the model training were selected from all the combinations.

    III. Estimating the prediction error with k-fold cross-validation: We trained models with the previously selected best parameters and then evaluated them with cross-validation. To compare the regressors, the matched sample approach was chosen, using the same data in each iteration.

4. Calculate domain-specific measure of accuracy: We defined the accuracy according to the appropriate tolerance levels for each nutrient, which were defined by inter-

national legalizations and regulations. In 2012, the European Commission Health and Consumers Directorate-General published a guidance document [26] in order to provide recommendations for calculating the acceptable differences between quantities of nutrients on the nutrient content labels of the food products and the ones established in Regulation EU 1169/2011 [27]. It is impossible for foods to contain the exact quantity of each nutrient on the printed labels; therefore, these tolerances for the food product labels are very important. These differences occur due to the natural variations of foods, and the variations occurring during production and the storage process. The accuracy is calculated according to the tolerance levels in Table 2.

Table 2. Results from the evaluation on Recipe1M.

Measure	Target	Algorithm					
		With Domain Heuristic			Without Domain Heuristic		
		Word2Vec	GloVe	Doc2Vec	Word2Vec	GloVe	Doc2Vec
Maximal accuracy (in %)	Fat	76.66	75.62	91.65	21.45	22.16	20.29
	Proteins	90.57	88.79	97.98	55.47	54.54	52.67
	Sugars	73.38	76.35	88.14	25.73	25.81	22.97
	Saturates	72.78	73.66	95.95	24.00	24.71	20.58
	Salt	43.34	41.79	52.35	36.28	33.10	19.43
Minimal accuracy (in %)	Fat	18.65	18.65	28.00	8.34	8.34	8.35
	Proteins	56.60	56.60	56.57	27.20	27.20	27.22
	Sugars	17.03	17.03	28.54	7.90	7.90	7.90
	Saturates	30.52	30.52	45.27	8.34	8.34	8.35
	Salt	19.24	19.24	45.27	9.44	9.44	9.44
Mean accuracy (in %)	Fat	37.96	48.03	68.99	12.54	13.22	14.70
	Proteins	68.02	78.99	86.12	37.19	36.85	39.54
	Sugars	38.58	47.13	56.05	13.61	14.03	15.07
	Saturates	60.10	67.37	78.32	12.84	13.18	14.11
	Salt	26.25	27.49	31.01	16.07	15.65	18.37

If the actual value of the ith instance from the test set on a certain iteration of the k-fold cross-validation is a_i, and the predicted value p_i of the same ith instance of the test set, then:

$$d_i = |a_i - p_i|, \tag{11}$$

and the absolute difference between the two set values is d_i. Then, *allowed* is a binary variable that is assigned a positive value if the predicted value is in the tolerance levels.

$$\begin{array}{c} allowed = 1\, if : \\ Salt : \begin{cases} a_i < 1.25,\ d_i < 0.375 \\ a_i \geq 1.25,\ d_i \leq 0.2\ \times\ a_i \end{cases} \\ Saturates : \begin{cases} a_i < 4,\ d_i < 0.8 \\ a_i \geq 4,\ d_i \leq 0.2\ \times\ a_i \end{cases} \\ Fat : \begin{cases} a_i < 10,\ d_i < 1 \\ 10 \leq a_i \leq 40,\ d_i \leq 0.2\ \times\ a_i \\ a_i \geq 40,\ d_i \leq 8 \end{cases} \\ Protein,\ Sugar : \begin{cases} a_i < 10,\ d_i < 2 \\ 10 \leq a_i \leq 40,\ d_i \leq 0.2\ \times\ a_i \\ a_i \geq 40,\ d_i \leq 8 \end{cases} \end{array} \tag{12}$$

The accuracy is calculated as the ratio of predicted values that are in the defined tolerance level, i.e., have *allowed* set to 1 :

$$Accuracy = \frac{\sum_{i=1}^{n} allowed}{n} \tag{13}$$

where n is the total number instances (recipes) in the test set. For the baseline mean and baseline median, the accuracy is calculated as the percentage of baseline values that falls in the tolerance level range (has *allowed* set to 1), calculated according to Equations (6)–(8), where the actual value of the ith instance from the test set is a_i, and instead of p_i, we have:

$$b = \begin{cases} \frac{\sum_{i=1}^{m} x_i}{m}, & \textit{the baseline is the mean} \\ \frac{\mathbf{X}_{[(m+1)/2]} + \mathbf{X}_{[(m+1)/2]}}{2}, & \textit{the baseline is the median} \end{cases} \quad (14)$$

where m is the total number of instances in the train set, and $\mathbf{X}$ is the sorted train set (in ascending order).

3.3. Evaluation Outcomes

In order to evaluate the performance of our domain-specific merging heuristic, we repeat the same experiments twice:

1. Merge the single-ingredient embeddings with the domain-specific heuristic, perform the predictive modeling, obtain the results, and calculate the defined accuracy.
2. Merge the single-ingredient embedding with conventional merging heuristics (sum and average) according to the merging heuristic used when obtaining the single-ingredient embeddings. In other words:
    a. The embedding on the recipe level is obtained by calculating an average of the embeddings on a single-ingredient level when they are obtained using Equation (4).
    b. The embedding on recipe level is obtained by summing the single-ingredient embeddings when they are obtained using Equation (5).

After obtaining the embedding on the recipe level, the same steps are applied: performing the single-target regressions, obtaining the predictions, and calculating the defined accuracy, of course, using the same experimental set-up described earlier.

The results from the evaluation showed that using the domain-specific heuristic yields higher accuracy percentages than the conventional merging techniques. In the following table (Table 2), the results from the evaluation are presented. For presentation purposes, for each embedding algorithm, we included only the maximal and minimal accuracies achieved for each nutrient (without the details of vector dimension, sliding window, and regression algorithm), and we also calculated the mean accuracy, calculated for each nutrient from all the accuracy percentages for the certain embedding algorithm (all possible vector dimensions, sliding windows, and regression algorithms). From the results, it is evident that the ML models that use the embeddings merged with the domain heuristic as input features outperform the models which use the embeddings merged with conventional merging heuristics as features. The differences between the two approaches in the accuracies were rather big. We can note that the prediction accuracies for salt content were rather lower than the other nutrients, but this is due to the fact that the salt content in 100 g from the food/recipe were rather low compared to the other nutrients (most often less than 1 g). It is also apparent that the prediction results for protein values were the best out of all the nutrient and that the Doc2Vec embedding algorithm outperforms the Word2Vec and GloVe algorithms.

For further interpretation of the results, we used the principal component analysis (PCA) [28] as a reduction technique to visualize some of the obtained embeddings. With closer inspection of the results, we observed that the maximal accuracies in most cases were obtained with the Doc2Vec algorithm when using the following parameters:

Vector dimension: 100,
Sliding window: 2,
Architecture: PV-DBOW,
Merging technique: sum.

All of the visualizations presented used the embeddings obtained from the Doc2Vec algorithm and these parameters.

First, we visualized the embeddings obtained with the domain-specific heuristic for 20 different recipes, with the same list of ingredients but same or different quantities. The visualization is presented in Figure 2. From the visualization, we can see how they grouped in the embedding space; there was a big chunk of recipes in the middle and then five of them very far from each other. To understand why this was happening, we searched for the nutrient values of these recipes, which are given in Table 3. We can see that the reduced embeddings for the recipe with title "The Best No Bake Cookies" and the recipe with title "No Bake Oatmeal Cookies" are overlapping, and from the table, we can see that they have almost identical nutrient values. Next, we can see that the reduced embedding for the recipe with title "Laura's House Famous Mud Pie" is far apart from the remaining 19, and from the table, we can see that it is because its sugar content is significantly higher than the rest. Then, the reduced embeddings for the recipes with titles "No Bake Cookies" and "No-Bake Cookies" were placed far apart, although they have almost identical titles (except the hyphen); however, from the table, we can see that this was logical because of their different nutrient values in fat and sugars.

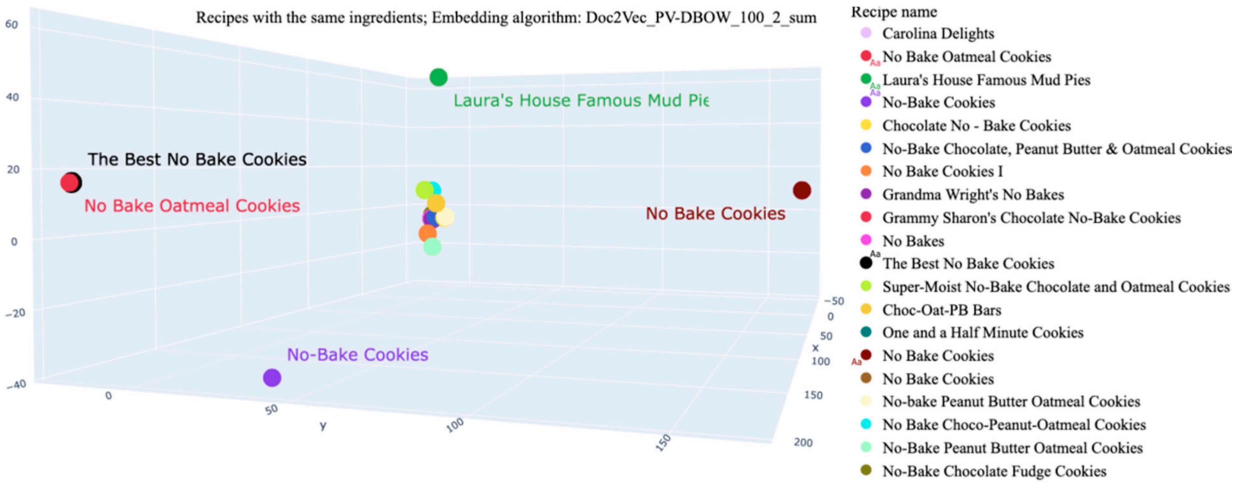

Figure 2. Visual representation of the vector representations when using the domain-specific heuristic of a group of recipes with the same ingredients.

Table 3. Nutrient values for recipes with the same ingredient list, but same or different quantities.

Recipe Title	Ingredients	Nutrients Per 100 g					
		Energy	Fat	Protein	Salt	Saturates	Sugars
The Best No Bake Cookies	"cocoa, dry powder, unsweetened", "milk, fluid, 1% fat, without added vitamin a and vitamin d", "oats", "peanut butter, smooth style, without salt", "sugars, granulated", "butter, without salt", "vanilla extract"	378.64	35.40	3.81	0.06	21.01	8.59
No Bake Oatmeal Cookies		378.44	35.37	3.83	0.06	21.00	8.58
Laura's House Famous Mud Pies		385.10	15.58	4.45	0.02	8.05	50.80
No Bake Cookies		323.39	15.26	14.74	0.03	6.41	19.38
No-Bake Cookies		317.04	22.93	13.34	0.06	5.53	11.53
Chocolate No–Bake Cookies		397.65	15.39	9.43	0.02	6.28	33.02

The same visualization was made for the embeddings on a recipe-level, obtained with the conventional merging heuristics (Figure 3). From the figure, we can see a very different placement of the embeddings in the space. For comparison purposes, only the names of the previously analyzed recipe embeddings were included. We can see that the embeddings for the recipes with titles "The Best No Bake Cookies" and "No Bake Oatmeal Cookies" are very far apart, even though they have almost identical nutrient values, while the embeddings for the "No Bake Oatmeal Cookies" recipe and "Chocolate No-Bake Cookies" are overlapping, even though we can clearly see from the table (Table 3) that they have very different nutrient values, i.e., the difference in each nutrient is very considerable (fat difference is two times, protein difference is three times, saturates difference is almost three times, and sugar difference is more than four times).

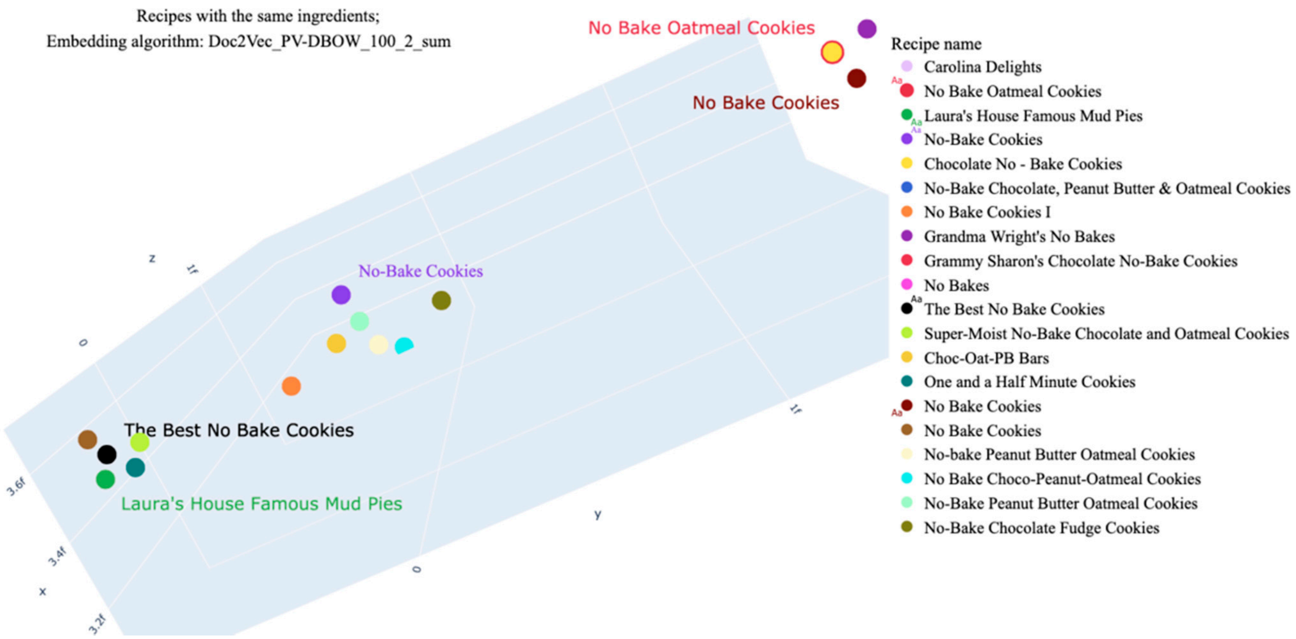

Figure 3. Visual representation of the vector representations without the domain-specific heuristic of a group of recipes with the same ingredients.

The next thing we can notice is how the recipes “Laura’s House Famous Mud Pie” and “The Best No Bake Cookies” were placed very close together when, judging from the table, they have very different nutrient values (the first recipe has two times less fat than the second, more than six times the amount of sugar, and almost three times less saturates). This just goes to show how important the heuristic is when merging word embeddings. Even though all these recipes have the same identical ingredient list, they differ significantly in nutrient content.

This depicts the differences in the feature space; to capture this difference in the performance space, we presented the results from the predictions for the same recipes in Table 4. From these results, we can see that the difference between the actual values of the nutrients and the predicted values of the nutrients when using the domain heuristic is smaller than the difference between the actual values of the nutrients and the predicted values of the nutrients without using the domain heuristic.

From the list of 20 recipes, in these visualizations, we can see that there are a lot of recipes with very similar names, if not with the same name. Given this observation, we dug deeper, and we gathered a list of recipes in the Recipe1M dataset that had the same identical name, “No Bake Cookies”; there were some recipes that included punctuation sign(s), but we omitted those. There were seven recipes in total with the name “No Bake Cookies”. We did the same two visualizations for the embeddings on a recipe-level, obtained with the domain-specific heuristic (Figure 4) and with the conventional merging heuristics (Figure 5). For each recipe, we included the five nutrient values of concern next to the reduced embedding point. We can see that when using the domain heuristic, the embeddings placed close together were for recipes that have very similar nutrient values for the five nutrients (i.e., which further helps the prediction task), while when using the conventional merging heuristic, the embeddings placed close together (i.e., making the prediction task difficult), or in this case overlapping (the black and grey), have very different nutrient content (the recipe represented with the grey marker has almost double the amount of sugar).

This not only proves that the domain-specific merging heuristics is a better approach when predicting nutrient values, but also how limited the information included in these short recipe descriptions is; having the same name, or even the same ingredient list, does, by any means, state that two recipes have the same or even comparable nutrient content.

Table 4. Differences in performance space (A–Actual value, DH–Predicted value when using the domain heuristic, No DH–Predicted values without using the domain heuristic).

	Recipe Title		**The Best No Bake Cookies**	**No Bake Oatmeal Cookies**	**Laura's House Famous Mud Pies**	**No Bake Cookies**	**No-Bake Cookies**	**Chocolate No–Bake Cookies**
Nutrients per 100 g	Fat	A	35.4	35.37	15.58	15.26	22.93	15.39
		DH	32.28	32.92	14.38	15.32	25.54	14.47
		No DH	23.11	12.44	29.33	16.78	16.78	8.79
	Protein	A	3.81	3.83	4.45	14.74	13.34	9.43
		DH	3.59	4.38	4.40	12.29	9.12	7.26
		No DH	8.96	19.67	2.34	12.11	7.34	15.67
	Salt	A	0.06	0.06	0.02	0.03	0.06	0.02
		DH	0.03	0.03	0.15	0.06	0.30	0.39
		No DH	1.78	0.30	0.20	0.28	0.20	0.38
	Saturates	A	21.01	21.00	8.05	6.41	5.53	6.28
		DH	18.21	18.23	7.21	6.11	6.79	5.74
		No DH	10.53	10.43	12.56	2.56	11.67	15.89
	Sugars	A	8.59	8.58	50.80	19.38	11.53	33.02
		DH	10.88	10.88	50.62	22.36	11.30	34.46
		No DH	27.45	18.99	24.75	2.33	21.74	27.85

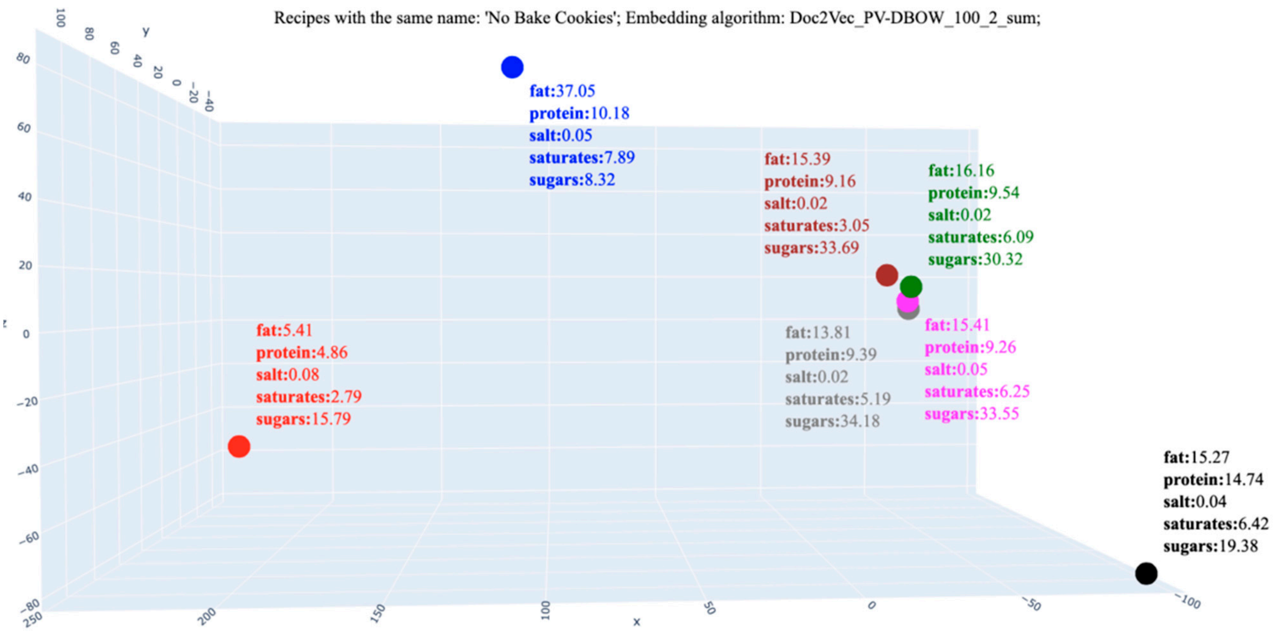

Figure 4. Visual representation of the vector representations with the domain-specific heuristic of a group of recipes with the same name.

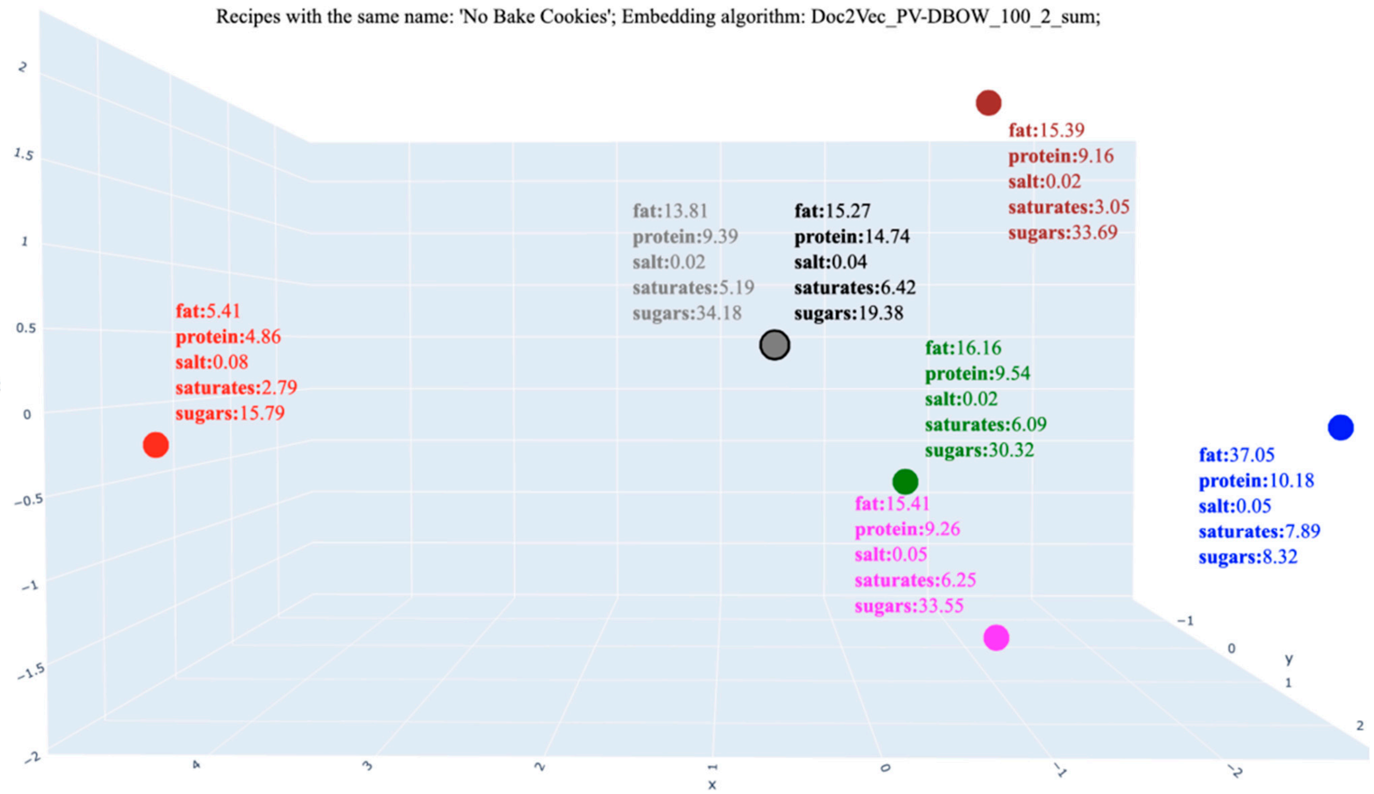

Figure 5. Visual representation of the vector representations without the domain-specific heuristic of a group of recipes with the same name.

4. Discussion and Future Work

The results from this study, following our two previous related studies [3,17], indicate the impact of integrating domain-driven knowledge into an ML pipeline using a nowadays common NLP tool, word embeddings. When put into effect, our task-specific embedding merging heuristic yields high accuracy results for a domain sensitive assignment, such as nutrient prediction.

We must note that the Recipe1M dataset is very thorough, and it can be regarded as an "outlier" dataset, since many datasets of such kind, i.e., recipe datasets, do not contain such exhaustive information, particularly talking about the "weight per ingredient in grams" data, which, in our case, is crucial. The most typical data for this information, which can be found on crosswise recipe datasets, would be just a list of ingredients, and it usually includes data about the quantity of each ingredient, the unit in which set ingredient is measured, and the ingredient itself, which is commonly combined, or if we put it into data terms, it is a coherent string. An important fact that should be noted here is that we are talking about recipe data, which is data that is usually retrieved from the Web; thus, these units are most certainly expressed in standard food/cooking household measurements. Therefore, in order to utilize this data and draw the information that is required, a few NLP techniques must be put into work:

- Named entity recognition (NER): to segmentize the strings into quantity and unit, which means we need rules of what represents a quantity, a resource with all the possible units, i.e., the common food household measurements, and lastly, a resource to identify the ingredients/food items [29,30].
- Normalizing the quantities: after the NER, the units (household measurements) need to be converted in grams, in order to have the same unit all across the dataset. This can be done using conversion tables [31], the problem that arises here is that these conversion tables are separate for liquid and dry ingredients, so the ingredients

need to be separated into liquid and dry beforehand. This can be viewed as a classification problem.

- Map the extracted unit to the proper unit in the conversion table, which can be viewed as simple string matching, but since there are multiple ways of writing a single household measure unit, it can be viewed as a slightly more complex task than string matching, for example, mapping strings based on lexical similarity [32].

5. Conclusions

In this work, we used Word2vec, GloVe, and Doc2Vec as algorithms for generating multi-word uncontextualized embeddings of ingredients, and we combined them, using a heuristic drawn from domain knowledge. In the evaluation for comparison purposes, we repeated the same steps with the same experimental setup but using conventional embedding merging heuristics (sum and average). The results from this study revealed the significant difference that this domain insight provided in the prediction results.

Even though the same single-ingredient embeddings were used, the results were drastically different when using the domain heuristics vs. the conventional merging heuristics, with up to +40% in accuracy, with the Doc2Vec embedding algorithm outperforming the Word2Vec and GloVe algorithms.

When dealing with data of any specific field, the fusion of domain and data driven knowledge is crucial for making performant vector representations.

Having a better prior understanding of the problem in hand and the domain is a key factor when dealing with a prediction task, and domain knowledge is the single, most important step in predictive modeling.

Author Contributions: Conceptualization, G.I., T.E. and B.K.S.; methodology, G.I. and T.E.; software, G.I.; validation, G.I. and T.E.; resources, B.K.S.; data curation, B.K.S.; writing—original draft preparation, G.I.; writing—review and editing, T.E. and B.K.S.; visualization, G.I.; supervision, T.E. and B.K.S.; project administration, B.K.S.; funding acquisition, B.K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Slovenian Research Agency (research core grant number P2-0098) and the European Union's Horizon 2020 research and innovation programme (FNS-Cloud, Food Nutrition Security) (grant agreement 863059). The information and the views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies, nor any person acting on their behalf, may be held responsible for the use that may be made of the information contained herein.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: http://pic2recipe.csail.mit.edu/ (accessed on 7 July 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ijaz, M.F.; Attique, M.; Son, Y. Data-Driven Cervical Cancer Prediction Model with Outlier Detection and Over-Sampling Methods. *Sensors* **2020**, *20*, 2809. [CrossRef] [PubMed]
2. World Health Organization. *Diet, Nutrition, and the Prevention of Chronic Diseases: Report of a Joint WHO/FAO Expert Consultation*; World Health Organization: Geneva, Switzerland, 2003; Volume 916.
3. Ispirova, G.; Eftimov, T.; Koroušić Seljak, B. P-NUT: Predicting NUTrient Content from Short Text Descriptions. *Mathematics* **2020**, *8*, 1811. [CrossRef]
4. Rand, W.M.; Pennington, J.A.; Murphy, S.P.; Klensin, J.C. *Compiling Data for Food Composition Data Bases*; United Nations University Press: Tokyo, Japan, 1991.
5. Greenfield, H.; Southgate, D.A. *Food Composition Data: Production, Management, and Use*; Food and Agriculture Org.: Rome, Italy, 2003; ISBN 978-92-5-104949-5.
6. Schakel, S.F.; Buzzard, I.M.; Gebhardt, S.E. Procedures for Estimating Nutrient Values for Food Composition Databases. *J. Food Compos. Anal.* **1997**, *10*, 102–114. [CrossRef]

7. Machackova, M.; Giertlova, A.; Porubska, J.; Roe, M.; Ramos, C.; Finglas, P. EuroFIR Guideline on Calculation of Nutrient Content of Foods for Food Business Operators. *Food Chem.* **2018**, *238*, 35–41. [CrossRef] [PubMed]
8. Yunus, R.; Arif, O.; Afzal, H.; Amjad, M.F.; Abbas, H.; Bokhari, H.N.; Haider, S.T.; Zafar, N.; Nawaz, R. A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques. *IEEE Access* **2018**, *7*, 2643–2652. [CrossRef]
9. Pouladzadeh, P.; Shirmohammadi, S.; Al-Maghrabi, R. Measuring Calorie and Nutrition from Food Image. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 1947–1956. [CrossRef]
10. Jiang, L.; Qiu, B.; Liu, X.; Huang, C.; Lin, K. DeepFood: Food Image Analysis and Dietary Assessment via Deep Model. *IEEE Access* **2020**, *8*, 47477–47489. [CrossRef]
11. Samsung Health (S-Health). Available online: https://health.apps.samsung.com/terms (accessed on 12 April 2021).
12. MyFitnessPal. Available online: https://www.myfitnesspal.com/ (accessed on 12 April 2021).
13. Kaur, S.; Malik, K. Predicting and Estimating the Major Nutrients of Soil Using Machine Learning Techniques. In *Soft Computing for Intelligent Systems*; Springer: Cham, Germany, 2021; pp. 539–546.
14. Wankhede, D.S. Analysis and Prediction of Soil Nutrients PH, N, P, K for Crop Using Machine Learning Classifier: A Review. In *International Conference on Mobile Computing and Sustainable Informatics*; Springer: Cham, Germany, 2020; pp. 111–121.
15. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
16. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
17. Ispirova, G.; Eftimov, T.; Seljak, B.K. Exploring Knowledge Domain Bias on a Prediction Task for Food and Nutrition Data. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020.
18. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
19. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21 June 2014; pp. 1188–1196.
20. Jiang, M.; Sanger, T.; Liu, X. Combining Contextualized Embeddings and Prior Knowledge for Clinical Named Entity Recognition: Evaluation Study. *JMIR Med Inform.* **2019**, *7*, e14850. [CrossRef] [PubMed]
21. Li, Y.; Wang, X.; Hui, L.; Zou, L.; Li, H.; Xu, L.; Liu, W. Chinese Clinical Named Entity Recognition in Electronic Medical Records: Development of a Lattice Long Short-Term Memory Model With Contextualized Character Representations. *JMIR Med. Inform.* **2020**, *8*, e19848. [CrossRef] [PubMed]
22. Rasmy, L.; Xiang, Y.; Xie, Z.; Tao, C.; Zhi, D. Med-BERT: Pretrained Contextualized Embeddings on Large-Scale Structured Electronic Health Records for Disease Prediction. *NPJ Digit. Med.* **2021**, *4*, 1–13. [CrossRef] [PubMed]
23. Marin, J.; Biswas, A.; Ofli, F.; Hynes, N.; Salvador, A.; Aytar, Y.; Weber, I.; Torralba, A. Recipe1m+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 187–203. [CrossRef] [PubMed]
24. Rehurek, R.; Sojka, P. Gensim—Statistical Semantics in Python. *NLP Cent. Fac. Inform. Masaryk Univ.* **2011**.
25. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
26. European Commission Health and Consumers Directorate-General Guidance Document for Competent Authorities for the Control of Compliance with EU Legislation on: Regulation (EU) No 1169/2011 of the European Parliament and of the Council of 25 October 2011 on the Provision of Food Information to Consumers, Amending Regulations (EC) No 1924/2006 and (EC) No 1925/2006 of the European Parliament and of the Council, and Repealing Commission Directive 87/250/EEC, Council Directive 90/496/EEC, Commission Directive 1999/10/EC, Directive 2000/13/EC of the European Parliament and of the Council, Commission Directives 2002/67/EC and 2008/5/EC and Commission Regulation (EC) No 608/2004Devlin. Available online: https://ec.europa.eu/food/sites/food/files/safety/docs/labelling_nutrition-supplements-guidance_tolerances_1212_en.pdf (accessed on 13 April 2021).
27. Commission, E. Regulation (EU) No 1169/2011 of the European Parliament and of the Council of 25 October 2011 on the Provision of Food Information to Consumers, Amending Regulations (EC) No 1924/2006 and (EC) No 1925/2006 of the European Parliament and of the Council, and Repealing Commission Directive 87/250/EEC, Council Directive 90/496/EEC, Commission Directive 1999/10/EC, Directive 2000/13/EC of the European Parliament and of the Council, Commission Directives 2002/67/EC and 2008/5/EC and Commission Regulation (EC) No 608/2004. *Off. J. Eur. Union L* **2011**, *304*, 18–63.
28. Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
29. Popovski, G.; Kochev, S.; Korousic-Seljak, B.; Eftimov, T. FoodIE: A Rule-Based Named-Entity Recognition Method for Food Information Extraction. In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, Prague, Czech Republic, 19–21 February 2019; pp. 915–922.
30. Cenikj, G.; Popovski, G.; Stojanov, R.; Seljak, B.K.; Eftimov, T. BuTTER: BidirecTional LSTM for Food Named-Entity Recognition. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 3550–3556.

31. Lynn Wright Cooking Measurement Conversion Tables. Available online: https://www.saga.co.uk/magazine/food/cooking-tips/cooking-measurement-conversion-tables (accessed on 13 April 2021).
32. Ispirova, G.; Eftimov, T.; Korousic-Seljak, B.; Korosec, P. Mapping Food Composition Data from Various Data Sources to a Domain-Specific Ontology. In Proceedings of the 9th International Conference on Knowledge Engineering and Ontology Development, Funchal, Portugal, 1–3 November 2017; pp. 203–210.