mathematics

MDPI

Article

P-NUT: Predicting NUTrient Content from Short Text Descriptions

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.: +386-14773519

Received: 14 September 2020; Accepted: 8 October 2020; Published: 16 October 2020

Abstract: Assessing nutritional content is very relevant for patients suffering from various diseases, professional athletes, and for health reasons is becoming part of everyday life for many. However, it is a very challenging task as it requires complete and reliable sources. We introduce a machine learning pipeline for predicting macronutrient values of foods using learned vector representations from short text descriptions of food products. On a dataset used from health specialists, containing short descriptions of foods and macronutrient values: we generate paragraph embeddings, introduce clustering in food groups, using graph-based vector representations, that include food domain knowledge information, and train regression models for each cluster. The predictions are for four macronutrients: carbohydrates, fat, protein and water. The highest accuracy was obtained for carbohydrate predictions – 86%, compared to the baseline – 27% and 36%. The protein predictions yielded the best results across all clusters, 53%–77% of the values fall in the tolerance-level range. These results were obtained using short descriptions, the embeddings can be improved if they are learned on longer descriptions, which would lead to better prediction results. Since the task of calculating macronutrients requires exact quantities of ingredients, these results obtained only from short description are a huge leap forward.

Keywords: macronutrient prediction; representation learning; machine learning; data mining; word embeddings; paragraph embeddings; single-target regression

1. Introduction

There is no denying that nutrition has become a core factor to today's society, and an undeniable solution to the global health-crisis [1–4]. The path towards making the average human diet healthier and environmentally sustainable is a fundamental part of the solution for numerous challenges from ecological, environmental, societal and economic perspective, and the awareness for this has just started to grow and be fully appreciated.

We live in a time of a global epidemic of obesity, of diabetes, of inactivity, all connected to bad dietary habits. Many chronic diseases such as high blood pressure, cardiovascular disease, diabetes, some cancers [5], and bone-health diseases are linked to, again – poor dietary habits [6]. Dietary assessment is essential for patients suffering from many diseases (especially diet and nutrition related ones), it is also very much needed for professional athletes, and because of the accessibility of meal tracking mobile applications it is becoming part of everyday habits of a vast majority of individuals, for health, fitness, or weight loss/gain. Obesity is spiking each day in developed western countries and this contributes to raised public health concern about some subcategories of macronutrients, specifically about saturated fats, and added or free sugar. Nutritional epidemiologists are also raising concern about micronutrients like – sodium, whose intake should be monitored for individuals suffering from

specific diseases like osteoporosis, stomach cancer, kidney disease, kidney; and fiber, whose intake is critical for patients suffering from irritable bowel syndrome (IBS).

Nutrient content from one food to another can vary a lot, even though they have roughly the same type of ingredients. This makes nutrient tracking and calculating very challenging, and predicting nutrient content very complicated. In this paper, we propose an approach, called P-NUT (Predicting NUTrient content from short text descriptions), for predicting macronutrient values of a food item considering learned vector representations of text describing the food item. Food items are generally unbalanced in terms of macronutrient content. When there is a broad variety of foods, they can go from one extreme to another for one macronutrient content, for example the content of fat can go from 'fat free' foods to 'fat based' foods (ex. different kinds of nut butters), which can be a good base for grouping foods. Therefore, a general model for prediction will not be efficient in macronutrient prediction. For this reason, we decided to apply unsupervised machine learning – clustering as a method to separate foods in order to obtain clusters (groups) of foods with similar characteristics. Subsequently, on these separate clusters we predict the macronutrients with applying supervised machine learning. Predicting macronutrients is not a task that has been approached in such a manner before, usually nutrient content of food is calculated or estimated from measurements and exact ingredients [7–9]. These calculations are pretty demanding, the detailed procedure for calculation of the nutrient content of a multi-ingredient food has a few major steps: selection or development of an appropriate recipe, data collection for the nutrient content of the ingredients, correction of the ingredient nutrient levels for weight of edible portions, adjustment of the content of each ingredient for effects of preparation, summation of ingredient composition, final weight (or volume) adjustment, and determination of the yield and final volumes. This is when all the ingredients and measurements are available. When the data for the ingredients are not available, this procedure gets more complicated [7,8].

With using just, short text descriptions of the food products – either a simple food or complex recipe dish, the results from this study show that this way of combining representation learning with unsupervised and supervised machine learning provides results with accuracy as high as 80%, compared to the baseline (mean and median – calculated from the values of a certain macronutrient of all the food items in a given cluster) in some cases there are differences in accuracies of up to 50%.

The structure of the rest of the paper is the following: In Section 2, we begin with the related work in Section 2.1 where we present the published research need to understand P-NUT, then Section 2.2 provides a structure and description of the data used in the experiments, and in Section 2.3, we explain the methodology in detail. The experimental results and the methodology evaluation are presented in the Section 3. In the Section 4, we review the outcome of the methodology, the benefits of such approach, and its novelty. At, the end, in Section 5, we summarize the importance of the methodology and give directions for future work.

2. Materials and Methods

To the best of our knowledge, predicting nutritional content of foods/recipes using only short text description has never been done before. There has been some work involving machine learning done in this direction, mainly involving image recognition: employing different deep learning models for accurate food identification and classification from food images [10], dietary assessment through food image analysis [11], calculating calorie intake from food images [12,13]. All this work in the direction of predicting total calories, strongly relies on textual data retrieved from the Web. There are numerous mobile and web applications, for tracking macronutrient intake [14,15]. Systems like these are used for achieving dietary goals, allergy management or simply, maintaining a healthy balanced diet. The biggest downside is the fact that they require manual imputation of details about the meal/food.

2.1. Related Work

In this subsection we present a review of the concepts relevant to P-NUT, the algorithms that were used, and recent work done in this area.

2.1.1. Representation Learning

Representation learning is learning representations of input data by transforming it or extracting features from it, which then makes it easier to perform a task like classification or prediction [16]. There are two different categories of vector representations: non-distributed or sparse, which are much older and distributed or dense, which have been in use for the past few years. Our focus is on distributed vector representations.

Word Embeddings

Word representations were first introduced as an idea in 1986 [17]. Since then, word representations have changes language modelling [18]. Following up is work that includes applications to automatic speech recognition and machine translation [19,20], and a wide range of Natural Language Processing (NLP) tasks [21–27]. Word embeddings have been used in combination with machine learning, improving results from biomedical named entity recognition [28], capturing word analogies [29], extracting latent knowledge from scientific literature and going towards a generalized approach to the process of mining scientific literature [30], etc. We previously explored the idea of applying text-based representation methods in the food domain for the task of finding similar recipes based on cosine similarity between embedding vectors [31]. Word embeddings are vector space models (VSM), that in a low-dimensional semantic space (much smaller than the vocabulary size) represent words in a form of real-valued vectors. Having distributed representations of words in vector space helps improve the performance of learning algorithms in for various NLP tasks.

- Word2Vec was introduced as word embedding method by Mikolov et al. in 2013 at Google [32], and it is a neural network based word embedding method. There are two different Word2Vec approaches, Continuous Bag of Words and Continuous Skip Gram [33]:
    - Continuous Bag-of-Words Model (CBOW) – This architecture consists of a single hidden layer and an output layer. The algorithm tries to predict the center word based on the surrounding words – which are considered as the context of this word. The inputs of this model are the one–hot encoded context word vectors.
    - Skip-gram Model (SG) – In the SG architecture we have the center word and the algorithms tries to predict the words before and after it, which make up the context of the word. The output from the SG model are C number of V dimensional vectors, where C is the number of context words which we want the model to return and V is the vocabulary size. The SG model is trained to minimize the summed prediction error and gives better vectors with increments of C [32,33].

  If compared, CBOW is a lot simpler and faster to train but SG performs better with rare words.
- GloVe [34] is another method for generating word embeddings. It is a global log-bilinear regression model for unsupervised learning of word representations, that has been shown to outperform other models on word analogy, word similarity, and named entity recognition tasks. It is based on co-occurrence statistics from a given corpus.

Paragraph Embeddings

In 2014 [35] an unsupervised paragraph embedding method, called Doc2Vec, was proposed. Doc2Vec in contrast to Word2Vec generated vector representations of whole documents, regardless of their length. The paragraph vector and word vectors are concatenated in a sliding window and the next word is predicted; the training is done with a gradient decent algorithm. The Doc2Vec algorithm also takes into account the word order and context. The inspiration, of course, comes from the Word2Vec algorithm: the first part, called Distributed Memory version of Paragraph Vector (PV-DM), is an extension of the CBOW model with an additional vector (Paragraph ID) added, with the

difference of including another feature vector, unique to the document, for the next word prediction. The word vectors represent the concept of a word, while the document vector represents the concept of a document.

The second algorithm, called Distributed Bag of Words version of Paragraph Vector (PV-DBOW), is similar to the Word2Vec SG model. In PV-DM the algorithm considers the concatenation of the paragraph vector with the word vectors for the prediction of the next word, whereas in the PV-DBOW the algorithm ignores the context words in the input, and the word are predicted by random sampling from the paragraph in the output.

The authors recommend using a combination of the two models, even though the PV-DM model performs better and usually will achieve state of the art results by itself.

Graph-Based Representation Learning

Besides word embeddings, there are methods that are used for embedding data represented as graphs, consequently named graph embedding. Usually, embedding methods learn vector embeddings represented in the Euclidean vector space, but as graphs are hierarchical structures, in 2017 the authors in [36] introduced an approach for embedding hierarchical structures into hyperbolic space – Poincaré ball. Poincaré embeddings are vector representations of symbolic data, the semantic similarity between two concepts is the distance between them in the vector space, and their hierarchy is waved by the magnitudes of the vectors. Graph embeddings have improved performance over many of the existing models on tasks such as text classification, distantly supervised entity extraction, and entity classification [37], they also have been used for unsupervised feature extraction from sequences of words [38]. In [39], the authors generate graph embeddings (Poincaré) for the FoodEx2 hierarchy [40]. FoodEx2 version 2 is a standardized system for food classification and description developed by the European Food Safety Authority (EFSA), it has domain knowledge embedded in it and it contains descriptions of a vast set of individual food items combined in food groups and more broad food categories in a hierarchy that exibits parent-child relationship. The domain knowledge contained in the FoodEx2 hierarchy is transcended through the graph embeddings, which later the authors use in order to group the food items from the FoodEx2 system in clusters. The clustering is done using the Partition Around Medoids algorithm [41], and the number of clusters is determined using the silhouette method [42].

2.2. Data

In our experiments we used a dataset that contains nutritional information about food items recently collected as food consumption data in Slovenia with the collaboration of subject-matter experts for the aims of the EFSA EU Menu project [43] – designed for more accurate exposure assessments and ultimately support of risk managers in their decision-making on food safety. The ultimate goal being – enabling quick assessment of exposure to chronic and acute substances possibly found in the food chain [44]. In this dataset there are 3265 food items, some of which are simple food products and others are recipes with short descriptions, a few instances are presented in Table 1 as an example.

From the dataset for each food item we have available: name in Slovene, name in English, FoodEx2 code, and nutrient values for: carbohydrates, fat, protein and water. We repeated our experiments for both English and Slovene names of the food products and the recipes.

Table 1. Subset from the dataset used in the experiments (SLO—Slovenian, ENG—English).

SLO Food Name	ENG Food Name	FoodEx2 Code	Energy (g)	Water (g)	Fat (g)	Carb (g)	Protein (g)
Zelenjavna rižota s parboiled rižem, sezonsko zelenjavo in repičnim oljem	Vegetable risotto with parboiled rice, seasonal vegetables and rapeseed oil	A041G# F04.A036V	90.07	79.36	1.55	16.77	1.79
Medenjaki iz pirine in ržene moke ter hojevega medu	Gingerbread biscuit made of spelt and rye flour and honey	A00CT# F04.A004H$ F04.A003J$ F04.A033K	423.68	0.00	1.81	91.41	8.96
Čokoladna rezina Kit Kat	Candies, KIT KAT Wafer Bar	A009Z	517.87	1.63	25.99	64.59	6.51
Zeleni ledeni čaj z medom, arizona	Tea, ready-to-drink, green iced tea, Arizona	A03LD	27.65	93.00	0.00	6.80	0.01

2.3. Methodology

On Figure 1 a flowchart of the methodology is presented. Our methodology is consisted of three separate parts: representation learning and unsupervised machine learning, conducted independently, and then combined in supervised machine learning.

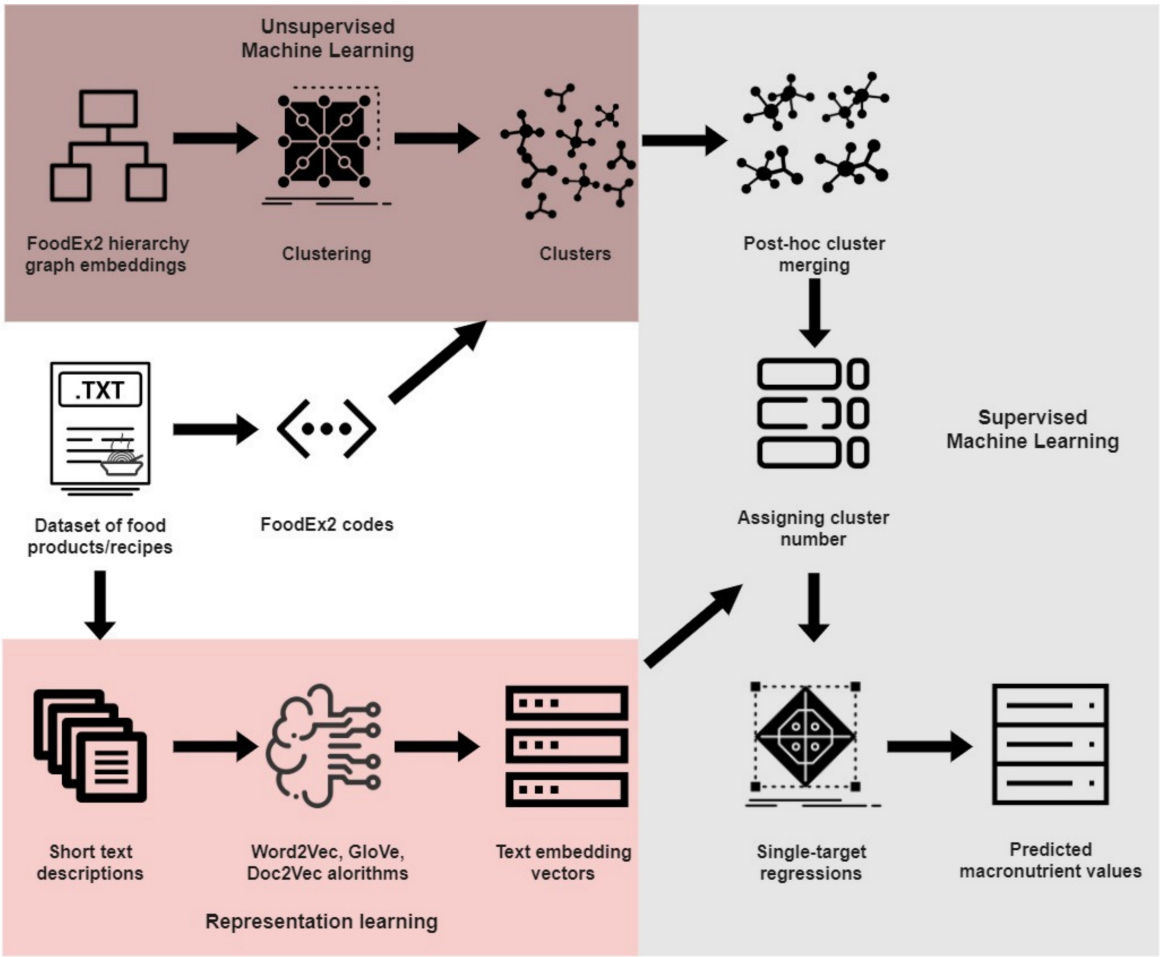

Figure 1. Flowchart of the methodology.

The idea is: (i) represent text descriptions in vector space using embedding methods, i.e., semantic embeddings at sentence/paragraph level of short food descriptions, (ii) cluster the foods based on their FoodEx2 codes [40] using graph embeddings [39], (iii) perform post-hoc cluster merging in order to obtain more evenly distributed clusters on a higher level of the FoodEx2 hierarchy, (iv) apply

different single-target regression algorithms on each cluster having the embedding vectors as features for predicting separate macronutrient values (carbohydrates, fat, protein and water), (v) evaluate the methodology by comparing the predicted with the actual values of the macronutrients.

2.3.1. Representation Learning

The starting point is the textual data, in our case the short text descriptions of the food products/recipes, alongside with their FoodEx2 codes and macronutrient values. For representing the textual data as vectors, the embeddings are generated for the whole food product name/description, using two different approaches:

1. Learning word vector representations (word embeddings) with the Word2Vec and GloVe methods – The vector representations of the whole description are obtained with merging separate word embeddings generated for each separate word in the sentence (food product name/description). If D as a food product description consisted of n words:

$$D = \{word_1, word_2, \ldots, word_n\} \tag{1}$$

And $E[word]$ is the vector representation (embedding) of a separate word:

$$E[word_a] = [x_{a1}, x_{a2}, \ldots, x_{ad}] \tag{2}$$

where $a \in \{1, \ldots, n\}$, n being the number of words in the description, and d is the dimension of the word vectors, which is defined manually for both Word2Vec and GloVe. These vectors are representations of words, to obtain the vector representations for the food product description we apply two different heuristics for merging the separate word vectors. Our two heuristics of choice are:

- Average – The vector representation for the food product description is calculated as an average from the vectors of the words from which it consists of:

$$E_{average}[D] = \left[\frac{x_{11} + \ldots + x_{n1}}{n}, \frac{x_{12} + \ldots + x_{n2}}{n}, \ldots, \frac{x_{1d} + \ldots + x_{nd}}{n}\right] \tag{3}$$

- Sum – The vector representation for each food product/recipe description is calculated by summing the vector representations of the words it consists of:

$$E_{sum}[D] = [x_{11} + \ldots + x_{n1}, x_{12} + \ldots + x_{n2}, \ldots, x_{1d} + \ldots + x_{nd}] \tag{4}$$

where $E_{average}[D]$ and $E_{sum}[D]$ are the merged embeddings, i.e., embeddings for the whole description. When generating the Word2Vec and GloVe embeddings, we considered different values for the dimension size and sliding window size. The dimension sizes of choice are [50,100,200], also for the Word2Vec embeddings we considered the two types of feature extraction available: CBOW and SG. For these dimensions we assign different values to the parameter called 'sliding' window. This parameter indicates the distance within a sentence between the current word and the word being predicted. The values of chose are [2,3,5,10] because our food product descriptions are not very long – the average number of words in a food product description in the dataset is 11, while the maximum number of words is 30). By combining these parameter values, 24 Word2Vec models were trained, plus considering the heuristics for combining, a total of 48 models, while with GloVe a total of 24 models were trained.

2. Learning paragraph vector representations with Doc2Vec algorithm – The Doc2Vec algorithm is used to generate vector representations for each description (sentence). If D is the description of the food product/description, then $E_{Doc2Vec}$ is the sentence vector representation generated with Doc2Vec is as follows:

$$E_{Doc2Vec}[D] = [x_1, x_2, \ldots, x_d] \tag{5}$$

where d is the predefined dimension of the vectors. Same as the two chosen word embedding methods, we considered different dimension sizes and sliding window sizes, specifically [2,3,5,10] for the sliding window and [50,100,200] for the dimension size. We also considered the two types architectures in the Doc2Vec model - PV-DM and PV-DBOW, and we used the non-concatenative mode (separate models for the sum option, and separate for the average option) because if we used the concatenation of context vectors rather than sum/average the result would be a much-larger model. Taking into account all these parameters there are 48 Doc2Vec models trained in total.

2.3.2. Unsupervised Machine Learning

Foods exhibit large variations in the nutrient content, therefore have very unbalanced macronutrient content. The dataset in our experiments includes a broad variety of foods, which implies that the content of a macronutrient can go from one extreme to another. Therefore, it goes without saying that in order to have better predictions for the content of macronutrients, food items should be grouped by some similarity. Here, the FoodEx2 codes that are available come into use, since they already contain domain knowledge, and based on them food items are grouped in food groups and broader food categories in the FoodEx2 hierarchy [40].

Independently of the representation learning process, we used the method presented in [39], where the FoodEx2 hierarchy is presented as Poincaré graph embeddings and then the FoodEx2 codes based on these embeddings are clustered into 230 clusters. This clustering process is performed on the bottom end of the hierarchy, i.e., on the leaves of the graph. Given that our dataset is rather small compared to the total number of FoodEx2 codes in the hierarchy, and the fact that when assigned a cluster number some of the clusters in our dataset will contain very few or no elements at all, we decided to do a post-hoc cluster merging. The post-hoc cluster merging is performed following a bottom up approach, the clusters are merged based on their top-level parents, going level deeper until we have as evenly distributed clusters as possible.

2.3.3. Supervised Machine Learning

The last part of the methodology is the supervised machine learning part, which on input receives the outputs from the representation learning part and the unsupervised machine learning part. This part consists of applying single-target regression algorithms in order to predict the separate macronutrient values.

Separate prediction models are trained for each macronutrient, because from the conducted correlation test (Pearson's correlation coefficient) we concluded that there is no correlation between the target variables. In a real-time scenario, it is somewhat hard to select the right machine learning algorithm for the purpose. The overall most accepted approach is to select few algorithms, select ranges for the hyper-parameters for each algorithm, perform hyper-parameter tuning, and evaluate the estimators' performances with cross-validation by the same data in each iteration, benchmark the algorithms and select the best one(s). When working with regression algorithms, the most common baseline is using mean or median (central tendency measures) of the train part of the dataset for all the predictions.

2.3.4. Tolerance for Nutrient Values

The main goal is obtaining macronutrient values which are expressed in grams, and by international legalizations and regulations can have defined tolerances. The European Commission Health and Consumers Directorate General in 2012 published [45], with the aim to provide advised recommendations for calculation of the acceptable differences between quantities of nutrients on the label declarations of food products and the ones established in Regulation EU 1169/2011 [46]. These tolerances for the food product labels are important as it is impossible for foods to contain the exact levels of nutrients that are presented on the labels, as a consequence of the natural variations of foods, as well as the variations occurring during production and the storage process. However,

the nutrient content of foods should not deviate substantially from labelled values to the extent that such deviations could lead to consumers being misled. From the tolerance levels stated in [45], for our particular case we used the tolerance levels for the nutrition declaration of foods that do not include food supplements, out of which we used the needed information presented in Table 2 – where the allowed deviations are presented for each of the four macronutrients, depending on their quantity in 100 grams of the food in matter. These tolerance levels are included at the very final step in our methodology in the determination on how accurate the predicted macronutrient values are.

Table 2. Tolerated differences in nutrition content in foods besides food supplements.

Quantity per 100 g/Macronutrient	Tolerances (Allowed Deviations in Quantity)			
	Carbohydrates	Protein	Water	Fat
<10 g per 100 g		±2 g		±1.5 g
10–40 g per 100 g		±20%		±20%
>40 g per 100 g		±8 g		±8 g

3. Results

The first step towards the evaluation is pre-processing of the data. Our dataset for evaluation is a subset from the original dataset, obtained by extracting the English food product descriptions, alongside the columns with the macronutrient values (carbohydrates, fat, protein and water). The text descriptions are tokenized. The punctuation signs and numbers that represent quantities are removed, whereas the percentage values (of fat, of sugar, of cocoa ...) which contain valuable information concerning the nutrient content, and stop words which add meaning to the description, are kept. The next step is word lemmatization [47], separate lemmatizers are used for the English names and the Slovene names. In Table 3 a few examples of the pre-processed data for the English names are presented.

Table 3. Examples of pre-processed English descriptions.

Original Description	Pre-processed Description
Potatoes, mashed, dehydrated, prepared from flakes without milk, whole milk and butter added	['potato', 'mashed', 'dehydrated', 'prepared', 'from', 'flake', 'without', 'milk', 'whole', 'milk', 'and', 'butter', 'added']
Milk chocolate with 30% cocoa, Gorenjka (250 g)	['milk','chocolate', 'with', '30', 'cocoa', 'gorenjka']

After obtaining the data in the desired format, the next step is to apply the algorithms for generating embeddings. For this purpose we used the Gensim [48] library in Python, and the corresponding packages for the Word2Vec and Doc2Vec algorithms. The embedding vectors represent our base for the next steps.

Independently of this process, the data is clustered, i.e., the instances are divided in clusters based on their FoodEx2 codes. In the beginning from the clustering in [39] there are 230 clusters, when assigned a cluster number, the instances in our dataset are clustered. From this initial clustering we can note that not all clusters have elements in them, and some of them have very few elements. Therefore, the post-hoc cluster merging is performed, where we merge the clusters following a bottom up approach. For our dataset we went for the parents on the third level in the FoodEx2 hierarchy and we obtained 9 clusters. In Table 4 a few examples from each cluster are given (the English names are given for convenience purposes).

Table 4. Example instances from each cluster.

Cluster Number	Example Food Products		
Cluster 1	Oil, industrial, mid-oleic, sunflower, principal uses frying and salad dressings	Homemade minced lard, Mesarija Kragelj	Margarine (with added vegetable sterols 0,75g/10g), line Becel pro-activ, Unilever
Cluster 2	Peanuts, all types, oil-roasted, with salt	Seeds, pumpkin and squash seed kernels, dried	Avocados, raw, California
Cluster 3	Cheese, processed, 60% fat in dry matter	Yogurt, fruit (peach, cereals), low fat 2.6% milkfat	Baby food, cottage cheese, creamed, fruit (strawberry, banana), FruchtZwerge, Danone
Cluster 4	Plums, canned, purple, light syrup pack, solids and liquids	Segedin cabbage with pork meat	Buckwheat porridge sauted with onion and garlic
Cluster 5	Fried chicken file (canola oil, without breadcrumbs)	Trout with parsley and garlic sauce	Beef, rib, whole (ribs 6–12), separable lean and fat, trimmed to 1/8 of an inch of fat, all grades, cooked, roasted
Cluster 6	Fruit tea infusion, with sugar and lemon	Soup made of turnip cabbage, peas and tomato (olive oil, stock)	Chicken stew with seasonal vegetables, without roux
Cluster 7	Fish, salmon, pink, canned, without salt, solids with bone and liquid	Salty anchovies in vegetable oil	Tuna with beans, canned
Cluster 8	Ham, sliced, regular (approximately 11% fat)	Chicken hot dog, pan-fried	Turkey ham, sliced, extra lean, prepackaged or deli-sliced
Cluster 9	Egg, whole, cooked, scrambled	Fried egg (olive oil)	Egg spread

The next step in our methodology is the machine learning part – applying single-target regressions according to the following setup:

1. Select regression algorithms – Linear regression, Ridge regression, Lasso regression, and ElasticNet regression (using the Scikit-learn library in Python [49]).
2. Select parameter ranges for each algorithm and perform hyper-parameter tuning – Ranges and values are a priori given for all the parameters for all the regression algorithms. From all the combinations the best parameters for the model training are then selected with GridSearchCV (using the Scikit-learn library in Python [49]). This is done for each cluster separately.
3. Apply k-fold cross-validation to estimate the prediction error – We train models for each cluster using each of the selected regression algorithms. The models are trained with the previously selected best parameters for each cluster and then evaluated with cross-validation. We chose the matched sample approach for comparison of the regressors, i.e., using the same data in each iteration.
4. Apply tolerance levels and calculate accuracy – The accuracy is calculated according to the tolerance levels in Table 2. If a_i is the actual value of the i^{th} instance from the test set on a certain iteration of the k-fold cross-validation, and p_i is the predicted values of the same, i^{th}, instance of the test set, then:

$$d_i = \left|a_i - p_i\right|, \tag{6}$$

d_i is the absolute difference between the two set values. We define a binary variable that is assigned a positive value if the predicted value is in the tolerance level.

$$\begin{gathered} allowed = 1\ if: \\ a_i \leq 10 \ \wedge \begin{cases} d_i \leq 2, \ for\ protein\ and\ carbohydrate \\ d_i \leq 1.5, \ for\ fat \end{cases} \\ a_i > 10 \ \wedge a_i \leq 40 \ \wedge \ d_i \leq 0.2 \times a_i \\ a_i > 40 \ \wedge \ d_i \leq 8 \end{gathered} \tag{7}$$

At the end we calculate the accuracy as the ratio of predicted values that were in the 'allowed' range, i.e., tolerance level:

$$Accuracy = \frac{\sum_{i=1}^{n} allowed}{n} \tag{8}$$

where n is the number of instances in the test set. The accuracy percentage is calculated for the baseline mean and baseline median as well – the percentage of baseline values (means and medians from each cluster) that falls in the tolerance level range, calculated according to Equations (6)–(8), where a_i is the actual value of the i^{th} instance from the test set on a certain iteration of the k-fold cross-validation, and instead of p_i we have:

$$b = \begin{cases} \frac{\sum_{i=1}^{m} x_i}{m}, & \textit{the baseline is the mean} \\ \frac{\mathbf{X}_{[(m+1)/2]} + \mathbf{X}_{[(m+1)/2]}}{2}, & \textit{the baseline is the median} \end{cases} \tag{9}$$

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

The accuracy percentages are calculated for each fold in each cluster, and at the end for each cluster we calculate an average of the percentages from each fold. In Table 5 the results obtained from the experiments with the embeddings generated from the English names are presented, and in Table 6 with the embeddings generated from the Slovene names.

Table 5. Accuracy percentages after k-fold cross validation on each cluster obtained with the embeddings for the English names of the food products. Target: C—Carbohydrates, F—Fat, P—Protein, W—Water. The bolded numbers in the table represent the overall best performance for each macronutrient in the given cluster.

Cluster	Target	Accuracy				
		Word2Vec	**GloVe**	**Doc2Vec**	**Mean**	**Median**
1	C	**59.21**	47.84	50.11	1.00	17.47
	F	44.26	35.95	**49.32**	5.05	10.21
	P	56.37	**60.32**	56.95	13.16	14.26
	W	40.32	**52.32**	48.21	8.05	9.26
2	C	**34.84**	34.32	33.22	10.95	13.27
	F	**67.22**	64.69	64.69	7.93	60.55
	P	**63.87**	61.34	59.22	7.58	31.89
	W	50.44	**52.83**	52.41	17.89	19.73
3	C	46.51	46.18	**46.98**	11.13	15.74
	F	**67.42**	63.62	64.00	6.84	59.81
	P	69.64	65.47	**70.74**	8.75	58.55
	W	56.68	**60.85**	58.70	12.18	29.83
4	C	40.92	**43.32**	40.53	12.95	16.40
	F	**68.28**	66.40	66.67	4.79	62.43
	P	**72.50**	70.85	71.71	7.23	66.07
	W	59.09	**61.51**	60.99	11.24	33.86
5	C	**46.38**	37.65	46.07	9.58	15.80
	F	**66.12**	62.38	62.38	4.57	42.43
	P	66.12	63.63	**66.83**	8.73	52.38
	W	49.87	48.95	**53.98**	12.80	21.53
6	C	29.46	30.55	**33.30**	7.90	10.24
	F	41.66	41.08	**43.26**	6.68	29.76
	P	53.35	54.37	**55.81**	15.09	20.11
	W	38.01	39.69	**41.28**	11.03	15.45
7	C	**72.78**	**72.78**	**72.78**	11.11	41.11
	F	42.78	48.33	**53.33**	5.56	11.11
	P	**73.89**	**73.89**	**73.89**	31.67	15.00
	W	46.67	48.89	**57.22**	15.56	20.00

Table 5. *Cont.*

Cluster	Target	Accuracy				
		Word2Vec	**GloVe**	**Doc2Vec**	**Mean**	**Median**
8	C	**58.31**	51.60	55.58	0.95	21.69
	F	48.27	39.74	**50.17**	6.58	15.15
	P	60.48	63.25	**67.06**	7.62	19.74
	W	41.60	48.27	**49.83**	11.34	11.26
9	C	**86.36**	81.82	72.73	27.27	36.36
	F	**50.00**	40.91	45.45	4.55	4.55
	P	**77.27**	72.73	63.64	36.36	31.82
	W	45.45	40.91	**50.00**	9.09	18.18

Table 6. Accuracy percentages after k-fold cross validation on each cluster obtained with the embeddings for the Slovene names of the food products. Target: C—Carbohydrates, F—Fat, P—Protein, W—Water. The bolded numbers in the table represent the overall best performance for each macronutrient in the given cluster.

Cluster	Target	Accuracy				
		Word2Vec	**GloVe**	**Doc2Vec**	**Mean**	**Median**
1	C	**61.37**	54.11	52.00	1.00	17.47
	F	**44.26**	37.00	41.26	5.05	10.21
	P	**58.26**	50.00	53.26	13.16	14.32
	W	34.05	**37.05**	33.89	8.05	9.26
2	C	27.47	24.43	**32.15**	10.95	14.00
	F	**70.28**	67.04	64.69	7.93	60.55
	P	**63.72**	60.12	59.22	7.58	31.54
	W	**49.96**	43.28	48.38	17.89	19.55
3	C	**47.28**	41.51	45.00	11.13	16.13
	F	**67.42**	63.99	63.62	6.84	59.81
	P	**69.27**	65.83	69.20	8.75	58.55
	W	52.86	43.97	**54.34**	12.18	29.44
4	C	34.78	28.49	**40.33**	12.95	16.93
	F	**70.13**	67.74	66.40	4.79	62.43
	P	**72.50**	69.58	70.38	7.23	66.07
	W	54.24	47.66	**55.79**	11.24	33.86
5	C	**47.63**	41.40	45.95	9.58	15.80
	F	**66.12**	62.80	62.38	4.57	42.43
	P	**66.12**	64.47	64.47	8.73	52.38
	W	48.18	41.48	**51.02**	12.80	21.12
6	C	31.42	25.61	**33.74**	7.90	10.75
	F	39.34	34.97	**44.42**	6.68	29.98
	P	53.36	50.73	**63.13**	15.09	20.33
	W	41.21	34.67	**41.85**	11.03	15.45
7	C	**72.78**	67.78	**72.78**	11.11	41.11
	F	**58.33**	37.78	48.33	5.56	11.11
	P	63.89	63.89	**69.44**	31.67	15.00
	W	**46.11**	36.67	41.11	15.56	20.00
8	C	56.41	49.91	**56.45**	0.95	21.69
	F	**47.32**	43.51	44.55	6.58	15.15
	P	**64.29**	59.70	61.43	7.62	19.74
	W	35.11	**36.80**	35.11	11.34	11.26
9	C	**86.36**	72.73	**86.36**	27.27	36.36
	F	**50.00**	31.82	**50.00**	4.55	4.55
	P	63.64	59.09	**68.18**	36.36	31.82
	W	54.55	36.36	45.45	9.09	18.18

In these tables we give the accuracy percentages from the predictions for each target macronutrient in each cluster. From these tables we can see that having the Word2Vec and Doc2Vec embeddings as features for the regressions yielded better results in more cases than having the GloVe embedding vectors as inputs to the regressions, but this difference is not big enough to say that these two embedding algorithms outperformed GloVe. In Figures 2–5 the results for each target macronutrient are presented graphically.

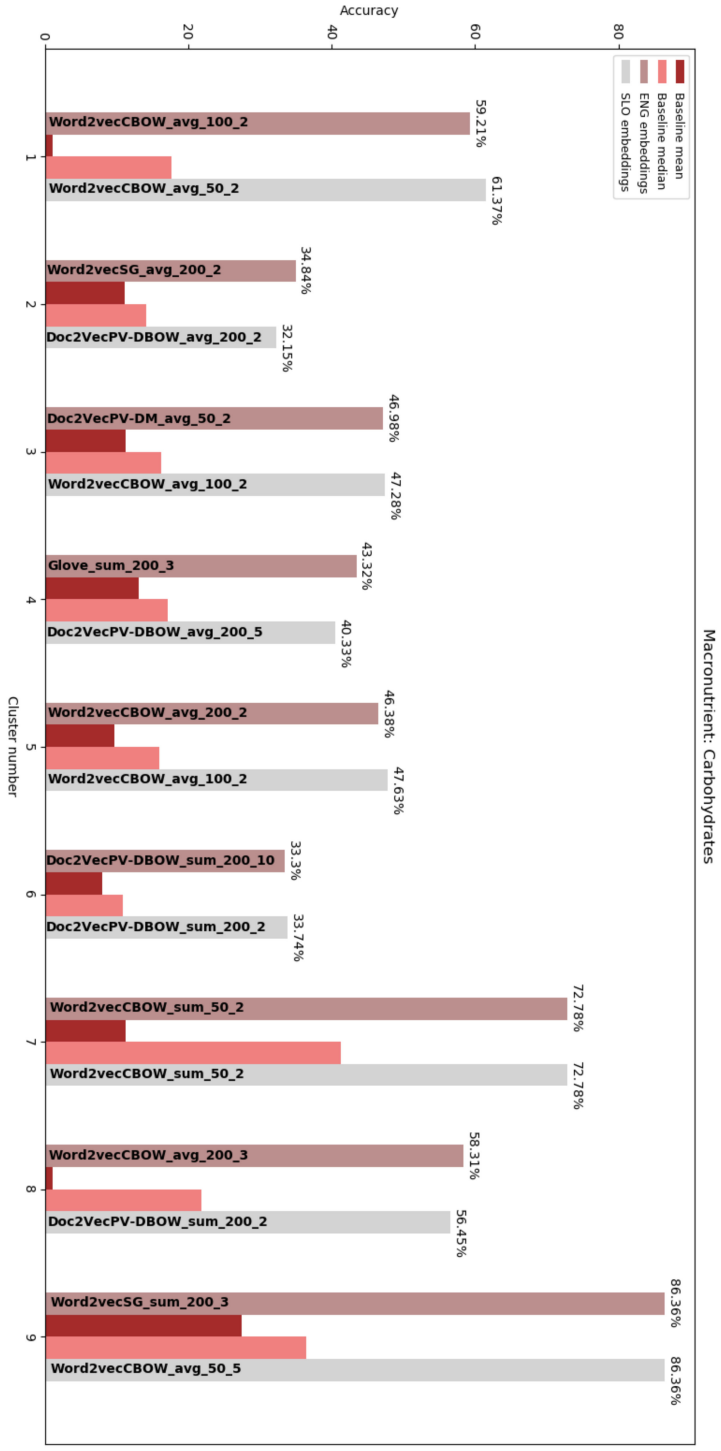

Figure 2. Best prediction accuracies for carbohydrates predictions obtained from the embeddings for the English names and Slovene names for each cluster compared to the baseline mean and median for the particular cluster.

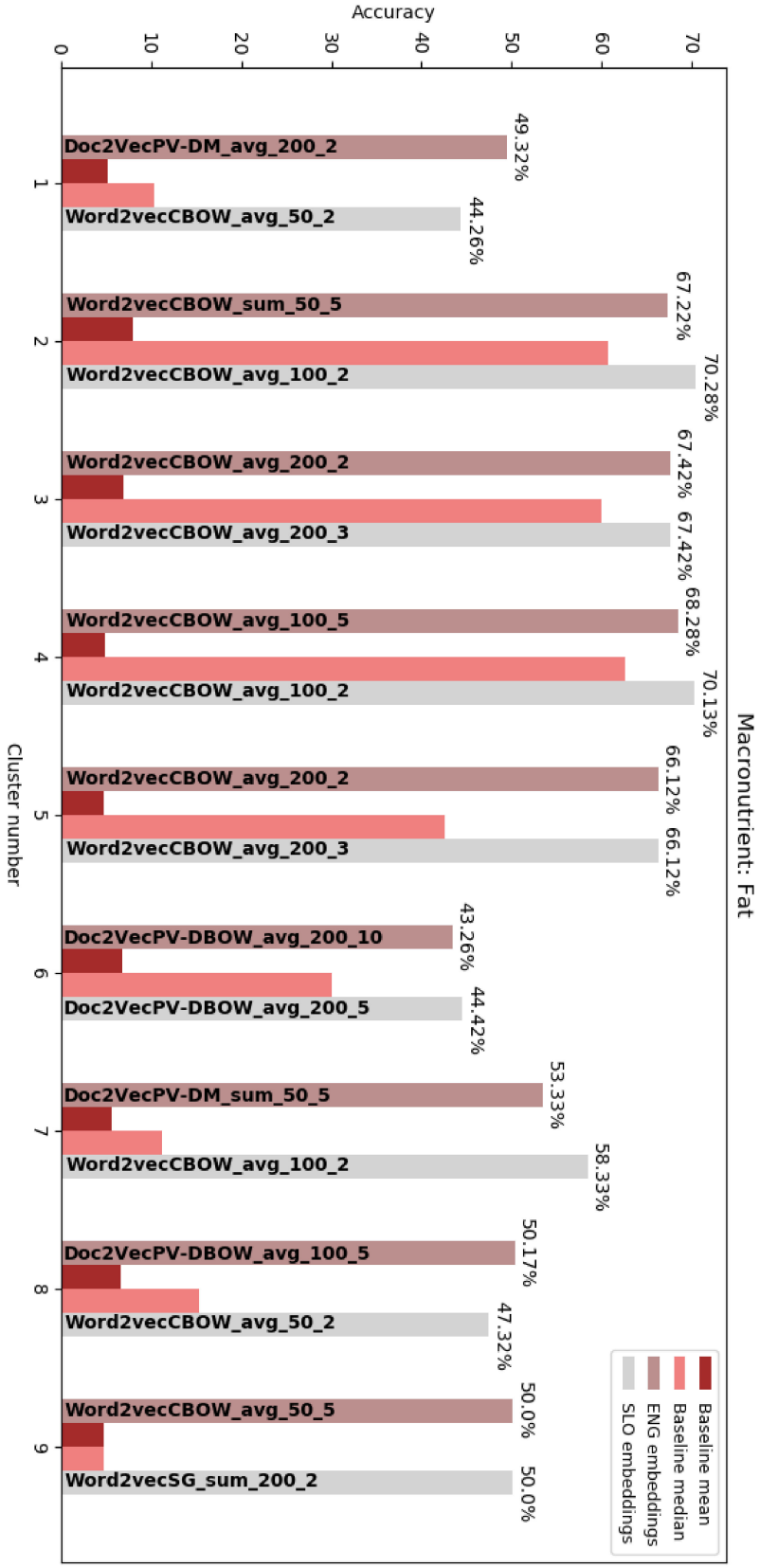

Figure 3. Best prediction accuracies for fat predictions obtained from the embeddings for the English names and Slovene names for each cluster compared to the baseline mean and median for the particular cluster.

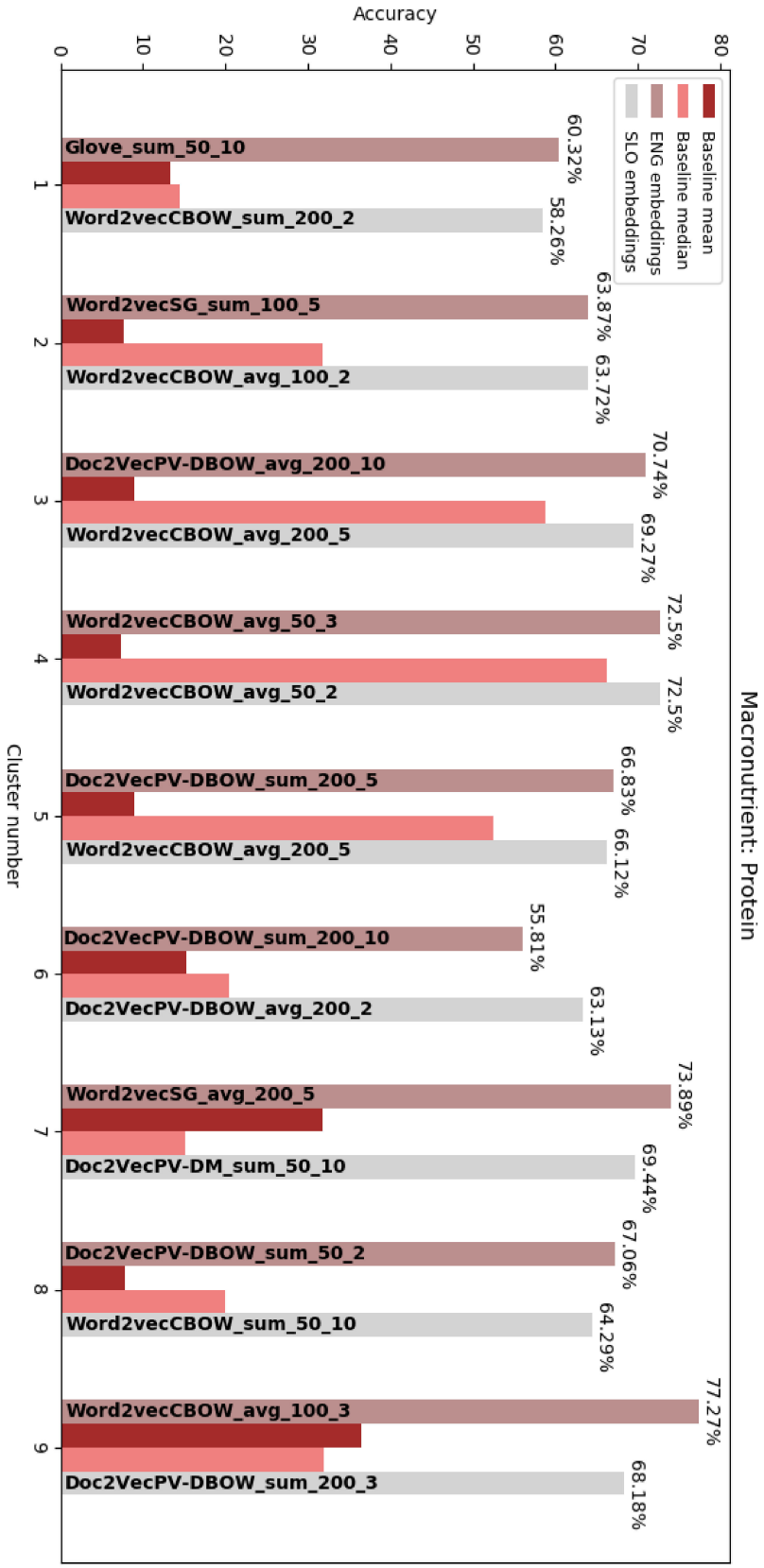

Figure 4. Best prediction accuracies for protein predictions obtained from the embeddings for the English names and Slovene names for each cluster compared to the baseline mean and median for the particular cluster.

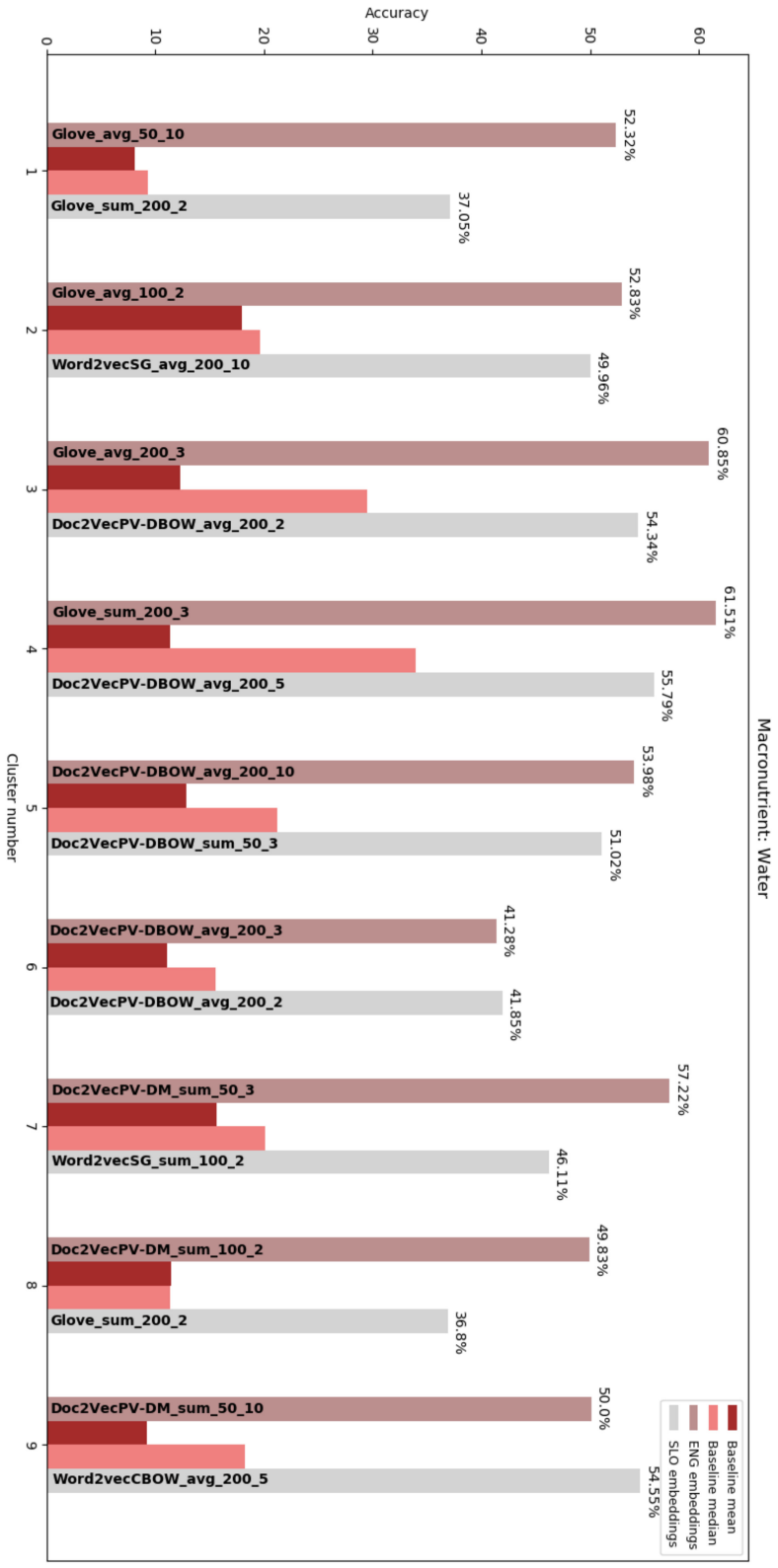

Figure 5. Best prediction accuracies for water predictions obtained from the embeddings for the English names and Slovene names for each cluster compared to the baseline mean and median for the particular cluster.

In the graphs, for each target macronutrient, for each cluster, we give the best result obtained with the embedding vectors from the English and Slovene names and compare them with the baseline

mean and median for the particular cluster. In the graphs the embedding algorithm that yields the best results alongside with the parameters and heuristic is given as:

$$E\_h\_d\_w, \begin{cases} h \in \{sum, average\}, \textit{ is the chosen heuristic} \\ d \in \{50, 100, 200\}, \textit{is the dimension} \\ w \in \{2, 3, 5, 10\}, \textit{is the sliding window} \end{cases} \tag{10}$$

where, E is the embedding algorithm (Word2Vec, GloVe or Doc2Vec). We can see that the embedding algorithm that yields the best results changes, but the in all cases the embedding algorithm gives better results than the baseline methods. In Table 7, we present the embedding algorithms (with all the parameter used) that gave the best results for each target macronutrient in each cluster, alongside with the regression algorithm used for making the predictions.

Table 7. Embedding and regression algorithms which yielded highest accuracies for each macronutrient prediction in each cluster. Target: C—Carbohydrates, F—Fat, P—Protein, W—Water.

Cluster	Target	Embedding Algorithm		Regression Algorithm	
		ENG	SLO	ENG	SLO
1	C	Word2VecCBOW_avg_100_2	Word2VecCBOW_avg_50_2	ElasticNet	Ridge
	F	Doc2VecPV-DM_avg_200_2	Word2VecCBOW_avg_50_2	Lasso	Ridge
	P	GloVe_sum_50_10	Word2VecCBOW_sum_200_2	Lasso	Ridge
	W	GloVe_avg_50_10	GloVe_sum_200_2	ElasticNet	Ridge
2	C	Word2VecSG_avg_200_2	Doc2VecPV-DBOW_avg_200_2	Ridge	Ridge
	F	Word2VecCBOW_sum_50_5	Word2VecCBOW_avg_100_2	Lasso	Ridge
	P	Word2VecSG_sum_100_5	Word2VecCBOW_avg_100_2	Ridge	Ridge
	W	GloVe_avg_100_2	Word2VecSG_avg_200_10	Ridge	Ridge
3	C	Doc2VecPV-DM_avg_50_2	Word2VecCBOW_avg_100_2	Ridge	ElasticNet
	F	Word2VecCBOW_avg_200_2	Word2VecCBOW_avg_200_3	Ridge	Ridge
	P	Doc2VecPV-DBOW_avg_200_10	Word2VecCBOW_avg_200_5	Ridge	Ridge
	W	GloVe_avg_200_3	Doc2VecPV-DBOW_avg_200_2	Ridge	ElasticNet
4	C	GloVe_sum_200_3	Doc2VecPV-DBOW_avg_200_5	Ridge	ElasticNet
	F	Word2VecCBOW_avg_100_5	Word2VecCBOW_avg_100_2	Lasso	Ridge
	P	Word2VecCBOW_avg_50_3	Word2VecCBOW_avg_50_2	Lasso	Ridge
	W	GloVe_sum_200_3	Doc2VecPV-DBOW_avg_200_5	Ridge	ElasticNet
5	C	Word2VecCBOW_avg_200_2	Word2VecCBOW_avg_100_2	Ridge	Lasso
	F	Word2VecCBOW_avg_200_2	Word2VecCBOW_avg_200_3	Ridge	Ridge
	P	Doc2VecPV-DBOW_sum_200_5	Word2VecCBOW_avg_200_5	ElasticNet	Ridge
	W	Doc2VecPV-DBOW_avg_200_10	Doc2VecPV-DBOW_sum_50_3	Ridge	Lasso
6	C	Doc2VecPV-DBOW_sum_200_10	Doc2VecPV-DBOW_sum_200_2	Ridge	Ridge
	F	Doc2VecPV-DBOW_avg_200_10	Doc2VecPV-DBOW_avg_200_5	Ridge	Ridge
	P	Doc2VecPV-DBOW_sum_200_10	Doc2VecPV-DBOW_avg_200_2	Ridge	Ridge
	W	Doc2VecPV-DBOW_avg_200_3	Doc2VecPV-DBOW_avg_200_2	Ridge	Ridge
7	C	Word2VecCBOW_sum_50_2	Word2VecCBOW_sum_50_2	Linear	Linear
	F	Doc2VecPV-DM_sum_50_5	Word2VecCBOW_avg_100_2	ElasticNet	Linear
	P	Word2VecSG_avg_200_5	Doc2VecPV-DM_sum_50_10	Linear	ElasticNet
	W	Doc2VecPV-DM_sum_50_3	Word2VecSG_sum_100_2	Linear	Linear
8	C	Word2VecCBOW_avg_200_3	Doc2VecPV-DBOW_sum_200_2	Ridge	Ridge
	F	Doc2VecPV-DBOW_avg_100_5	Word2VecCBOW_avg_50_2	Lasso	Ridge
	P	Doc2VecPV-DBOW_sum_50_2	Word2VecCBOW_sum_50_10	ElasticNet	Ridge
	W	Doc2VecPV-DM_sum_100_2	GloVe_sum_200_2	Ridge	Ridge
9	C	Word2VecSG_sum_200_3	Word2VecCBOW_avg_50_5	Lasso	Linear
	F	Word2VecCBOW_avg_50_5	Word2VecSG_sum_200_2	Linear	Linear
	P	Word2VecCBOW_avg_100_3	Doc2VecPV-DBOW_sum_200_3	Linear	Lasso
	W	Doc2VecPV-DM_sum_50_10	Word2VecCBOW_avg_200_5	Lasso	Linear

4. Discussion

From the obtained results we can observe that the highest percentage of correctly predicted macronutrient values is obtained in cluster 9, for the prediction of carbohydrates: 86,36%, 81,82% and 72,73% for the English names and 86,36%, 72,73% and 86,36% for the Slovene names, and for the Word2Vec, GloVe and Doc2Vec algorithms appropriately, whereas the baseline (both mean and median) is more than half less. Following these results are the predictions for protein quantity in the same cluster, and then the predictions for protein and carbohydrates in cluster 7. When inspecting these two clusters, we concluded that these were the only two clusters that were not merged with other ones, therefore, the FoodEx2 hierarchy is on a deeper level, and the foods inside these clusters are more similar to each other compared to food in other clusters. Cluster 9 consists of types of egg products, and simple egg dishes – each of these foods have almost identical macronutrients because they only contain one ingredient – eggs. Cluster 7, on the other hand contains fish products, either frozen or canned. If we do not consider the results from these two clusters, then the best results are obtained for protein predictions in cluster 4 (70%–72%) and fat predictions (66%–68%), but compared to the baseline median of that cluster, they are not much better, but if we look at the results from the protein predictions in cluster 8 (60%–67%) we can see that the obtained accuracies are much higher than the baseline mean and median for this cluster. Cluster 8 mainly contains types of processed meats, which can vary notably in fat content, but have similarities in the range of protein content.

For comparison reasons, we also ran the single-target regressions without clustering the dataset. The results are presented in Figure 6.

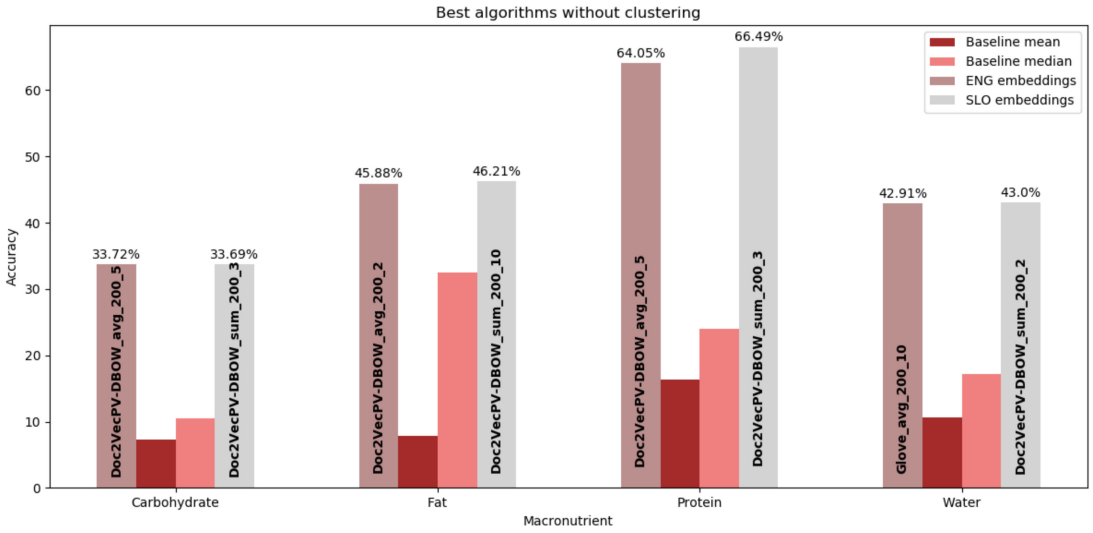

Figure 6. Best prediction accuracies for each macronutrient obtained from the embeddings for the English and Slovene names compared to the baseline mean and median from the whole dataset.

From this graph we can conclude the same – the embedding algorithms give better results than the baseline mean and median (in this case of the whole dataset), for each target macronutrient. The best results, again, are obtained for the prediction of protein content (62%–64%).

In Table 8, we give the parameters for the embedding algorithms and the regressors with which the best results were obtained without clustering the data.

From these results, it is worth arguing that modeling machine learning techniques on food data previously clustered based on FoodEx2 codes would yield better results than predicting on the whole dataset. If we compare the performances of the three embedding algorithms, it is hard to argue if one outperformed the others, or if one underperformed compared to the other two. This outcome is due to the fact that we are dealing with fairly short textual descriptions.

Table 8. Embedding and regression algorithms which yielded highest accuracies for each macronutrient prediction on the whole dataset (without clustering). Target: C—Carbohydrates, F—Fat, P—Protein, W—Water.

Target	Embedding Algorithm		Regression Algorithm	
	ENG	SLO	ENG	SLO
C	Doc2VecPV-DBOW_avg_200_5	Doc2VecPV-DBOW_sum_200_3	Ridge	Ridge
F	Doc2VecPV-DBOW_avg_200_2	Doc2VecPV-DBOW_sum_200_10	Lasso	ElasticNet
P	Doc2VecPV-DBOW_avg_200_5	Doc2VecPV-DBOW_sum_200_3	Ridge	Ridge
W	GloVe_avg_200_10	Doc2VecPV-DBOW_sum_200_2	Ridge	Linear

Given the fact that the results with the clustering are better than the results without, and we rely so strongly on having the FoodEx2 codes in order to cluster the foods, the availability of the FoodEx2 codes is of big importance and therefore a limitation of the methodology. For this purpose, we can rely on a method such as StandFood [50], which is a natural language processing methodology developed for classifying and describing foods according to FoodEx2. When this limitation is surpassed, the application of our method can be fully automated.

From a theoretical viewpoint this methodology considers the benefits of using representation learning as the base of a predictive study, and proves that dense real-valued vectors can capture enough semantics even from a short text description (without including the needed details for the task in question – in our case, measurements or exact ingredients) in order to be considered in a predictive study for complicated and value-sensitive task such as predicting macronutrient content. This study offers a fertile ground for further exploration of representation learning and considering more complex embedding algorithms – using transformers [51,52] and fine tuning them for this task.

From a managerial viewpoint the application of this methodology opens up many possibilities for facilitating and easing the process of calculating macronutrient content, which is crucial for dietary assessment, dietary recommendations, dietary guidelines, macronutrient tracking, and other such tasks which are key tools for doctors, health professionals, dieticians, nutritional experts, policy makers, professional sport coaches, athletes, fitness professionals, etc.

5. Conclusions

We live in a modern health crisis. We have a cure for almost everything, and yet the most common causes of biggest mortality factor – cardiovascular diseases are nutrition and diet related. Knowing what is in our food, and understanding its nutritional content (macro and micronutrients) is the first step, that is in our power, towards the prevention of diet-related diseases. There is an overwhelming amount of nutrition-related data available, and most of it comes in textual form, structured and unstructured. Data Science can help us utilize this data for our benefit. We presented a methodology that combines representation learning and machine learning for the task of predicting macronutrient values from short textual descriptions of food data – a combination of food products and recipes. Taking learned vector representations of the descriptions as features, and applying different regression algorithms on separate clusters of the data obtained by clustering based on Poincaré graph embeddings from the FoodEx2 codes of the data, and obtaining results with as high as 86% accuracy, this approach proves to be very effective for this task. For our future work we intend to extend this methodology with the state-of-the-art embeddings based on transformers – Bert Embeddings [51], clustering on an upper level of the FoodEx2 hierarchy, and including methods for obtaining FoodEx2 codes, when they are not available [50], as well evaluating it on a bigger dataset, with longer, more detailed descriptions.

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.

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

References

1. Willett, W.; Rockström, J.; Loken, B.; Springmann, M.; Lang, T.; Vermeulen, S.; Garnett, T.; Tilman, D.; DeClerck, F.; Wood, A.; et al. Food in the Anthropocene: The EAT–Lancet Commission on healthy diets from sustainable food systems. *The Lancet* **2019**, *393*, 447–492. [CrossRef]
2. Branca, F.; Demaio, A.; Udomkesmalee, E.; Baker, P.; Aguayo, V.M.; Barquera, S.; Dain, K.; Keir, L.; Lartey, A.; Mugambi, G.; et al. A new nutrition manifesto for a new nutrition reality. *The Lancet* **2020**, *395*, 8–10. [CrossRef]
3. Keeley, B.; Little, C.; Zuehlke, E. *The State of the World's Children 2019: Children, Food and Nutrition–Growing Well in a Changing World*; UNICEF: New York, NY, USA, 2019.
4. Mbow, H.-O.P.; Reisinger, A.; Canadell, J.; O'Brien, P. *Special Report on Climate Change, Desertification, Land Degradation, Sustainable Land Management, Food Security, and Greenhouse Gas Fluxes in Terrestrial Ecosystems (SR2)*; IPCC: Geneva, Switzerland, 2017.
5. 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]
6. 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.
7. 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.
8. Greenfield, H.; Southgate, D.A. *Food Composition Data: Production, Management, and Use*; Food and Agriculture Organization: Rome, Italy, 2003; ISBN 978-92-5-104949-5.
9. 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]
10. 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]
11. 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]
12. Pouladzadeh, P.; Shirmohammadi, S.; Al-Maghrabi, R. Measuring calorie and nutrition from food image. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 1947–1956. [CrossRef]
13. Ege, T.; Yanai, K. Image-based food calorie estimation using recipe information. *IEICE Trans. Inf. Syst.* **2018**, *101*, 1333–1341. [CrossRef]
14. Samsung Health (S-Health). Available online: https://health.apps.samsung.com/terms (accessed on 11 May 2020).
15. MyFitnessPal. Available online: https://www.myfitnesspal.com/ (accessed on 11 May 2020).
16. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef]
17. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Cogn Modeling* **1988**, *5*, 1. [CrossRef]
18. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
19. Mikolov, T. *Statistical Language Models Based on Neural Networks*; Presentation at Google, Mountain View, 2nd April 2012; Brno University of Technology: Brno, Czech Republic, 2012; Volume 80.
20. Caracciolo, C.; Stellato, A.; Rajbahndari, S.; Morshed, A.; Johannsen, G.; Jaques, Y.; Keizer, J. Thesaurus maintenance, alignment and publication as linked data: The AGROVOC use case. *Int. J. Metadatasemantics Ontol.* **2012**, *7*, 65–75. [CrossRef]

21. Weston, J.; Bengio, S.; Usunier, N. Wsabie: Scaling up to large vocabulary image annotation. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
22. Socher, R.; Lin, C.C.; Manning, C.; Ng, A.Y. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 129–136.
23. Glorot, X.; Bordes, A.; Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the Proceedings of the 28th International Conference on Machine Learning (ICML-11); Omnipress: Madison, WI, USA, 2011; pp. 513–520.
24. Turney, P.D. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Trans. Assoc. Comput. Linguist.* **2013**, *1*, 353–366. [CrossRef]
25. Turney, P.D.; Pantel, P. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res.* **2010**, *37*, 141–188. [CrossRef]
26. Mikolov, T.; Yih, W.; Zweig, G. Linguistic regularities in continuous space word representations. In Proceedings of the Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 746–751.
27. Eckart, C.; Young, G. The approximation of one matrix by another of lower rank. *Psychometrika* **1936**, *1*, 211–218. [CrossRef]
28. Habibi, M.; Weber, L.; Neves, M.; Wiegandt, D.L.; Leser, U. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* **2017**, *33*, i37–i48. [CrossRef]
29. Drozd, A.; Gladkova, A.; Matsuoka, S. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In Proceedings of the Coling 2016, the 26th International Conference on Computational Linguistics: Technical papers, Osaka, Japan, 11–17 December 2016; pp. 3519–3530.
30. Tshitoyan, V.; Dagdelen, J.; Weston, L.; Dunn, A.; Rong, Z.; Kononova, O.; Persson, K.A.; Ceder, G.; Jain, A. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **2019**, *571*, 95–98. [CrossRef]
31. Ispirova, G.; Eftimov, T.; Seljak, B.K. Comparing Semantic and Nutrient Value Similarities of Recipes. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 5131–5139.
32. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
33. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
34. Pennington, J.; Socher, R.; Manning, C. 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.
35. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
36. Nickel, M.; Kiela, D. Poincaré embeddings for learning hierarchical representations. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6338–6347.
37. Yang, Z.; Cohen, W.; Salakhudinov, R. *Revisiting Semi-Supervised Learning with Graph Embeddings.*; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 40–48.
38. Ristoski, P.; Paulheim, H. Rdf2vec: Rdf graph embeddings for data mining. In Proceedings of the International Semantic Web Conference, Hyogo, Japan, 17–21 October 2016; Springer: Cham, Switzerland, 2016; pp. 498–514.
39. Eftimov, T.; Popovski, G.; Valenčič, E.; Seljak, B.K. FoodEx2vec: New foods' representation for advanced food data analysis. *Food Chem. Toxicol.* **2020**, *138*, 111169. [CrossRef]
40. European Food Safety Authority The food classification and description system FoodEx2 (revision 2). *EFSA Supporting Publ.* **2015**, *12*, 804E.
41. Van der Laan, M.; Pollard, K.; Bryan, J. A new partitioning around medoids algorithm. *J. Stat. Comput. Simul.* **2003**, *73*, 575–584. [CrossRef]

42. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
43. The European Food Safety Authority. Available online: https://www.efsa.europa.eu/en/data/food-consumption-data (accessed on 11 May 2020).
44. Authority, E.F.S. Use of the EFSA comprehensive European food consumption database in exposure assessment. *EFSA J.* **2011**, *9*, 2097. [CrossRef]
45. 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 11 May 2020).
46. European Commission. 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.
47. Korenius, T.; Laurikkala, J.; Järvelin, K.; Juhola, M. Stemming and lemmatization in the clustering of finnish text documents. In Proceedings of the thirteenth ACM international conference on Information and knowledge management, Washington, DC, USA, 8–13 November 2004; pp. 625–633.
48. Rehurek, R.; Sojka, P. *Gensim—Statistical Semantics In Python*; NLP Centre, Faculty of Informatics, Masaryk University: Brno, Czech Republic, 2011.
49. 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.
50. Eftimov, T.; Korošec, P.; Koroušić Seljak, B. StandFood: Standardization of foods using a semi-automatic system for classifying and describing foods according to FoodEx2. *Nutrients* **2017**, *9*, 542. [CrossRef]
51. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
52. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. Ernie: Enhanced representation through knowledge integration. *arXiv* **2019**, arXiv:1904.09223.