Ingredient substitution for improved recipe health using different clustering algorithms

Bikowski, Jannik
University of Regensburg
Chair of Information Science
Regensburg, Germany
jannik.bikowski@stud.uni-regensburg.de

Emme, Till
University of Regensburg
Chair of Information Science
Regensburg, Germany
till.emme@stud.uni-regensburg.de

Scheuer, Franz
University of Regensburg
Chair of Information Science
Regensburg, Germany
franz.scheuer@stud.uni-regensburg.de

ABSTRACT

Some recipes are very popular due to their simplicity or their final taste. But sometimes you would wish for a healthier alternative. In this paper we try to substitute unhealthy with healthier ingredients by comparing similar recipes and building clusters from the varying ingredients. We chose this approach because some ingredients are essential for a recipe and therefore we don't want to replace those, irrespective of their healthiness. The clustering algorithms we chose to compare for this approach are LSI, LDA, Word2Vec and FastText. From these four algorithms FastText provided the best results.

KEYWORDS

Probability, Expert Systems, Machine Learning, Information Retrieval, Recommender Systems, Recipes, Ingredient Substitution, Health, Clustering

ACM Reference Format:
Bikowski, Jannik, Emme, Till, and Scheuer, Franz. 2018. Ingredient substitution for improved recipe health using different clustering algorithms. In *Proceedings of Ing. sub. for improved recipe health (Chair of Information Science).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

The internet has become a prominent resource for finding and revising recipes for a number of reasons. The large scale and social nature of online recipe communities (including Allrecipes.com) give users a large amount of easy to access knowledge, intuitive search and a large number of recipes to choose from; Allrecipes.com alone hosts over 60,000 recipes. However, a known issue present in online recipe communities is that of recipe health; a majority of the recipes present in the Allrecipes.com database have been judged as unhealthy by two health standards supplied by the world health organization and the Food Standards Agency of the United Kingdom [17]. While recipe recommendation has attempted to integrate recipe health in recommendations, observed positive correlations between the high recipe ratings (and popularity) and low health scores limit the efficacy of systems to recommend both popular (desirable) and healthy recipes [17].

Chair of Information Science, Summer 2018, Regensburg, Germany

ACM ISBN 123-456-789-0.
https://doi.org/10.475/123_4

This work attempts to circumvent this trade-off by recommending popular recipes where the ingredient has been autonomously modified with the goal of improving the overall recipe health. To achieve this we first gathered all data from the database. We processed the ingredients and assigned a score to every single one to divide them into healthy and unhealthy. To retrieve the nutritional information to calculate the health score we used an API of the "United States Department of Agriculture". Afterwards we compared the ingredients of similar recipes and assumed that the matching ingredients would be essential to a recipe, irrespective of their healthiness. Because of that we decided to only substitute differing ingredients. Finally, the list of unequal ingredients was clustered by four different algorithms LSI, LDA, Word2Vec and FastText and the results were compared and evaluated.

A review of relevant literature is presented in the following section (2) followed by the substitution methodology in section 3. Lastly, results are presented and discussed in section 4 and conclusions are provided in section 5.

2 LITERATURE REVIEW

While there has been a large amount of work in the field of recommender systems[2, 14, 14], recipe recommendation has only recently received a large degree of focus by the research community.

Even less research has explicitly attempted to identify valid substitutions to create novel, feasible recipes [1, 7]. In general, this is due to the complex task required to appropriately identify "valid" ingredient substitutions [16]. It is difficult to explicitly identify the substitution relationships between recipes and the validity of a substitution will ultimately be subject to the user who is judging the substitution. At present, valid substitutions have been identified by crawling the comments on a recipe database (Allrecipes.com) and mining the text for indicators of substitution information [16] and quantifying the co-occurrence of ingredient pairs using the *PMI* [6]. Achananuparp and Weber (2016) quantified the substitutability of ingredients based on co-occurrence of an ingredient and the context (breakfast, lunch and dinner) is found in using the modified PMI_{sig} termed PMI_s [8].

In a similar fashion, recent work has also explored the ability of an autonomous system to *create* new recipes from scratch. Cromwell et al. (2015) used a data-driven approach to train a decision stump model to recognize the most popular recipe between two recipes. Recipes were characterized using the ingredient complement network of [16] and the flavor complement network of [3] for input into the decision stump model.

Another approach of making food healthier in general is recommending the user a recipe between "want to eat" and "should eat" as Elsweiler et al. (2015) stated. It is important to not overwhelm people with completely different recipes but giving them slightly healthier versions of their desired recipes. In this way the new recipe will more likely get positive feedback and will actually be an alternative[10].

The basic idea of our method to only substitute differing ingredients of similar recipes is supported by the work of Akkoyunlu et al. (2017). They drew to the conclusion that two items are highly substitutable if they are consumed in similar contexts and less substitutable if they are consumed together [4].

3 METHODOLOGY

The idea behind our approach is to compare similar recipes, take the same ingredients as essential to not change the essence of the recipe and look for healthier substitutions for the differing ingredients. In the following section (3.1) the method will be described and explained step by step.

3.1 Approach

3.1.1 Data acquisition. The first step was to gather data from all recipes in the database. Therefore, we created a recipe corpus consisting of all recipes and their parsed ingredient names and stored them in lists. In the beginning, we started by using all of the database's ingredients, but later it turned out that better clusters are generated if only the top 1000 ingredients are taken into consideration. After retrieving the top 1000 ingredients, we removed stop words with the following stop word list: "grilled", "canned", "ground", "raw", "baked", "cooked", "steamed", "crumb", "sliced", "chopped", "diced", "fresh", "whole", "dried", "roasted", "processed", "red", "green", "yellow", "lb", "oz", "%", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0". By using these stop words, we removed most of the prefixes of all ingredients. After stop word and duplicates removal 876 distinct ingredients remained. Next, we improved the recipe corpus by iterating over every recipe, removing the stop words from every ingredient and checking, if all the recipes ingredients occur in the top 1000 ingredients. If an ingredient didn't occur in the top list, the ingredient was split into sub-strings. All these sub-strings were then again compared against the top 1000. If a match was found, the original ingredient was replaced by the sub-string and if no match was found, the ingredient was removed completely. In the final corpus, around 80.000 of the approximately 2.2 million ingredients of all recipes were dismissed.

3.1.2 Health score. Secondly, we needed a score to divide the ingredients into healthy and unhealthy. Because of the limited amount of information and the difficulty of calculating a score for single ingredients, we used our own modified version of the FSA scoring scheme. In our take, we only used the energy, sodium, sugar and fatty acid levels to calculate a score. While the standard FSA scoring scheme also uses fruit, vegetable and nut content of a recipe for calculating the final score, we decided to omit these contents out of two reasons: First, no information about these contents was given in the database that we used to retrieve the nutritional information of the ingredients. Second, no exact information about the healthiness of an ingredient is needed for our approach because it suffices that our calculated score indicates which ingredients are healthier than others. Manual inspection of the results showed that our scores fitted well. We got these nutritional values by using the API of the "United States Department of Agriculture", which is a department of the US government dedicated to agriculture and food related topics. They are hosting a collection of databases called "USDA Food Composition Databases", which contain nutritional information about ingredients of certain manufacturers and about unmanufactured ingredients.

3.1.3 Similarity of recipes. In order to find possible substitutions for all ingredients, we made the assumption that those ingredients which differ in similar recipes are possible substitutions for each other. To be able to compare recipes with similar ingredients, we decided to build a vector space model containing all recipes by calculating a simplified tf-idf value for every ingredient in every recipe. The tf-idf measure is described by the following term [15]:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

where t is a term, or in our case an ingredient and d is a document (recipe). $tf(t,d)$ refers to the frequency of a term in a document, but in the case of recipes, every ingredient of a recipe would have a term-frequency of 1 because it is contained only once per recipe. A possible improvement of the term-frequency for ingredients would be to incorporate the mass of an ingredient into the score. Due to the fact that many different mass schemes like lb, cups etc. are used to describe the mass of an ingredient, we decided to skip the mass and use boolean term-frequency instead. Inverse document frequency (idf) is used to describe how often a term (ingredient) appears in all documents (recipes). Because our term-frequency is always 1, our final tf-idf measure is described by[15]:

$$\text{tf-idf}_{t,d} = \text{idf}_t = log\frac{N}{\text{df}_t}$$

where N is the number of recipes in the collection and df_t is the number of recipes the ingredient appears in.

Using the tf-idf measure we generated a vector-space model from the collection of recipes. A vector-space model is a N-dimensional vector space, where N corresponds to the number of distinct ingredients of all recipes. Each recipe is then represented as a vector in the N-dimensional space, depending on the ingredients of the recipe. To achieve our goal of finding recipes similar to a given one we used the cosine-similarity to find those recipes, whose vectors are closest to the vector of the given recipe. The cosine-similarity is the cosine of the angle between two recipes and is described by [15]

$$sim(r1, r2) = \frac{\vec{V}(r1) \cdot \vec{V}(r2)}{|\vec{V}(r1)||\vec{V}(r2)|}$$

where $r1$ and $r2$ are two different recipes. Using this method, we compared every recipe of the recipe corpus to its 10 most similar neighbours and created lists containing only ingredients that differed between those similar recipes.

3.1.4 Generating clusters. With the collected data we now tried to generate clusters for every ingredient containing only the ingredients, that were mostly used as an alternative for the original ingredient in the similar recipes. To achieve this, we tested four

different clustering algorithms, LDA, LSI, Word2Vec and FastText. After testing and tuning all algorithms the latter two showed significantly better results than the other algorithms, so we decided to use the combined results of FastText and Word2Vec for our ingredient substitution. A comparison of the results of all four algorithms is described in section 3.2.5. We also decided to choose only the three most similar results from each cluster as possible substitutions for an ingredient in order to get only ingredients that are often replaced by each other.

3.1.5 Ingredient substitution. For the final ingredient substitution our method needs the previously computed ingredient clusters, the list of all ingredients, their computed FSA Score and the vector space model. Given a recipe all the recipe's ingredients are requested from the database by selecting their parsed ingredient names. Afterwards, stop words are removed from the ingredients. Then, to identify essential ingredients for the recipe, the recipe's ingredients are compared with the ingredients of the original recipe's 10 most similar recipes. These similar recipes are found as well by calculating the cosine similarity described in section 3.1.3. After testing various thresholds, we identified those ingredients that appear in more than 80% as essential for a recipe. For all ingredients below this threshold we assumed that these ingredients can be substituted. For each of these substitutable ingredients the clusters of FastText and Word2Vec are retrieved and combined. This has the advantage that if no cluster is retrieved by one of them, the cluster of the remaining one can still be used for substitution. We also removed ingredients containing the words "salt" or "pepper", because these two ingredients appear in almost every recipe and therefore occur in many clusters. Because they also have a good health score, they are often used for substitution if they are not removed from the clusters beforehand. Then, the health scores of all ingredients in these clusters are retrieved and compared to the health score of the original ingredient. If an ingredient with a lower health score is found, the original ingredient is substituted with this healthier ingredient. After substituting all ingredients below the threshold our method returns the modified recipe containing both its essential and substituted ingredients.

3.2 Algorithms

In this section, we briefly want to explain the four algorithms LSI, LDA, Word2Vec and FastText, which were used to cluster the ingredients. For the actual computations, we used Gensim (Meaning: "generate similar"), a free Python library. The library is specifically designed to handle large text collections and to realize unsupervised semantic modelling from plain text [13].

3.2.1 LSI. Latent Semantic Indexing (LSI), also known as Latent Semantic Analysis (LSA), uses singular-value decomposition [9]. It takes a large matrix of term-document association data and constructs a "semantic" space wherein terms and documents that are closely associated are placed near one another. In our approach the semantic spaces are the clusters we want to build.

3.2.2 LDA. Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus [5]. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The topics describe the clusters of suitable replacements for an ingredient.

Table 1: Results for cream

LSI	LDA	Word2Vec	FastText
unsalted butter	butter	whipping cream	whipping cream
worcestershire sauce	milk	heavy whipping cream	heavy whipping cream
kosher salt	quick cooking oat	evaporated milk	cream cheese
thyme	egg	cream cheese	evaporated milk

Table 2: Results for ketchup

LSI	LDA	Word2Vec	FastText
pork	salsa	hamburger bun	barbecue sauce
egg yolk	onion	liquid smoke	bbq sauce
spinach	tomato	bbq sauce	steak sauce
extra virgin olive oil	pepper	catsup	worcestershire sauce

Table 3: Results for almond

LSI	LDA	Word2Vec	FastText
banana	ginger root	slivered almond	slivered almond
apple	tea	walnut	walnut
shrimp	water	sunflower seed	peanut butter chip
raisin	egg	blueberry	blueberry

3.2.3 Word2Vec. Word2Vec takes a text corpus as input and produces the word vectors as output [12]. It first constructs a vocabulary from the training text data and then learns vector representation of words. Every ingredient of each recipe relates to a word or rather a vector in the model.

3.2.4 FastText. FastText is a library for efficient learning of representations and sentence classification. It transforms text into continuous vectors that can later be used on any language related task [11]. The comparison of the vectors, in this case our ingredients, corresponds to the best substitution choice in our method.

3.2.5 Comparison. To test our methods, we chose to look at the top substitution suggestions for three random ingredients cream, ketchup and almond. The algorithms provide very different results that can be seen in Table 1,2 and 3. These results yield that Word2Vec and FastText performed way better than LSI and LDA. A closer investigation of LSI and LDA showed that even after trying out

Table 4: Original vs. substituted recipes

almond butter		mommas pasta and shrimp salad		easy beef goulash		chicken and pumpkin goulash	
original	substituted	original	substituted	original	substituted	original	substituted
olive oil	extra virgin olive oil	elbow macaroni	elbow macaroni	water	water	ground cinnamon	vanilla extract
almond	almond	italian salad dressing	italian salad dressing	tomato paste	tomate puree	olive oil	extra virgin olive oil
honey	honey	onion	onion	dry onion soup mix	condensed cream of celery soup	water	water
		tomato	roma tomato	beef stew meat	beef stew meat	cornstarch	cherry pie filling
		shrimp	scallop			ground ginger	ginger root
		mayonnaise	mayonnaise			brown sugar	molass
		celery	pea			ground cumin	ground cumin
		cucumber	english cucumber			onion	onion
						salt	salt
						ground coriander	ground coriander
						tomato	roma tomato
						garbanzo bean	great northern bean
						chicken breast	shredded chicken
						pumpkin	pumpkin

different parameters for these two algorithms, their results were still worse than the results of Word2Vec and FastText, so we decided to concentrate on the latter two algorithms. Due to the fact that FastText is built on Word2Vec, it is not surprising that these two achieve similar results. In our opinion, both algorithms produce good substitutions for the ingredients presented in tables 1, 2 and 3. The advantage of FastText over Word2Vec is the fact that FastText uses n-grams of the terms. This allows FastText to find a cluster to a given ingredient by using its n-grams, which almost always leads to a valid result, while Word2Vec returns an error if the ingredient is not found in any of the clusters. After further testing of both algorithms we found that Word2Vec produces better results than FastText if a high minimum count of terms was used.

4 RESULTS AND DISCUSSION

To test our methodology described in section 3.1.5, we decided to perform ingredient substitutions on the following four recipes:

(1) almond-butter
(2) mommas-pasta-and-shrimp-salad
(3) easy-beef-goulash
(4) chicken-and-pumpkin-goulash

In the following section, we will describe the best and worst substitutions made by our system. An overview of all substitutions can be seen in table 4.

The substituted "almond butter" recipe doesn't differ much from the original one. Only olive oil is substituted by extra virgin olive oil. In our opinion, honey is also an ingredient that could be substituted, but it appears in 9 of 10 similar recipes. The reason for this might be that we didn't incorporate the mass into our tf-idf

formula. Therefore, the recipe vectors in the vector space model are only represented by terms that either appear or don't appear. In conclusion, the most similar recipes to a given recipe are the ones that contain nearly the same ingredients. Because the "almond butter" recipe only consists of three different ingredients, the most similar recipes almost all contain almonds and honey.

In the "mommas pasta and shrimp salad" recipe the prominent substitutions are scallops instead of shrimps and peas instead of celery. These substitutions are actually fitting because they both belong in the same category, meaning fish is replaced by fish and vegetable is replaced by vegetable. Although the essence of the shrimp salad might be changed by using scallops, this substitution is in our opinion valid because the taste will mostly stay the same because both are sea foods. For the ingredient mayonnaise we received healthy alternatives at first but as described in 3.1.1 we had to only incorporate the top 1000 ingredients to achieve an overall better performance. With regards to this there weren't terms like light mayonnaise present in our recipe corpus, which could serve as a substitution. Furthermore, reoccurring terms in an alternative or wrong spelling like "mayo" or "mayonaise" made it hard to find suitable replacements for certain ingredients. The remaining ingredients either stay the same or are only changed very little, e.g. roma tomato for tomato.

Looking at the recipe "easy beef goulash" one can observe that tomato paste was substituted for a slightly healthier variant of a processed tomato product. Furthermore, dry onion soup mix was replaced by condensed cream of celery soup, which has however a much better health score than the original ingredient. In terms of taste there is not that big of a difference between the two ingredients because every substituted ingredient stays in the same class.

The substituted ingredients in the "chicken and pumpkin goulash" recipe behave similar to the preceding substitutions. The large part doesn't change at all or only slightly but there still are some total differences. Vanilla extract instead of ground cinnamon and molass instead of brown sugar can be substituted in terms of taste but in terms of a better overall health score there isn't much need for a substitution. The most conspicuous substitution is cherry pie filling instead of cornstarch, which isn't fitting at all regarding to taste.

All in all, the substituted ingredients are mostly from the same food category and therefore don't change the taste and essence of the recipes. However, some essential ingredients don't get recognized as essential, because mass isn't included in the vector space. Therefore, some ingredients get substituted unnecessarily. On the other hand, substituting single ingredients often leads to suitable substitutions. Additionally, many clusters contain unsuitable ingredients because the data in the database is very unstructured, e.g. the ingredient mayonnaise occurs with many different writings like "mayo", "mayonaise" etc. Furthermore, the quality of the cluster is lowered by the fact that our assumption of finding suitable substitutions in similar recipes does not always apply because often similar recipes only contain additions or deletions of ingredients.

5 CONCLUSIONS

The approach described in this paper shows that clustering of ingredients of similar recipes can lead to suitable substitutions. Looking at differences between similar recipes can yield information about which ingredients are often replaced by one another.

Future work could improve the approach described herein by finding a way to incorporate ingredient mass into the vector space model and therefore improve the comparison of similar recipes. Furthermore, the process of clustering ingredients can be improved by incorporating substitutions suggested in recipe comments into the clusters as described by Teng[16]. Also, a better preprocessing of the raw data in the database could improve the performance of this system by reducing multiple occurrences of ingredients and spelling mistakes.

REFERENCES

[1] Palakorn Achananuparp and Ingmar Weber. 2016. Extracting food substitutes from food diary via distributional similarity. *arXiv preprint arXiv:1607.08807* (2016).

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.

[3] Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. 2011. Flavor network and the principles of food pairing. *Scientific reports* 1 (2011), 196.

[4] Sema Akkoyunlu, Cristina Manfredotti, Antoine Cornuéjols, Nicolas Darcel, and Fabien Delaere. 2017. Investigating substitutability of food items in consumption data. In *Second International Workshop on Health Recommender Systems*. 27.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[6] Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics* 16, 1 (1990), 22–29.

[7] Erol Cromwell, Jonah Galeota-Sprung, and Raghuram Ramanujan. 2015. Computational creativity in the culinary arts. In *FLAIRS Conference*. 38–42.

[8] Om P Damani and Shweta Ghonge. 2013. Appropriately incorporating statistical significance in PMI. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 163–169.

[9] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.

[10] David Elsweiler, Morgan Harvey, Bernd Ludwig, and Alan Said. 2015. Bringing the "healthy" into Food Recommenders. In *DMRS*.

[11] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* (2016).

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[13] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.

[14] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.

[15] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press.

[16] Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. 2012. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 298–307.

[17] Christoph Trattner and David Elsweiler. 2017. Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 489–498.