Automatic ingredient replacement in
digital recipes: combining machine
learning with expert knowledge

_Swaan Dekkers_

swaan.dekkers@student.uva.nl
10437495

June 23, 2017

Master Information Studies
Data Science

Faculty of Science
University of Amsterdam

Internally supervised by
**Dr. Maarten Marx**
University of Amsterdam

Externally supervised by
**Dr. Vladimir Nedović**
Flavourspace

Universiteit van Amsterdam

Flavourspace
Abstract

Over the past years, people have become increasingly aware of their eating habits and the corresponding effects on mental and physical health. This thesis argues that eating habits could be further improved by user empowerment, which could be achieved by the possibility of people adjusting recipes to their needs and preferences. One of the most prominent examples of recipe adjustment is ingredient substitution when the proposed ingredient is unavailable or contains allergen substances. This thesis explores how to identify suitable ingredient substitutes in a broad range of recipes. The following two methods have been implemented and evaluated: the traditional use of domain expert knowledge and data-driven methods of using word embeddings. The main assumption of the word embedding method was inspired by the distributional hypothesis in linguistics, namely that words that are often used in similar contexts tend to convey similar meanings. In the context of cooking, this hypothesis suggests that ingredients consumed in similar recipes are likely to be suitable substitutes of each other. The two methods were extensively evaluated with a user test, which resulted in an adapted and therefore better performance of the expert knowledge model.
1 Introduction

1.1 Motivation

Over the past years, there has been an increasing awareness of the effects of eating habits on mental and physical health. People, but also companies and the government, are more and more concerned with consuming and promoting 'healthy' eating standards. Improvement of eating habits could be further encouraged by user empowerment. This thesis argues that user empowerment could be achieved by providing people the possibility to easily adjust recipes to their needs and preferences.

Studies show that home cooking is a complex process. Recipe choice and food consumption is influenced by different factors such as time, food preferences, allergies, eating cultures, cooking equipment and dependency upon seasonal products. A way to simplify cooking and enhance user empowerment is to provide different substitutes for ingredients that are for example unavailable or contain allergen substances. However, how can we offer the right substitute for certain ingredients in such a complex process?

Flavourspace (FS), an Amsterdam based company, is currently working on this issue. This company aims to improve cooking practices using artificial intelligence. In collaboration with Flavourspace, this research has resulted in an ingredient substitution system to enhance their recipe search engine. This engine assists users in finding a fitting recipe, which could be further improved by making the recipes adaptable, so customers can easily change ingredients to their own tastes. If a user for example dislikes coriander, the system can offer a substitute ingredient such as parsley. Other applications of this could be if the user wants to reduce calories the calories of a certain dish or prefers a vegetarian or vegan version.

This research has used two methods for obtaining ingredient substitutes, namely using expert knowledge (EK) from a domain expert and the data-driven method of word embeddings (WE). The method of expert knowledge is based on the implicit and explicit substitution rules derived from the expert knowledge source Cook’s Thesaurus, which will be explained later on. Using word embedding to extract substitution rules is inspired by the distributional hypothesis in linguistics: words that occur in the similar contexts tend to convey similar meanings. When applying this hypothesis on cooking practices, ingredients used in similar recipes are likely to be a suitable substitute of each other. The two methods are evaluated with a user test and will be combined in an ingredient substitution system.
1.2 Research Question

The design of the ingredient substitution system is based on the following question:

- *How can we develop a system that effectively determines ingredient substitutes for recipes based on a combination of word embeddings and expert knowledge?*

To answer this question, this research has proposed the following sub-questions:

1. *Can the expert data also support implicit substitution rules, next to the explicit substitution rules defined by the expert?*

2. *Which similarity formula can best be applied when calculating the distance between two vectors in the word embedding model, intending to rank the vectors to find ingredient substitutes?*

3. *Which of the following systems results in a better performance: the expert knowledge system or the word embedding system?*

1.3 Overview

The following section 2 provides an extensive literature review on earlier studies on this issue. Based on this theoretical background, several experiments are designed in which the two ingredient substitution systems are evaluated. Section 3 is an in-depth explanation of the methodology of the substitution system and the designed experiments. The results of these experiments are discussed in Section 4. Finally, Section 5 presents a conclusion based on the outcome of these experiments.

2 Related work

2.1 Ingredient substitution in recipes

Earlier studies have explored different methods for obtaining ingredient substitutes in recipes. One method used to extract substitute ingredients from recipes is the use of network analysis [37]. The substitute network is derived from user generated suggestions for modifications. This substitution network is then further used to uncover novel recommendation algorithms suitable for recipe recommendations. However, it has not been evaluated on itself.
Another approach to extract substitutes is with the use of a statistical model [5]. This method shows that topic densities of recipe document’s mixtures can be used for ranking candidate substitutes from expert generated rules. The model uses a topic space in which recipes are placed and rankings are generated by considering the closest topics according to the Kullback-Leibler divergence.

Thirdly, Shidochi et al. [35] proposed an algorithm to extract substitution ingredients from recipes. Their method was based on matching the cooking practices with the corresponding ingredient. According to their hypothesis, substitutable ingredients are subjected to the same processing methods.

Furthermore, other research explores the effectiveness of using ontologies in improving ingredient substitution [9, 2, 7, 15, 41, 2]. The first preliminary results of this research were positive, however, this method would need a more extensive evaluation. Additionally, it is a very costly method to implement.

To the best of our knowledge, this research is the first to extensively evaluate the used methods and conduct an extensive user test. Also, using word embedding, the substitution relationship is directly identified from the data instead of relying on external knowledge sources [5, 37, 9, 2, 7, 35]. This is a very cost-effective approach due to its reliance on data rather than human expertise.

2.2 Expert knowledge

Expert knowledge is used in a wide range of scientific domains like ecological research [18, 40], hydrology [8], and medic [1]. It provides a valuable source of information, often because of the complexity of problems or lack of data. Expert knowledge is defined as “the substantive information on a particular topic that is not widely known by others” [25].

With the help of the knowledge-representation theory, expert knowledge can be transformed into a representation such that is understandable for computer systems. In knowledge representation, different types of knowledge are defined, such as objects, events, procedures, relations, mental states and meta knowledge [33]. These types provide different sorts of information. A specific type of object information is category and subcategory information. From this categorical information, an taxonomy can be constructed in which all categories and subcategories are structured. By identifying specific properties of the objects and relations between the objects, the researcher can construct an ontology, resulting in the substitution system as mentioned
Related work

In regard to cooking practices, expert knowledge is most often used as source for ingredient substitute information. In recipes provided by recipe books or websites, authors often provide complementary or possible replacements of the proposed ingredients. The quality of such an substitute depends on the subjective factor of taste. Until now, there has been no prior research on the satisfaction of users with the offered substitutes in recipes.

2.3 Word embedding

Vector based word representations have a long tradition of usage in the Natural Language Processing (NLP) research community. The vectors are used to compute similarity between terms, but can also be used as representational basis for downstream NLP tasks like clustering, POS tagging, classification and sentiment analysis. Recently, there has been an increasing usage of word embeddings as an input in machine learning tasks [29, 20, 3].

Word embeddings are based on the idea that contextual information constitutes a representation of linguistic items. This idea is derived from the distributional linguistic hypothesis: words occurring in similar contexts tend to have similar meanings [14]. This hypothesis found its origins in several published works the 1950s by Zelling Harris, John Firth, and Ludwig Wittgenstein [14, 12, 39] and has been previously explored by Flavourspace using factor analysis and topic modelling [5].

A popular training technique for word embeddings, word2vec [30], consists of using a 2-layer neural network that is trained to identify a certain word in relation to its context. A neural network can’t be fed with words strings, so the word should be transformed so that the network can make sense of it. Every unique word in the training corpus is represented as a one-hot vector. This vector will have the length of the number of unique words in the corpus. It will have a “1” in the position of the corresponding word, and 0s in all other positions. The output of the network is a single vector with the length of the number of unique words in the training corpus. If two different words are used in similar contexts - that is, when the same words are likely to appear in the same setting - the model will output similar results for these words.

Word2vec has been widely used because it results high accuracy on a wide range of tasks [32] and is robust across a wide range of semantic tasks [34]. The aim of this thesis is finding ingredient substitutes with word2vec. Although this exact task has never been examined before, there have been experiments with finding lexical substitutions with word embeddings which
led to positive results \([27, 26, 4, 17]\).

A very fascinating property of word2vec is the ability to capture linguistic relationships of both semantic (king to man is like queen to woman) and syntactic nature (ran to run is like laughed to laugh) \([29, 22]\). Analogical reasoning is therefore a promising line of research, since it can be used for many tasks like word sense disambiguation \([11]\), morphological analysis \([19]\), semantic search \([6]\), and even for broad-range detection of both morphological and semantic features \([21]\). However, it remains unclear to what extent word embedding models are able to capture relations between words. Research has shown that derivational and lexicographic relations such as synonymy remain a major challenge \([16, 13]\). This thesis examines whether analogical reasoning improves the task of finding ingredient substitutes.

3 Methodology

Figure 2 illustrates the flowchart of the proposed method of this research. The following Section 3.1 will first explain the data that has been used. The logic combiner, where all input comes together, ranks different substitutions and eventually outputs an ordered list of substitutes. The first input in the logic combiner consists of the the substitution rules derived from the domain expert knowledge. The methodology of this data will be described in Section 3.2. This section also describes the methodology of the conducted experiments to find implicit substitution rules, next to the explicitly defined rules by the expert, addressing first sub-question proposed in this research.

The second input consists of the word embedding substitution rules, derived from WE models and explained in Section 3.3. An experiment has been conducted to determine which combination of model and a similarity measure performs best, addressing sub-question 2. A more detailed explanation of the logic combiner is given in Section 3.4. Finally, a user test has been performed to decide how to rank potential substitution candidates within the logic combiner. The outcome of the user test will answer the third sub-question of this research. An explanation of the user test evaluation method is given in Section 3.5.

3.1 Data

For this research, Cooks Thesaurus (CT) will function as the source of expert knowledge. CT is a cooking encyclopedia that contains information about
2373 ingredients. The data has been crawled, parsed and tokenized for this thesis.

The CT proposes possible substitutes for certain ingredients, often accompanied with a note on how this substitute should be prepared and would influence the end result of the dish. It also provides synonyms for the ingredient. An example CT substitute can be found in Figure 3. As explained in Section 2.2, there are different types of knowledge of which various can be found in the CT. The first one is object knowledge, which consists of both specific and general object knowledge. Specific object knowledge are the notes given on a certain ingredient. General object knowledge are the category and subcategory an ingredients belongs to. For example, *chili bean* is of the subcategory *dry beans*. *Dry beans* is again a category of *legumes & nuts*. In this manner a taxonomy, a system of categories and subcategories, is constructed. Next to the object information, the main reason to use CT as source of information is the relational information between the objects it provides. Object relations that can be derived from the expert data are substitute relationships and synonym relationships. For example; SubstituteOf(*pinto bean, chili bean*) and SynonymOf(*pink bean, chili bean*).
3 Methodology

The second data source used is the collection of 112,000 recipes from Flavourspace, used to train the word embedding model. The collection is obtained by scraping and parsing recipes from different websites (i.e., All-recipes, Epicurious, MyRecipes, Jamie Oliver, and others) and are represented as lists of ingredients. For example, a Caesar salad would be represented as: *tuna, vegetable oil, kosher salt, black pepper, tomato, egg, ciabatta bread, mayonnaise, olive tapenade*.

3.2 Expert knowledge based substitutions

The ingredients and the corresponding substitute relationships form an ingredient network, with the ingredients as nodes and the substitute relationships as edges. The ingredient network is based on ingredient relations derived from the expert knowledge and consists of directed and undirected edges. If ingredient A is a substitute for ingredient B, there will be an edge going from A to B. Some ingredient pairs are linked in both ways, thus directed edges are going in both ways. Therefore, these could also be interpreted as an undirected edge. For other pairs, the relationship is only given in one direction. The number of edges going into a node is known as the indegree of the corresponding node and the number of edges coming out of a node is known as the outdegree.

3.2.1 Explicit substitution rules

The expert provides substitutes for certain ingredients: the explicit substitution rules. CT provided substitution rules for 2373 unique ingredients and 14,536 explicit substitution rules. Because a domain expert has written the explicit substitutes relations, they are considered to be very reliable. The explicit substitution rules that...
can be derived from the example given by Figure 3, can be seen in Figure 4.

<table>
<thead>
<tr>
<th>Categories</th>
<th>ID</th>
<th>Ingredient</th>
<th>Substitute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>legumes &amp; nuts, dry beans</td>
<td>175</td>
<td>chili bean</td>
<td>pinto bean</td>
<td>Explicit expert</td>
</tr>
<tr>
<td>legumes &amp; nuts, dry beans</td>
<td>176</td>
<td>chili bean</td>
<td>rattlesnake bean</td>
<td>Explicit expert</td>
</tr>
<tr>
<td>legumes &amp; nuts, dry beans</td>
<td>177</td>
<td>chili bean</td>
<td>red kidney bean</td>
<td>Explicit expert</td>
</tr>
<tr>
<td>legumes &amp; nuts, dry beans</td>
<td>178</td>
<td>pink bean</td>
<td>pinto bean</td>
<td>Explicit expert</td>
</tr>
<tr>
<td>legumes &amp; nuts, dry beans</td>
<td>179</td>
<td>pink bean</td>
<td>rattlesnake bean</td>
<td>Explicit expert</td>
</tr>
<tr>
<td>legumes &amp; nuts, dry beans</td>
<td>180</td>
<td>pink bean</td>
<td>red kidney bean</td>
<td>Explicit expert</td>
</tr>
</tbody>
</table>

Fig. 4: Substitution rules derived from the explicit substitution rules of *chili bean*, given in Figure 3.

The first column contains the categories of the ingredient, starting with the main category and ending with the subcategory. The second column contains the ID of the ingredient. As mentioned before, an ingredient can have multiple names. To prevent offering synonyms to the user, the first ingredient mentioned by the expert is taken as ID, because it is the most common name used. The third column contains the ingredient name of all synonym names given. In the fourth column the substitute ingredients are listed. The last column contains the type of ingredient, so for example the explicit substitution rule, implicit substitution rule or word embedding substitution rule. All ingredients are lemmatized to simplify future comparison. So, if the expert would have given the substitute pinto beans, lemmatization would have turned it into pinto bean.

3.2.2 Implicit substitution rules

Other substitution rules can be derived by implicit substitution rules. There has been no proof for these implicit substitution rules in cooking literature. Various experiments were conducted as part of the research in order to determine whether or not to apply these implicit substitution rules. These experiments answer our first sub-question.

The first implicit substitution hypothesis is based on symmetry. The application of this rule would mean that all nodes who have either a directed edge from A to B, from B to A or in both ways, would get an undirected edge. This would turn the graph in an undirected graph. To determine whether or not to apply this rule, an experiment is designed to validate this on the CT data. For each node v, the amount of edges of a substitute in both
direction, \( \text{deg}^{-}(v) \), is divided by the indegree, \( \text{deg}^{+}(v) \). When referring to the symmetry of the whole ingredient graph \( G \), the average symmetry of all nodes \( N \) in the network is calculated using the following formula:

\[
S(G) = \frac{1}{N} \sum_{v=0}^{N} \frac{\text{deg}^{-}(v)}{\text{deg}^{+}(v)}
\]

The second implicit substitution hypothesis is based on triadic closure \[^{10} \]. The principle of triadic closure applied on this ingredient network entails that if two ingredients in the ingredient network have a substitute relation to a certain ingredient in common, there is an increased likelihood that they also have an substitute relation. A method to measure the presence of this triadic closure is the clustering coefficient \[^{31, 38} \]. The clustering coefficient of node \( A \) is defined as the probability that two randomly selected nodes that have an edge with \( A \) will also have an edge with each other. The clustering coefficient of a node ranges from 0 (when none of the nodes substitutes are substitutes from each other) to 1 (when all of the nodes substitutes are substitutes from each other). For note \( v \), let \( \lambda_{G}(v) \) be the number of triangles on \( v \in G \) for undirected graph \( G \). That is, \( \lambda_{G}(v) \) is the number of subgraphs of \( G \) with 3 edges and 3 vertices, one of which is \( v \). Let \( \tau_{G}(v) \) be the number of triples on \( v \in G \). That is, \( \tau_{G}(v) \) is the number of subgraphs with 2 edges and 3 vertices, one of which is \( v \) and such that \( v \) is incident to both edges. Then, the clustering coefficient is defined as:

\[
C_{i} = \frac{\lambda_{G}(v)}{\tau_{G}(v)}.
\]

When referring to the clustering coefficient of the whole ingredient graph \( G \), the average clustering coefficient of all nodes in the network is calculated as:

\[
C(G) = \frac{1}{N} \sum_{v=0}^{N} C(v)
\]

When applying the rule of triadic closure, all the missing edges would be added to the database. To determine if this rule should be applied, one calculates the average clustering coefficient over all of the nodes in the network. This calculation is applied on the directed graph based on the explicit substitution rules and on the undirected graph constructed after applying rule one.

The third rule is based on the taxonomy that can be derived from the Cook’s Thesaurus. The taxonomy of CT is the classification and subclassification of ingredients in an ordered system. Distance through the taxonomy denotes a form of gastronomic similarity. The assumption holds that
Methodology

Ingredients in the same category, e.g. types of nuts or types of greens, have a higher probability of being interchangeable [9, 2, 7, 15, 41, 2]. To determine the strength of this assumption in the case of CT, the taxonomic strength is calculated. To calculate the taxonomic strength of node A, the incoming edges from nodes with the similar subcategory as node A, $\text{deg}^+_{\text{similar}}(v)$, is divided by the total indegree, $\text{deg}^+(v)$.

The substitutions of that ingredient with the same category as the ingredient are divided by the total amount of substitutions of that ingredient. When referring to the taxonomic strength of the whole ingredient graph G, the average taxonomic strength of all nodes N is calculated as followed:

$$C(G) = \frac{1}{N} \sum_{i=0}^{N} \frac{\text{deg}^+_{\text{similar}}(v)}{\text{deg}^+(v)}$$

3.3 Word embedding based substitutions

Two models are used in order to obtain ingredients through word embedding. The first word embedding model is based on the 112 000 ingredient lists of recipes mentioned before and is referred to as the Recipe model.

The second word2vec model is trained by Google and is freely available [30, 29, 28]. The dictionary is based on a 100 million word corpus from Google News. Using this dictionary may seem like an surprising choice since this research only needs very specific information on ingredients. However, Google News also offers a lot of recipes. It is not possible to look into the exact corpus the Google model uses, but because the corpus is so extremely extensive, we assume that it will contain the information needed for this task.

To find ingredients that are substitutes of each other, the two above models are compared. For calculating similarities between vectors, three possible similarity measures are implemented, explained in Section 3.3.1. The performances of the combination of the two models and three similarity measures is evaluated with three evaluation metrics, see Section 3.3.2.

3.3.1 Similarity measures

To find substitutes for ingredient A, first the vector location of A is obtained from the word embedding model. Possible substitutes are found by finding vectors that are similar to vector A. For calculating similarity, different formulas are used. Finally, the output is a list of vectors, ranked from high to low similarity.
The first similarity measure is cosine similarity. For ingredient A, possible substitutes B are calculated with the following cosine similarity formula:

$$cos(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The other two proposed similarity measures are based on the linguistic regularities, explained in Section 2.3. The idea is that words which have a specific relation, will have a similar offset in the vector space. This is still a relatively new idea. This thesis further examines whether this holds for searching ingredient substitutes.

To calculate relational similarity, a word pair is needed of which the offset resembles the ingredient substitute relationship. The method of finding this word pair is to look at the cosine similarity of all the ingredient substitution pairs proposed by the domain expert. Finally, the top hundred pairs with the highest cosine similarity are selected.

Given the analogy 'A is to B like C is to D', the following measure is proposed for finding a possible substitute D for ingredient C, given the predefined substitution pair (A, B). The first formula, CosAdd, proposed by Mikolov et al. [29], calculates the similarity and ranks according to the following formula:

$$CosAdd(A : B, C : D) = \cos(B - A + C, D)$$

CosAdd measure can be deconstructed into the summation of three cosine similarities, where in practice one of the three terms often dominates the sum. To overcome this bias, Levy and Goldberg [23] proposed the CosMul formula:

$$CosMul(A : B, C : D) = \frac{\cos(B, D) \cos(C, D)}{\cos(A, D) + \epsilon}$$

To determine whether the Google or Recipe model should be used and which similarity method then should be applied, three quantitative performance metrics are evaluated. For CosAdd and CosMul, the metrics are calculated for hundred selected word pairs.

The outcome of this evaluation shows if using linguistic regularities is an improvement compared to using the cosine similarity. Additionally, it also shows which word pair performs best and which of the two models performs best. The model that performs best will be used in the user test and implemented in the logic combiner.
### 3.3.2 Quantitative performance evaluation

As metric for evaluating the different similarity methods, recall, precision and Mean Average Precision (MAP) are used. The substitution rules derived from the expert knowledge are used as the ground truth. For this task, precision is the fraction of retrieved substitute ingredients that are suitable replacements. Precision could be defined with the following formula:

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Recall is the fraction of the relevant ingredients that are successfully retrieved, calculated with the following formula:

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Precision and recall are single-value metrics based on the whole list of ingredients returned by the system. The system implemented in this thesis is a ranked system according to the measure of similarity between the query ingredient vector and the found substitute ingredient vector. For systems that return a ranked sequence, it is desirable to also consider the order in which the returned documents are presented. That is why MAP is implemented.

When calculating the MAP, one first needs to calculate the average precision for each query. This is defined as the mean of the precision at K values computed after each relevant document was retrieved. The final MAP value is defined as the mean of average precision of all queries in the test set.

$$ap@n = \sum_{k=1}^{n} \frac{(P(k) \times rel(k))}{\min(m, n)}$$, where $rel(k) = \begin{cases} 0 & \text{if the item at position } k \text{ is a relevant document} \\ 1 & \text{otherwise} \end{cases}$

Whereas $P(k)$ refers to the precision at cut-off k in the item list, $rel(k)$ is an indicator function equaling 1 if the item at rank k is a relevant document and zero otherwise. This summation is divided by the minimum of all relevant retrieved documents retrieved (m), or the number of predicted ingredients (n). The mean average precision for N ingredients at position n is the normalized average precision over all ingredients:

$$MAP@n = \frac{\sum_{i=1}^{N} ap@n_i}{N}$$
3 Methodology

For this task, precision is more important than recall. The system does not have to return all possible substitute ingredients, as only the top two are used. It is most important that the correct substitutes are offered at the top of the rank, making MAP the most important evaluation metric. For this reason, all metrics are considered up to rank two (@2).

In addition to the ingredient level, evaluation is performed at a level of sub-categories. This evaluation is based on the taxonomy hypothesis described in 3.2.2. For example, instead of the expert ground truth that *chili bean* can be replaced by *rattlesnake bean*, it now could be substituted by all ingredients from the subcategory *dried beans*, because *rattlesnake bean* belongs to the subcategory *dried beans*.

3.4 The logic combiner

The final ingredient substitute is determined using a logic combiner, which ranks methods according to their reliability. The reliability of EK and WE approaches is, in turn, determined by a user test.

3.5 User test evaluation

A user test is conducted to decide if the expert knowledge and word embedding substitutes are good enough to use in a system. And, if so, which method preforms better. This user test is designed with help of professional market researchers and extensive testing.

The user test consist of seventeen questions. The first two questions are focused on assessing the user’s cooking level and cuisine preferences. The other 15 questions, of which Figure 5 is an example, are used to evaluate the two methods.

For each of the four proposed substitutes, the user rates on a scale from 1 to 6 the extent to which they think the ingredient is a suitable substitute for the missing ingredient (1 = *I would never use this as a replacement*, 6 = *This would be a perfect substitute for the missing ingredient*). Two of these substitutes are provided by the EK model, and two by using the WE model. In the example, *mustard* and *roasted garlic* are proposed by the EK model and *cream cheese* and *sour cream* by the WE model.

The supermarket case is used for various reasons. Firstly, it is a context all users know. Providing a clear context to respondents improves the accuracy of the answer, because respondents rely less on their own imagination. Secondly, it excludes a lot of side factors as the proposed ingredients are all available in the supermarket. Beforehand, the user is also asked to
Suppose you prepare a caesar salad, but the mayonnaise is sold out. Below you see four ingredients that are still available. Please indicate the extent to which you think the following ingredients would be a right substitute for mayonnaise.

(1 = I would never use this as a replacement, 6 = This would be a perfect substitute for the missing ingredient)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Don’t know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sour cream</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mustard</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cream cheese</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roasted garlic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5: Example question of the user test for the ingredient mayonnaise, asking the user to rate four possible substitutes of which two are derived from the WE model and two from the EK model.

Assume that all ingredients are equal in price. In this manner, the user test tries to capture the effect of the quality of the substitution ingredient and externalize other influences like money or availability. Furthermore, users can choose the don’t know option if they are unfamiliar with the product. In this manner, the results show if one of the methods proposes a lot of relatively unknown ingredients, indicating a lower usability of the method.

When all the results are collected, the average rates of the answers are calculated. If assuming that users randomly answer questions, the probability mass function for the rating of $A$ will be:

$$P(A_i = x) = \begin{cases} \frac{1}{6}, & x \in \{1, 2, 3, 4, 5, 6\} \end{cases}$$

This results in an expected mean of $\mu_o = E(A_i) = 3.5$ and expected variance of $\sigma_o^2 = Var(A_i) = 2.917$. The hypotheses of this experiment will therefore be $H_0: \mu = 3.5$ and $H_1: \mu \neq 3.5$.

Since the sample of respondents will be large, the results should be approximately standard normal under the null hypothesis if the given rates of the respondents are random. Conducting a two-tailed Z-test, the p-value is calculated. Using $\alpha = 0.05$, a p-value smaller than 0.05 results in the rejection of $H_0$, which would imply that the results are not random and can be meaningfully interpreted.
When calculating the mean, variance and p-value per question, the score of that question for the WE model is based on all the ratings from the user on the two answers obtained from the same model. So in the example question, the ratings of mustard and roasted garlic are combined. The don’t know answer is not taken into account, when determining the N used in the Z-test. When calculating the overall mean, variance and p-value, the ratings from all 15 questions on the answers proposed by the same model are combined.

4 Results

4.1 Expert knowledge experiments

To determine whether to apply the symmetry rule on the expert data, this research used the experiment as explained in Section 3.2.2. The result of this experiment is an average symmetry coefficient of 0.43. This number implies that on average, for almost halve of the substitute relations of an ingredient, also the reverse substitution relationship is given by the expert. When manually checking the cases where the assumption does not hold, the missing substitute is often a less known ingredient that is not offered by CT, probably because the expert suggests more common ingredients. When applying this rule on the data-set, 3831 new substitute relationships can be added. These 3831 new substitute relationships hold the substitute rules for 689 ingredients.

For the second experiment, concerning triadic closure, the average clustering coefficient is calculated. The average clustering coefficient, calculated on the original directed graph based on CT, is 0.216. After applying the first implicit substitution rule, which turns the graph in an undirected network, the average clustering coefficient becomes 0.469, locating 1587 triangles. An example representing a small part of the ingredient network can be found in Figure 6. Alphonso olive node has a clustering coefficient of 1 because the two neighbours are also connected with each other. Kalamanta olive has a cluster coefficient of $\frac{2}{6}$ because of the six possible connections between the neighbours, only two are actually made.

The third experiment is based on the taxonomy of the ingredients. The average taxonomic strength calculated with the formula presented in 3.2.2, is 0.72 when looking at the subcategory. At the category level, this is 0.877. This implies that on average, 72 to 87 percent of the substitutes of an given ingredient, have the same subcategory or category as that ingredient.

Finally, only the first rule concerning symmetry is applied to the data
4.2 Experiments on word embedding models and similarity measures

The results of the MAP, Recall and Precision evaluated on the level of the ingredient and the level of the subcategory can be found in the appendix A.1. As explained before, MAP is considered the most important metric, therefore those results can be found in Table 1.

The highest results for all performance metrics are achieved by the Google model while applying the CosAdd formula. The word pair used in this similarity measure is: cointreau - curacao, both fruit liqueurs. Other word pairs that scored high on the evaluation metrics are: button mushroom - oyster mushroom, black mustard seed - brown mustard seed, white chocolate - milk chocolate. Overall, the metrics do not score very high. When analyzing the metrics at the level of the subcategory, an improvement can be found. In both cases, the highest score is achieved by using the Google trained model and using the CosAdd formula to calculate similarity. Therefore, these settings are used in the user test.
Tab. 1: Results of Mean Average Precision of the Google and Recipe model, using three different similarity measures.

(a) MAP applied on the ingredient level

<table>
<thead>
<tr>
<th></th>
<th>CoSim</th>
<th>CosAdd</th>
<th>CosMul</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>Google 0.047</td>
<td>0.055</td>
<td>0.055</td>
</tr>
<tr>
<td></td>
<td>Recipe 0.031</td>
<td>0.050</td>
<td>0.045</td>
</tr>
</tbody>
</table>

(b) MAP applied on the subcategory level

<table>
<thead>
<tr>
<th></th>
<th>CoSim</th>
<th>CosAdd</th>
<th>CosMul</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>Google 0.208</td>
<td>0.229</td>
<td>0.222</td>
</tr>
<tr>
<td></td>
<td>Recipe 0.174</td>
<td>0.173</td>
<td>0.166</td>
</tr>
</tbody>
</table>

(a) Zoom-in on the representation various type of onions. (b) Zoom-in on various types of rice and the ingredient red lentil.

Fig. 7: A 2-dimensional visualization of Recipe word2vec model, constructed with the t-SNE algorithm. This algorithm is developed for visualizing high dimensional data [24].

4.3 Results user test

In the user test, there is a total of 333 respondents. Each respondent answered 15 questions consisting of 4 proposed substitutes, two obtained from the expert knowledge and two from the word embedding model, resulting in 333*15*4 = 19980 answers.

The result of the combined ratings of the 15 questions can be found in Table 2. When analyzing the overall mean, the EK model performs slightly better than the WE model as the WE model scores slightly below 3.5 and the EK almost an average rate of 3.6.

The variance of the WE model is 0.05 higher then the EK model. For
Tab. 2: Results of the user test when all ratings from the 15 questions are summed per model. * means significant with $\alpha = 0.05$, ** means significant with $\alpha = 0.1$. DK explains the percentage of respondents that did not know the proposed substitute.

<table>
<thead>
<tr>
<th>Expert knowledge</th>
<th>Word embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Var</td>
</tr>
<tr>
<td>3.59</td>
<td>3.08</td>
</tr>
</tbody>
</table>

both models, the variance is high, which indicates that the data points are spread out from the mean and from one another. This could be caused by the subjectivity of a taste judgment, leading to strong differences between the rates of the substitutes. The ingredients proposed by the expert were unknown for 1274 times, which is 18.0% of the cases. For the WE model, this is 19.9%. In this respect, the usability of the expert knowledge model is slightly better.

The results of the user test showed per question can be found in Table 3. For each question, the mean, variance and p-value is calculated. Although the overall mean of the EK model is higher, the WE model scores better at the average mean per question due to its higher variance. Of 15 questions, the mean of the WE was higher in eight questions. Four times, both models had a mean below 3.5, which is considered insufficient. For both models, 10 of the 15 calculated means are significant. Concluding, there is no model obviously outperforming the other model, but the EK model performs slightly better than the WE model because the overall mean is higher and the percentage of Don’t know is lower.
Tab. 3: Results of the user test per individual question. * means significant with $\alpha = 0.1$, ** means significant with $\alpha = 0.05$. DK refers to the percentage of respondents that did not know the proposed substitute.

<table>
<thead>
<tr>
<th></th>
<th>Expert knowledge</th>
<th></th>
<th></th>
<th></th>
<th>Word embedding</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>var</td>
<td>DK</td>
<td>p-value</td>
<td>mean</td>
<td>var</td>
<td>DK</td>
</tr>
<tr>
<td>q1</td>
<td>2.90</td>
<td>2.67</td>
<td>43.2%</td>
<td>0.001**</td>
<td>3.32</td>
<td>3.02</td>
<td>25.8%</td>
</tr>
<tr>
<td>q2</td>
<td>2.42</td>
<td>2.45</td>
<td>18.9%</td>
<td>0.000**</td>
<td>2.71</td>
<td>2.92</td>
<td>15.0%</td>
</tr>
<tr>
<td>q3</td>
<td>2.99</td>
<td>2.85</td>
<td>36.3%</td>
<td>0.003**</td>
<td>3.77</td>
<td>2.88</td>
<td>12.7%</td>
</tr>
<tr>
<td>q4</td>
<td>3.72</td>
<td>2.53</td>
<td>19.3%</td>
<td>0.139</td>
<td>3.04</td>
<td>3.19</td>
<td>34.5%</td>
</tr>
<tr>
<td>q5</td>
<td>3.50</td>
<td>3.01</td>
<td>11.7%</td>
<td>0.992</td>
<td>2.74</td>
<td>2.34</td>
<td>13.6%</td>
</tr>
<tr>
<td>q6</td>
<td>2.88</td>
<td>2.84</td>
<td>7.4%</td>
<td>0.000**</td>
<td>3.78</td>
<td>2.78</td>
<td>8.9%</td>
</tr>
<tr>
<td>q7</td>
<td>3.01</td>
<td>2.88</td>
<td>30.9%</td>
<td>0.003**</td>
<td>3.71</td>
<td>3.23</td>
<td>28.8%</td>
</tr>
<tr>
<td>q8</td>
<td>4.75</td>
<td>1.76</td>
<td>3.2%</td>
<td>0.000**</td>
<td>4.14</td>
<td>3.26</td>
<td>4.4%</td>
</tr>
<tr>
<td>q9</td>
<td>3.77</td>
<td>2.95</td>
<td>16.5%</td>
<td>0.064*</td>
<td>3.96</td>
<td>2.77</td>
<td>28.9%</td>
</tr>
<tr>
<td>q10</td>
<td>3.45</td>
<td>2.60</td>
<td>20.8%</td>
<td>0.719</td>
<td>4.31</td>
<td>2.29</td>
<td>5.5%</td>
</tr>
<tr>
<td>q11</td>
<td>4.02</td>
<td>2.67</td>
<td>12.7%</td>
<td>0.000**</td>
<td>2.57</td>
<td>2.43</td>
<td>20.6%</td>
</tr>
<tr>
<td>q12</td>
<td>3.88</td>
<td>2.90</td>
<td>13.6%</td>
<td>0.008**</td>
<td>2.89</td>
<td>2.45</td>
<td>29.7%</td>
</tr>
<tr>
<td>q13</td>
<td>3.52</td>
<td>2.79</td>
<td>8.5%</td>
<td>0.873</td>
<td>3.48</td>
<td>2.84</td>
<td>10.0%</td>
</tr>
<tr>
<td>q14</td>
<td>3.33</td>
<td>2.75</td>
<td>20.1%</td>
<td>0.271</td>
<td>3.46</td>
<td>3.32</td>
<td>39.8%</td>
</tr>
<tr>
<td>q15</td>
<td>4.95</td>
<td>1.18</td>
<td>7.0%</td>
<td>0.000**</td>
<td>3.48</td>
<td>3.30</td>
<td>30.1%</td>
</tr>
</tbody>
</table>

5 Conclusion

This research aimed to answer the following main question:

- *How can we develop a system that effectively determines ingredient substitutes for recipes based on a combination of word embeddings and expert knowledge?*

This question is answered by the following sub-questions:
1. *Can the expert data also support implicit substitution rules, next to the explicit substitution rules defined by the expert?*

2. *Which similarity formula can best be applied when calculating the distance between two vectors in the word embedding model, intending to rank the vectors to find ingredient substitutes?*

3. *Which of the following systems results in a better performance: the expert knowledge system or the word embedding system?*

To answer the first sub-question, three experiments were designed to test the three implicit substitution rules. This resulted into the implementation of the first rule based on symmetry. The two other rules, based on triadic closure and taxonomically, were not applied. While evidence on the rules was found in the CT data, the application of these rules caused the network to become too generic and no other positive effects, like adding new ingredients, were found.

In order to answer the second sub-question, the performance of three similarity measures and two word embedding models were evaluated. The best results were achieved when applying the model trained by Google and using the CosAdd similarity formula, showing that the implementation of analogical reasoning improves the results of the substitute search task.

A user test was conducted in order to determine which model performed best, answering the third sub-question. The user test showed that the expert knowledge model performed slightly better. The overall mean of the EK ratings are slightly higher then the WE ratings. Yet, the WE model scores better at the average mean per question due to its higher variance. However, not all results were significant and the WE model also had a higher percentage of substitutes that were unknown by the user. Concluding, if the EK model and the WE model suggest different substitutes, this research shows that the EK model should be applied first. The final flow of the logic combiner is visualized in Figure 8.

### 5.1 Discussion

Conducting a user test, both the results and the proposed questions should be critically examined. Firstly, because the context of the Flavourspace recipe search engine may differ from the context of choosing a substitute in the supermarket, extensively A/B testing the questionnaire can add to developing a more precise context and formulation of the questions.
Fig. 8: Final flow of the logic combiner. The ingredient that the user wants to substitute is the input of the logic combiner. The output is a ranked list of possible substitutes.

Secondly, the vast majority of the user test respondents are Dutch citizens and the expert knowledge is derived from an American website. These two factors could possibly lead to a bias towards the traditional Northern American cuisine and other similar cuisines like the Northern European cuisine. This bias should be taken into account when using the substitute system.

Lastly, a reason why the Google model outperformed the Recipe model might be explained by the training corpus of Google, which is much larger than the training corpus of the Recipe model. If a certain word is not frequently used in the corpus, the probability in the model of that word is going to be lower anyway, irrespective of the actual substitute quality. Extending the Recipe corpus with more ingredient lists will improve the quality of the Recipe model.

5.2 Future work

The quality of a substitute ingredient also depends on the recipe it is used in. In these models, the substitution rules are determined based on the ingredient that should be replaced. Further research is needed to explore how the recipe could be taken into account when ranking the final substitutes.

Another improvement of the model is to incorporate more generic user characteristics, like vegetarianism or lactose-intolerance. Using the taxonomy derived from the expert knowledge, the ingredient tree can exclude all categories and subcategories of meat or dairy. To incorporate this into the word embedding model, the similar theory of linguistic relations could be used.

Furthermore, the logic combiner could be developed in more detail. A differentiation can be made between implicit and explicit expert substitu-
tions of which both the individual quality could be tested.

Also, the final decision of the user in the Flavourspace engine in regard to the substitute they prefer to use can also be incorporated in the system. Over time, the system would be able to improve itself based on user judgments.

Lastly, the system could also provide the subcategory of the substitutes, instead of the specific substitute itself. Instead of offering two types of beans as a substitute for Chili beans, the system would then offer a *type of dry bean*, as all given substitutes belong to that subcategory.

Concluding, this thesis has contributed to user empowerment in cooking practices by providing people the possibility to easily adjust recipes to their needs and preferences with a newly developed ingredient substitution system. This research has used two methods for obtaining ingredient substitutes by using expert knowledge from a domain expert and the data-driven method of word embeddings. The expert knowledge, derived from Cook’s Thesaurus, has been interpreted and tested in order to be able to use by the Flavourspace search engine. At the same time, this research has also shown how a substitute could be derived based on recipe data without the need of human expertise. This would be a highly cost-effective method for finding new ingredient substitutes.
References


REFERENCES


A Appendix

A.1 Performance results of MAP, recall and precision.

Tab. 4: Results of MAP, recall and precision of the Google and Recipe model, using three different similarity measures.

(a) Metrics applied on the ingredient level

<table>
<thead>
<tr>
<th></th>
<th>CoSim</th>
<th>CosAdd</th>
<th>CosMul</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>0.047</td>
<td>0.055</td>
<td>0.055</td>
</tr>
<tr>
<td>Recipe</td>
<td>0.031</td>
<td>0.050</td>
<td>0.045</td>
</tr>
<tr>
<td>Recall</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>0.056</td>
<td>0.066</td>
<td>0.065</td>
</tr>
<tr>
<td>Recipe</td>
<td>0.038</td>
<td>0.052</td>
<td>0.050</td>
</tr>
<tr>
<td>Precision</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>0.054</td>
<td>0.067</td>
<td>0.067</td>
</tr>
<tr>
<td>Recipe</td>
<td>0.035</td>
<td>0.052</td>
<td>0.050</td>
</tr>
</tbody>
</table>

(b) Metrics applied on subcategory level

<table>
<thead>
<tr>
<th></th>
<th>CoSim</th>
<th>CosAdd</th>
<th>CosMul</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>0.208</td>
<td>0.229</td>
<td>0.222</td>
</tr>
<tr>
<td>Recipe</td>
<td>0.174</td>
<td>0.173</td>
<td>0.166</td>
</tr>
<tr>
<td>Recall</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>0.227</td>
<td>0.253</td>
<td>0.247</td>
</tr>
<tr>
<td>Recipe</td>
<td>0.186</td>
<td>0.188</td>
<td>0.182</td>
</tr>
<tr>
<td>Precision</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>0.206</td>
<td>0.251</td>
<td>0.246</td>
</tr>
<tr>
<td>Recipe</td>
<td>0.150</td>
<td>0.178</td>
<td>0.172</td>
</tr>
</tbody>
</table>