Sentiment Analysis on Ingredients

Hristo Minkov
Maastricht University
h.minkov@student.maastrichtuniversity.nl

Zhecho Mitev
Maastricht University
z.mitev@student.maastrichtuniversity.nl

Abstract

Often people find themselves in a situation when they wonder if adding a certain component to a meal will improve or worsen its taste. This problem leads to the question of whether a specific ingredient combination tastes delicious or disgusting. This research will scrape and bring together several datasets about various recipes from worldwide cuisines. Each recipe has a list of ingredients. The goal of this paper is to create a model that can evaluate correctly the tastiness of the ingredient combination. Thus, we call th idea Sentiment Analysis on ingredients.

1 Introduction

Nowadays, cooking becomes much easier thanks to the Internet and to the online recipes that are available. Anyone can browse for a product to figure out how exactly it can be prepared. However, if somebody would like to experiment and mix several different products, there is a big chance that the exact ingredients do not exist in a combination. Does it mean that such dish would not taste well?

The key to the answer of this question is hidden in the fact that a recipe can be represented as an embedding. It is possible that the embedded recipe is derived from a picture (Salvador et al., 2017; Carvalho et al., 2018) or computed from the word representation of each recipe ingredient, which is closer to the field of research of this paper. Several applications of representing a word as vector already exist, as one of them, which utilizes the means of word2vec, creates a food similarity map and a recipe embedding map, which can be used for recommendation system for cooks (Altosaar, 2017). In this paper we propose and compare Machine Learning and NLP techniques such as bag-of-words (Yin Zhang, 2010), tf-idf (Salton, 1989) and word2vec (Tomas Mikolov, 2013), in order to predict if a combination of ingredients can turn out to be a delicious one or not. The expectations are that the bag-of-words model would perform well, however due to the sparsity of its matrix it might not be the best one. A model using tf-idf must output a better result and if a word2vec representation is included, this might increase the scores even more.

2 Related Work

The topic of detecting, evaluating, predicting and combining ingredients from recipes has always been an meaningful matter concerning the food industry. Therefore, various research groups and individuals conducted analysis on plenty of food domain tasks.

(Luis Herranz and Jiang, 2018) proposed integrating external knowledge for food recipes into food-oriented applications, with special focus on recipe analysis and retrieval. In the essence of the research, they put emphasis on the ingredients as the main component of food and continue with specifying certain types of ingredients - not directly observable (e.g. salt) which could be represented not as an essential part of the recipe but as a feature to all other recipe elements. We take that into account in our method by using the Tf-idf addition to the transformer.

(Almeida, 2015) focuses on content-based methods for food recommendations in his research. He specifically mentions that there are various term weighting schemes, but the Tf-idf is perhaps the most commonly used among them (Salton, 1989). We use this vector representation for the same reason.

This paper targets mainly the method we use to represent ingredients as feature vectors for the task of ingredient sentiment analysis. (Altosaar, 2017) has done significant amount of research towards building the best food specific word representations. He highlights the importance and flexibility of word embeddings and mentions that by just a small change in the definition of the context of our corpus, we can apply these learned vector representations to a different kind of task. Our paper accommodates this view by using the Word2vec model as a way to improve the performance of the model.

3 Data

In order to make the sentiment analysis work, we need to obtain many recipes which have a set of ingredients. In addition, our research closely examines some of the dataset features - e.g. the distribution of ingredient list lengths for each recipe (see Figure 1). The paper extracts the needed data from several sources:

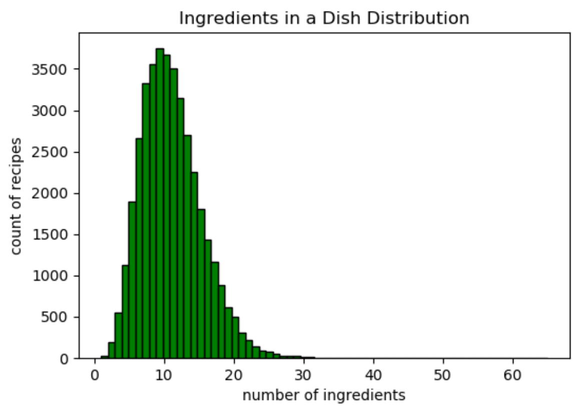

Figure 1: Distribution of ingredient lengths.

3.1 Epicurious Dataset

'Epicurious Recipes with Rating and Nutrition' (Darwood, 2017) is a Kaggle dataset that explores over 20,000 recipes, listed by recipe rating, nutritional information and assigned category. The ingredients were extracted from the categories of each recipe. If a category is not an actual ingredient then it is not added to the set of ingredients, otherwise it is included. For example a recipe with categories 'fish, olive, tomato, low-carb, dinner, healthy' has ingredients 'fish, olive, tomato'. Although this is not the full list of ingredients in the current case, such category extraction helps the model to recognize which main products go well with each other.

3.2 What's Cooking Dataset

'Whats cooking?' (Yummly, 2016) is a Kaggle dataset that includes almost 50,000 recipes categorized by the recipe id, type of cuisine and the list of ingredients. This list contains extensive number of ingredients from different regions of the world, which makes our dataset very diverse.

3.3 Data Scraping

Moreover, a scraping machine was built to extract data from a Bulgarian and other Balkan websites (Food, a,c,b), and at the end, 65 typical Balkan recipes were added. The purpose for including them is to enrich the data by adding new ingredients such as chubritsa, kajmak, ect., which are not present in other types of cuisines.

3.4 Data Preprocessing

After gathering this information into a combined dataset, preprocessing is performed. More specifically, recipes with one or no ingredient are removed, because they do not add value to the model. Furthermore, the text between two round brackets is removed for simplicity. In addition, all ingredients are normalized by removing the accent from the letters with such, in order to bring the text in standard form, with no unknown characters.

3.5 Data Analysis

The most important part of our combined dataset is the set of ingredients needed for preparing a dish. The name of a recipe is optional, since it cannot play a role in improving the model, however it is good to know what the final product of the combination of several ingredients is. Another optional criterion is the rating of a recipe, due to the fact that it is almost impossible to find reliable data for the rating of a dish. Only one of the dataset contains ratings and they are in the form of unusual categorical values (see Figure 2). It can be observed that around 2000 of the recipes are rated with 0.0, which is not a good indication. Moreover, it is unknown how were the recipes evaluated. Therefore, since the sample of the people who gave the rating might be too small, it was decided to disregard the rating of a recipe.

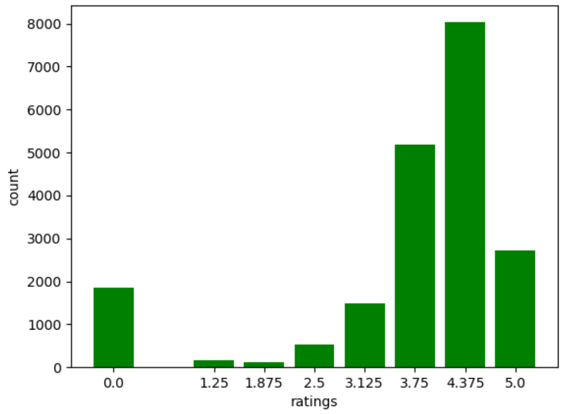

Figure 2: Distribution of rating scores from Epicurious dataset.

3.6 Negative Data Generation

The binary classification problem from this paper still requires negative training examples. However, the datasets only include positive cases of existing combinations of ingredients. The approach towards building a corpus of negative examples involves random sampling from the respective datasets. In general, our approach aims to construct a well-balanced corpus. Accordingly, each of the proposed datasets (of positive ingredient combinations) needs a negative counterpart of the same length. Initially, the algorithm for building the contrary data takes into consideration the distribution of the lengths of each recipe. Then, it draws a collection of random ingredients with size equal to a randomly chosen number from the previously mentioned probability distribution. This frames the balanced training corpus for our problem.

3.7 Train/Test Data

After collecting and cleaning the data, it needs to be splitted into train and test sets. These account for 85% and 15% respectively. The training data is then divided into training and validation sets (again 85% and 15% respectively) for each of the four approaches. The richness of the data makes these figures justifiable.

4 Models

This research focuses on implementing two basic solutions for the ingredient sentiment analysis. Subsequently, it explores the utilization of a particular Word2vec embedding model and how this could potentially improve the performance. All approaches below use the Random Forest classifier with default parameter values as an end-estimator of the respective pipelines. This protects our experiments from bias due to the concrete final estimator's complexity.

4.1 Bag-of-Words Model

This algorithm contains an implementation of the Bag-of-words model as a way of transforming the input raw text data with ingredients into features that can be fed to the final estimator. This model typically disregards grammar and word order but preserves multiplicity. It works significantly well because the raw input contains independent ingredients, splitted by a comma. Therefore, it's reasonable to disregard the order and the grammar of each data point.

4.2 Tf-idf Model

The difference between the BOW and the Tf-idf models is that Tf-idf keeps track not only of the frequency of an ingredient, but also its importance for the current recipe. This will prevent certain generally frequent ingredients of the whole corpus (e.g. salt, onion, water, etc.) from having high impact on the vector representations. For this reason, this approach will improve the performance on the task.

4.3 Word2vec Models

Distributed dense representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words (Tomas Mikolov, 2013). Word2vec is a two-layer neural net that transforms words from a text corpus into feature vectors.

4.3.1 Word2vec Implementation

The implementation of the Word2vec model in this paper do not use pre-trained embeddings since that could potentially introduce corpus-specific bias. Moreover, the size of the training data ($\approx 130,000$ recipes) is sufficient to train a well-performing Word2vec model. It computes 300 features for each word, the minimum word count is 2 and the context window of the model is 10 (the mean size of a recipe ingredient list).

The Word2vec model's behaviour is acceptable if one inspects several examples of most similar words. (see Table 1)

Word	Score
coffee	0.9448
vanilla	0.8981
pastry	0.8492
liqueur	0.8418
candy	0.8334
milk	0.8238
berry	0.8103

Word	Score
pork	0.9663
cabbage	0.9420
vinegar	0.8905
meat	0.8818
rice	0.8504
cooking oil	0.8479
white rice	0.8461

Table 1: Example similarity words for chocolate (left) and beef (right).

4.3.2 Plain Word2vec Model

This model explores the basic Word2vec approach where every ingredient is turned directly into a feature vector and then fitted to the estimator. Therefore, the algorithm pipeline consists of a Word2vec transformer and a Random Forest classifier with default parameters. The power of this approach is that it uses very large feature embeddings for each ingredient which allows for detecting more complex relations in recipes.

4.3.3 Tf-idf Word2vec Model

The tf-idf representation of the Word2vec model provides an additional feature of accounting for ingredient importance. As explained for the basic tf-idf model, some ingredients occur so often, that they should be considered with lower weight when computing the feature vector representations. This will eliminate the risk of every element being very similar to one specific ingredient that just happen to appear more often in various recipes.

5 Evaluation

This section mainly focuses on analysis of the four embedding models. All of the approaches are evaluated on the previously defined test set and the whole classification report is taken into consideration. The paper assumes that people are significantly good at measuring whether it is justified to put several ingredients together. Therefore, this research targets towards the threshold of 95% in order to claim achieving human-baseline score.

5.1 Results

The evaluation procedure reached the following results after running the test evaluation on all four distinct models:

BOW	*precission*	*recall*	*f1-score*
neg	**0.96**	**0.89**	**0.92**
pos	**0.89**	**0.97**	**0.93**
accuracy	*N/A*	*N/A*	**0.93**
macro avg	**0.93**	**0.93**	**0.93**
weighted avg	**0.93**	**0.93**	**0.93**

Table 2: Classification report of the Basic Bag-of-words model.

Tf-idf	*precission*	*recall*	*f1-score*
neg	**0.96**	**0.95**	**0.96**
pos	**0.95**	**0.96**	**0.96**
accuracy	*N/A*	*N/A*	**0.96**
macro avg	**0.96**	**0.96**	**0.96**
weighted avg	**0.96**	**0.96**	**0.96**

Table 3: Classification report of the Basic Tf-idf model.

W2V	*precission*	*recall*	*f1-score*
neg	**0.97**	**0.97**	**0.97**
pos	**0.97**	**0.97**	**0.97**
accuracy	*N/A*	*N/A*	**0.97**
macro avg	**0.97**	**0.97**	**0.97**
weighted avg	**0.97**	**0.97**	**0.97**

Table 4: Classification report of the Plain Word2vec model.

W2V (Tf-idf)	*precission*	*recall*	*f1-score*
neg	**0.97**	**0.97**	**0.97**
pos	**0.97**	**0.97**	**0.97**
accuracy	*N/A*	*N/A*	**0.97**
macro avg	**0.97**	**0.97**	**0.97**
weighted avg	**0.97**	**0.97**	**0.97**

Table 5: Classification report of the Tf-idf Word2vec model.

6 Analysis

After observing the achieved results from the four proposed models, this segment will focus on analyzing and clarifying the respective outcomes. The preferable evaluation metric for the justifications is the weighted average f1-score as it is the most general of all mentioned measures. One more thing to notice here is that all evaluations are based on the same final estimator with default parameters (Random Forest classifier).

The Basic Bag-of-words model performs the worst (93% weighted average f1-score) since it has the simplest representation with no added dependencies between different ingredients. The Basic Tf-idf model outperforms it with the significant 3% which makes us conclude that adding a Tf-idf transformation for this specific task can greatly benefit the performance.

Both the Plain and Tf-idf Word2vec implementations perform exactly the same for all available evaluation metrics (97%). This supports our initial hypothesis that a plain Word2vec model (with a big feature vector) is able to learn that certain words appear so often, that their weighted importance for the recipes should be lowered (the exact purpose of the Tf-idf transformer). This makes the Tf-idf addition to the word2vec redundant for this task. Therefore, this research regards the Plain Word2vec model as the simplest solution with the highest score for ingredient sentiment analysis.

7 Conclusion

This research gathered recipes data from various sources to construct a corpus with acceptable combinations of ingredients. We trained four distinct models and eventually picked the Plain Word2vec one since it equal score with the Tf-df Word2vec model but it is simpler in terms of implementation. It managed to score 97% which is regarded as approaching the human-level performance for the ingredient sentiment analysis task. Our research can be further incorporated into the creativity components of food recommendation or new recipes generation systems. We hope that the food industry can make use of our research and make people more aware of what they consume - 'Tell me what you eat and I will tell you what you are.' (Brillat-Savarin, 1826).

8 Acknowledgement

We thank Dr. Gerasimos (Jerry) Spanakis and Tonio Weidler for providing us with the ground knowledge for Natural Language Processing and also giving us rigorous feedback and ideas of how to polish our research.

References

Jorge Almeida. 2015. Personalized food recommendations. exploring content-based methods.

Jaan Altosaar. 2017. food2vec - augmented cooking with machine intelligence.

Jean Anthelme Brillat-Savarin. 1826. *The Physiology of Taste*.

Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. 2018. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. *CoRR*, abs/1804.11146.

Hugo Darwood. 2017. Epicurious recipes with rating and nutrition.

BG Food. a. Bulgarian food.

EU Food. b. European cuisines.

SRB Food. c. Serbian recipes.

Weiqing Min Luis Herranz and Shuqiang Jiang. 2018. Food recognition and recipe analysis: integrating visual content, context and external knowledge.

G. Salton. 1989. Automatic text processing.

Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marn, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning cross-modal embeddings for cooking recipes and food images. pages 3068–3076.

Kai Chen Gred Corrado Jeffrey Dean Tomas Mikolov, Ilya Sutskever. 2013. Distributed representations of words and phrases and their compositionality.

Zhi-Hua Zhou Yin Zhang, Rong Jin. 2010. Understanding bag-of-words model: A statistical framework.

Yummly. 2016. Whats cooking?