Ludwig Maximilian University of Munich

Master's Thesis in Computational Linguistics

Computational Analysis of Food Using Distributional Semantics

Author:
Nicolai Ruhnau

Supervisor:
Prof. Dr. Hinrich Schütze

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Institute for Information and Language Processing

Working period: September 9, 2018 — January 23. 2019

Declaration of Authorship

I, Nicolai Ruhnau, declare that this thesis titled "Computational Analysis of Food Using Distributional Semantics" and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Munich, January 23, 2019:

LUDWIG MAXIMILIAN UNIVERSITY OF MUNICH

Abstract

Institute for Information and Language Processing

Master of Science

Computational Analysis of Food Using Distributional Semantics

by Nicolai Ruhnau

Often the evaluation of food text representations is still done by hand, in lack of good automated evaluation methodologies. Visualizations of distributional food semantics are often not used to their full potential. There are only a few works that use dense word embeddings and more complex food corpus model techniques in general, such as topic modeling.

This thesis gives an overview of the existing literature and helps define the rather new field of research of the computational analysis of food using distributional semantics. The use of various food text representations is investigated, creating embeddings and successfully conducting new experimental benchmarks in order to evaluate them. The methodology of these automated benchmarks is explained step-by-step, showcasing powerful Python libraries. Latent topics in a food dataset are extensively explored, with various visualizations techniques for a more intuitive understanding.

It is shown that a smaller domain specific corpus can produce embeddings with similar and sometimes better food category prediction capabilities as embeddings based on very large corpora such as Wikipedia or Google News. An interesting find is how beneficial subword-level embeddings are in the context of food and what the reasons for this are. Complicated embeddings that include food image and recipe instructions information did generally perform not better and sometimes much worse than a simpler baseline embedding based on the same corpus. It is shown how visualizations can be used to very effectively to explain results and describe datasets in an intuitive way. The presented detailed list of the available linguistic food resources helps others in rapidly applying the techniques mentioned in this thesis to other datasets.

Acknowledgements

I would like to thank **Prof. Dr. Hinrich Schütze** for the opportunity to write this thesis under his supervision.

I would like to thank **PhD Candidate and M.Sc. Ehsaneddin Asgari** for his advice, his technical help and his ideas, influencing many parts of this thesis.

I would also like to thank my friends and family for their spiritual support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Food made the news in early 2019: Be it emotional outbursts from a football player eating a gold-leaf steak in Dubai (Press 2019) or the US-Administration celebrating the win of an American football college team in the White House serving 3.000 $ worth of fast food next to golden candelabras (Bump 2019; Maura Judkis 2019). Also just now one of Berlin's staples, the currywurst - the famous bratwurst with tomatoey curry sauce - has been given a 70 years commemoration coin, reflecting the cultural importance of food even in Germany, cf. Frost 2019.

But the importance of food for human kind transcends centuries and trends, taking for example potatoes, that are now being cultivated even on the moon (BBC News 2019): Similar to how the imported south American tuber was an essential element of Prussia's rising economy in the 18 hundreds, it is now China investing in it - showing its political importance as the government tries to make it more popular with the people due to its comparatively low water consumption (Deuber 2019; Ankenbrand et al. 2015).

There has been interesting natural language processing (NLP) research around food in recent years, for example the prediction of food prices based on the words in restaurant menus presented in a book called "The Language of Food" (Dan Jurafsky 2014, 2015). Furthermore, technology has advanced enough to attempt tackling more fundamental problems, such as the translation and transformation of cuisine styles (Kazama et al. 2018; Nobumoto et al. 2017). There have also been promising attempts in revealing latent topics within linguistic food resources (Min et al. 2018a, p.e.).

But there is still a preference for using manual evaluations, even though they are expensive and error prone. How can one evaluate distributional semantics like in the works by (Kazama et al. 2018; Nobumoto et al. 2017) in an automated fashion? How to evaluate ingredient similarity, how to compare "apples with oranges in the context of recipes" (Teng et al. 2011, cf. p. 6)?

1.2 Definitions

1.2.1 Computational Analysis of Food Texts

Up until 6 years ago, there had "only been little research examining the usefulness of NLP in tasks related to the food domain" (Wiegand et al. 2012b, p. 1). August 2018 saw the release of "the first comprehensive survey that targets the study of computing technology for the food area", while at the same time outlining the field of research in a fundamental way (Min et al. 2018b, p. 1-4). Computational analysis of food texts is in my own definition - drawing from ibid., p. 3 - about using natural language processing tools like statistics, machine learning and visualization techniques on linguistic food resources with the goal of solving tasks such as prediction, retrieval or recommendation.

As Manning et al. 2008, p. xxxi explain, "Statistical NLP as we define it comprises all quantitative approaches to automated language processing, including probabilistic modeling, information theory, and linear algebra". Computational analysis of food data without

the constraint for textual data was coined the term ”Food Computing” by Min et al. 2018b, p. 3, which includes the important area of computer vision and image related tasks. The tasks need not be limited to vision or text, but can be multimodal, too.

ibid. see the goals of ”Food Computing” mostly about solving issues like monitoring, teaching and so called ”guiding” measures especially in the area of healthy food and medicine, mention also agriculture, gastronomy, satisfying cultural curiosity or advancing biological and neurological sciences.

But it can be argued that the area of research could bring substantial improvements in excess of satisfying basic needs like the absence of diseases or an efficient food industry, by instead directly allowing individuals to experience even more joy eating and drinking. Kazama et al. 2018 mention the personal aspect of food and how a cook's or guest's preference could be better reflected by the use of a system that translates recipes to another cuisine, a work based on word vectors.

This is in line with the recently growing interest in food related tasks like the creation of ”culinary art” or works investigating how to get hold of the semantics in recipe texts, cf. Carvalho et al. 2018, p. 2. Yamakata et al. 2013, p. 1 not only mention culinary art, but also entertainment, human communication as goals, while Trattner et al. 2017a, p. 1 use the terms ”leisure” and highlight that there has been surprisingly little importance given to it comparing it to the attention it receives in other domains. ibid., p. 1 speak in contrast to the authoritatively ”guiding” and institution-sympathetic position of Min et al. 2018b, p. 3, when they p.e. speak of recommender systems as ”empowering” the individual user finding the recipes that fit her ”life-style preferences”.

The end user perspective of online recipe websites is being extensively treated in M. Chang et al. 2018, p. 4, which may include professional chefs trying to reverse engineer an unusual dish or experienced business researchers correlating the amount of de facto standardization of the preparation for a specific food with its market saturation. Wiegand et al. 2012c, p. 508 think of ”every-day-life shopping scenarios” when it comes to use cases for NLP in the food domain.

The French author Despres 2014, p. 27 underlines the importance of web technologies in the field of computational analysis of food, calling it the apparently established term ”la cuisine numérique”. As she is creating an ontology, the computational modeling should encompass all relevant aspects of food, not only images and ingredient names and measurements, but also the taste, texture, smell, season, regional origin, relation to base recipes, cuisine representativity, conservation measures and more (ibid., p. 30-31). Some of these aspects can already be found in existing databases and websites, especially with regards to textual resources presented later.

1.2.2 Distributional Food Semantics

Distributional semantics are about using the statistical measure of co-occurrence of terms in text corpora to build semantic representations of words and larger texts, ”a mainstream research paradigm in computational linguistics (Lenci 2018, p. 151). As ibid., p. 152 cites Harris 1954, p. 156 verbatim, the hypothesis is that “difference of meaning correlates with difference of distribution.”, who builds on Wittgenstein's “the meaning of a word is its use in the language” (Wittgenstein 1953, PI 43), cited in (Daniel Jurafsky et al. 2018, p. 106), as did Joos and Firth in the 1950s with important works (ibid., cf. p. 106 for more details). Along these lines, distributional food semantics will be about defining the meaning of ingredients p.e. by the ingredients and instructions context in recipes.

As words like ”bank” or ”space” are ambiguous (Schütze et al. 1995, p. 1), the meaning of ingredients such as wheat flour depends on the situation they are used in, p.e. to thicken a sauce versus being the main ingredient in baked goods, which in themselves can be very diverse. What ”tortilla” means depends p.e. on whether the main ingredients are eggs, potatoes and olive oil or instead water and flour - salt being present in both.

The general field is also known as vector space semantics (Lenci 2018, p. 152) or simply "vector semantics" and uses unsupervised or automatically learning methods, as opposed to creating representations manually(Daniel Jurafsky et al. 2018, p. 101-102). Building on the vector space model by Salton et al. 1975, the use of vectors as a measure for co-occurrence offers the advantage of a continuous "mathematical encoding of the distributional properties of lexemes", which opens up the convenient way of computing the semantical similarity of two texts by computing the cosine similarity (Lenci 2018, p. 153, see also Daniel Jurafsky et al. 2018, cf. p. 106; Manning et al. 2008, ch. 6.3). The resulting numbers between 1 for identical vectors and -1 for completely inverse meanings can not only be used for conducting ingredient level comparisons, but applied to recipes as well p.e. by averaging all ingredient vectors per recipe. Moreover, working with vectors provides a straightforward way to visualization p.e. via reduction of the number of dimensions to two (Daniel Jurafsky et al. 2018, cf. p. 123). The vectors are also commonly called embeddings, as they signify words being embedded in a specific vector space (ibid., cf. p. 107).

A relatively simple but useful way of assigning meaning to a recipe would lie in creating a sparse matrix (only some columns per instance are non-zero) with ingredients as columns with each ingredient being defined as the term-frequency inside a recipe weighted by the inverse-document-frequency. This means assigning importance to more discriminative ingredients like special spices rather than basic ingredients like salt or eggs (ibid., cf. p. 114).

Another, rather "direct" approach to model meaning by co-occurrence for a recipe could consist in creating a sparse matrix with the ingredients in the specific corpus vocabulary as row and column dimensions, counting the number of co-occurring ingredients inside a k-sized window - although this would usually not be computationally sensible because of the high number of dimensions (Schütze et al. 1995, cf. p. 5).

More practical approaches achieve dense matrices p.e. by the use of n-grams (Schütze 1993, cf. p. 2) or second-order co-occurrence, i.e. measuring the shared neighbors of two ingredients in the corpus instead of measuring the simple fact of two ingredients being neighbors in a recipe (Schütze 1992, 1998; Schütze et al. 1995, cf.). In recent years very popular implementations of dense word vectors are "Word2Vec"(Mikolov et al. 2013), GloVe (Pennington et al. 2014) and fastText (Bojanowski et al. 2016), as well as the document-level implementations Doc2Vec (Le et al. 2014) and Latent Dirichlet Allocation (Blei et al. 2003), as p.e. mentioned in Daniel Jurafsky et al. 2018, cf. p. 127; Gomez et al. 2018, p. 6; Lenci 2018, p. 159-160).

One of the main research questions in distributional semantics is the question of how adequate the various possible distributional representations of meaning are, in other words how to compare and benchmark them (Lenci 2018, cf. p. 160-161). Existing research on this with regards to the field of distributional food semantics and general NLP research on the computational analysis of food will be presented in the next section.

1.3 NLP Research on Computational Analysis of Food

1.3.1 Food Representations Literature

As one of the earlier works in the domain of computational analysis of food, Ahn et al. 2011 created a flavor network, with each ingredient being represented by the vector of its chemical flavor compounds. The goal was to test whether ingredients are paired more often if they share flavor compounds. Examining the data a convincing case was made that Western cuisine labeled recipes tend to use ingredients with shared flavor compounds in contrast to East Asian and Southern European recipes (ibid., p. 3). They also used ingredient frequency to visualize the main ingredient of a cuisine and ingredient type categories. The data had been obtained from a book about flavor compounds on one hand, and one Korean as well as two US-American as recipe websites on the other hand ("menupan.com," "allrecipes.com" and "epicurious.com") and made public by the authors.

Ahn et al. 2011, p. 4 remark representatively for the time that the availability of recipe data sets was still limited. This in turn limited the popularity of working on food vectors, as most data had to be crawled and parsed by hand, p.e. in the case of Wiegand et al. 2012a from German cooking website forum "chefkoch.de". ibid., p. 22 emphasize that "the usage of co-occurrence measures is only effective if large amounts of data, for instance the web, are used as a dataset". They extract relations such as suitable substitutions for an ingredient and evaluate the results such as the co-occurrence word window size by the ability to retrieve items from a manually compiled list (ibid., p. 26), because as they had stated before: "But in order to find robust methods for these tasks manually labeled data are required for evaluating the output of automated systems" Wiegand et al. 2012c, p. 507.

This obviously requires a large amount of effort (Wiegand et al. 2012b, p. 4), while specialized meanings may not get covered or only at prohibitively high cost (Schütze et al. 1995, cf. p. 2). Moreover, manual evaluation in the food domain can be highly subjective, because even within a certain target group great differences are to be accounted for, such as very experienced versus novice home cooks (Kusu et al. 2017, cf. p. 5).

Marin et al. 2018 present the largest freely available data set so far crawled from more than two dozen websites: 1 million recipes with texts and images. Their multimodal "im2recipe" model jointly combines image with ingredient word vectors based on "Word2Vec" plus instruction vectors based on the "skip-thoughts" method (Kiros et al. 2015) in order to predict recipes by images. Nutritional information is added to the model via another database, semantic categories are added by choosing the most common titles and categories from an image dataset called "food 101" (Bossard et al. 2014). "t-Distributed Stochastic Neighbor Embedding" (t-SNE) visualization (Maaten et al. 2008) is used to display food healthiness and semantic categories (Marin et al. 2018, p. 9). Human approval of the embeddings via Amazon Mechanical Turk service ("mturk.com") is measured ibid., p. 9.

Teng et al. 2011 created a network of ingredient complements and substitutes in a different way than Ahn et al. 2011 by using co-occurrence of ingredients. The data was recipes and accompanying reviews scraped from a cooking "allrecipes.com", the method based on pointwise mutual information, a sparse vector method similar to "term frequency - inverse document frequency (TF-IDF), cf. Daniel Jurafsky et al. 2018, p. 116. This means an ingredient that occurs in every recipe will be given a smaller weight than an ingredient that occurs only in some recipes. They benchmarked recipe representations using the ingredient network information by predicting which recipe of recipe pairs would receive a higher rating according to the previous ratings of a user (Teng et al. 2011, p. 6). These predictions were learned via support vector machines (Cortes et al. 1995, SVM, cf.) and stochastic gradient boosting trees (Friedman 2002).

Targeted towards recommendation engines, Nezis et al. 2018 applied Doc2Vec on crawled recipes from "allrecipes.com" combined with healthiness information derived from the UK Food Standards Agency. Automated evaluation metrics were similar to (Teng et al. 2011) rating based. Trattner et al. 2017b also used data from "allrecipes.com" to predict healthiness of a recipe. In another work using data from this website but also from German "kochbar.de" Trattner et al. 2017a successfully employed a complicated and astonishingly rich feature vector for recipes (ibid., p. 27) to help predict and recommend the most popular recipes.

Kusmierczyk et al. 2016b predicted food types and ingredients usage evaluated again by recommender systems metrics with recipes crawled from the German site "kochbar.de". Ingredient and ingredient category occurrence distributions were the basis for recipe vectors. Improving recommender engines for healthy recipes was the goal of Trattner et al. 2017b, who benchmarked a variety of algorithms on recipes and user profiles from the "allrecipes.com" website, with LDA performing competitively ibid., p. 494.

Using distribution probabilities, Sajadmanesh et al. 2016 used normalized ingredient as well as flavor occurrences for each cuisine to obtain feature vectors. The data used was a list of "yummly.com" recipes of various languages translated via "Google translate API" and

crawled ingredients from the ”BBC food” website (”bbc.co.uk/food”) in order to have a reference list of ingredients for the translated recipes. Diversity of ingredients per cuisine, the complexity of cuisines as well as the similarity of ingredients and cuisines was then evaluated by the use of Jensen-Shannon- and Kullback-Leibler-divergence (ibid., p. 4-5). SVM and Neural Network classifiers were trained, too, to predict cuisines and merged cuisine ”regions” on a separate test data set, with each recipe being represented by a sparse TF-IDF vector of all ingredients in the data set (ibid., p. 4-6).

As well using an SVM cuisine classifier, Su et al. 2014 assign each recipe a sparse Boolean vector - over 5000 ingredients as either 0 or 1 depending if it occurs in the recipe or not -, to which they apply singular-value decomposition with recipes crawled from ”food.com”. Another cuisine prediction, now with a logistic regression classifier, was done by Kicherer et al. 2017 with data crawled from the German cooking website ”chefkoch.de”. Different sparse vectors based on various Boolean ingredient, instruction word lists or context words are compared on the ability to predict cuisines and ingredient type categories.

Kalajdziski et al. 2018 used recipes from ”yummly.com” available via the ”Whats cooking?” competition on ”kaggle.com” converted to vectors of pointwise mutual information to predict cuisines with SVM, Naive Bayes and Neural Network classifiers.

Recipe generation is another area of currently very active research that uses food vectors. Kiddon et al. 2016, p. 334 downloaded the so called ”Now You’re Cooking” data from ”ffts.com” in the structured ”Meal-Master” format, preprocessed it via ”Word2Vec”and generated recipe sequences, while human evaluation was conducted via Amazon Mechanical Turk in lack of good automated meaning evaluation metrics. Very similar is the work of Willis et al. 2017 who also used ”Word2Vec”but added a k-means clustering approach of sparse TF-IDF recipe vectors, again with human evaluation but on a much smaller scale. Using again the same corpus, Parvez et al. 2018, p. 8 evaluated their recipe generator using an interesting metric where the system is supposed to fill in blanks for ingredients in recipes.

Sauer et al. 2017 used the corpus supplied by Ahn et al. 2011 with fastText embeddings to determine ingredient complements and substitutes as well as to generate recipes, without an automated evaluation of meaning. Ingredients are visualized via t-SNE (Sauer et al. 2017, p. 12). They also mention Altosaar 2017a his use of ”fastText” on this dataset as well as a larger ”yummly.com” one from ”kaggle.com”, who features a blog post (ibid.) and a two year old GitHub project (Altosaar 2017b) dubbed ”food2vec”. The ambitious idea is executed as simple proof of concept presenting examples and t-SNE visualizations, with the online demo unfortunately not entirely working currently and no considerations of how to benchmark. In addition to the ”Meal-Master” corpus mentioned above, Bostan 2017 downloaded the freely available corpus of ”openrecip.es” and applied hierarchical clustering of Boolean recipe matrices with regards to cuisines. They then also used ingredient-pair co-occurrence matrices weighted by pointwise mutual information and reduced by Singular Value Decomposition to achieve semantic ingredient vector analogies which they explored visually and with examples. An automated measure to asses the quality of the recipe sequence generation undertaken was left to future work, p.e. on tasks such as Question&Answer, ibid., p. 49.

A multimodal recipe Question&Answer corpus and benchmark was achieved by Yagcioglu et al. 2018 using data from ”instructables.com” that contains images and instructions for every instruction step. Among other things, they used the method proposed in Marin et al. 2018 to detect recipe ingredients and trained a Doc2Vec model to represent recipe titles and descriptions.

One of the most interesting recent research ideas is about food translation: using analogies of known food items to explain foreign menus by Nobumoto et al. 2017 or the transformation of a recipe to another cuisine style by Kazama et al. 2018.

One set of recipe vectors (Nobumoto et al. 2017) is created via a custom method that uses as dimensions the counts of the characteristic words of manually compiled categories

like foods, cooking utensils or season. Another set of vectors based on "Word2Vec"is then compared by hand to determine which vector method produces more correct and imaginable dish similarities, p.e. translating Japanese "Sashimi" to Italian "Carpaccio", computed with cosine similarity (Nobumoto et al. 2017, p. 6 with great examples for Japanese, Italian, Korean, American dishes). Data comes from a variety of sources, mainly the Japanese site "cookpad.com" and Wikipedia. Kazama et al. 2018 propose a system based on "Word2Vec"enhanced by incorporating cuisine information to transfer p.e. the Japanese recipe "Sukiyaki", a hot pot dish, to French style cooking. A professional chef reviews the suggested recipe, because as they state, "Rating by experts is the standard approach for assessing novel generative artifacts, e.g., in studies of creativity (Jordanous 2012), but going forward it is important to develop other approaches for assessment." (Kazama et al. 2018, p. 6). t-SNE is used to visualize ingredients and cuisines ibid., p. 6.

1.3.2 Food Topic Modeling Literature

The technique to discover latent topics in a corpus called "Topic Modeling" is based on the "Latent Dirichlet Allocation" (LDA) by Blei et al. 2003, p. 996, an interesting food embedding technique based on hierarchical Bayesian probabilities, discussed in a separate subsection here as it exhibits a variety of particularities to the representations presented so far. Documents are embedded by latent topics in the corpus - a great and maybe more intuitive way to explore a dataset. More technical details are presented in chapter 2.

Different automatic metrics have been developed to find the best number of topics. These include logarithmic likelihood (Griffiths et al. 2004), average correlation between pairs of topics to ensure discriminability (Cao et al. 2009, p. 3) and symmetric Kullback-Leibler divergence combined with singular-value decomposition (Arun et al. 2010). The "coherence" metric by Mimno et al. 2011, according to the assessment of domain experts of the used corpus "corresponds well with human coherence judgments" (ibid., p. 262) while not requiring a separate held-out evaluation corpus (ibid., p. 266). J. Chang et al. 2009 presented an evaluation of perplexity and log-likelihood that showed even a negative correlation with human coherence and relevance assessments.

Focusing on dietary information like nutritional values, Kusmierczyk et al. 2016a are one of the few to apply topic modeling to the food domain. They investigate inter-nutrient correlations, p.e. between proteins and kcal., based on detailed data crawled from "allrecipes.com", creating nutritional topics by way of designing a multi-label logistic regression enhanced LDA model. A standard LDA is evaluated against this via automated nutrition prediction and manual topic analysis of 5 topics. The number of topics equaling 5 is chosen by visualization considerations (ibid., p. 2015).

Min et al. 2018a presented a multi modal cuisine summarization framework based on a Bayesian probabilities topic model, using recipes crawled from "yummly.com" that include information like the course category. Performance was evaluated using image retrieval, comparing their technique with LDA among other methods, as well as retrieving the top five cuisine and course labels for recipes (ibid., p. 8, p. 12). They include beautiful visualizations such as topic word clouds summarizing different cuisine recipes and a cuisine network (ibid., p. 10-12). The number of topics is selected by comparing the perplexity score of models with a number between 20 and 200 topics on a separate test set (ibid., p. 8).

1.4 Linguistic Resources Available for the Food Domain

1.4.1 Recipe and Ingredient Datasets

One of the main issues seem to be about finding linguistically useful food data. A lot of the times researchers still tend to crawl their own datasets, without publishing them. The overview about food-related datasets presented in Min et al. 2018b, table 1. is written from the point of view of multimodal or image only research, as important freely available linguistic datasets are missing as well as details about the texts.

The largest collection of recipe texts as well as images comes from Marin et al. 2018. Recipes were crawled from more than two dozen cooking websites and preprocessed, checking p.e. for duplicates (ibid., p. 3). The structured .json files available contain recipes with the identifiers linking them to the image files, as well as basic ingredient lists and instructions. Semantic categories added to the recipes in the algorithmic system like the most common titles or nutritional information are not provided for the recipes in the publicly available data. The ingredient lists are well cleaned and there is even an additional file containing all recipes with simplified ingredients only, i.e. measurements removed and words like "bell pepper" formatted as "bell_pepper", as this is needed for the training pipeline. This result of the so called "ingredient extraction" is described (ibid., p. 5), although the implementation not included in the publicly available code on GitHub, neither on the newer PyTorch code[1] nor the older Torch7 Code[2]. The dataset is very large, clean and consistent, but does not contain categorized or labeled recipe data.

Another useful dataset is created by Min et al. 2018a Yummly66K"[3], named because it contains 66 thousand recipes from the website "yummly.com". The publicly available text and image dataset contains 10 .json files representing the 10 cuisines, that have many labels such as flavors, cuisine, course and well written titles, although flavor and course information tend to be missing from time to time. Nine out of the 10 files have the ingredients removed from additional words like measurements, whereas the "American.json" recipes still need to be parsed with ingredients such as "1/2 pound medium raw shrimp, peeled and deveined OR boneless, skinless chicken breast, cut into 1- inch cubes".

The Japanese-English "cookpad" dataset has its English parts as well almost completely from the website "yummly.com" (Harashima et al. 2016, p. 2458). It could be interesting for Japanese-themed food vectors and is available by email application.

The dataset by Ahn et al. 2011 is despite its age one of the most interesting ones, as it combines a variety of attributes of ingredients such as type categories as well as flavor information. A t-SNE visualization as well as a good way to download the dataset is offered by Cheng 2016, where the different food labels are prepared in a better readable format compared to the data available on Yong-Yeol Ahn's website[4]. The data parts are nicely explored in De Clercq 2014, p. 27. Similar to combining disjoint subsets to form a larger, more diverse but also more balanced corpus, the recipes were collected from the Korean and English language communities (Ahn et al. 2011, for details see p. 2). Its size of 56 thousand recipes makes it still one of the largest freely available data sets to this day, especially considering the high quality and consistency p.e. with regards to the vocabulary. This appeal met with response by the research community which has been the "flavor network" and "food pairing hypothesis" based on it on a regular basis as well as using the data itself, p.e. in Sauer et al. 2017. The downside of the available data is that the recipes are very simple lists of ingredients without measurements or descriptions. There are no instructions, recipe titles or anything else, except for very valuable cuisine labels.

Another very interesting and free linguistic resource is the "Meal Master" dataset named after the recipe organizing software[5] and corresponding .mms format[6], also known as "Now You're Cooking Recipe Software". It is as of now more than 158 thousand recipes large[7]. It has been popular among recipe generation researchers and was p.e. used by Willis et al. 2017, Kiddon et al. 2016, Parvez et al. 2018 and Bostan 2017.

[1] "Recipe1M+", github.com/torralba-lab/im2recipe-Pytorch
[2] "Recipe1M", github.com/torralba-lab/im2recipe
[3] isia.ict.ac.cn/dataset/Yummly-66K.html
[4] yongyeol.com/2011/12/15/paper-flavor-network.html
[5] ffts.com
[6] ffts.com/mmformat.txt
[7] ffts.com/recipes.htm

Bostan 2017 also used the ”Open Recipes” dataset[8]. Its is a non profit endeavor that aims to ”Prevent good recipes from disappearing when a publisher goes away.”[9]. There are over 170 thousand recipes, although the download link is sometimes hart to find, hidden in the ”issues” section on GitHub. Unfortunately the quality and consistency of the set varies extremely and requires a lot of parsing to be useful for almost any distributional semantics tasks. ”RecipeRescuer”[10], available online[11], cleaned and reduced the ”Open Recipes” amount to around 90 thousand recipes. Each recipe contains the original website address it was taken from, p.e. a personal blog, and a detailed ingredient lists.

1.4.2 Special Linguistic Food Resources

This section is about food text resources different to the classical recipes-with-ingredients datasets described before. A new and interesting data set is ”RecipeQA”, a multimodal corpus using data from ”instructables.com” with 20 thousand recipes and 36 thousand ”context-question-answer-triplets” (Yagcioglu et al. 2018, p. 9). It is unique in that it contains very detailed and well structured ”how-to” instructions with separate titles and images for each instruction step, with the ingredients inside these steps and not listed separately.

Ontologies are another source of linguistically interesting food text that could be explored if they could bring value to distributional semantics of food, p.e. in the form of food type labels or hierarchical information about ingredients. A particularly well written and regularly updated one is ”FoodOn”[12]. It can be downloaded for free in the .owl format and conveniently explored online[13].

Marin et al. 2018, p. 4 used the United States Department of Agriculture Food Composition Databases[14] to add nutritional information to their ”im2recipe” embeddings. The database contains as of now almost 250.000 finely specified ingredients according p.e. to whether they are cooked or raw. It can readily be queried online or simply downloaded[15].

Mostly Canadian ”FooDB” is another government supported project[16], offering an impressive and comprehensive insight into chemical food constituents. It can be downloaded for free as well, either as .csv or SQL files. There are also various online query services.

Another noteworthy resources for rich text information about food is Wikipedia, p.e. the main Wikipedia article about ”Lists of food and beverage topics”[17] or the specialized, albeit since 2016 not much maintained specialized WikiBooks website[18].

Since the popular BBC Food website was about to be shut down, although it has been confirmed it will remain open Snowdon 2018, there have been initiatives about crawling the data before it gets lost klausbatb 2019. The HTML data offers a total of 74 thousand recipes with rich categorical labels like season, course, or diet, but needs custom parsing and preprocessing.

The modern database ”FoodRepo.org” offers a great .json API access that currently holds over 37 thousand food products. The products data comes from grocery companies, specifically Lidl and the Swiss Migros and Coop Lazzari et al. 2018. It was initiated by the Digital Epidemiology Lab of the École polytechnique fédérale de Lausanne[19] and funded by the

[8] openrecip.es
[9] github.com/fictivekin/openrecipes
[10] sagar794.github.io/RecipeRescuer/
[11] github.com/sagar794/RecipeRescuer
[12] github.com/FoodOntology/foodon
[13] www.ebi.ac.uk/ols/ontologies/foodon
[14] ndb.nal.usda.gov
[15] ndb.nal.usda.gov/ndb/search/list
[16] foodb.ca
[17] wikipedia.org/wiki/Lists_of_food_and_beverage_topics
[18] wikibooks.org/wiki/Cookbook:Table_of_Contents
[19] www.epfl.ch

Kristian Gerhard Jebsen Foundation[20]. A product contains detailed information that would reflects the one on its packaging, including nutritional information, several images and very often even the country of origin per ingredient or the product as a whole. Reflecting its Swiss background, every product is available in French, German and Italian.

1.4.3 Publicly Available Textual Datasets Summary

Dataset	Year	# Recipes	# Images	# Ingredients	Format
Ahn Flavours	2011	56 K, only ingr.	No	1384	5* preprocessed .tsv
Recipe1M	2017	1.029 M	887 K	16 K	2* preprocessed/1* raw .json; .jpg
Yummly66K	2017	66 K	66 K	2416	10* .json files, 9* preprocessed
RecipeQA	2018	1.9 K, 36K Q&A	25 K	Yes	1963* raw .json and .jpg
Meal Master	2013	158 K	-	Yes	.mmf, i.e. structured but raw .txt
RecipeRescuer	2016	93 K	>160 K	Yes	1* somewhat parsed .json
FoodRepo.org	2018	-	37 K	37 K products	37 K .json files via convenient API
BBC.co.uk Food	2016	74 K	74 K	Yes	folders with raw .html and .jpg
FoodOn Ontology	2018	-	-	9 K products	A single .owl file
FooDB	2018	-	Many	722	20* raw/preprocessed .csv or SQL
Wiki Food/Bev.	2018	Many Hundreds	Many	Sometimes	raw .xml/SQL
Wiki Books	2016	2.6 K	Many	Yes	raw .xml/SQL

Table 1.1: Publicly available Textual Datasets Summary

The table shows an overview over the most important available text datasets with the main features being the year of release or last update, the number of recipes and images as well as ingredients. The last column "format" summarizes the file structure of the dataset and aims aims to answer the practical consideration of how easily the dataset can be applied to actual distributional semantics of food. For more details please consult the subsections above.

Dataset	Textual Metadata	Data/Metadata Origin
Ahn Flavours	Ingr.-to-type, ingr.-to-chemical-flavor-compounds, ingr.-list-to-cuisine	Fenaroli's Handbook of Flavor Ingredients; 3 popular cooking websites
Recipe1M	Recipe: title and instructions	<24 popular cooking websites
Yummly66K	Recipe: title, instructions, cuisine, flavors (some missing), course	yummly66K cooking website; the website computed flavors itself
RecipeQA	Recipe: title, detailed step-by-step instructions with image link	instructables.com website
Meal Master	Recipe: course, occasion, cuisine, diet, techniques and more	Anonymous users or commercial entities since at least 1998
RecipeRescuer	website address each recipe was crawled from	Many different cooking websites; cleaned version of open recipes
FoodRepo.org	Product: origin, energy, fat, carbohydr., sugars, fiber, protein, salt	Coop, Migros, Lidl (.de, .fr, .ita)
BBC.co.uk Food	Recipe: course, occasion, cuisine, diet, dish, season, chef	BBC, also from TV shows
FoodOn Ontology	Product: very detailed taxonomy	International group of researchers funded in part by Canada
FooDB	Ingr.: flavor and other chem. compounds, nutrients, descriptions, types	Extensive and ongoing literature research funded by Canada
Wiki Food/Bev.	Recipe: foods, beverages, techniques, dining, etc.; dishes by ingredient	Wikipedia Authors
Wiki Books	Recipe: cuisine, diet, dish, technique, food group, price	Wikibooks Authors

Table 1.2: Publicly available Textual Datasets Metadata

This table is about the metadata of the datasets. As it is these categories that can potentially bring the most interesting insights they are listed in detail, while the last column hints at their source and credibility.

1.5 Summary of Distributional Food Semantics Research and Knowledge Gaps

There has been already quite a lot of research in the area of distributional semantics of food. 2018 has been an especially active year, with major overview works such as Min et al. 2018b and the largest publicly available multimodal data set available yet (Marin et al. 2018), as well as with interesting visions of how the field could advance towards the future benefiting the society and individuals alike by translating the meaning of food across cultures (Kazama et al. 2018; Nobumoto et al. 2017)

[20] kgjf.org

Still there are knowledge gaps and opportunities to contribute to solving them. More experiments of how word and subword level representations can be applied to food corpora should be conducted. The knowledge gap of where to get the linguistic food resources needed has been alleviated by the datasets summary presentation before. As the popular metrics like ”BLEU” and ”METEOR” are hardly able to satisfyingly measure adequacy of meaning of p.e. recipes (Kiddon et al. 2016, cf.), it would be valuable to find more suitable automated measures to compare different implementations. The Benchmarking methodology thus appears to be improvable, which may be connected to the available data issue. Many times, assessing the quality of very interesting embeddings seems to still heavily rely on manual evaluation schemes p.e. by professionally assigning many laypersons (Kazama et al. 2018) or assigning an expert (Marin et al. 2018), to mention just two of many presented works.

Finding better automated benchmarks would make comparing embeddings much easier. It would p.e. be interesting to see if general domain embeddings indeed generally perform much worse than embeddings learned on food corpora, as reading Wiegand et al. 2012b, p. 2 might suggest: ”In the past, most research in NLP has been carried out on news corpora [15]. The topic that is predominant on this text type are political affairs rather than food-related issues. Consequently, this text type would be of little value for knowledge extraction of food relations.”. Use-cases like recommendations might profit from a different approach than the current, where using previous ratings of users might be affirming old preferences instead of creating new favorites.

Visualizations should be discussed more with regards to usefulness for comparing different implementations of distributional semantics. They can also help set food text embeddings into context and present the knowledge contained within in a more intuitive and accessible way.

There has been a lot of work using sparse vectors in the food domain, yet more complex models such as topic modeling are not as widely used, which makes more showcases desirable. Thus exploring a food corpus to present topic models, giving a practical approach of how to choose a good range for the number of food topics using the various metrics, would be another interesting and meaningful task.

1.6 Contributions

In this thesis a number of contributions are made to the computational analysis of food using distributional semantics:

- helping define the field of research
- review of current literature
- summary of publicly available textual datasets
- creating food embeddings based on the 1 million recipes dataset
- creating cuisine and ingredient prediction benchmarks and applying the created food embeddings as well as general domain vectors
- evaluation and analysis of classification results
- benchmarking and evaluating number of topics
- exploring a food topic model showcasing the ”tmtoolkit” library
- creating t-SNE visualizations of the various embeddings for a more intuitive understanding
- analysis and evaluation of the t-SNE visualizations
- giving an outlook for future directions

1.7 Thesis Structure

The materials and approach used for the distributional food semantics benchmarks, the food topic discovery and the visualizations will be presented in the next chapter. This includes mentioning the most important programming libraries employed and how visualizing the data will work.

First the main embedding techniques used, ”Word2Vec” and ”fastText” will be explored mathematically. Then their application to the 1 million recipes dataset will be shown, which includes describing how the existing ”im2recipe” algorithm was modified to yield a word-level embedding. Then the general domain embeddings from the corpora ”Google News” and ”Wikipedia” will be presented.

After the food representations, the datasets to use for creating benchmarks and topic discovery will be explored. The best options will be evaluated and the most apt corpus according to the prediction of categorical variables will be chosen. Statistics will be printed to help investigate the characteristics of the datasets.

Then the computational models used for benchmark creation, topic modeling as well as visualization will briefly mathematically be explored. The multi-class prediction classifiers include linear classification as well as logistic regression. Topic modeling means the use of the Latent Dirichlet Allocation (LDA) method, which will be described. The relatively modern ”t-Distributed Stochastic Neighbor Embedding” (t-SNE) dimensionality reduction technique will be described and compared to older techniques.

After this theoretical background of the computational models, the concrete applications will be described. This is the ingredient type classification using linear classification, the cuisine classification using linear classification and logistic regression, cuisine topic discovery and visualization of the datasets and embeddings. Parameters used will be explained and the general approach presented in detail, explaining the reasons for choosing certain libraries such as tmtoolkit”.

The results will be shown with minimal comments in the next chapter, putting some results that could distract the reader into the appendices. A comprehensive discussion follows in ensuing chapter, ordering them to coherent sections and evaluating them in detail while comparing them to existing research results where applicable. Four main results will be discussed: the ingredient type prediction and visualization, the cuisine prediction, the topic discovery and the cuisine visualization. This includes explaining what results are surprising, which are not, and how they can be interpreted with what arguments. Arguments as to why a certain model instance is decided to be the best will be presented. The most interesting finds will be highlighted.

Finally conclusions will be drawn, giving an outlook on how to continue and advance the ideas of this thesis.

Chapter 2

Materials and Approach

2.1 Food Representations

2.1.1 Embedding Methods

The ”Word2Vec” embedding (Mikolov et al. 2013) will be applied to food texts and thus briefly described. It is a feed-forward neural network architecture with input, hidden and output layers. That means that it is a probabilistic non-linear classifier with ”N” one-hot encoded input words (only one of the ”V” dimensions is 1, where ”V” is the vocabulary size) (ibid., p. 3). Each word is embedded in a real valued vector, usually with a much smaller number of dimensions than the size of the vocabulary; the classifier is a product of conditional input-output probabilities (Bengio et al. 2003, cf. p. 1139). The parameters of that joint probability function of word sequences are iteratively trained according to the maximum logarithmic likelihood via stochastic gradient descent and backpropagation methods (Goldberg 2015, p. 5 and 13) - those parameters are at the same time the word feature vectors that define the embedding (Bengio et al. 2003, cf. p. 1139).

A great example of a feed-forward neural network function, also called ”Multi Layer Perceptron” (MLP) is given by (Goldberg 2015, p. 12):

$$\begin{aligned} NN_{MLP1}(\mathbf{x}) = g(\mathbf{x}\mathbf{W^1} + \mathbf{b^1})\mathbf{W^2} + \mathbf{b^2} \\ \mathbf{x} \in \mathbb{R}^{d_{in}}, \\ \mathbf{W^1} \in \mathbb{R}^{d_{in} \times d_1}, \quad \mathbf{b^1} \in \mathbb{R}^{d_1}, \\ \mathbf{W^2} \in \mathbb{R}^{d_1 \times d_2}, \quad \mathbf{b^2} \in \mathbb{R}^{d_2} \end{aligned} \tag{2.1}$$

$\mathbf{x}$ is the input, $\mathbf{W^1}$ and $\mathbf{b^1}$ are the weight matrix and bias to be trained that represent the first linear transformation of the input. The g is the non-linear so called *activation function*), crucially important for representing complex word relations, with $\mathbf{W^2}$ and $\mathbf{b^2}$ as matrix and bias weights of the second linear transformation that yields the d_2 dimensional output vector (ibid., cf. p. 12).

Here a 2-layer feed-forward network example with additional linear and non-linear transformations by (ibid., p. 12):

$$NN_{MLP2}(\mathbf{x}) = (g^2(g^1(\mathbf{x}\mathbf{W^1} + \mathbf{b^1})\mathbf{W^2} + \mathbf{b^2}))\mathbf{W^3} \tag{2.2}$$

Using intermediary variables, the aforementioned input, hidden and output layers are more visible (ibid., p. 13):

$$\begin{aligned} NN_{MLP2}(\mathbf{x}) = \mathbf{y} \\ \mathbf{h^1} = g^1(\mathbf{x}\mathbf{W^1} + \mathbf{b^1}) \\ \mathbf{h^2} = g^2(\mathbf{h^1}\mathbf{W^2} + \mathbf{b^2}) \\ \mathbf{y} = \mathbf{h^2}\mathbf{W^3} \end{aligned} \tag{2.3}$$

$\mathbf{y}$ is the output layer, $\mathbf{h^1}$ and $\mathbf{h^2}$ are the two so called hidden layers.

A visualization can be seen in (ibid., fig. 2):

The circles are so called neurons of the network. They are a metaphor for a computational unit, or in other words a function with input and output, whose weights or variables are to

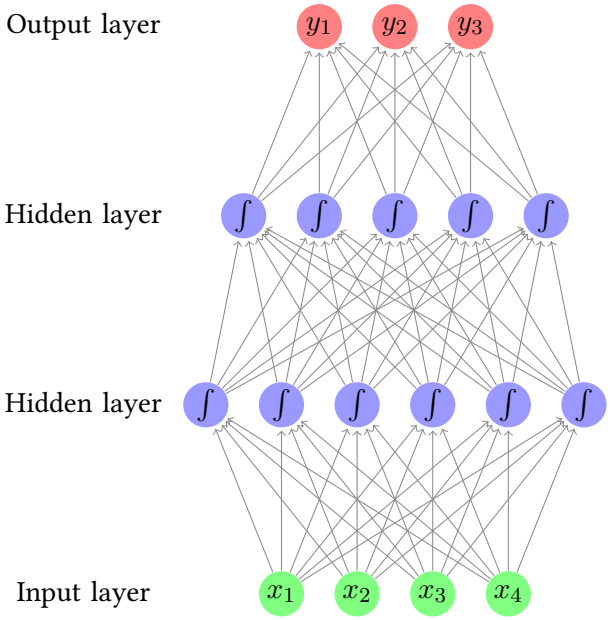

FIGURE 2.1: Feed-forward neural network with two hidden layers.

be trained by the network (Goldberg 2015, cf. p. 11). The sigma symbols inside the hidden layers represent the non-linear "sigmoid" function, characteristically "S"-shaped as can be seen in the figure created by Qef 2008:

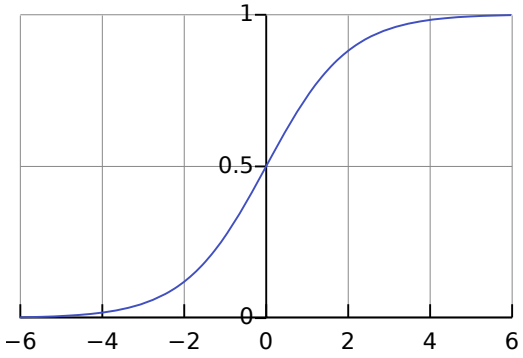

FIGURE 2.2: Sigmoid Curve by Qef 2008.

The dense word-level vectors of the "Word2Vec" system are produced by two main architectural parts called "continuous bag-of-words" (CBOW) and "Skip-gram" (Mikolov et al. 2013, p. 5). CBOW is about predicting the current word based on the context, p.e. in the context of recipes the CBOW input could be four ingredient or instruction words before and after an ingredient, while the prediction would be about the ingredient in the center of these two contexts; the Skip-gram" part in this recipe example would predict a certain number of the surrounding words given a recipe word (ibid., cf. p. 4-5). CBOW is called continuous, as the vectors are an array of real numbers, in contrast to the standard bag-of-words model being an array of count frequencies. The name of the also continuous "Skip-gram" model is due to its relation to the "n"-gram language models, that assign probabilities to a "n" tokens long sequence of words (Daniel Jurafsky et al. 2018, p. 38)

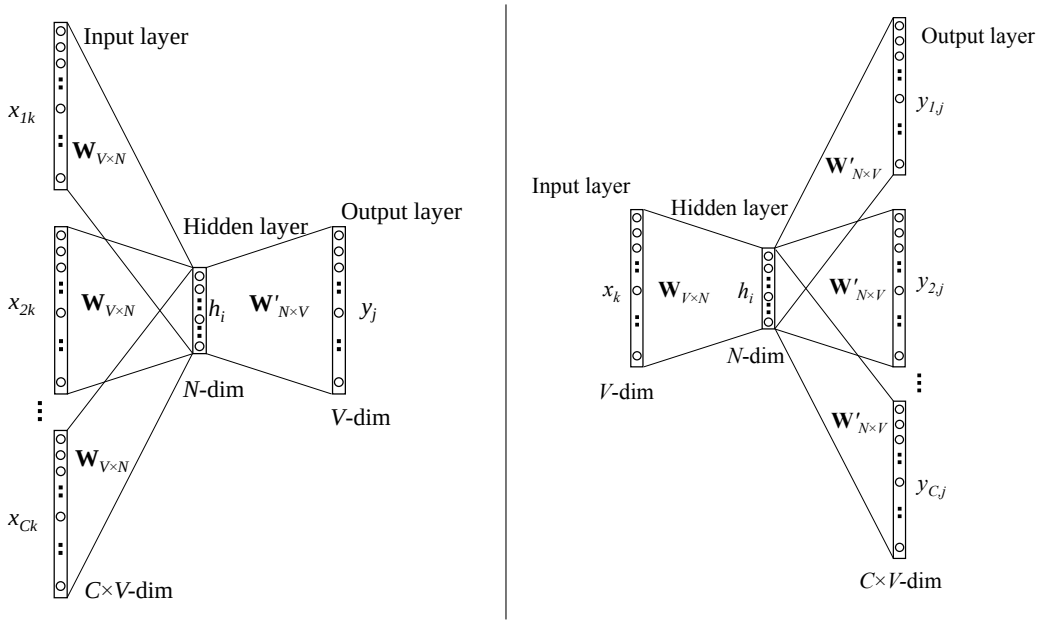

FIGURE 2.3: CBOW and "Skip-gram" model, taken from Rong 2014.

In the case of "Skip-gram", given a set D of all word and context pairs extracted from the text, here the formula to maximize the probabilities $p(c|w)$ (probability of the context word c given word w) by adjusting the parameters θ of $p(c|w;\theta)$ (Goldberg et al. 2014, p. 1):

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(c|w;\theta) \tag{2.4}$$

The conditional probability $p(c|w;\theta)$ can be parameterized as follows (ibid., cf. p. 2):

$$p(c|w;\theta) = \frac{e^{v_c \cdot v_w}}{\sum_{c'\in C} e^{v_{c'} \cdot v_w}} \tag{2.5}$$

with the vector representations v_c and $v_w \in R^d$ of v_c and $v_w \in R^d$, and with the set of all available contexts C.
The last layer consists p.e. of the "soft-max" function, i.e. the log-likelihood maximization objective, predicting only one context word c:

$$\arg\max_{\theta} \sum_{(w,c)\in D} \log p(c|w) = \sum_{(w,c)\in D} (\log e^{v_c \cdot v_w} - \log \sum_{c'} e^{v_{c'} \cdot v_w}) \tag{2.6}$$

The loss function for CBOW can be described like this (Rong 2014, p. 6):

$$E = -\log p(w_O|w_{I,1}, \cdots, w_{I,C}) \tag{2.7}$$

More explanations can be read in ibid., who also gives a great visualization of the "Word2Vec" embedding, available online[1]:

[1] https://ronxin.github.io/wevi/

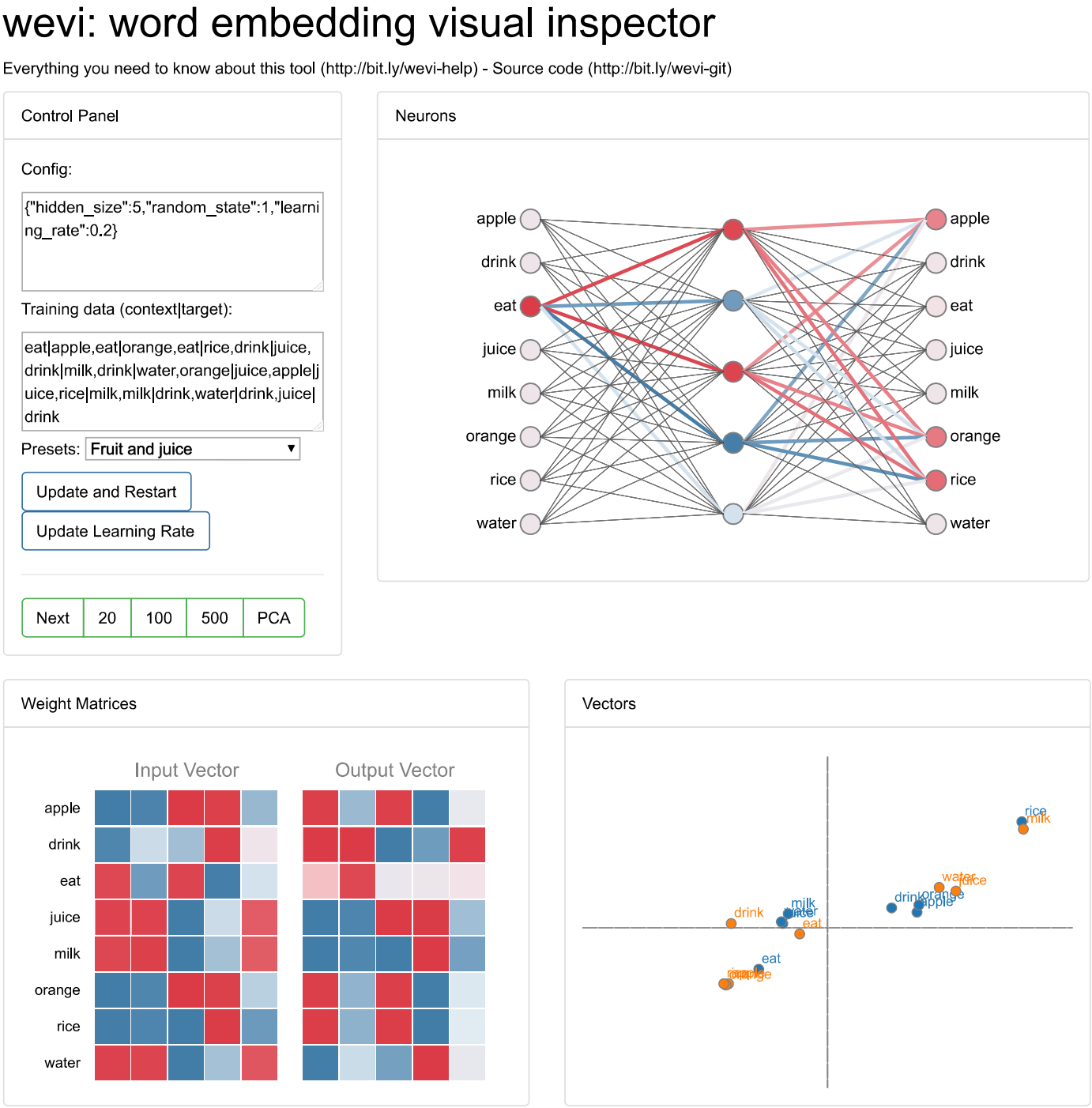

FIGURE 2.4: Word2Vec model visualized by Rong 2014's "wevi-online".

Another useful resource fore intuitive explanations around NLP word embeddings and Neural Networks linking them p.e. to functional programming can be found in the blog of Christopher Olah 2019.

The free open-source "fastText" library[2] is based on a similar idea as "Word2Vec", but with the important difference that it is a subword-level embedding. Basing on an idea of Schütze 1993, words are represented by a bag of character n-grams of size G (Bojanowski et al. 2016,

[2] fasttext.cc

p. 4). Denoting $\mathcal{G}_w \subset \{1, \ldots, G\}$ as the set of n-grams appearing in a given word w, each n-gram g gets assigned a vector representation $\mathbf{z}_g$, while the word embedding equals the sum of the n-gram vectors (ibid., p. 4). This yields the following loss function to be optimized as the embeddings get trained:

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c. \tag{2.8}$$

Though simple, this model allows shared embeddings across words; by doing this, even rare and unknown words but with known n-grams can be embedded (ibid., p. 4). These embeddings require less corpus data to train compared to a "Word2Vec" baseline while delivering very usable unknown word embeddings ((ibid., cf. p. 7)). For memory efficiency, n-grams are hashed, which means a word is represented by its set of hashed n-grams (ibid., p. 4).

2.1.2 Food Domain Representations

In order to benchmark word embeddings that are representative for the food domain, excellent existing embeddings or a linguistic resource as large as possible was necessary. The 1 Million Recipes work by Marin et al. 2018 satisfied both requirements, in that it brings several word vectors as well as the largest collection of recipes in a consistent and easy to process manner.

As the "im2recipe" algorithm requires Word2Vec preprocessing of the ingredients and instructions, the correspondent 300 dimensional Word2Vec embedding was ready to be used in the repository online, consisting of all 30.167 instruction words. This vocabulary would en passant include all the 18.252 ingredient words, with the code already in place to extract these unique ingredients from the separate list of recipes with measurements and comments removed ("det_ingrs.json") that was preprocessed by an algorithm unfortunately not in the repositories as mentioned in Chapter (1.3.1). The resulting embedding will be called "im2rec_base" in this thesis, serving as a baseline for comparisons between embeddings based on this corpus.

The "im2recipe" vectors on the other hand, being joint recipe and image vectors with only an retrieval and input image/recipe pipeline supplied in the repositories by the original authors, could not directly be used as word vectors. The "test.py" script offers a way to input a list of joint multimodal recipe embeddings and get the output embeddings in order to compare them. The input is a multimodal trio embedding: minimal ingredient lists, with each ingredient represented by its Word2Vec vector, instructions represented by its skip-thoughts embedding as well as images represented by its image embedding. In order to get a word level representation of the embedding weights learned in the "im2recipe" task, the existing pipeline to save embeddings from these weights was therefore modified.

The "test.py" script and the "im2recipe" model defining "trijoint_ingredient.py" with its ingredients and instructions LSTMs (Hochreiter et al. 1997) were modified to accept recipes that contain only one word as input, i.e. inputting as many recipes as there are words in the Word2Vec vocabulary. Image embeddings would not be computed, while the instructions that are concatenated with the ingredients vectors by the "im2recipe" model were handled in two ways: One embedding cut the instructions part all together, setting its embedding to a 0 vector, calling it "im2rec_joint_null". The other embedding computed the average of all recipe instructions available in the original embedding, calling it "im2rec_join_avg". Both embedding have the same size of 1024 dimensions.

As a side note, the original corpus was left as is, as the available code is depending on extensive recipe preprocessing such as the removal of measurements. Both implementations, but the earlier Torch7 version (see Chapter 1.4.1) in particular - which contains the instructions transformation via skip-thoughts - rely on old libraries which may conflict with current Python and "CUDA" libraries as well as newer Linux Kernels.

The embeddings mentioned before are on word level, which has the disadvantage that applying it to another corpus works best if this other corpus contains exactly the same words.

This is possibly especially problematic with regards to ingredients, where usually each word has a lot of importance to meaning but there may be many different variations of them, including rare ones. It could p.e. deteriorate classification performance if the embedding's vocabulary contained the words "green_olives", "olive_oil", but not the ingredient from the corpus used for classification named "green_olive_pesto".

Thus, to compare how big of an issue this would actually represent in a benchmark application, a "fastText" embedding on the same 1 million recipes corpus would be trained. It had already been successfully applied on food datasets by **{sauer_cooking_2017}** and Altosaar 2017a.

The default "n"-grams size 3 to 6 was used[3] together with 300 dimensions, because this would compare nicely against the equally dimensioned "im2rec_base". Training took less than an hour thanks to a powerful AMD 2700X processor with 16 threads, 32GB RAM and NVME SSD. It was named "im2rec_fasttext".

Additionally a simple TF-IDF representation was produced using the "scikit-learn" library function "TfidfVectorizer" with "l1" norm[4], using no further preprocessing and tokenization as the dataset is already preprocessed. This embedding was dubbed "tfidf" and evaluated separately.

2.1.3 General Domain Embeddings

With the goal of comparing the four embeddings based on a food domain corpus, general domain corpus based embeddings would have to be used as well. The "Word2Vec" Google News embedding from[5] seemed like a good candidate to see if the in Chapter 1 mentioned doubts of Wiegand et al. 2012b, p. 2 of aptitude for food domain tasks would be justified.

The original Word2Vec algorithm used the Google News corpus of 6 billion single tokens (p.e. "New York" represent two tokens), restricting vocabulary to the 1 million most frequent words Mikolov et al. 2013, p. 6. The 300 dimensional embedding for around 3 million words and phrases was named "googlenews" for further purposes.

To have a corresponding general domain subword-level embedding to "im2rec_fasttext", the 300 dimensional "fastText" vector trained on a 2016 dump of Wikipedia (Bojanowski et al. 2016, p. 4) was chosen. It was downloaded[6] and named "wiki_fasttext".

2.2 Datasets

2.2.1 Ingredients Dataset

The aim was to evaluate the embeddings a different corpus with interesting categorical information. The 1 million recipes corpus does not contain interesting labels as described in Chapter 1.4.1 and could induce a bias as the embeddings were already trained on it. A couple of options were considered: The "Open Recipes" has an impressive number of recipes, but lacks interesting categories and would need a lot of preprocessing of the ingredients to make matches with the vocabularies of the embeddings. The "Meal Master" data not only has even more recipes, but a variety of interesting metadata that could serve for classifying. The main downside would be the considerable amount of preprocessing of the .mms (essentially structured .txt files) with regards to the ingredients, like removing measurements and comments. "Yummly66K" has interesting Metadata like course and flavor information, plus it was well preprocessed, except for a central cuisine, "American". The "Ahn" dataset offers ingredient level metadata as well as a respectably sized corpus of recipes. They are simple, but consistent and well preprocessed, making application of an embeddings vocabulary easy.

[3] fasttext.cc/docs/en/options.html
[4] scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[5] code.google.com/archive/p/Word2Vec/
[6] github.com/facebookresearch/fastText

It was decided to keep the benchmark setup as simple as possible, leaving out additional complexity such as extensive preprocessing using p.e. an ingredient phrase tagger (github.com/NYTimes/ingredient-phrase-tagger appears to be a good implementation of such systems). This left the ”Yummly66K” and ”Ahn” corpus as choices.

As one of the largest sources of supply and demand for online recipe resources is US-American based, it seemed preferable to use the ”Ahn” corpus which offers many well cleaned recipes labeled ”NorthAmerican” where every ingredient is just one token, where ”Yummly66K” lacks the preprocessing steps like removal of measurements. Beyond its appealing simplicity, the ”Ahn” dataset furthermore has the advantage of offering ingredient-level annotations, something that is a unique and interesting information.

For these reasons, the ”Ahn” ingredient type annotations as well as the ”Ahn” recipes with cuisine region labels were chosen to be used. The datasets are hence called the ”ingredients dataset” and ”cuisine dataset”.

The ingredients dataset is a simple .csv file containing 1530 ingredients. Consisting of 14 different types, the dataset is highly skewed towards plants and plant derivatives:

	Ingredient Type	Number of occurrences	Rounded relative Size
1	plant_derivative	424	27.7%
2	plant	313	20.4%
3	fruit	186	12.1%
4	vegetable	104	6.7%
5	herb	90	5.8%
6	flower	66	4.3%
7	meat	57	3.7%
8	fish/seafood	56	3.6%
9	spice	55	3.5%
10	alcoholic_beverage	50	3.2%
11	dairy	39	2.5%
12	cereal/crop	39	2.5%
13	nut/seed/pulse	33	2.1%
14	animal	18	1.1%
	Total number of labeled ingredients	1530	100%

Table 2.1: The 14 ingredient type occurrences.

These two classes alone make almost half of the data, while ”animals” accounts for only 1.1% and many other classes being only 2-5% large. The different types are very fine grained categories, p.e. ”plant_derivative” ingredients often times representing the oil of ”plant” ingredients.

If this data skew would make learning predictions of the right ingredient type for the rarer classes using the embeddings described before a challenging or plain impossible task, would be interesting to find out in the results section.

2.2.2 Cuisine Dataset

Similarly to the ingredients dataset, the cuisine dataset is highly skewed:

	Cuisine label	Number of ingredient lists	Relative Size
1	NorthAmerican	41524	73.4%
2	SouthernEuropean	4180	7.3%
3	LatinAmerican	2917	5.1%
4	WesternEuropean	2659	4.7%
5	EastAsian	2512	4.4%
6	MiddleEastern	645	1.1%
7	SouthAsian	621	1.0%
8	SoutheastAsian	457	0.8%
9	EasternEuropean	381	0.6%
10	African	352	0.6%
11	NorthernEuropean	250	0.4%
	Total number of labeled ingredient lists	56498	100%
	with a unique vocabulary size of	381	

Table 2.2: The 11 cuisine occurrences.

The most common cuisine, "NorthAmerican", makes almost three quarts of the dataset, while the next largest cuisine, "SouthernEuropean", makes 7.3% and the smallest six cuisines, "MiddleEastern", "SouthAsian", "SoutheastAsian", "EasternEuropean", "African" and "NorthernEuropean" together make just 5.2%.

De Clercq 2014, p. 18-19 suggest four groups that clearly feature very similar recipes, measured by hierarchical clustering of a binary embedding. The co-occurrence investigation by Ahn et al. 2011, figure 4, although not covering all cuisines, does to a degree affirm the results of the dendrogram by De Clerq.

As initial training in the draft stages predicting all cuisines took at the very minimum many hours while having an overwhelming bias towards the largest class, "NorthAmerican", it was decided to merge cuisines as suggested (De Clercq 2014, p. 19):

	Merged Cuisine	Recipes	Rel. Size	Old Cuisine Labels
1	Western	44814	79.3%	NorthAm., WesternEur., NorthernEur., EasternEur.
2	Southern	8094	14.3%	African, LatinAmerican, MiddleEastern, Southern
3	Eastern	2969	5.2%	EastAsian, SoutheastAsian
4	SouthAsian	621	1%	SouthAsian
	Ing. lists	56498	100%	

Table 2.3: The four merged cuisine occurrences.

"Western" makes for more than three quarts of the data now, while the smallest class, "SouthAsian" is only 1% large.

The other two cuisines, "Southern" and "Eastern" are representing around 1/7 and 1/20 of the data. The dataset still exhibits a large skew, but now with far fewer but more distinct classes.

"SouthAsian" was not merged, despite its small size, hinting at the use of the most unique ingredients compared to the original cuisine labels. The other three classes, "Western", "Southern" and "Eastern" consist of at least two of the old cuisines as they had very similar ingredients in their recipes. This way, the classification will still offer an interesting and challenging task, but will be more manageable and with less sample size bias.

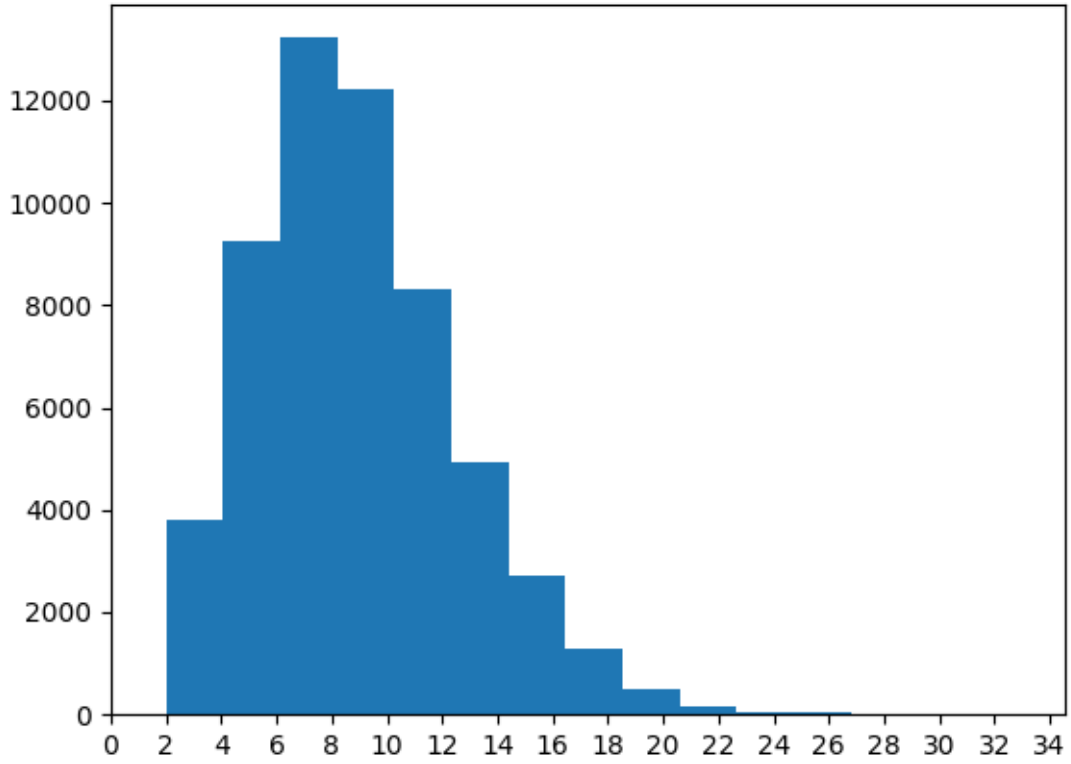

FIGURE 2.5: Recipe Length Histogram.

The histogram of the length of each recipe shows a peak of around 7 ingredients per recipe, with a considerable amount of recipes, around 10% has only 2-4 ingredients and around 15% have 12 or more ingredients, up to 26.

It was decided to reuse the dataset from before for the topic modeling task, as there was already a form of familiarity with it and it would comparing visualizations easier. Initial trials using topic modeling on the original cuisine dataset did perform with seemingly reasonable results in a very acceptable amount of training time of less than one hour, so all 10 instead of the 4 merged cuisine regions were to be used. The cuisine label of each recipe is added as an additional "ingredient" as the topic modeling input. As described in the previous chapter, a varying number of topics should be metered, to have a better basis for a topic model decision.

2.3 Computational Models

2.3.1 Classification Models

The two classification techniques applied will be linear classification (LC) and logistic regression (LR). As explained by Yuan et al. 2012, cf. p. 2:

"given training data

$$(y_i, \mathbf{x_i}) \in \{-1, +1\} \times \mathbb{R}^n, i = 1, \ldots, l$$

where y_i is the label and $\mathbf{x_i}$ is the feature vector" (word vector), for both algorithms a decision function d can be constructed:

$$d(\mathbf{x}) \equiv [w]^T \phi(\mathbf{x}) + b \tag{2.9}$$

$\mathbf{w}$ stands for the the trainable weight, b the bias.

Non-linear classifiers such as logistic regression or support vector machines (SVM) solve classification problems by mapping each feature instance $\mathbf{x}$, p.e. an ingredient word, via a non-linear function to a higher dimensional feature space $\phi(\mathbf{x})$ if the data is not linearly separable, where a linear surface (hyperplane) can be constructed that separates the classes, p.e. cuisines Yuan et al. 2012, cf. p. 2; Cortes et al. 1995, cf. p. 1. Due to potentially very large dimensions, training can be expensive compared to linear classifiers, but generally with better prediction accuracy Yuan et al. 2012, cf. p. 2. On the other hand linear classification tends to be faster to train and test than for large-scale applications, as they don't map data points, i.e.:

$$\phi(\mathbf{x}) = \mathbf{x}. \tag{2.10}$$

Still linear classifiers perform comparably in the context of documents ibid., p. 2.

The optimization problem of linear classifiers is as follows ibid., cf. p. 3:

$$\min_{\mathbf{w},b} f(\mathbf{w},b) \equiv r(\mathbf{w}) + C\sum_{i=1}^{1} \xi_{LC}(\mathbf{w},b;\mathbf{x_i},y_i) \tag{2.11}$$

where $r(\mathbf{w})$ the regularization parameter, $\xi(\mathbf{w},b,\mathbf{x_i},y_i)$ the loss function for a wrongly-classified observation $(\mathbf{x},y)$ and the balancing parameter C for the regularization and summed loss that emphasizes classification errors according to its value. Here the simplified version without bias:

$$\min_{\mathbf{w}} f(\mathbf{w}) \equiv r(\mathbf{w}) + C\sum_{i=1}^{1} \xi_{LC}(\mathbf{w};\mathbf{x_i},y_i) \tag{2.12}$$

The simplified loss function for the per definition non-linear logistic regression classification is as follows (ibid., p. 3):

$$\xi_{LR}(\mathbf{w};\mathbf{x};y) \equiv log(1+e^{-y\mathbf{w}^T\mathbf{x}}) \tag{2.13}$$

Logistic regression is geometrically similar to maximizing the posterior probability with a Laplacian or Gaussian prior of $\mathbf{w}$ from a Bayesian perspective (ibid., p. 4).
The classifiers can solve "primal" or "dual" problems in the sense of Lagrangian duality. The linear classification optimization mentioned before can be more easily solved in situations where the number of instances is much smaller than the number of features (ibid., p. 5). This option can often be set via a parameter in programming functions, p.e. in scikit-learn it is named "dual"[7].

2.3.2 Topic Models

The topic model discovery is based on the "Latent Dirichlet Allocation" (LDA) of Blei et al. 2003, p. 996. It is also a recipe embedding, as recipes are embedded by the topics as the number of dimensions. As describe ibid., p. 996, "LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D:"

1. Choose a sequence of N words $\sim Poisson(\xi)$.
2. Choose $\theta \sim Dirichlet(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim Multinomial(\theta)$.
   (b) Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

A graphical model representation of the LDA model is given by ibid., p. 997 as follows:

[7] scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

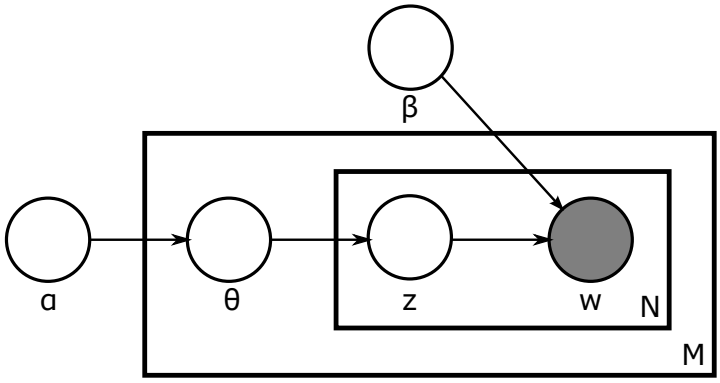

FIGURE 2.6: LDA graphical model Bkkbrad 2008.

The outer rectangle represents documents, the inner rectangle the repeated choice of topics and words within a document. α and β are estimated using Bayesian statistics (ibid., p. 997). "With scientific documents, a large value of β will lead the model to find a relatively small number of topics, perhaps at the level of scientific disciplines, whereas smaller values of β will produce more topics that address specific areas of research.", as explain Griffiths et al. 2004, p. 5231.

2.3.3 Visualization Models

The goal was to produce interesting and easily interpretable graphical visualizations that would help in understanding the datasets used, but also the benchmark results. "t-Distributed Stochastic Neighbor Embedding" (t-SNE) seemed like a good choice due to its recent uses in the food domain described in chapter 1.

It is based on the "Stochastic Neighbor Embedding" (SNE) by Hinton et al. 2003 but uses a modified cost function that is symmetrized and simpler to train, using a heavily tailed "Student-t distribution" instead of a "Gaussian" in order to alleviate crowding problems, computing similarity in a lower dimensional space (Maaten et al. 2008, p. 2583). The t-SNE system converts input vectors with a high number of dimensions into p.e. two or three dimensions, while preserving as much "of the significant structure of the high-dimensional data as possible" (ibid., p. 2580).

Different to traditional dimensionality reduction techniques such as "Principal Components Analysis" (PCA) (Hotelling 1933), t-SNE is non-linear.

t-SNE was developed as these techniques were unsatisfactory in visualizing high-dimensional data (Maaten et al. 2008, cf. p. 2580) - t-SNE in contrast can retain the local structure of the data while revealing structures such as clusters (ibid., cf. p. 2599).

Conditional probabilities in the higher and lower dimension p_{ij} and q_{ij} are jointly minimized using Kullback-Leibler divergence (ibid., p. 2583). The cost function of symmetric SNE looks like this (ibid., p. 2583):

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{2.14}$$

Pairwise similarities in the low-dimensional map q_{ij} are computed like this (ibid., p. 2584):

$$q_{ij} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} exp(-\|y_i - y_j\|^2)\prime} \tag{2.15}$$

Pairwise similarities in the high-dimensional space p_{ij} are elegantly computed via the symmetrized conditional probabilities in order to avoid problems with outliers:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2.16}$$

The perplexity parameter influences the effective number of neighbors, with typical numbers between 5 and 50 (Maaten et al. 2008, cf. p. 2582). For great examples examining various values for the perplexity see[8].

2.4 Ingredient Type Classification

The Python library "scikit-learn" was used for the majority of the implementation as it is simple yet efficient and covers a variety of machine learning techniques (scikit-learn.org).

The data was first split into a training and a separate test set using 10-fold cross-validation ("StratifiedKFold"-function). The ingredient type distribution was thus approximately kept in both sets.

Then a training pipeline was created using the "LinearSVC" function, a supervised linear support vector classifier implemented in the "liblinear" library. It is supposed to scale better to large number of samples (Fan et al. 2008, p. 1871) than the "SVC" function with parameter kernel=linear that uses "libsvm" (csie.ntu.edu.tw/c̃jlin/libsvm/), see[9]. Almost 60.000 recipes with hundreds of embedding dimensions and much less ingredients but more ingredient types with the similar embedding sizes are assumed by me to be relatively large, thus opting for the "LinearSVC" function.

A hyperparameter grid search ("GridSearchCV") using 10-fold cross-validation was used to find the best parameters, optimizing the loss for macro-F1. The grid search enables parallel computation of several model candidates at once using all 16 available cores of the computers at hand via the "n_jobs" parameter[10]. It applies 10-fold cross-validation to evaluate the models in order to avoid overfitting and make the classification results more generalizable. This means separating the training data into 10 different parts with approximately the same class balance, training a model candidate on 9 parts of it and evaluate on the spare 10th set, repeating this process 10 times (so that each 1/10 was once the evaluation set). A good illustration is given by the Institute for Genomics and Bioinformatics - Graz University of Technology et al. 2005:

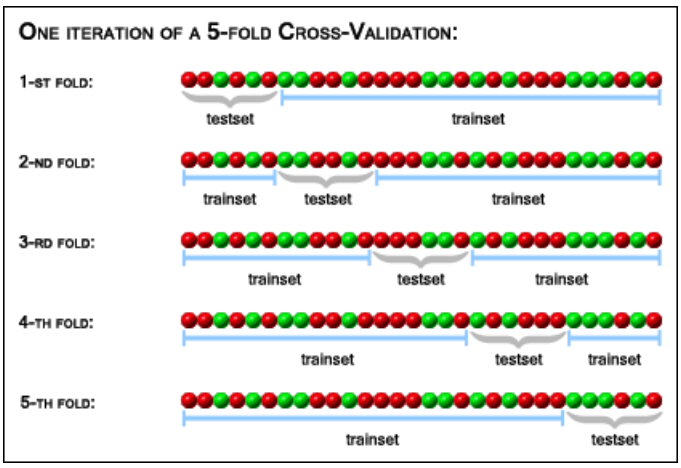

Figure 2.7: Cross-validation illustration (Institute for Genomics and Bioinformatics - Graz University of Technology et al. 2005).

The hyperparameters were tuned using a range of values in particular for the aforementioned classification error balancing parameter "C", as well as a "True" and "False" value for the "dual" parameter.

A large grid of possible values was chosen in order to find the best possible parameters, with 3.000 iterations as training was relatively fast:

[8] distill.pub/2016/misread-tsne/

[9] scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html and scikit-learn.org/stable/modules/svm.html#svm-classification

[10] scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

- 'C': [1000, 500, 200, 100, 50, 20, 10, 5, 2, 1, 0.2, 0.5, 0.01, 0.02, 0.05, 0.001]
- 'penalty': ['l2'] # Default
- 'dual': [False, True] # Default=False
- 'loss': ['squared_hinge'] # Default
- 'max_iter': [3000] # Default=1000
- 'tol': [1e-06, 1e-04] # Default=1e-4
- 'class_weight': ['balanced', None] # Default=balanced
- 'multi_class': ['ovr', 'crammer_singer'] # Default=ovr
- 'random_state': [0]

"ovr" and "crammer_singer" determine the strategy for solving the multi-class prediction problem [11]).

"ovr" trains one-vs-rest classifiers: With k classes in the training data, it constructs k binary classifiers that are trained against all other classifiers and assigns an instance $\mathbf{x}$ the class with the highest decision value (Yuan et al. 2012, cf. p. 11). "crammer_singer" on the other hand directly predicts the class with only one trained classifier, using the "dual" method (Crammer et al. 2002), but is more expensive to compute[12].

"classweight"="balanced" adjusts the "C" parameter by weighting it inversely to the class frequency[13]. "penalty" specifies the loss penalization norm. "loss" specifies the loss function, here the square of the hinge loss. "tol" is the tolerance value for training stopping criteria - if the training does not show better improvements than the value specified over a certain number of iterations, the training will stop before the maximum iterations are run.

Training would be conducted using 16-threaded intel i7 machines with 64 GB Ram for each embedding offered via the university network, using the "Slurm Workload Manager"[14].

The results would be computed using a slightly modified "scikit-learn" "classification_report", in order to compute all the desired micro- and macro-avg in the Python Dictionary format. The resulting dictionaries would then transformed to "Pandas Dataframes"[15], a table like format with powerful convenient manipulation options, p.e. a "to_latex()" function. The trained classifier weights would be saved via the "joblib" library for efficient compression[16].

The "classification_report" (scikit-learn 2019b) gives a detailed view of the results per ingredient type. Using a tutorial by "scikit-learn" (scikit-learn 2019a) a normalized confusion matrix would be created. The normalization would make interpreting the wrongly classified ingredient types easier, considering the extreme skew of the data.

2.5 Cuisine Classification

The Python library "scikit-learn" was again used for the majority of the implementation (scikit-learn.org). After merging the cuisines, the dataset was split using cross-validation as described before. As the embeddings mentioned are word or "n"-gram based, not word list based as is the case with the recipes in this dataset, a summation strategy was applied:

[11] scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[12] scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[13] scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[14] slurm.schedmd.com
[15] pandas.pydata.org
[16] pypi.org/project/joblib/

One variation was to simply sum up all the without weighting using the function ”sum” from Python library ”NumPy” (numpy.org). Additionally a simple sparse term frequency-inverted document frequency (TF-IDF) vector was produced using the function ”TfidfVectorizer” with ”l1” norm, using no further preprocessing and tokenization as the dataset is already preprocessed. This embedding was dubbed ”tfidf” and evaluated separately.

It was also used to create TF-IDF weighted sums instead of simple sums for the other embeddings via application of the NumPy function for the dot product, appending ”-tfidf” to the embedding names as the named reference, as well concatenated to these TF-IDF weighted vectors using the NumPy function ”concat” and naming these embeddings ”-concat”.

A ”LinearSVC” classifier was used with a smaller hyperparameter tuning grid to keep training times less than 10 hours per embedding:

- ’C’: [500, 200, 100, 50, 20, 10, 5, 2, 1, 0.5, 0.01, 0.05]
- ’dual’: [False, True] # Default=False
- ’class_weight’: [’balanced’] # Default
- ’random_state’: [0]
- ’max_iter’: [1000] # Default

The ”LogisticRegression” classifier of ”scikit-learn” was used with a similar hyperparameter tuning grid adjusting the number of iterations:

- ’C’: [500, 200, 100, 50, 20, 10, 5, 2, 1, 0.5, 0.01, 0.05]
- ’dual’: [False, True]
- ’multi_class’: [’auto’] # Default=ovr
- ’random_state’: [0]
- ’max_iter’: [500] # Default=100

The ”multi_class” parameter ”auto”, set per default in future versions, will choose ”one-versus-rest” loss here due to the chosen default solver[17]

2.6 Cuisine Topic Discovery

There are a variety of topic modeling implementations available, the Java library ”MAchine Learning for LanguagE Toolkit” (MALLET) being one of the more popular and established ones[18] using ”Gibbs” sampling, a Markov chain Monte Carlo algorithm (MCMC). There is also a MALLET wrapper in Python by the ”Gensim” library[19].

Alternative topic model implementations often use variational inference that has advantages but also severe drawbacks compared to ”Gibbs” sampling with MCMC: ”Thus, variational inference is suited to large data sets and scenarios where we want to quickly explore many models; MCMC is suited to smaller data sets and scenarios where we happily pay a heavier computational cost for more precise sample.” (Blei et al. 2017).

As the initial training runs using the ”Gensim” wrapper were relatively fast, it was decided to prefer a ”Gibbs” sampling implementation. Unfortunately, a good tutorial to implement metrics like log-likelihood on a separate cross-validation set was not found, not for Java and neither for Python (cf. Konrad 2017). While this setup could be implemented via the ”scikit-learn” library in a relatively straightforward manner[20], its LDA function uses variational Bayesian inference[21].

[17] see scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[18] http://mallet.cs.umass.edu/topics.php
[19] radimrehurek.com/gensim/models/ldamallet.html
[20] https://mail.python.org/pipermail/scikit-learn/2017-March/001309.html
[21] scikit-learn.org/stable/related_projects.html

A tutorial about measuring the ”coherence” score with the Gensim MALLET wrapper was found[22], but an even more powerful alternative was chosen, the ”tmtoolkit” library[23] by the ”Wissenschaftszentrum Berlin für Sozialforschung”[24]. It is the free and open source ”Text Mining and Topic Modeling Toolkit for Python”, notably offering various evaluation metrics including a wrapper for ”gensim”’s coherence calculation as well as multiple visualizations based p.e. on ”matplotlib”[25], ”wordcloud”[26] and ”Pillow”[27]. Topic modeling is calculated by the ”Gibbs” sampling method via the ”lda” package[28].

A well written tutorial by the main Author Markus Konrad (ibid.) was customized to fit the cuisine data and to measure also logarithmic-likelihood, giving a total of four metrics:

- log-likelihood (Griffiths et al. 2004)
- average correlation between topic-pairs (Cao et al. 2009)
- symmetric Kullback-Leibler divergence (Arun et al. 2010)
- coherence (Mimno et al. 2011)

Between 2 and 252 topics in steps of 10 were optimized with 2.000 maximum iterations and scored according to these metrics, conveniently displayed as a graph using the library capabilities. The effect of using different α and β values was to be explored as well by running two different variants, one with default values α=1.0/k & constant β=0.1 and one with a variable β=1/10k, as suggested by Konrad 2017.

A topic model of each run will be explored by a sample of the topics, while the subjectively more fitting and interesting one will be explored in more detail.

The examples of the library ”tmtoolkit”[29] would serve to create measures like ”distinctiveness and saliency” (”word_and_topic_scores.ipynb”), topic word clouds and document-topic distribution heatmaps (”lda_visualization.ipynb”). It is important to note that lemmatization should in contrast to these examples not be used here in order to avoid p.e. the ingredients ”rose” and ”clove” present in the ingredients type dataset being transformed to the verbs ”rise” and ”cleave”.

The ”distinctiveness” score describes how informative a word is for the topic generation compared to a random word, i.e. if a word occurs in all topics, it’s distinctiveness would be low (Chuang et al. 2012, p. 75). Saliency is distinctiveness weighted by the frequency of the word.

Additionally to the static graphs, a screenshot of the interactive topic model visualization ”pyLDAvis”[30] was made adjusting it to the output of the ”tmtoolkit”. This will be especially useful in comparing marginal topic distributions.

2.7 Visualization of the Ingredient Types and Cuisines

Several runs of t-SNE visualizations were to be expected as there are a lot of variables to be accounted for, such as the ”perplexity” parameter or the color scheme. The parameters would be set to the default (α=1/k, β=0.1, 1000 iterations) except for perplexity, for which values between 5 and 50 would be inserted. The ingredient t-SNE would get more maximum

[22] machinelearningplus.com/nlp/topic-modeling-gensim-python/#17howtofindtheoptimalnumberoftopics-forlda
[23] github.com/WZBSocialScienceCenter/tmtoolkit
[24] wzb.eu/
[25] matplotlib.org
[26] github.com/amueller/word_cloud
[27] github.com/python-pillow/Pillow
[28] pydoc.io/pypi/tmtoolkit-0.7.2/autoapi/lda_utils/tm_lda/index.html and pythonhosted.org/lda/api.html
[29] github.com/WZBSocialScienceCenter/tmtoolkit/tree/master/examples
[30] github.com/bmabey/pyLDAvis

iterations as it would only take seconds to train. Each run of the cuisine data representations on the other hand was going to be resource intensive, as it was about the algorithmic dimension reduction for each of the over 56.000 recipe embeddings from sometimes more than 1.000 to two dimensions:

Transformations using the single CPU-core limited "scikit-learn" library usually took 10-30 minutes (AMD 2700X processor with 16 threads, 32GB RAM and NVME SSD) which would slow down development. A good tutorial for t-SNE visualizations of topics was found and customized (Shuai 2016) that mentioned a multi-core implementation by "Gensim"[31]. As the other code was so far was based on the "scikit-learn" library, the similarly interfaced "MulticoreTSNE" library[32] was instead used, reducing conversion times to less than five minutes.

Three settings for the t-SNE "perplexity" parameter would be evaluated for the cuisine dataset: the minimal usual perplexity 5, the default 30 and the largest usual value, 50. Based on these results the ingredient type visualization perplexity would be adjusted and the best graph displayed in the results section.

For the actual graph plot, the libraries "matplotlib"[33] and "seaborn" were used[34], specifically the "lmplot" function[35].

For each set, the "ingredients dataset", the "cuisine dataset" as well as its variant the "merged cuisine dataset", a custom color palette as well as markers list was carefully composed, in order to maximize the visibility of every class (Maaten et al. 2008, cf. figures 2-5).

Colors such as the following often yielded the best results: pink and turquoise, orange, yellow and black, shades of blue, as well as a bright red and shades of green. These would be iteratively optimized for each t-SNE graph. Very often appearing classes would be given smaller and rarer classes larger markers, in a preference of readability over precision.

[31] https://radimrehurek.com/gensim/models/ldamulticore.html
[32] github.com/DmitryUlyanov/Multicore-TSNE
[33] matplotlib.org
[34] seaborn.pydata.org
[35] seaborn.pydata.org/generated/seaborn.lmplot.html

Chapter 3

Results

3.1 Ingredient Type Classification

3.1.1 Results

Embedding	micro-p	micro-r	micro-f1	macro-p	macro-r	macro-f1
googlenews	0.684	0.684	0.684	0.624	0.662	0.636
wiki_fasttext	0.721	0.721	0.721	0.659	0.694	0.673
im2rec_joint_null	0.300	0.300	0.300	0.199	0.240	0.214
im2rec_joint_avg	0.300	0.300	0.300	0.200	0.235	0.213
im2rec_base	0.772	0.772	0.772	0.687	0.709	0.695
im2rec_fasttext	0.768	0.768	0.768	0.747	0.700	**0.718**

TABLE 3.1: LinearSVC ingredient type classification results overview (train data).

Embedding	micro-p	micro-r	micro-f1	macro-p	macro-r	macro-f1
googlenews	0.720	0.720	0.720	0.736	0.715	0.678
wiki_fasttext	0.746	0.746	0.746	0.718	0.696	0.691
im2rec_joint_null	0.189	0.189	0.189	0.121	0.230	0.153
im2rec_joint_avg	0.216	0.216	0.216	0.133	0.269	0.172
im2rec_base	0.675	0.675	0.675	0.630	0.711	0.636
im2rec_fasttext	0.767	0.767	0.767	0.770	0.678	**0.698**

TABLE 3.2: LinearSVC ingredient type classification results overview (test data).

Table 3.1 shows the results overview of the linear classifier predicting ingredient types using 10-fold cross-validation on the training data set.

”micro-” or ”macro-p/r/f1” stand for the average micro- or macro-precision-/recall-/f1-score.

Table 3.2 shows the results overview of the linear classifier predicting ingredient types using the separate test set.

im2rec_fasttext	precision	recall	macro-f1-score	support
alcoholic_beverage	0.926	0.844	0.883	45.0
animal	0.000	0.000	0.000	17.0
cereal/crop	0.870	0.750	0.805	36.0
dairy	0.918	0.944	0.931	36.0
fish/seafood	0.880	0.862	0.871	51.0
flower	0.685	0.400	0.505	60.0
fruit	0.781	0.785	0.783	168.0
herb	0.648	0.432	0.518	81.0
meat	0.979	0.903	0.940	52.0
nut/seed/pulse	0.838	0.866	0.852	30.0
plant	0.603	0.794	0.686	282.0
plant_derivative	0.869	0.874	0.872	382.0
spice	0.680	0.640	0.659	50.0
vegetable	0.779	0.712	0.744	94.0
micro avg	0.768	0.768	0.768	1384.0
macro avg	0.747	0.700	0.718	1384.0

Table 3.3: LinearSVC ingredient type classification best embedding detailed results (train data).

Table 3.3 shows the detailed results per ingredient classification of the best embedding by the macro-average-F1 standard (im2rec_fasttext) using 10-fold cross-validation on the training data set.

im2rec_fasttext	precision	recall	macro-f1-score	support
alcoholic_beverage	1.000	0.800	0.888	5.0
animal	0.000	0.000	0.000	1.0
cereal/crop	0.750	1.000	0.857	3.0
dairy	1.000	1.000	1.000	3.0
fish/seafood	0.750	0.600	0.666	5.0
flower	0.750	0.500	0.600	6.0
fruit	0.789	0.833	0.810	18.0
herb	0.555	0.555	0.555	9.0
meat	1.000	1.000	1.000	5.0
nut/seed/pulse	1.000	0.666	0.800	3.0
plant	0.589	0.741	0.657	31.0
plant_derivative	0.904	0.904	0.904	42.0
spice	1.000	0.200	0.333	5.0
vegetable	0.700	0.700	0.700	10.0
micro avg	0.767	0.767	0.767	146.0
macro avg	0.770	0.678	0.698	146.0

Table 3.4: LinearSVC ingredient type classification best embedding detailed results (test data)

3.1.2 Best Embedding (Macro-F1) in Detail

Table 3.4 shows the detailed results per ingredient classification of the best embedding by the macro-average-F1 standard (im2rec_fasttext) using the separate test set.

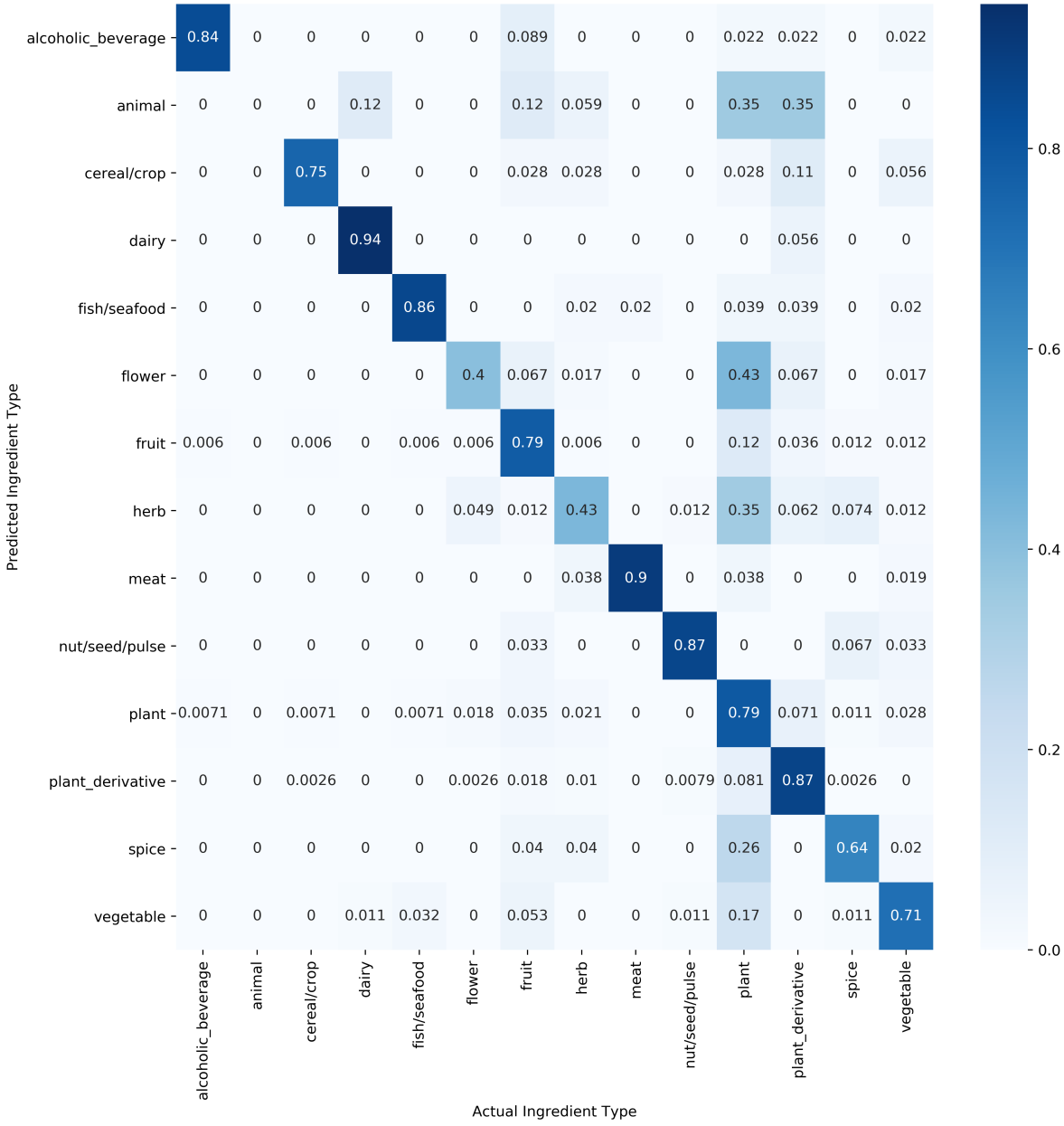

FIGURE 3.1: Confusion matrix ingredient type classification (train data).

Figure 3.1 shows the normalized confusion matrix per ingredient classification (LinearSVC) of the best embedding by the macro-average-F1 standard (im2rec_fasttext) using 10-fold cross-validation on the training data set. If 100 percent of the classifications are correct, the matrix cell displays 1.0.

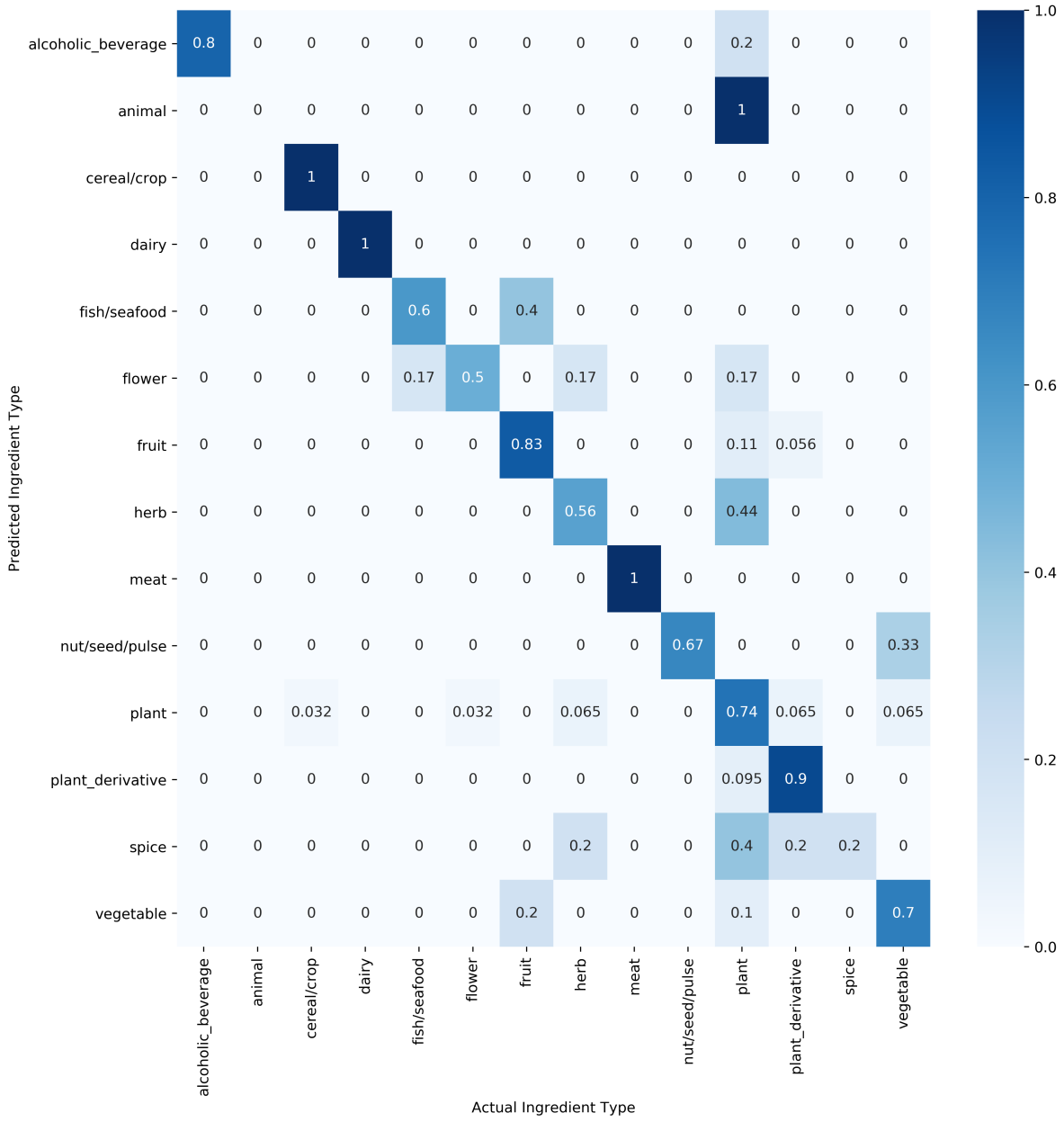

FIGURE 3.2: Confusion matrix ingredient type classification (test data).

Figure 3.2 shows the normalized confusion matrix per ingredient classification (LinearSVC) of the best embedding by the macro-average-F1 standard (im2rec_fasttext) using the separate test set. If 100 percent of the classifications are correct, the matrix cell displays 1.0.

3.2 Cuisine Classification

3.2.1 Results

Recipe Embedding	micro-p	micro-r	micro-f1	macro-p	macro-r	macro-f1
tfidf	0.815	0.815	0.815	0.618	0.737	0.663
googlenews-sum	0.815	0.815	0.815	0.612	0.738	0.657
googlenews-tfidf	0.810	0.810	0.810	0.607	0.733	0.655
googlenews-concat	0.807	0.807	0.807	0.593	0.740	0.644
wiki_fasttext-sum	0.818	0.818	0.818	0.615	0.730	0.655
wiki_fasttext-tfidf	0.816	0.816	0.816	0.626	0.724	0.666
wiki_fasttext-concat	0.814	0.814	0.814	0.629	0.726	0.669
im2rec_joint_null-sum	0.818	0.818	0.818	0.627	0.727	0.667
im2rec_joint_null-tfidf	0.813	0.813	0.813	0.615	0.727	0.659
im2rec_joint_null-concat	0.812	0.812	0.812	0.608	0.737	0.656
im2rec_joint_avg-sum	0.823	0.823	0.823	0.654	0.705	**0.677**
im2rec_joint_avg-tfidf	0.815	0.815	0.815	0.623	0.715	0.662
im2rec_joint_avg-concat	0.811	0.811	0.811	0.618	0.723	0.661
im2rec_base-sum	0.819	0.819	0.819	0.660	0.686	0.672
im2rec_base-tfidf	0.815	0.815	0.815	0.627	0.704	0.660
im2rec_base-concat	0.812	0.812	0.812	0.622	0.699	0.656
im2rec_fasttext-sum	0.828	0.828	0.828	0.663	0.693	0.676
im2rec_fasttext-tfidf	0.822	0.822	0.822	0.641	0.699	0.666
im2rec_fasttext-concat	0.813	0.813	0.813	0.611	0.748	0.660

Table 3.5: LinearSVC cuisine classification results overview (train data).

Table 3.5 shows the results overview of the LinearSVC classifier predicting the four merged cuisine regions using 10-fold cross-validation on the training data set. ”micro-” or ”macro-p/r/f1” stand for the average micro- or macro-precision-/recall-/f1-score.

Recipe Embedding	micro-p	micro-r	micro-f1	macro-p	macro-r	macro-f1
tfidf	0.788	0.788	0.788	0.564	0.643	0.591
googlenews-sum	0.799	0.799	0.799	0.558	0.662	0.584
googlenews-tfidf	0.789	0.789	0.789	0.548	0.644	0.570
googlenews-concat	0.789	0.789	0.789	0.544	0.664	0.578
wiki_fasttext-sum	0.790	0.790	0.790	0.546	0.654	0.566
wiki_fasttext-tfidf	0.792	0.792	0.792	0.575	0.642	0.601
wiki_fasttext-concat	0.790	0.790	0.790	0.584	0.639	**0.606**
im2rec_joint_null-sum	0.795	0.795	0.795	0.574	0.611	0.581
im2rec_joint_null-tfidf	0.796	0.796	0.796	0.564	0.630	0.579
im2rec_joint_null-concat	0.786	0.786	0.786	0.545	0.646	0.576
im2rec_joint_avg-sum	0.803	0.803	0.803	0.594	0.594	0.586
im2rec_joint_avg-tfidf	0.794	0.794	0.794	0.560	0.629	0.575
im2rec_joint_avg-concat	0.790	0.790	0.790	0.567	0.655	0.601
im2rec_base-sum	0.782	0.782	0.782	0.648	0.539	0.566
im2rec_base-tfidf	0.797	0.797	0.797	0.573	0.659	0.604
im2rec_base-concat	0.792	0.792	0.792	0.569	0.617	0.582
im2rec_fasttext-sum	0.788	0.788	0.788	0.568	0.569	0.562
im2rec_fasttext-tfidf	0.797	0.797	0.797	0.549	0.605	0.546
im2rec_fasttext-concat	0.784	0.784	0.784	0.550	0.666	0.586

Table 3.6: LinearSVC cuisine classification results overview (test data).

Table 3.6 shows the results overview of the LinearSVC classifier predicting the four merged cuisine regions using the separate test set.

3.2.2 Best Embedding (Macro-F1) in Detail

wiki_fasttext-concat	precision	recall	macro-f1-score	support
Eastern	0.726	0.841	0.779	2673.0
SouthAsian	0.415	0.658	0.509	559.0
Southern	0.470	0.540	0.503	7285.0
Western	0.905	0.864	0.884	40333.0
micro avg	0.814	0.814	0.814	50850.0
macro avg	0.629	0.726	0.669	50850.0

TABLE 3.7: LinearSVC cuisine classification best embedding detailed results (train data).

Table 3.7 shows the detailed results per merged cuisine classification (LinearSVC) of the best embedding by the macro-average-F1 standard (im2rec_fasttext) using 10-fold cross-validation on the training data set.

wiki_fasttext-concat	precision	recall	macro-f1-score	support
Eastern	0.603	0.601	0.602	296.0
SouthAsia	0.430	0.645	0.516	62.0
Southern	0.423	0.444	0.433	809.0
Western	0.881	0.867	0.874	4481.0
micro avg	0.790	0.790	0.790	5648.0
macro avg	0.584	0.639	0.606	5648.0

TABLE 3.8: LinearSVC cuisine classification best embedding detailed results (test data).

Table 3.8 shows the detailed results per merged cuisine classification (LinearSVC) of the best embedding by the macro-average-F1 standard (wiki_fasttext_concat) using the separate test set.

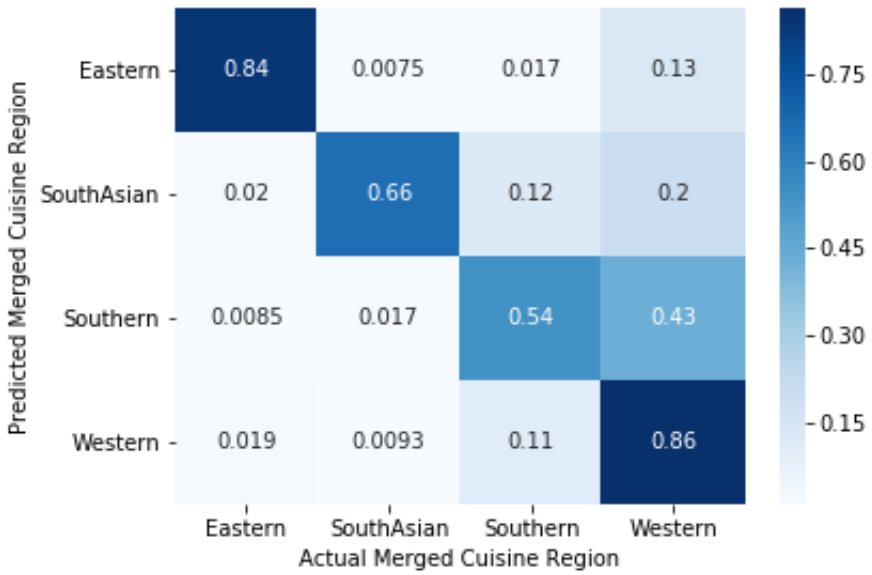

FIGURE 3.3: Confusion matrix ingredient type classification (train data).

Figure 3.3 shows the normalized confusion matrix per merged cuisine classification (LinearSVC) of the best embedding by the macro-average-F1 standard (wiki_fasttext_concat) using 10-fold cross-validation on the training data set. If 100 percent of the classifications are correct, the matrix cell displays 1.0.

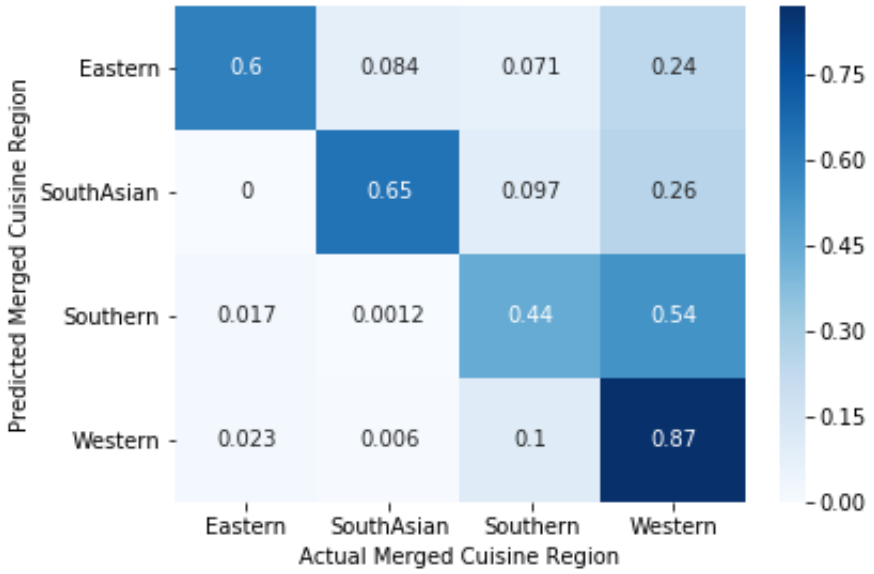

FIGURE 3.4: Confusion matrix ingredient type classification (test data).

Figure 3.4 shows the normalized confusion matrix per merged cuisine classification of the best embedding by the macro-average-F1 standard (wiki_fasttext_concat) using the separate test set. If 100 percent of the classifications are correct, the matrix cell displays 1.0.

3.3 Discovered Food Topics

3.3.1 Discovered Food Topics

Topic №	Top ten words
1	butter NorthAmericanCUISINE olive_oil garlic chicken_broth mushroom parmesan_cheese cream white_wine SouthernEuropeanCUISINE
2	NorthAmericanCUISINE wheat egg butter vanilla cocoa cane_molasses milk lard walnut
3	onion black_pepper carrot garlic beef thyme WesternEuropeanCUISINE NorthAmericanCUISINE butter olive_oil
4	cayenne cilantro lime_juice LatinAmericanCUISINE onion garlic tomato olive_oil NorthAmericanCUISINE avocado
5	egg wheat butter milk almond vanilla SouthernEuropeanCUISINE WesternEuropeanCUISINE cream cinnamon
6	NorthAmericanCUISINE wheat egg cinnamon vanilla vegetable_oil butter cane_molasses walnut oat
7	garlic SoutheastAsianCUISINE cayenne fish lime_juice cilantro vegetable_oil ginger rice soy_sauce
8	cayenne onion tomato LatinAmericanCUISINE garlic cheddar_cheese beef corn cumin cheese
9	wheat NorthAmericanCUISINE yeast egg butter milk vegetable_oil whole_grain_wheat_flour honey seed
10	vinegar cane_molasses NorthAmericanCUISINE onion tamarind black_pepper mustard garlic vegetable_oil egg

Table 3.9: The top ten words for the first ten topics of topic model №1

The ten top words for the first ten topics of chosen Topic Model №1, which is based on an LDA training with the parameters α=1/k and β=0.1.

Topic №	Top ten words
1	butter NorthAmericanCUISINE olive_oil garlic chicken_broth parmesan_cheese cream white_wine mushroom SouthernEuropeanCUISINE
2	NorthAmericanCUISINE wheat egg milk butter cocoa vanilla cream coffee vegetable_oil
3	NorthAmericanCUISINE lemon lime orange_juice orange pineapple lime_juice rum lemon_juice honey
4	NorthAmericanCUISINE butter WesternEuropeanCUISINE cream bread onion cream_cheese rice egg olive
5	NorthAmericanCUISINE wheat egg butter milk cream vanilla lemon_juice lemon cream_cheese
6	NorthAmericanCUISINE egg wheat cinnamon butter milk nutmeg cane_molasses vanilla ginger
7	cumin turmeric coriander pepper fenugreek onion garlic SouthAsianCUISINE vegetable_oil cayenne
8	cayenne onion LatinAmericanCUISINE garlic tomato cumin corn cheese cheddar_cheese bell_pepper
9	NorthAmericanCUISINE wheat egg cinnamon butter vanilla walnut cane_molasses vegetable_oil raisin
10	NorthAmericanCUISINE onion thyme celery sage black_pepper butter rosemary chicken_broth bread

Table 3.10: The top ten words for the first ten topics of topic model №2

The ten top words for the first ten topics of chosen Topic Model №2, which is based on an LDA training with the parameters α=1/k and β=1/10k.

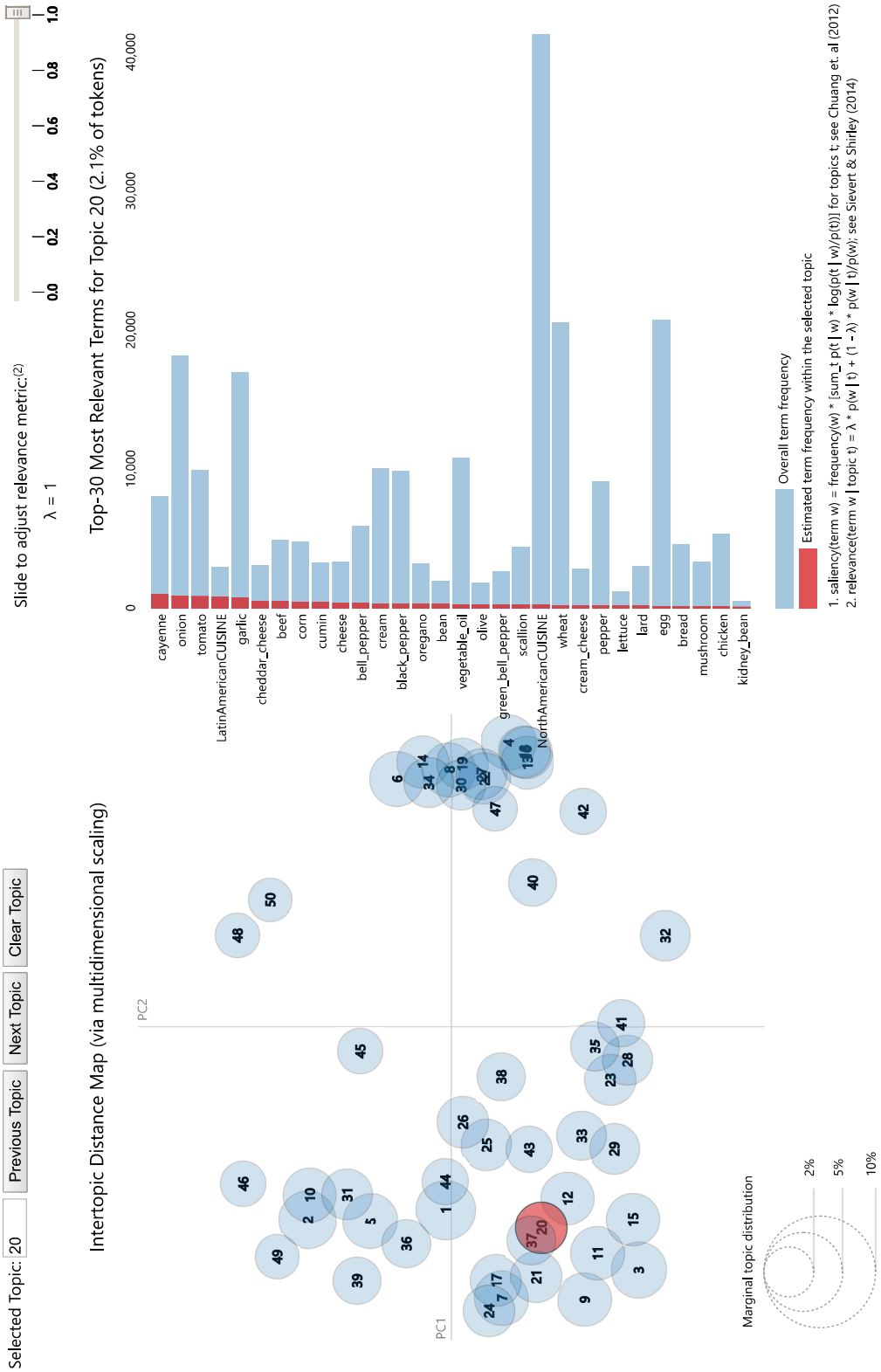

FIGURE 3.5: pyLDAvis Visualization Topic Model №1

Figure 3.5 shows a screenshot of the interactive pyLDAvis HTML Visualization for Topic Model №1. Topics are numbered according to the marginal topic distribution.

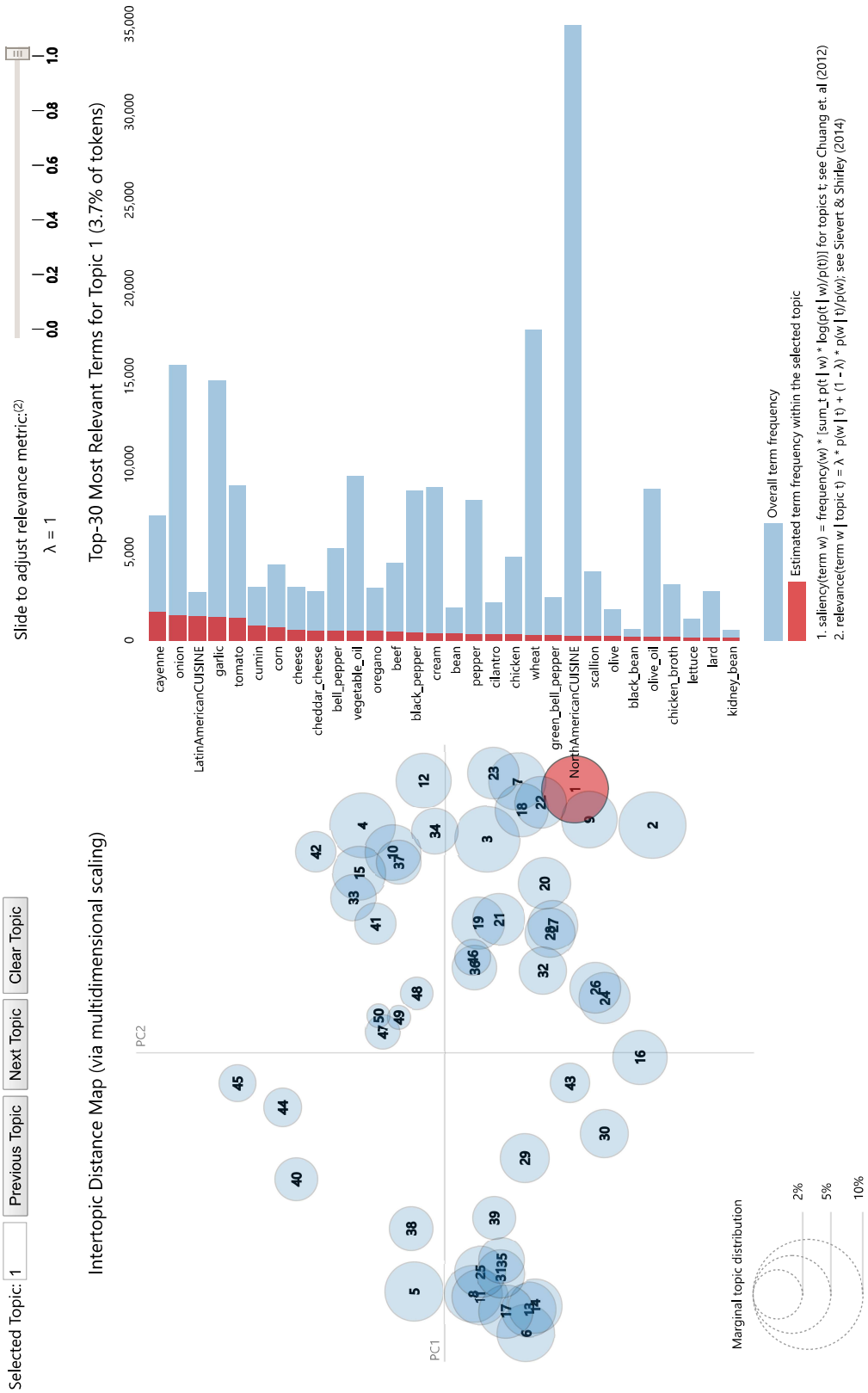

FIGURE 3.6: pyLDAvis Visualization Topic Model №2 (high marginal topic distribution topic)

Figure 3.6 shows a screenshot of the interactive pyLDAvis HTML Visualization for Topic Model №2 with the topic with the highest marginal topic probability selected. Topics are

numbered by their marginal topic distribution.

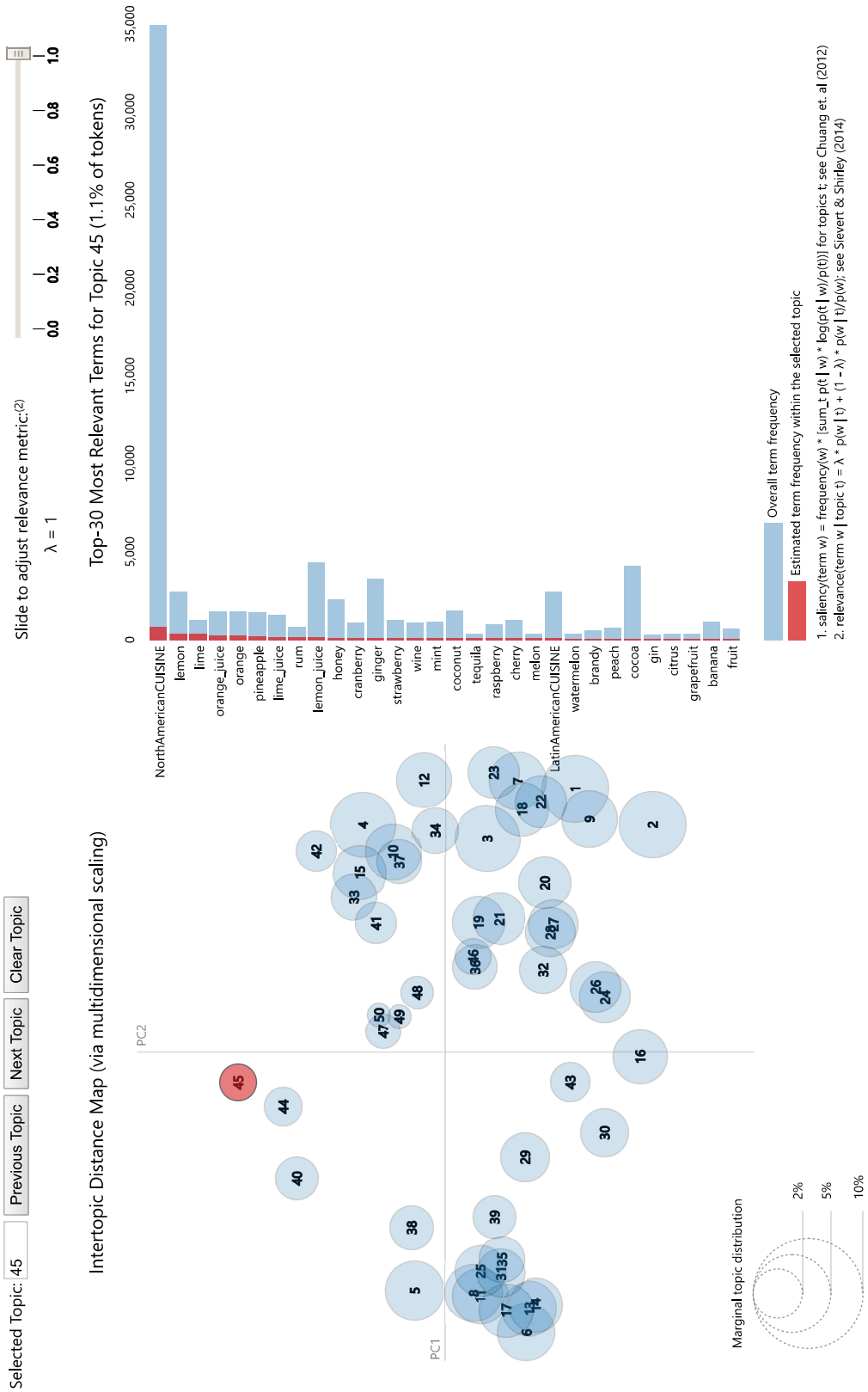

FIGURE 3.7: pyLDAvis Visualization Topic Model №2 (low marginal topic distribution topic)

Figure 3.7 shows a screenshot of the interactive pyLDAvis HTML Visualization for Topic Model №2 with a topic with a low marginal topic probability selected.

3.3.2 Distinct and Salient Words, TF-IDF

	Most distinct words	Least distinct words
1	lemongrass	sturgeon_caviar
2	SoutheastAsianCUISINE	NorthAmericanCUISINE
3	fenugreek	roasted_hazelnut
4	lovage	roasted_pecan
5	gin	roasted_nut
6	celery_oil	jamaican_rum
7	seaweed	muscat_grape
8	galanga	pelargonium
9	wasabi	mate
10	red_bean	angelica

Table 3.11: Top ten most and least distinct words according to Chuang et al. 2012.

The top ten most and least distinct words according to ibid. for Topic Model №1.

	Most salient words	Least salient words
1	wheat	sturgeon_caviar
2	egg	NorthAmericanCUISINE
3	milk	roasted_hazelnut
4	vanilla	roasted_pecan
5	butter	roasted_nut
6	olive_oil	jamaican_rum
7	vinegar	muscat_grape
8	NorthAmericanCUISINE	pelargonium
9	tomato	mate
10	garlic	angelica

Table 3.12: Top ten most and least salient words according to ibid.

The top ten most and least salient words according to ibid. for Topic Model №2.

	Most Common Words	TF-IDF Score		Least Common Words	TF-IDF Score
1	NorthAmericanCUISINE	1.307927	1	angelica	11.248831
2	egg	1.991989	2	beech	11.248831
3	wheat	2.000088	3	durian	11.248831
4	butter	2.001918	4	emmental_cheese	11.248831
5	onion	2.142075	5	geranium	11.248831
6	garlic	2.207264	6	jamaican_rum	11.248831
7	milk	2.477306	7	jasmine_tea	11.248831
8	vegetable_oil	2.658387	8	lilac_flower_oil	11.248831
9	cream	2.716454	9	mate	11.248831
10	tomato	2.740679	10	muscat_grape	11.248831

Table 3.13: Ten most and least common words according to the TF-IDF metric

The ten most and least common words according to the TF-IDF metric; the lower the score the more common.

3.3.3 Recipe-to-Topic Heatmaps

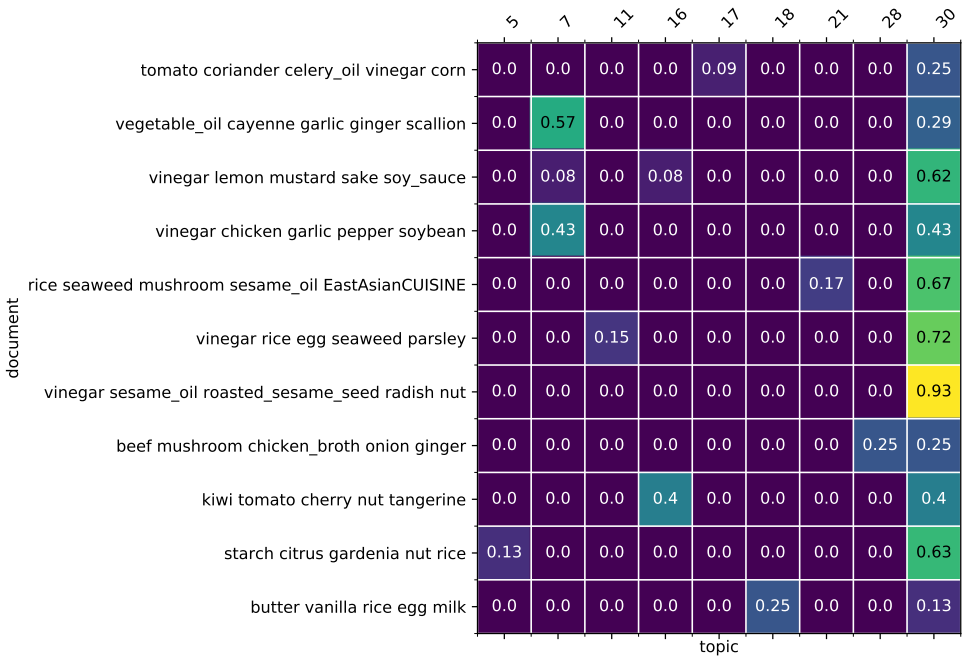

FIGURE 3.8: Topic heatmap of twelve "EastAsian" recipes.

An example recipe-to-topic heatmap for twelve "EastAsian" recipes for Topic Model №1. Rows are recipes, columns are a sample of the 50 topics shown as word clouds later.

FIGURE 3.9: Topic heatmap of twelve "SouthernEuropean" recipes.

An example recipe-to-topic heatmap for twelve "SouthernEuropean" recipes for Topic Model №1. Rows are recipes, columns are a sample of the 50 topics shown as word clouds later.

3.3.4 Marginal Topic Distribution

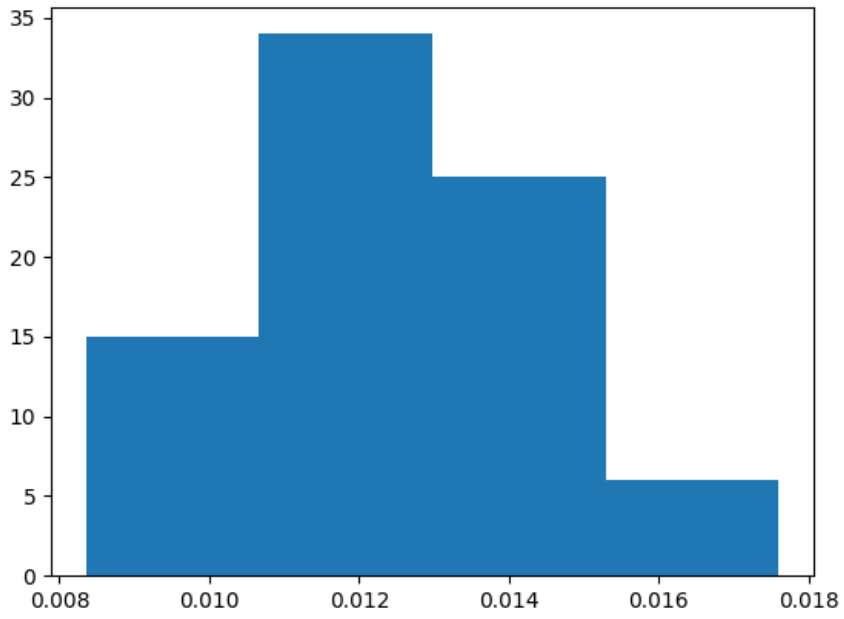

Figure 3.10: Marginal Topic Distribution Histogram

The marginal topic distribution histogram for Topic Model №1. The y-axis shows the number of topics, the x-axis the marginal probability of these topics, i.e. how much % they cover of the corpus (Sievert et al. 2014, cf. the size of the circles in the system described on p. 68). The histogram is approximated to four bins for better readability.

3.3.5 Word Clouds

Word clouds as created by the ”tmtoolkit” library for topics 5, 7, 11, 16, 17, 18, 21, 28 and 30. The bigger the word, the more it defines the topic.

Figure 3.11: Word Cloud of Topic №5 with 20 top Words

Figure 3.12: Word Cloud of Topic №7 with 20 top Words

Figure 3.13: Word Cloud of Topic №11 with 20 top Words

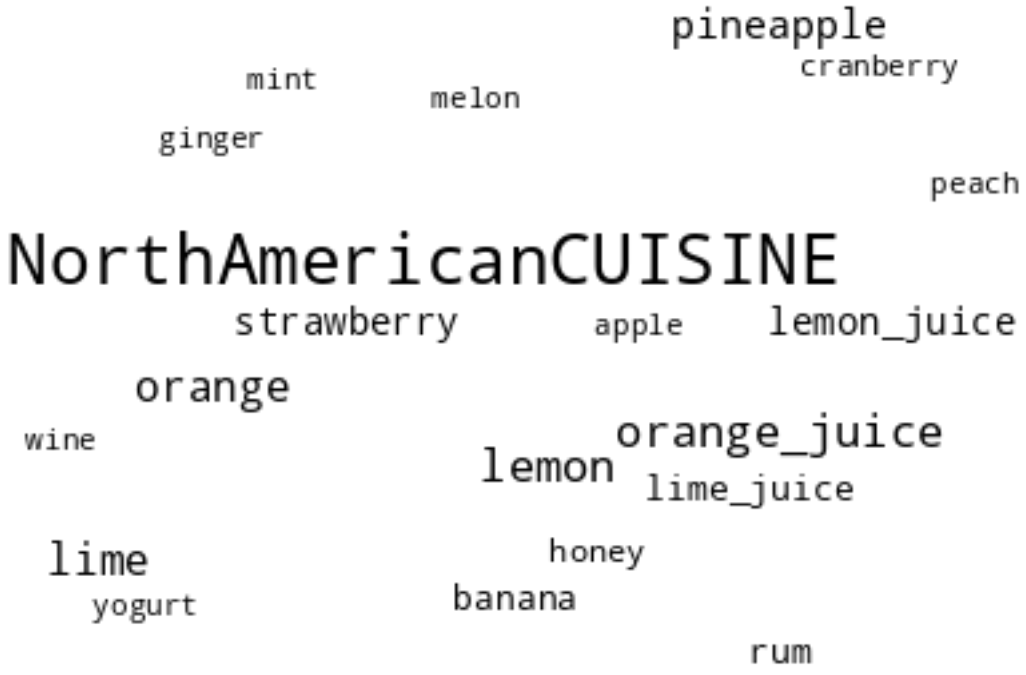

Figure 3.14: Word Cloud of Topic №16 with 20 top Words

Figure 3.15: Word Cloud of Topic №17 with 20 top Words

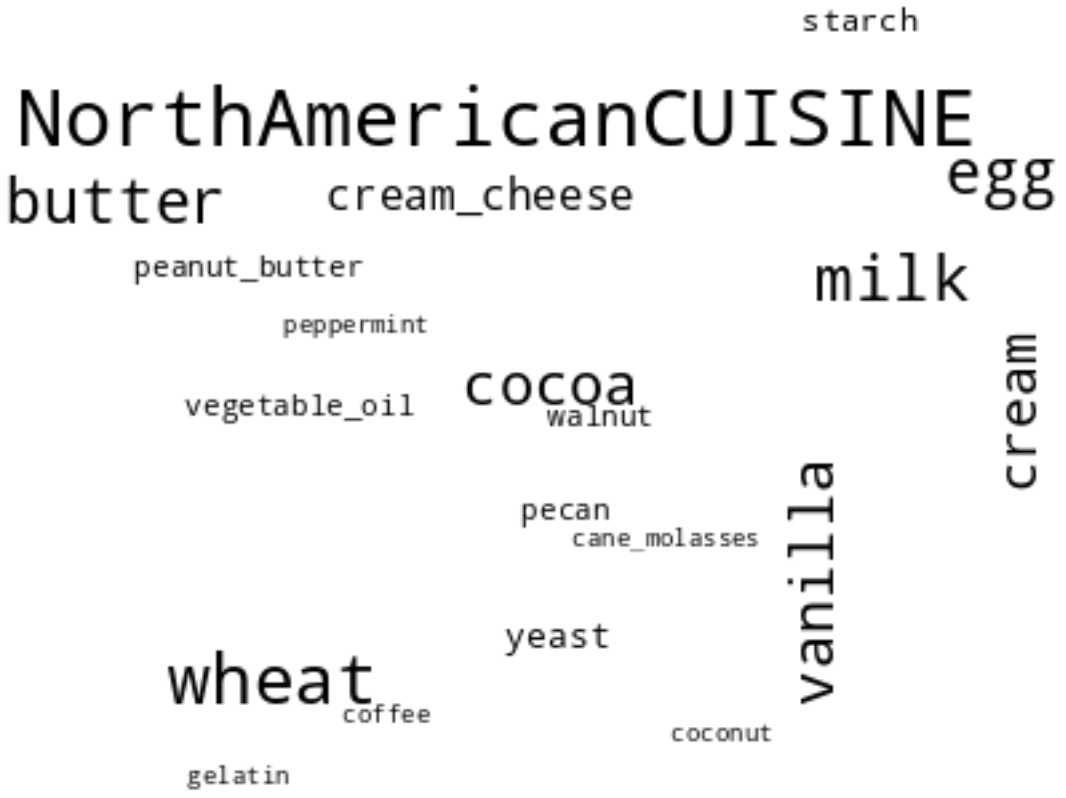

Figure 3.16: Word Cloud of Topic №18 with 20 top Words

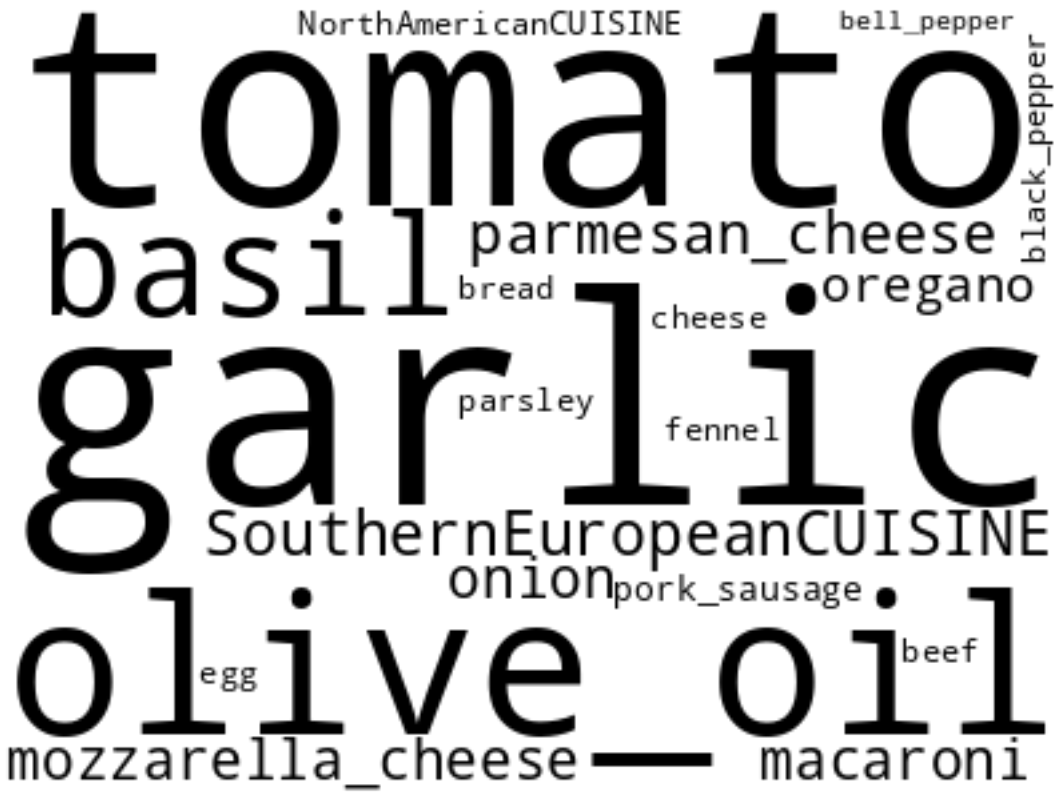

Figure 3.17: Word Cloud of Topic №21 with 20 top Words

Figure 3.18: Word Cloud of Topic №28 with 20 top Words

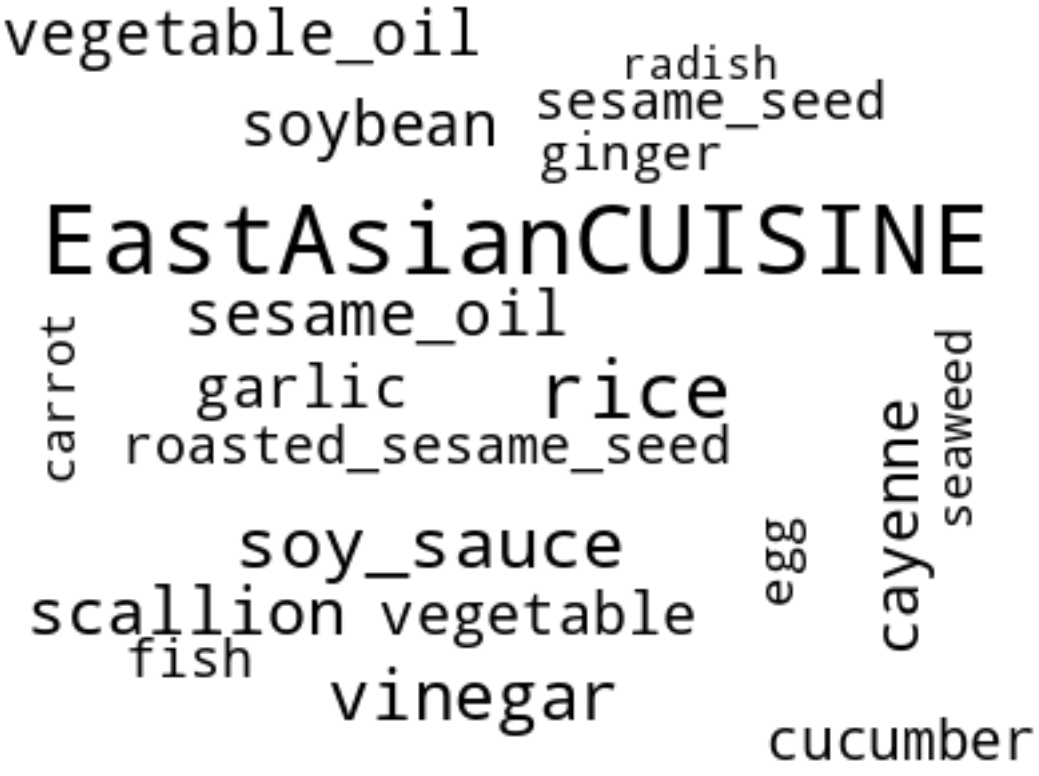

Figure 3.19: Word Cloud of Topic №30 with 20 top Words

3.4 Visualizations

3.4.1 Ingredient Type Visualization

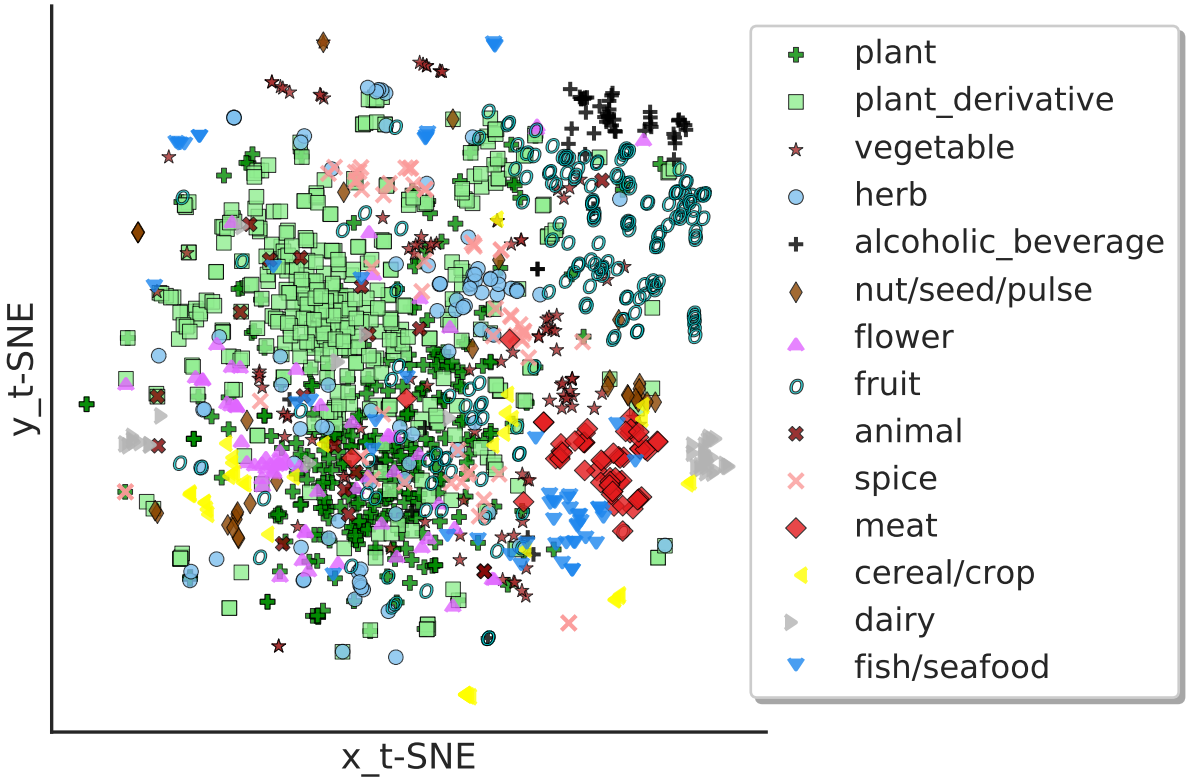

FIGURE 3.20: t-SNE (40 perplexity) of the ingredient types

Figure 3.20 visualizes the t-SNE conversion to two dimensions of the best embedding by the macro-average-F1 standard (im2rec_fasttext) which actually has 300 dimensions, using the parameters α=1/k, β=0.1, perplexity=40, iterations=3.000.

3.4.2 Visualization of the Cuisine Topic Embedding

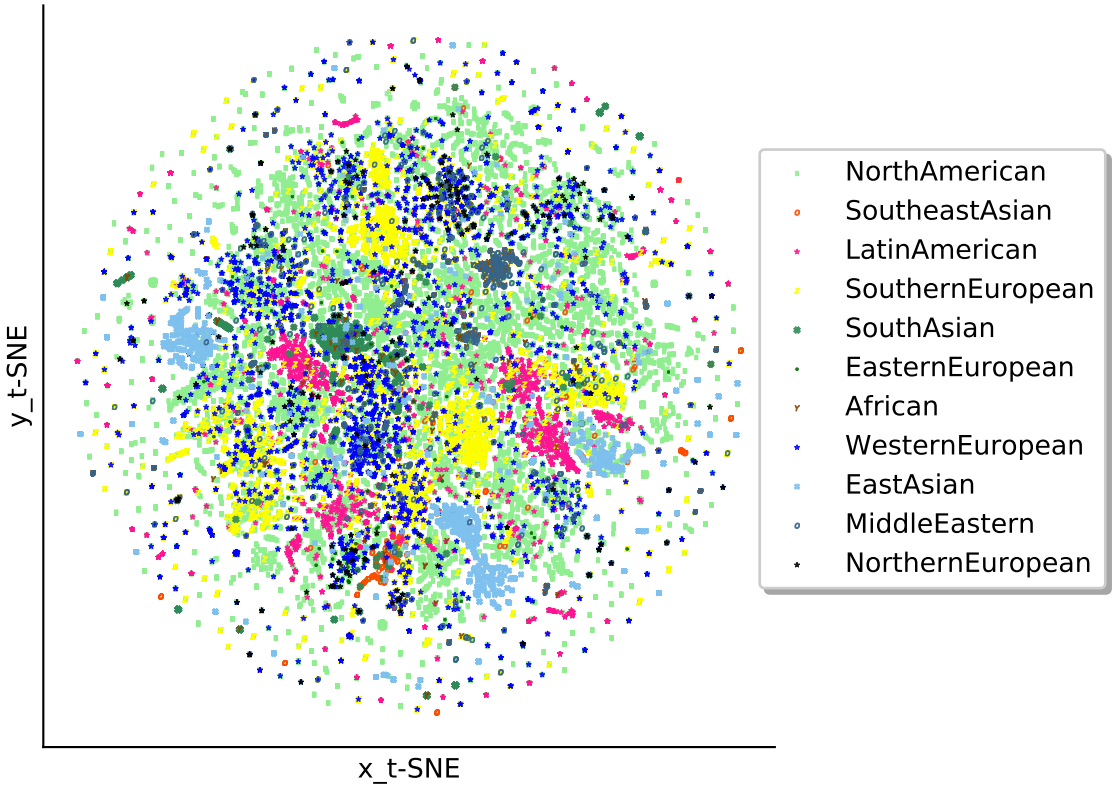

Figure 3.21: t-SNE (5 perplexity) of topic model №1

The 2-dimensional t-SNE of topic model №1 embedding's 50-dimensional recipe representations, colored by the 11 cuisine regions, using perplexity=5.

w

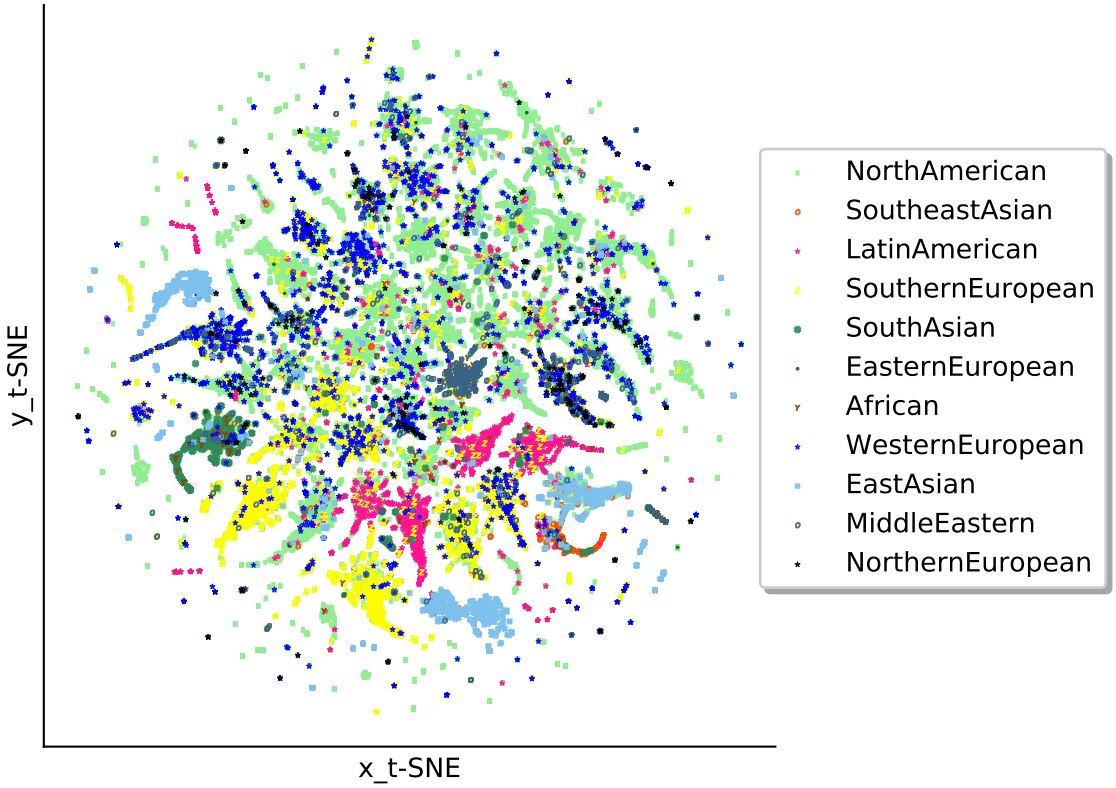

FIGURE 3.22: t-SNE (30 perplexity) of topic model №1

The 2-dimensional t-SNE of topic model №1 embedding's 50-dimensional recipe representations, colored by the 11 cuisine regions, using perplexity=30.

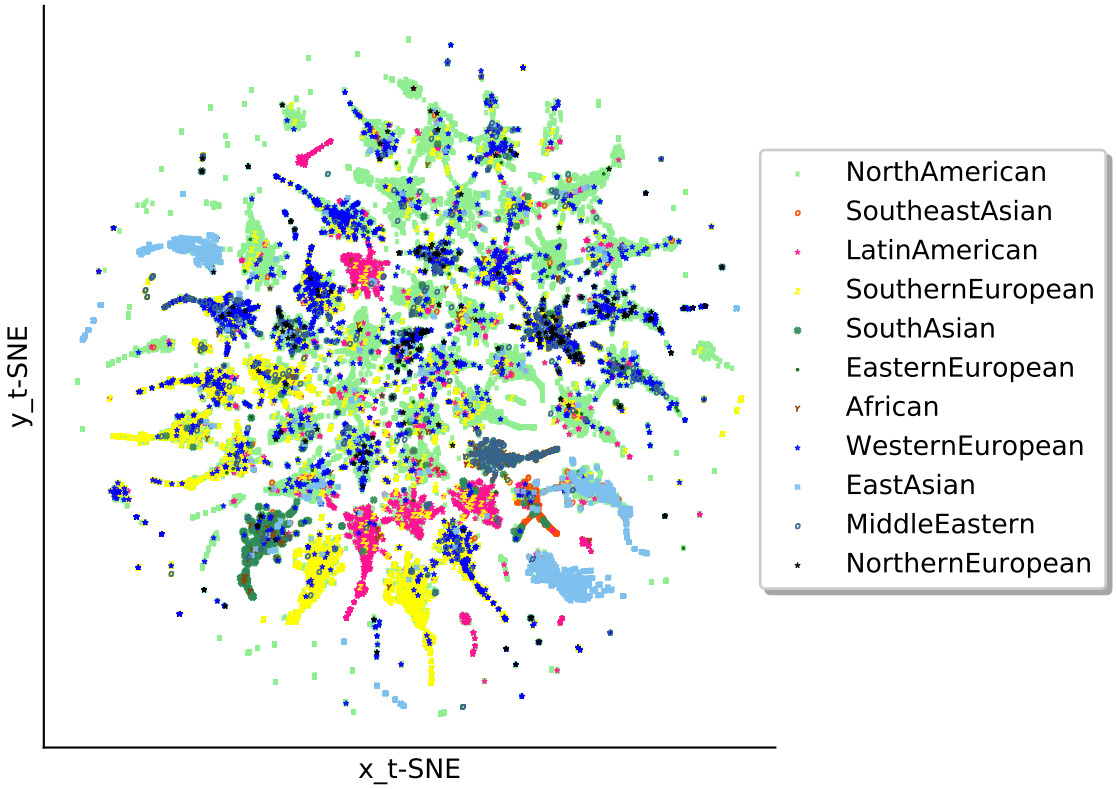

FIGURE 3.23: t-SNE (50 perplexity) of topic model №1

The 2-dimensional t-SNE of topic model №1 embedding's 50-dimensional recipe representations, colored by the 11 cuisine regions, using perplexity=50.

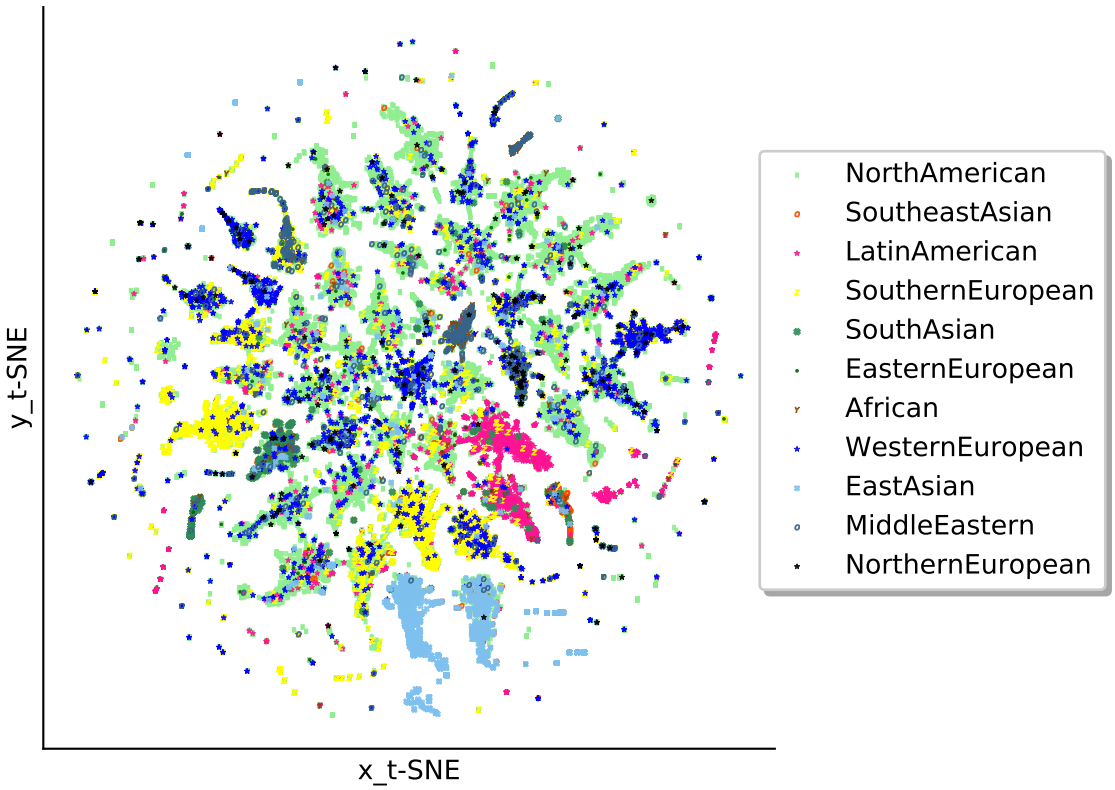

Figure 3.24: t-SNE (50 perplexity) of topic model №2

The 2-dimensional t-SNE of topic model №2 embedding's 50-dimensional recipe representations, colored by the 11 cuisine regions, using perplexity=50.

3.4.3 Visualization of the Cuisine TF-IDF Embedding

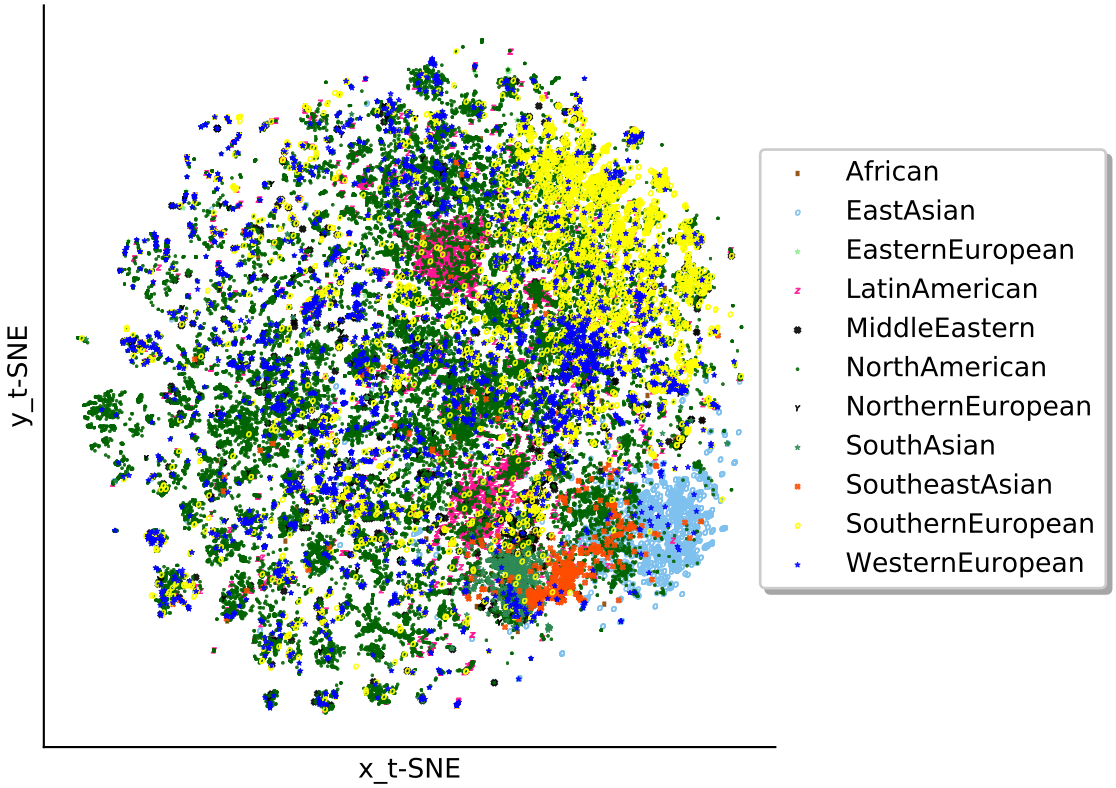

FIGURE 3.25: t-SNE (50 perplexity) of the TF-IDF recipe embedding with 11 cuisines.

The 2-dimensional t-SNE representation of the 381-dimensional TF-IDF recipe embedding of the 11 cuisines, using perplexity=50.

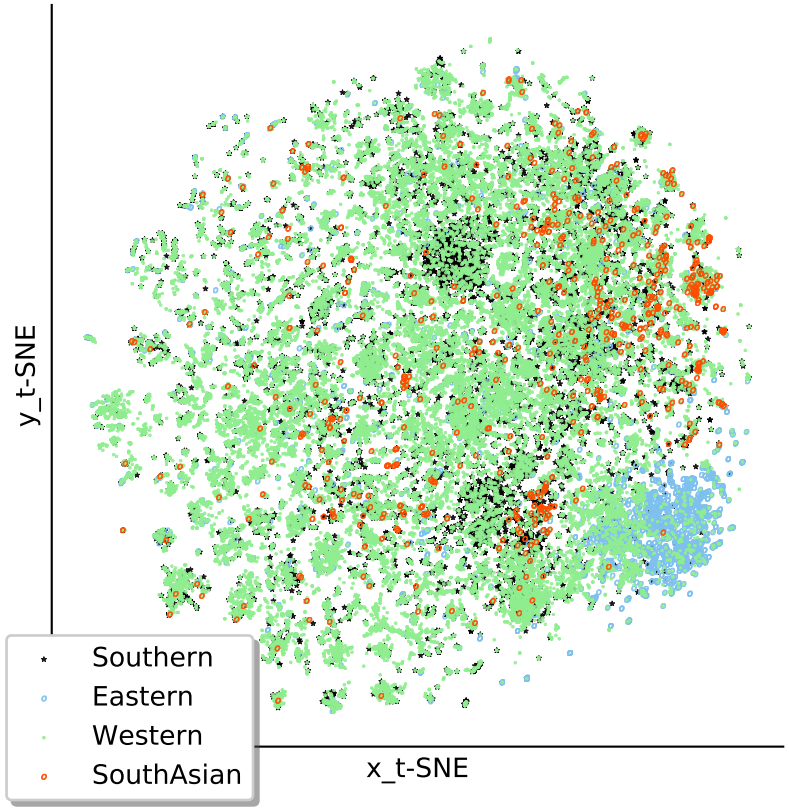

FIGURE 3.26: t-SNE (50 perplexity) of the TF-IDF recipe embedding with 11 cuisines, colored only by the four merged cuisine regions.

Figure 3.25 colored only by the four merged cuisine regions.

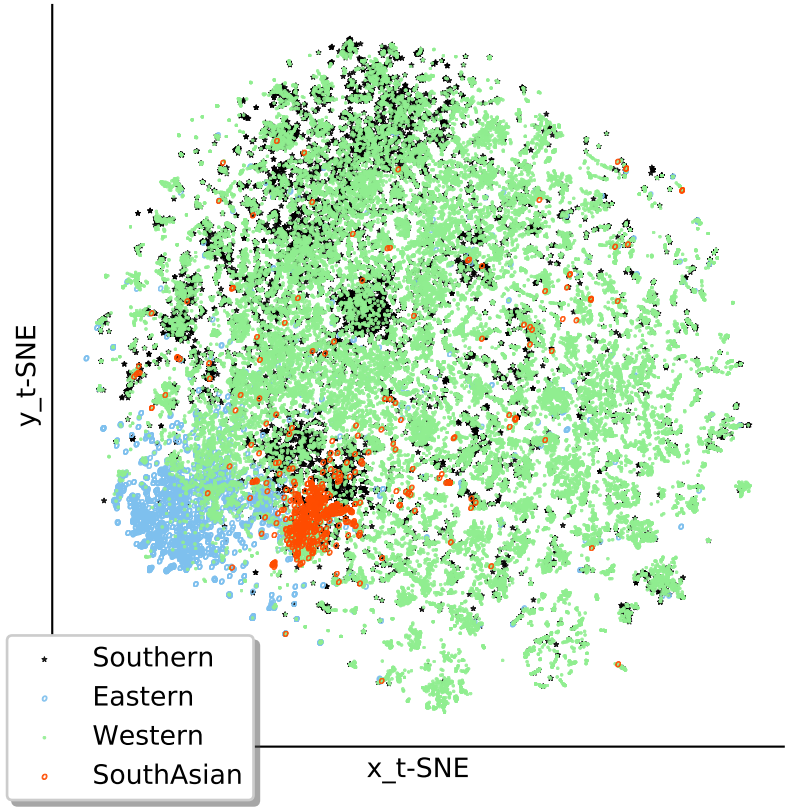

FIGURE 3.27: t-SNE (50 perplexity) of the TF-IDF recipe embedding with four merged cuisine regions.

The 2-dimensional t-SNE representation of the 381-dimensional TF-IDF recipe embedding of the four cuisines, using perplexity=50.

3.4.4 Visualization of the Best Cuisine Embedding

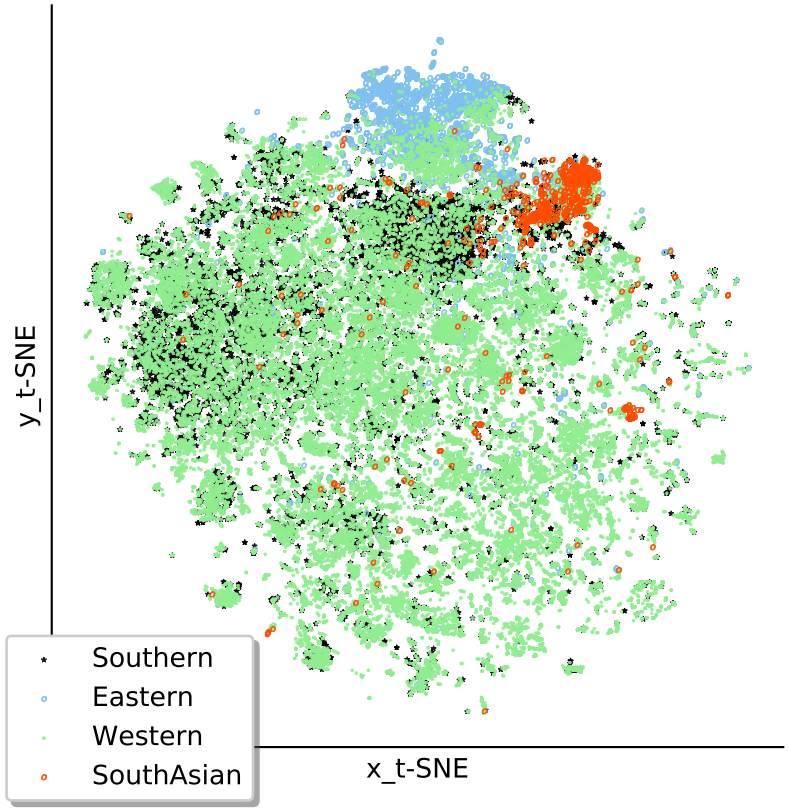

FIGURE 3.28: t-SNE (50 perplexity) of the best recipe embedding with four merged cuisine regions.

The 2-dimensional t-SNE representation of the 300-dimensional ”wiki_fasttext_concat” recipe embedding of the four cuisines, using perplexity=50.

Chapter 4

Discussion

4.1 Ingredient Type Classification

4.1.1 Results

The micro-average performances of predicting the right ingredient type are almost always higher than the macro-average ones, for both datasets train and test. I infer this is because of the skew of the dataset described in Chapter 2 combined with the fact that the number of the rarer ingredient types is very low, p.e. only 18 for the type ”animal”. This problem is exacerbated for the word based embeddings (Chapter 2), as their vocabulary matches only a subset of the dataset, making rare types potentially even rarer.

As can be calculated from the ”X_cvtrain” and ”X_test Shape” columns in table A.1 on page 71, the non-”fastText” embeddings’ vocabulary based on the 1 million recipes dataset cover only around one fourth of the ingredients dataset. This doesn’t necessarily need to lead to terrible results precision and recall wise, as ”im2rec_base” shows with 63% macro-average-F1 performance on the test set, given that the best performance is only 7 percent points better with 69.8% (im2rec_fasttext) and the similarly disadvantaged ”googlenews” embedding achieves 67.8%.

The performance of the two worst embeddings, ”im2rec_joint_null” and ”-_avg”, can therefore only in part be explained by these sample size issues. Their results are surprising given their large number of dimensions and that they perform so much worse than the embedding they are to an extent based upon, ”im2rec_base”. Creating a classifier with the rule of only returning the class ”plant_derivative” for every ingredient would probably lead to a similar micro-performance on the training dataset and a possibly better one on the test dataset, given the low performance of only 18.9% and 21.6% there and that this class makes up for 30.6% of the dataset.

Because of the imbalance of the sample size of the different classes, the macro-average performances are more challenging, but potentially more interesting and representative of the whole dataset and predictive task. It is possible that the impact of the images originally used to train the weights that represent ”im2rec_joint_null” and ”-_avg” dimensions can explain some of the confusion these models have in predicting the right type. In the ”im2recipe” task the neural network may have learned rather practical meanings of the different ingredients, setting them in the context of the way of preparing them, how they look in prepared recipes and how this appearance can help identify the recipe name. In this sense a rather theoretic or abstract differentiation between different types of ingredients like plant, plant derivative and vegetable may not have been learned. The image contextualization may thus have been detrimental to the ability to predict the right type of ingredient according to the ingredients dataset.

Training and test dataset results are generally similar, with important differences for particular embeddings (table 3.1 and table 3.2 on page 29).

Coming from the train results and looking at the test macro-average-F1 scores, the ”googlenews” embedding gains ca. 4, ”wiki_fasttext” ca 2 percent points, whereas it is the opposite for ”im2rec_base” and ”im2rec_fasttext” with losses of 5 and 2 percent points. Judging from these better performing embeddings, I thus infer the successfully learned relationships transfer relatively well to the unseen dataset. The already poor performing two embeddings ”im2rec_joint_null” and ”-_avg” lose another hefty 6 and 4 percent points. That the ”im2rec_joint-_avg” embedding performs better may be because the knowledge of the recipe preparation steps may help learning the right class. This part of the ”im2rec_joint_null” embedding is set to null, as described in Chapter 2.

All of the 1 million recipe based embeddings tend to overfit to the training data, resulting in often significantly worse test data results. Still even for them, probably because of the cross-validation procedure, the results are always at worst 6 percent points apart, meaning some of the training success is generalizable. The reason for the ”googlenews” and ”wiki_fasttext” embeddings better performance on the test set may lie in the fact that the texts their neural networks are based on a more diverse corpus in contrast to the ”im2rec” based embeddings being only based on recipes or in part food images. This helps ”googlenews” and ”wiki_fasttext” to have an easier time learning generalizable features like the prediction of the ingredient type here. It would be important to verify this interpretation on a larger test set however, because as seen on table A.1 the test set is at most 146, and at least only 37 ingredients large. Similarly it has to be kept in mind that the with regards to macro-average-F1 performance best embeddings, ”wiki_fasttext” and ”im2rec_fasttext”, had around 4 times as many training examples to learn their ability to predict ingredient types. This may be one of the reasons why the ”fastText” version of the 1 million recipe based embeddings outperforms the baseline version using ”Word2Vec” by more than 6 percent points, or around 10%.

The embeddings based on the two other corpora, Wikipedia and Google News, are relatively close together, with only around 1.3 percent points apart. It was to be expected that a Wikipedia embedding performed well on the ingredient type prediction task, as this is exactly the kind of knowledge an encyclopedia comprises. That ”googlenews” performs well is kind of surprising, given that the sample sizes are small due to vocabulary limitations and it is not immediately intuitive that news articles make for good knowledge about ingredient types. The embedding still may have been trained on the occasional recipe in newspapers articles inside its Google News corpus or by news about p.e. the commodities market or laws about food.

With 77% macro-average-precision, ”im2rec_fasttext” outperforms the next best embeddings ”googlenews” and ”wiki_fasttext” by 3 and respectively 5 percent points. 67% recall performance on the other hand is around 2 to almost 4 percent points worse than the aforementioned embeddings. This means the ”im2rec_fasttext” embedding not only has the best macro-F1 average but also the best macro-F1-precision average over all classes, with a marginally worse recall performance.

4.1.2 The Best Embedding’s Results

Looking at the detailed results reports of ”im2rec_fasttext” for the cross-validation and the test data, see table 3.3 on page 30 and table 3.4 on page 30, one can see the results for each ingredient types separately.

Judging by the train results, apparently the type ”animal” was impossible to predict, although sample sizes of 17 for the train data set also mark the smallest proportion of all ingredient type samples, making training hard. The types ”flower”, ”herb”, ”plant” and ”spice” make the second most difficult to predict classes, with F1-scores between 50 and 69 percent. The types ”fruit”, ”vegetable” and ”cereal/crop” are the second most easy to predict classes with scores between 74 and 80.5 percent. Lastly, predictions for ”alcoholic_beverages”, ”dairy”, ”fish/seafood”, ”meat”, ”nut/seed/pulse” and ”plant_derivative” seem to be the most successful with macro-F1-scores of at least 87 and up to 93 percent.

The test results correspond only to some extend directly to these training observations. The single sample of ”animal” was not correctly classified, as was to be expected by the training results. Interestingly, almost all of the other classes perform similarly well or often better on the test data. The type ”spice” however performs significantly worse with only 33% instead of 65% F1-score, with an especially bad value of 20% recall - keeping in mind that the test size for this type is only 5 samples. Other than ”spice”, only ”fish/seafood”, ”nut/seed/pulse” and ”vegetable” take a hit. But is merely a small one with 5 percent points worse F1-score and they remain relatively well predicted with a F1-score of at least 70%.

Looking at the confusion matrix of the cross-validation test data, figure 3.1 on page 31 ”animal” erroneously often seems to be predicted as ”plant” or ”plant_derivative”. The confusion between p.e. ”animal” typed ingredients like ”lard” or ”honey” are most likely similarly used to how the ”plant_derivative” types like ”olive_oil” or ”maple_syrup” are being used. And as there are some ”animal” ingredients containing ”_oil”, they will probably be wrongly classified likewise as that is usually the clear indicator for a ”plant_derivative” as opposed to p.e. a ”plant” type ingredient. Other than I infer that the reasons for the bad prediction performance lie in the fairly diverse (”egg”, ”gelatin”) and sometimes odd kinds of items like ”musk” in the ”animal” type ingredients, while at the same time being by far the smallest of all groups.

”flower” and ”herb” type ingredients are often mistaken for ”plant” types. This may be because all three groups contain Latin derived ingredient names. P.e. ”ormenis_multicaulis” is of type ”plant”, an ingredient also known as Wild Moroccan Chamomile, while ”chamomile” is labeled as ”flower” in the dataset. Also of type ”flower” is ”rosa_centifolia”, a type of rose with edible flesh and petals; ”piper_chaba”, an ingredient known as a type of long pepper as well as ”myrtus_communis”, better known as myrtle, are labeled ”herb”. The ”plant” type ingredients are the second largest group of ingredients and yet seem to cover many diverse aspects of food, making exact predictions for similar types difficult.

The ”fish/seafood” type, being rather well predicted on the training data (figure 3.2 on page 32), gets confused with fruit 40% of the time when using the test data. I find it hard to pinpoint the exact reason for this; the rare as most words in the test set should be rather obvious to classify (p.e. ”whitefish, tuna, sweetfish). The only rare ”fish/seafood” labeled word in the test set is ”trassi”, a shrimp paste, while there are no ”flower” words beginning with ”tra” so an explanation is not apparent.

Also worth mentioning about the test data confusion matrix is the fact that ”spice” type ingredients get most often confused with ”plant” ingredients, which might be because two of the five spice ingredients in the test data have a Latin name (p.e. ”zanthoxylum_piperitum”, also known as the Japanese Sansho pepper), like most of the ”plant” ingredients.

4.1.3 Ingredient Type Visualization

Figure 3.20 on page 49 displays what became apparent before in the results sections about the ”plant” and ”plant_derivative” type ingredients: even the best performing ”im2rec_fasttext” embedding has difficulties with many outliers or in other words not clearly divided clusters.

Especially the ”plant_derivative” type seems to be represented almost everywhere, despite having a great center cluster in the center left. Still good clustering can be identified, despite the large number of classes (14) and the overlaps between them. It should be noted that lower perplexity values than 40 for the t-SNE worsened the visibility of clusters.

The ”herb” and ”spice” type ingredients seem to form rather bad clusters, although occasionally still local aggregations form. Yet they are on the other hand divided from the more concentrated clusters to at least some degree. This can not be said about the ”animal” type ingredients which are all over the graph - some buried in the ”plant” type or coinciding with ”plant_derivative” ingredients which don’t make a clearly defined group either - which explains the bad predictive power observed before. Great clusters are being formed by ”fruit” (upper right), ”alcoholic_beverages” (far upper right), ”meat” (right), ”dairy” (far right and

far left), and ”nut/seed/pulse” (lower left); ”fish/seafood” has a nice cluster on the lower right but occasional fragments on the top of the plot.

The graph compares nicely to the Principal Component and the Kernel Canonical Correlation Analysis based visualizations of the flavor embeddings of the same dataset by De Clercq 2014, p. 35.

Figure 3.20 on page 49 is more airy, space-utilizing and with more distinct clusters, while ibid. her graphs offer the advantage of showing the ingredient names. The grayish and light blueish or light greenish colors used by ibid. make distinguishing the right ingredient type hard on the other hand. The PCA algorithm tends to create generally much denser visualizations that are more difficult to decipher looking at the figures in ibid., p. 29, 31.

Figure 2 in Ahn et al. 2011, p. 3 is a great example how the existing figure 3.20 on page 49 could be improved: adding names for a sample of statistically significant or in other ways representative ingredients without losing much if any clarity and distinguishability. In the same line, different sizes according to ingredient importance could be added, as shown in ibid., p. 3. Still the t-SNE visualization at hand fares remarkably well against the manually designed network graph by ibid., p. 3.

The t-SNE visualization of fastText embeddings of the same dataset by Sauer et al. 2017, p. 12 is in contrast much harder to read. Most likely the visualization could benefit from a different, possibly larger perplexity value as will be seen later in the cuisine dataset visualizations results.

More t-SNE projections using the ”im2recipe+” embedding that is also based on the 1 million recipes as the one at hand are given by Marin et al. 2018, p. 9. The visualization at hand, figure 3.20, is easier to read than these, not only because of the larger size, but also the more distinguishing choice of colors and the added marker symbols. Even though the semantic categories used by them are likable, applying the learned embedding to an external dataset like here resulted in a more appealing visualization that could furthermore also be used to compare different versions of their ”im2recipe” algorithm based embeddings with and without semantic categories.

4.2 Cuisine Classification

4.2.1 Results

The cuisine dataset is even more skewed than the ingredients dataset. The merged cuisine dataset is skewed as well. With ”Western” accounting for almost three fourths of all ingredient lists/recipes, and ”SouthAsian” only accounting one percent, the macro-results will thus again be covered in more detail than the micro-scores.

Interestingly about the micro-scores this time though is the fact that every embedding made the LinearSVC classifier perform better than a simple model predicting only the most common class all the time, with no micro-score worse than 78% on either train or test data, see table 3.5 on page 33 and table 3.6 on page 33. This may have to do with the much larger sample size per class compared to the ingredients type prediction. On top of this, the comparison between different embeddings is made easier, as the sample sizes per class are practically identical between the embeddings, see the ”X_cvtrain” and ”X_test Shape” columns in table A.2 on page 72.

The rare classes are larger in absolute terms as well: The rarest merged cuisine class, ”Eastern”, accounts p.e. for 62 ingredient lists in the case of ”wiki_fasttext_concat”, see table 3.8 on page 34. Nonetheless vocabulary size still matters, as embeddings with a very small one might have many ingredient lists with only one representable ingredient left, while the ”fastText” embeddings will retain all or most of them and will hence often have a better representation of the recipe.

The macro-F1-scores of the training range from 64 to almost 68 from worst to best, while the ones of the test data from 54 to 60.6%. I infer that the features learned to predict the merged cuisine region are not as easily transferable to unseen data. Taking into account only the best of the three embedding variants per embedding with the exception of ”tfidf” which has no variants, the macro-F1 scores on the training data vary only between 65 and 68 percent, the test scores between 58.1 and 60.6.

What is striking is that the relatively simple ”tfidf” embedding produces almost as good a prediction F1-score average as the best model, ”wiki_fasttext_concat”. The efficiency of the sparse TF-IDF embedding results in the by far fastest training time of only 6 minutes. Even accounting for the fact that this training benefits from preprocessing and caching as it is done as the last embedding to train and thus tripling that number could be considered (equaling 18 minutes), this is still around four times as fast as the next best training time, ”wiki_fasttext_concat” with 73 minutes. This is especially impressive considering it doesn’t even require the large corpora and neural network the embeddings needed.

Regarding training time and model complexity to cuisine prediction efficiency, the ”im2rec_joint_null” and ”-tfidf” embeddings show disappointing results. The extra dimensions needed cause an extremely long training duration, without delivering better results than the much more efficient ”im2rec_base_tfidf”. Even so this time they offer much better predictions compared to ingredient type prediction, with the ”im2rec_joint_avgconcat” embedding even faring favorably against ”tfidf” considering only the average F1-score. The instruction representation of the 1 million recipes data being present in the ”im2rec_joint_avg” embedding may finally show palpable advantage compared to the simpler the ”_null” variant.

Neither the TF-IDF weighting of each ingredient to compute the sum of each embedding ingredient vector nor the concatenation with the ”tdidf” embedding itself necessarily lead to better results. ”googlenews” and ”im2rec_joint_null” perform around one percent point worse when adding TF-IDF information, while the other embeddings, ”wiki_fasttext”, ”im2rec_joint_avg” and ”im2rec_fasttext” benefit considerably with at least 2, but up to 4 percent points.

Surprisingly, from all the 1 million recipes embeddings ”im2rec_base” performs best. Regarding macro-precision and macro-F1 averages, the best embedding is ”wiki_fasttext_concat” with a 60.6% average F1-score.

The logistic regression prediction on the test set performed similarly to the linear classifier, but with worse results generally, see appendix A. The training times were similar as well with the chosen grid search hyperparameter tuning. For these reasons, only the linearSVC classifier is explored in more detail.

4.2.2 The Best Embedding's Results

The detailed results reports for "wiki_fasttext_concat" for the cross-validation training and the separate test data set are fairly similar, see table 3.7 on page 34 and figure 3.3 on page 34. However the ability to predict the merged "SouthAsian" and "Southern" cuisine regions is considerably worse, from 77 to 60% F1-score and 50 to 43% respectively. I infer that the training did not generalize well to the unseen test data set and the classifier overfits to features of the training set for these regions, despite using 10-fold cross-validation. The test results for the other two merged cuisine regions, "Eastern" and "Western", are almost the same as the training results. It is noteworthy that "SouthAsian", by far the smallest class of all four with only one percent of the whole data, achieves a respectable 51% F1-score on the test set. I infer that this cuisine is quite distinctive and homogeneous and can be used to classify unseen data even though there is much less training data available compared to p.e. the "Southern" merged cuisine region. Speaking of merging, the original cuisines inside "Eastern" (table 2.3 on page 20), "EastAsian" and "SoutheastAsian" may be too diverse to infer their characteristics to unseen data, especially given the still relatively small size of this merged cuisine region.

As can be inferred from figure 3.3 on page 34 and figure 3.4 on page 35, the classifier overfits to the "Western" cuisine and falsely predicts ingredient lists labeled for all other cuisines as it. This makes sense, as the "Western" data dominates with a weighty 79.3%. Distinguishing between "Western" and "Southern" seems to be especially hard for the classifier. In almost half the cases during training and more often than not during testing the classifier will falsely predict the "Western" cuisine instead. I infer the ingredients used in both recipe lists must be relatively simple, so that the classifier can't reliably tell the difference and will then make a safe bet for the most common cuisine, "Western".

The "Western" cuisine also seems to incorporate many ingredients similar to the "Eastern" and "SouthAsian" cuisine regions. The classifier has even more difficulty successfully applying the learned weights to the unseen test data, see figure 3.4. Just like for the "Southern" cuisine, the "safe bet" will lead to the wrong prediction in around one fourth of the time where it should instead have been "Eastern" or "SouthAsian".

Comparing other cuisine classification confusion matrices such as Kazama et al. 2018, p. 4 that are barely colored, the use of a heatmap visualization such as the ones at hand clearly helps interpreting. It can also be seen that a less fine grained matrix like here with only four cuisine regions makes the interpretation much easier, while still presenting a compelling task. Comparing to the cuisine classification confusion matrix by Kalajdziski et al. 2018, p. 1248 that does not have labels, it can be seen that using labeled axis benefits interpretation largely. Even though their matrix offers color variations when a threshold is met, the confusion matrix at hand is more intuitive to understand. This affirms the initial assumption that a heatmap would be a great way to visualize a confusion matrix.

4.3 Discovered Food Topics

4.3.1 Choosing the Number of Topics

As can be seen figure B.1 on page 77, all of the four metrics for topic models with α=1/k and constant β=0.1 show roughly the same general trend of steep decline followed by a downward sloping curve as the number of topics goes up to 252. That being said, the first two metrics, "arun_2010" and "cao_juan_2009" are to be minimized in order to get the best topic models, whereas according to "coherence_mimno_2011" and "loglikelihood" the best topic models are found where their graphs are at their maximum.

For "coherence_mimno_2011" and "loglikelihood", the highest values are between 12 and 32 models, although there is a very long plateau between 62 and 162 topics that should be considered as well.

But at the same time "arun_2010" and "cao_juan_2009" are at their highest between 12 and 32 topics, too. After the steepest decline is over at 32 topics, they are not immediately reaching a plateau but rather much more slowly downward sloping. "arun_2010" is already for the most part at its lowest at 152 topics. "cao_juan_2009" on the other hand is at the lowest from 232 topics onwards. It seems hard finding a good balance between the four metrics or even ruling out the worst number of topics.

Because of this uncertainty, additional weight is decided to be given to "coherence" due to its correlation to human evaluation as described in chapter 2. Thus topic numbers of more than 162 should be avoided, while the metrics seem to stabilize only after more than around 42 topics.

In this sense the best topic models for the parameters α=1/k and β=0.1 are judged to lie between 40 and 160 topics.

The four metrics for topic models with variable α as before and variable β=1/10k paint a very different picture, figure B.2 on page 78. The metrics to be minimized, "arun_2010" and "cao_juan_2009" are steadily ascending and maximizing at 252 topics while being at their lowest between 12 and 52 topics.

The metrics to be maximized, "coherence_mimno_2011" and "loglikelihood" are almost the inverse. The first plateaus at high values up until 62 topics before almost linearly descending until 252 topics, while the latter steeply declines from 2 topics onwards until 52 topics, where it almost plateaus until reaching its lowest values from 172 to 252 topics.

Combining these results, "coherence_mimno_2011" and "loglikelihood" depict good models between 12 and 172, with a preference for models up to around 102, while "arun_2010" and "cao_juan_2009" suggest models between 12 and around 82, but preferable lower than 152. Trying to balance while avoiding unstable metrics while again opting to making the coherence metric tip the scales, the best models are judged to have between 50 and 150.

Using the similarly sized although a bit more complex "Yummly66K" dataset, Min et al. 2018a, p. 8 found 20 - 200 topics reasonable using the perplexity score on a separate test set. This compares nicely with the presented metrics of the models at hand.

From both parameter sets and models, two specific instances are chosen and explored in the following subsection, naming them topic model №1 and №2 - both with 50 topics.

4.3.2 Choosing Topic Model №1 or №2

Comparing the top ten words for the first ten of 50 topics for topic model №1 and №2 (table 3.9 on page 36 and table 3.10 on page 36), a couple of differences as well as similarities emerge:

All but one topic of model №2 start with a cuisine as the most important term, while for topic model №1 only two out of this sample are like that. This general trend is also seen over all topics, see appendix B on page 77.

Both sets of topics show a number of topics with several cuisines in the first ten words. There is also a comparably reasonable division between p.e. sweet and savory ingredient topics. The ingredients seem to make sense for the cuisine named, with the topics seemingly defining generic variations of certain types or styles of dishes, p.e. ingredients for a sweet cake or a hearty casserole.

The heavier use of cuisine terms imply that topic model №2 its topics are more generic, as the ”ingredient” ”NorthAmericanCuisine” is much more generic than a ”real” ingredient. This confirms the assumption about the β parameter, as topic model №1 with its relatively larger β is producing more specific topics.

This is also reflected in the t-SNE visualizations, figure 3.23 on page 51 and figure 3.24 on page 52, where the clusters for each cuisine are a bit more clearly separated. P.e. the pale green ”NorthAmerican” cuisine clusters are a bit more pronounced or less ”intruded” by other cuisines. On top of this many of the outliers recipe points around the main clusters at the far end of the graph coincide with a recipe of another cuisine for the t-SNE of topic model №2, while in the case of №1 they are mostly nicely separated points.

The pyLDAvis screenshots (figure 3.5 on page 37 and figure 3.5 on page 37) show a relative even marginal topic contribution for topic model №1, while topic model №2 tends to have more variation with a few larger and smaller circles in the graph. It is important to note that the topic numbers are not the same, as they are numbered by marginal contribution only in the pyLDAvis visualization.

The topics represented by larger circles must be relatively generic if they contribute so much to a lot of recipes. The smaller circles, an example given in figure 3.7 on page 40, are specific, but still seem to feature the very broad term ”NorthAmericanCuisine” as the most important term, confirming the arguments above.

Both models exhibit sound topics, which imply that 2.000 iterations as used are a good amount for this dataset given the time spent - less than two hours for both model sets. The use of 3.000 iterations like in Min et al. 2018a, p. 8 might still be more appropriate given more complex datasets as they used the ”Yummly66K” corpus, which on top of cuisine also includes course labels.

It is decided that the better specificity of topic model №1 it is more appealing and intuitive and therefore describes the dataset better. As such it will be examined in more detail below.

4.3.3 Exploring and Analyzing Topic Model №1

As can be seen in figure 3.10 on page 43 showing the marginal topic distribution histogram of topic model №1, most topics cover around 1.2% of the corpus, while there are ca. 5 topics that cover more than 1.5% and 15 topics that cover less than 1.1%, with no bin of topics covering much more than 1.7 or much less than 0.07%. It is deduced that the topics are fine-grained and evenly specific reflecting the variety of recipes in the corpus, with less dominating generic topics.

In the top ten most and least distinct words of table 3.11 on page 41, the two cuisines listed, ”SoutheastAsian”, ”SouthAsian” and ”NorthAmerican” are interesting finds that strike:

As the majority of recipes by far is of type ”NorthAmerican”, it makes sense that a lot of topics are generated to describe different aspects of it in relative detail compared to other cuisines with much fewer recipes. There, more fine-grained distinctions between several latent cuisine ”subtopics” such as desserts, vegetable dishes or soups may be relatively much more difficult given the relatively small sample size. Thus the appearance of many topics with the ”ingredient” ”NorthAmericanCUISINE” intuitively makes sense, which implies the word ”NorthAmericanCUISINE” is a relatively vague indicator of the actual latent topic.

The Southeast Asian cuisine on the other hand will by nature of the input skew be more rare. But this result is still speaking of the high distinctiveness of the two cuisines, as three

even rarer cuisines such as ”EasternEuropean”, ”NorthernEuropean” or ”African” are not in this top ten list.

Apropos Asian cuisines: the ingredients ”lemongrass”, ”fenugreek”, ”seaweed”, ”galanga” and ”wasabi” are frequently used in East to Southeast Asia (”lemongrass” and ”galanga” especially in Southeast Asia), and seem to be quite specific in describing topics. This confirms the intuition that the appearance of one of these ingredients will immediately, to a relatively large degree make only certain types of dishes or dish categories be probable, p.e. a savory East or Southeast Asian recipe.

Weighting the distinctiveness of an ingredient by its frequency, the most salient ingredients in table 3.12 on page 41 leave the right side of the previous list identical. This is because all of these words except for ”NorthAmerican” are very rare, see table 3.13 on page 41 and therefore do not get multiplied with a large number of frequency. ”NorthAmerican” is the most frequent ”ingredient”, but most likely because of its low ”distinctiveness” value described before still low on the ”saliency”. The least salient words statistics would probably reveal more interesting ingredients if there were no ingredients with only one occurrence over all recipes, which unfortunately is the case here.

”wheat”, ”egg”, ”milk”, ”butter” and other more very versatile ingredients get boosted on the other hand, because they appear very often in a variety of recipes and by that in a lot of topics. Interestingly, the list is very similar to the TF-IDF scores.

But there is a slight but important difference to be recognized, p.e. the most two common words may be ”egg” and ”wheat” (TF-IDF), but according to their frequency in the generated topics that describe the recipes, the order is reversed. This implies that despite ”egg” being more common than ”wheat” in the recipes themselves, ”egg” is still more distinctive than ”wheat” in terms of defining a latent topic, in other words ”egg” adds more information to a topic (or about a recipe for that matter).

The word clouds offer a good way to see the importance of a word contributing to a topic, making it easier to detect the general theme. A sample of the most interesting latent topics are described here:

- Topic 5 is interpreted to be about cakes or desserts from Southern or Western Europe - it compares nicely to topic 68 by ibid., p. 9 featuring similar ingredients (based on the ”Yummly66K” corpus).
- Topic 7 is about ingredients that resemble Thai or Vietnam dishes.
- Topic 11 features typical Italian ingredients with a seafood touch that are also popular in the United States due to the Italian heritage.
- Topic 16 is about fruits and to a lesser degree alcohol; a similarity also mentioned by Ahn et al. 2011, p. 2 - while the t-SNE visualization of figure 3.20 on page 49 puts both types also on the same general direction of the graph (upper right).
- Topic 17 describes the main spices of a curry mixture that for the most part reasonably gives more importance to spices that get used more liberally (p.e. cumin vs fenugreek), while adding the fitting ingredients such as coconut, onion, chicken, rice and yogurt make it an authentic South Asian dish.
- Topic 18 is about truly North American ingredients, suitable p.e. for cup cakes, featuring native ingredients such as peanut butter, pecan nuts and cane molasses.
- Topic 21 could be interpreted to be about pizza and pasta dishes, again popular in Southern Europe and North America due to the Italian heritage.
- Topic 28 is especially interesting, showing ingredients typical for the Middle Eastern such as chickpea, mint, lamb, cinnamon, olive oil, parsley and lemons - these could as well be taken from an actual Moroccan recipe, thereby intuitively explaining the ”AfricanCuisine” match as well.

- Last but not least, topic 30 is about a latent topic that would describe Japanese Sushi variations well - it compares nicely to topic 68 by Min et al. 2018a, p. 9 featuring similar ingredients (based on the ”Yummly66K” corpus).

The recipe-to-topic heatmaps for a sample of eleven cuisine specific recipes and the nine topics just mentioned (figure 3.8 on page 42 and figure 3.9 on page 42) display the idea of how recipes are embedded by topics. Only the first five ingredients per recipe are shown, there can actually be many more. The actual embedding would contain all 50 topics so that the column probability values per row sum up to one.

The heatmap of ”EastAsian” recipes shows a clear centering around topic number 30, which makes a lot of sense given the above description.

Recipe number 2 and 4 are a lot about topic 7, assumingly because of the ingredient ”garlic”.

The third to last recipe matches topic 16 highly, as it contains several fruits (kiwi, cherry, tangerine).

The second to last recipe could be about some sort of East Asian dessert (gardenia is also known as the flavoring for Jasmine tea), being defined to a certain degree by the ”Southern European Dessert” topic number 5, probably due to the citrus component.

The last recipe seems to also be about desserts, making it match topic 18 and to a lesser degree topic 30 - probably because of the ”rice” ingredient.

The ”SouthernEuropean” recipes shown in the second heatmap are less centered around one single topic, although topic number 11 seems to describe some of the recipes quite well.

Starting with the last three recipes, they all seem to some extent be defined by the ”Southern European Dessert” topic, the third to last probably because of the ”hazelnut” ingredient despite describing a savory dish as it seems with olive oil and thyme. The second to last's use of fruit - ”watermelon” and ”lemon” to be exact - as well as ”mint” most likely cause it to be defined by topic 16 as well.

The sixth recipe interestingly matches topic 30 which was described as a ”Sushi”-style topic - it is assumed that this is because of its seafood element (”squid”).

The first recipe consists of many spices and is thus fittingly embedded to a large degree by the ”curry topic” number 17.

4.4 Cuisine Visualizations

4.4.1 Visualization of the Cuisine Topic Embedding

Three variations a 2 dimensional projection and visualization of topic model №1 can be seen in figure 3.21 on page 50, figure 3.22 on page 51 and figure 3.23 on page 51.

Using perplexity=5, the first graph paints a relatively dense picture of the cuisines, with a very high number of recipes being outside the main cluster seemingly isolated and random.

Perplexity=30 alleviates the outliers and loosens the central cluster, while perplexity=50 loosens the central cluster even more while at the same time binding more outliers to existing clusters.

The perplexity=5 visualization is the least clear and makes distinction between cuisines hard, as they directly border each other. The many outliers with many recipes of different cuisines coinciding is difficult to interpret as well.

The visualizations with a larger perplexity value are much more coherent. It is also much easier to detect the different cuisines. The largest perplexity value, 50, seems to fit the topic model embedding the best, as there are the least outliers, the outliers are mostly single recipes of only one cuisine at a time, and the clusters inside are the most distinguishable.

4.4.2 Visualization of the TF-IDF and the Best Cuisine Embedding

There are three visualization for the TF-IDF embeddings, figure 3.25 on page 53, figure 3.26 on page 54 and figure 3.27 on page 55.

The first graph shows all cuisines, the other two graphs only the four merged cuisine regions. In the first graph it is a bit more difficult to see all the different cuisines compared to the previous topic model based ones. This reflects the much simpler representation that the sparse "tfidf" is, compared to the dense and elaborate topic model embedding.

Still looking a bit more closely, several clusters can be detected: "EastAsian" in pale purple "o"s on the bottom right, next to the orange cluster of "SoutheastAsia", which is next to the light green cluster of "SouthAsia", next to a smaller and a little less concentrated black cluster for the "MiddleEastern" cuisine.

There are also two pink clusters in the upper and lower middle representing the "LatinAmerican" cuisine, and a very large yellow cluster for "EasternEuropean". The "WesternEuropean" and "NorthAmerican" cuisine recipes are very close together, confirming the hierarchical clustering results described in Chapter2. While this is also the case for the topic model embedding visualization, the topic model manages to form more distinct clusters that even though mixing the two cuisines are often clearly dominated by one of the two. "African" recipes can almost not be recognized in the TF-IDF graph, while more easily in the topic model one (center bottom in figure 3.23 on page 51).

The visualizations described so far compare nicely to similar cuisine labeled recipe or ingredient visualizations: The use of special markers as well as a well defined set of colors as used in this work definitely improves visibility of a complex graph like this with 11 cuisines, comparing it p.e. to a visualization used in Altosaar 2017a.

Kazama et al. 2018, p. 5 applied an interesting approach using a newton diagram using spectral drawing that might be useful if the number of cuisines becomes even larger, while they resorted to a large zoom to ensure visibility when a similarly complex graph was shown (ibid., p. 6).

figure 3.26 on page 54 and figure 3.27 on page 55 is based on the same t-SNE transformation as the one before, but this time only colored with the four merged cuisine regions. It can be seen that black "Southern" and light green "Western" are barely separated, while sea green colored "Eastern" exhibits a well defined cluster on the bottom right. Red colored "SouthAsian" is showing a larger, not as much coherent cluster centered more to the right, but also scattered over the whole graph.

The t-SNE projection that is fed the number of classes to be plotted right from the start, see figure 3.27 on page 55, shows a much better cluster for "SouthAsian" in this regard. All the other cuisines are little bit better separated as well, with the black colored "Southern" cuisine now concentrating more clearly on the top to top left, although still being intertwined with the light green "Western".

It is remarkably similar to the t-SNE projection of the best cuisine prediction embedding, "wiki_fasttext_concat", see figure 3.28 on page 56. The black colored "Southern" cuisine region recipe cluster appear just a little bit more concentrated here, maintaining a clearer separation from the red colored "SouthAsian" cuisine and leaving slightly less recipes on the lower right green "Western" regions. This reflects the similar, but slightly better classification performance examined before.

Chapter 5

Conclusions and Outlook

The food embeddings based on the 1 million recipes dataset presented and reviewed showed that a smaller, but domain specific corpus can produce embeddings similarly capable of predicting ingredient types and cuisines compared to embeddings based on much larger corpora such as Wikipedia or Google News. Nonetheless, even general domain embeddings revealed great and sometimes the best performance in predicting ingredient types and cuisines. This may come as somewhat of a surprise especially regarding the "google-news" embedding, as intuitively news are less associated with food.

"n"-gram based embeddings using "fastText" performed best, showing their advantages applied to the particularities of linguistic food domain resources as ingredient words are more likely to be detected. On the other hand, a very simple and sparse TF-IDF representation was only slightly worse in predicting ingredient types. To put the performance of this benchmark corpus intrinsic representation into perspective, a benchmark of it on a separate corpus would be beneficent.

The results of the complex and long to train "im2recipe-joint"-embeddings were surprising, as in the ingredient prediction task they were by far the worst of all embeddings, while in cuisine prediction being basically only just as good as the much simpler "Word2Vec" embedded "im2recipe" baseline. Concatenating TF-IDF information to an already good embedding - "wiki_fasttext" - on the other hand yielded the best embedding for the cuisine prediction task.

Furthermore it would be interesting to see, if the general domain embedding's performances could be tuned by adding either domain specific corpora to the general domain corpus and training on this enlargened corpus, or by combining them with the already trained domain specific embeddings.

The general setup of creating a food embedding benchmark was successful and is generic enough to be easily applied to more food text resources. Techniques such as detailed results per class and confusion matrix heatmaps showed how even complicated prediction task results can be clearly arranged. The description at hand of how the powerful yet free libraries were used makes reproduction of the presented results very feasible to the interested researcher. Combining this with the extensive list of available resources of chapter 2 section1.4.1 this should help in creating new food embedding benchmarks in a much faster way.

While crowd-sourced recipes such as the ones presented offered interesting insights, it would also be interesting to find out how different embeddings would fare that would be based on an even more specialized corpus written by experts like professional chefs - consequently benchmarking different "food chef embeddings".

The visualizations of the two datasets used, ingredient types and cuisines, have been shown to offer a further way of comparing embeddings. The intuitively understandable t-SNE projections can easily be tuned to the peculiarities of the dataset and subsequent benchmark results. The care put into the graphs such as the use of custom color palettes and marker sets customized to best explain the complex datasets explored paid off in the form of well readable visualizations. This should hopefully inspire others in creating similar food embedding visualizations, improving the understanding of distributional food semantics via transformation to semantic graphical distributions.

One part of the ”Ahn” corpus that would be interesting to use for another embedding benchmark is the flavor compounds data. As Mouritsen et al. 2017 write, the application of chemistry to the gastronomy can create a view of the ”Chef as Flavor Scientist”.

The presented datasets overview lists a couple of other great resources around flavors, although the prediction task would most likely be more difficult to conduct. The ingredients type and cuisine classifiers were about a multi-class prediction task with less than 15 classes, with only one class per sample - a flavor compound prediction would often revolve around lists of hundreds if not thousands of possible flavor compound labels per sample. This would not only make preprocessing more complex, but also loss calculation and evaluation metrics.

Benchmarking the number of topics as well as the model creating parameter β yielded in both chosen topic models interesting and intuitively understandable topics, with the smaller β valued topic model producing more generic topics confirming previous assumptions.

The various metrics about finding the right number of topics were easy to apply thanks to the ”tmtoolkit” library. There is also a recently added function for measurements on a held-out set of documents (Konrad n.d.), that has not been used so far in the examples section of the library, but would be interesting to benchmark and compare with the existing ones. The metrics at hand could use more a thorough investigation in order p.e. to automatically assign them a weight so that a meta score function could be generated, helping to automate the evaluation process further and narrowing down the range of viable topic numbers.

The chosen topic model with 50 topics showed intuitively understandable results in characterizing the recipes of the dataset. The topics were easy to associate with kitchen styles around the world. It was shown how Topic Modeling provides an excellent technique to explore, represent and visualize food datasets. Again thanks to the powerful ”tmtoolkit” library, various statistics could easily be plotted, the topic word clouds being probably the most decorative results, worthy of being shown in a gourmet’s kitchen.

Going forward, the very interesting ideas of using distributional semantics to translate the meaning of food investigated by (Kazama et al. 2018; Nobumoto et al. 2017) could at least indirectly be measured in an automated fashion using the presented benchmark and visualization methodology - under the assumption, that embeddings that are better than others in predicting a certain cuisine and other labels are more appropriate in formulating this kind of semantic translation, too.

Appendix A

Classification Results Appendix

A.1 Support Vector Machine Results

A.2 Ingredient Classification Tuned Hyperparameters

Embedding	C	class_weight	dual	max_iter	multi_class	tol	minutes	X_cvtrain	X_test Shape
googlenews	0.02	balanced	False	3000	ovr	1e-06	7	(456, 300)	(43, 300)
wiki_fasttext	0.05	balanced	False	3000	ovr	1e-06	41	(1384, 300)	(146, 300)
im2rec_joint_null	1	None	False	3000	crammer_s	1e-06	20	(339, 1024)	(37, 1024)
im2rec_joint_avg	0.01	None	False	3000	crammer_s	0.0001	20	(339, 1024)	(37, 1024)
im2rec_base	0.05	balanced	False	3000	ovr	1e-06	6	(339, 300)	(37, 300)
im2rec_fasttext	0.05	None	False	3000	ovr	1e-06	27	(1384, 300)	(146, 300)

Table A.1: LinearSVC ingredient type classification tuned GridSearch hyperparameters.

Table A.1 shows the input shapes and the tuned hyperparameters of the LinearSVC classifier using cross-validation grid search. As described in Chapter 2, the training and test data was split using stratified sampling with a ten to one ratio before beginning the training pipeline. Then each of the the six embeddings was used to convert the ingredients of the training and test set into a representation.

The number of convertible ingredients and the number of dimensions can be seen in the columns ”X_cvtrain” and ”X_test Shape”. The LinearSVC classifier with the ”googlenews” embedding p.e. had only 456 matches of the ingredient names in its vocabulary and could therefore be used as training input, with each ingredient being represented by 300 dimensions. The test data was limited to 43 ingredients.

The ”fastText” embeddings were able to use all 1384 ingredients as input. This is because they are able to represent also ingredients which are not exactly matched in their vocabulary due to their ”n”-gram nature, as explained in Chapter 2.

The minutes column shows the time it took for the whole training pipeline to complete. ”max_iter” shows the maximum iterations the training was run.

A.3 Cuisine Classification LinearSVC Hyperparameter Tuning

Recipe Embedding	C	class_weight	dual	max_iter	minutes	X_cvtrain	X_test Shape
tfidf	100	balanced	True	1000	6	(50850, 381)	(5648, 381)
googlenews-sum	0.1	balanced	True	1000	188	(50828, 300)	(5648, 300)
googlenews-tfidf	10	balanced	True	1000	84	(50828, 300)	(5648, 300)
googlenews-concat	5	balanced	True	1000	72	(50828, 614)	(5648, 614)
wiki_fasttext-sum	0.1	balanced	True	1000	137	(50850, 300)	(5648, 300)
wiki_fasttext-tfidf	10	balanced	True	1000	78	(50850, 300)	(5648, 300)
wiki_fasttext-concat	10	balanced	True	1000	78	(50850, 681)	(5648, 681)
im2rec_joint_null-sum	5	balanced	True	1000	388	(50836, 1024)	(5647, 1024)
im2rec_joint_null-tfidf	100	balanced	True	1000	184	(50836, 1024)	(5647, 1024)
im2rec_joint_null-concat	50	balanced	True	1000	148	(50836, 1326)	(5647, 1326)
im2rec_joint_avg-sum	10	balanced	True	1000	394	(50836, 1024)	(5647, 1024)
im2rec_joint_avg-tfidf	100	balanced	True	1000	184	(50836, 1024)	(5647, 1024)
im2rec_joint_avg-concat	100	balanced	True	1000	150	(50836, 1326)	(5647, 1326)
im2rec_base-sum	2	balanced	True	1000	149	(50836, 300)	(5647, 300)
im2rec_base-tfidf	50	balanced	True	1000	80	(50836, 300)	(5647, 300)
im2rec_base-concat	50	balanced	True	1000	69	(50836, 602)	(5647, 602)
im2rec_fasttext-sum	1	balanced	True	1000	138	(50850, 300)	(5648, 300)
im2rec_fasttext-tfidf	50	balanced	True	1000	76	(50850, 300)	(5648, 300)
im2rec_fasttext-concat	10	balanced	True	1000	73	(50850, 681)	(5648, 681)

Table A.2: LinearSVC cuisine classification tuned GridSearch hyperparameters.

Table A.2 shows the input shapes and the tuned hyperparameters of the LinearSVC classifier using cross-validation grid search. As described in Chapter 2, the training and test data was split using stratified sampling with a ten to one ratio before beginning the training pipeline. Then each of the the six embeddings was used to convert the ingredients of the training and test set into a representation.

The number of convertible ingredient lists and the number of dimensions can be seen in the columns "X_cvtrain" and "X_test Shape". A recipe was used as long as at least one ingredient could be converted.

The LinearSVC classifier with the "googlenews" embedding p.e. had the subset of 50828 ingredient lists as training input, with each list being represented by 300 dimensions. The test data was composed of 5648 ingredient lists. The concatenated dimensions are 300 dimensions from its original shape plus 314 dimensions of the "tfidf" ingredient embedding which were in the "googlenews" embedding's vocabulary.

The "fastText" embeddings were able to use all 381 ingredients as additional "tfidf" concatenated embedding size and all ingredient lists. This is because they are able to represent also ingredients which are not exactly matched in their vocabulary, as explained in Chapter 2.

The minutes column shows the time it took for the whole training pipeline to complete. "max_iter" shows the maximum iterations the training was run.

A.4 Cuisine Classification Logistic Regression Results

A.4.1 Tuned Hyperparameters

	C	class_weight	dual	iter	multi_cl.	solver	minutes	X_cvtrain	X_test Shape
tfidf	500	balanced	True	500	auto	liblinear	8	(50836, 302)	(5647, 302)
googlenews-sum	2	balanced	True	500	auto	liblinear	177	(50828, 300)	(5648, 300)
googlenews-tfidf	200	balanced	True	500	auto	liblinear	77	(50828, 300)	(5648, 300)
googlenews-concat	200	balanced	True	500	auto	liblinear	72	(50828, 614)	(5648, 614)
wiki_fasttext-sum	5	balanced	True	500	auto	liblinear	170	(50850, 300)	(5648, 300)
wiki_fasttext-tfidf	100	balanced	True	500	auto	liblinear	84	(50850, 300)	(5648, 300)
wiki_fasttext-concat	100	balanced	True	500	auto	liblinear	80	(50850, 681)	(5648, 681)
im2rec_joint_null-sum	100	balanced	True	500	auto	liblinear	390	(50836, 1024)	(5647, 1024)
im2rec_joint_null-tfidf	500	balanced	True	500	auto	liblinear	160	(50836, 1024)	(5647, 1024)
im2rec_joint_null-concat	500	balanced	True	500	auto	liblinear	148	(50836, 1326)	(5647, 1326)
im2rec_joint_avg-sum	50	balanced	True	500	auto	liblinear	389	(50836, 1024)	(5647, 1024)
im2rec_joint_avg-tfidf	500	balanced	True	500	auto	liblinear	160	(50836, 1024)	(5647, 1024)
im2rec_joint_avg-concat	500	balanced	True	500	auto	liblinear	148	(50836, 1326)	(5647, 1326)
im2rec_base-sum	10	balanced	True	500	auto	liblinear	170	(50836, 300)	(5647, 300)
im2rec_base-tfidf	200	balanced	True	500	auto	liblinear	78	(50836, 300)	(5647, 300)
im2rec_base-concat	200	balanced	True	500	auto	liblinear	70	(50836, 602)	(5647, 602)
im2rec_fasttext-sum	5	balanced	True	500	auto	liblinear	169	(50850, 300)	(5648, 300)
im2rec_fasttext-tfidf	200	balanced	True	500	auto	liblinear	81	(50850, 300)	(5648, 300)
im2rec_fasttext-concat	200	balanced	True	500	auto	liblinear	77	(50850, 681)	(5648, 681)

Table A.3: LogistigRegression cuisine classification tuned GridSearch hyperparameters.

A.4.2 Results Overview

Embedding	micro-p	micro-r	micro-f1	macro-p	macro-r	macro-f1
tfidf	0.812	0.812	0.812	0.610	0.733	0.656
googlenews-sum	0.817	0.817	0.817	0.629	0.729	0.669
googlenews-tfidf	0.809	0.809	0.809	0.626	0.713	0.663
googlenews-concat	0.809	0.809	0.809	0.626	0.709	0.661
wiki_fasttext-sum	0.821	0.821	0.821	0.680	0.693	0.686
wiki_fasttext-tfidf	0.815	0.815	0.815	0.639	0.720	0.674
wiki_fasttext-concat	0.816	0.816	0.816	0.642	0.720	0.676
im2rec_joint_null-sum	0.827	0.827	0.827	0.664	0.692	0.676
im2rec_joint_null-tfidf	0.814	0.814	0.814	0.615	0.731	0.659
im2rec_joint_null-concat	0.814	0.814	0.814	0.615	0.729	0.658
im2rec_joint_avg-sum	0.823	0.823	0.823	0.644	0.712	0.673
im2rec_joint_avg-tfidf	0.814	0.814	0.814	0.616	0.730	0.660
im2rec_joint_avg-concat	0.814	0.814	0.814	0.615	0.729	0.658
im2rec_base-sum	0.823	0.823	0.823	0.653	0.707	0.677
im2rec_base-tfidf	0.814	0.814	0.814	0.616	0.729	0.659
im2rec_base-concat	0.814	0.814	0.814	0.615	0.727	0.658
im2rec_fasttext-sum	0.822	0.822	0.822	0.661	0.718	**0.686**
im2rec_fasttext-tfidf	0.815	0.815	0.815	0.640	0.727	0.677
im2rec_fasttext-concat	0.814	0.814	0.814	0.640	0.726	0.676

Table A.4: LogisticRegression cuisine classification results overview (train data)

Embedding	micro-p	micro-r	micro-f1	macro-p	macro-r	macro-f1
tfidf	0.785	0.785	0.785	0.549	0.637	0.576
googlenews-sum	0.802	0.802	0.802	0.573	0.636	0.591
googlenews-tfidf	0.793	0.793	0.793	0.570	0.618	0.585
googlenews-concat	0.793	0.793	0.793	0.571	0.623	0.588
wiki_fasttext-sum	0.799	0.799	0.799	0.596	0.594	0.589
wiki_fasttext-tfidf	0.788	0.788	0.788	0.563	0.644	0.594
wiki_fasttext-concat	0.785	0.785	0.785	0.558	0.611	0.576
im2rec_joint_null-sum	0.798	0.798	0.798	0.593	0.614	**0.598**
im2rec_joint_null-tfidf	0.787	0.787	0.787	0.552	0.626	0.570
im2rec_joint_null-concat	0.785	0.785	0.785	0.551	0.634	0.576
im2rec_joint_avg-sum	0.797	0.797	0.797	0.577	0.614	0.587
im2rec_joint_avg-tfidf	0.786	0.786	0.786	0.552	0.626	0.570
im2rec_joint_avg-concat	0.786	0.786	0.786	0.552	0.634	0.577
im2rec_base-sum	0.797	0.797	0.797	0.583	0.627	0.596
im2rec_base-tfidf	0.785	0.785	0.785	0.554	0.636	0.580
im2rec_base-concat	0.786	0.786	0.786	0.553	0.635	0.578
im2rec_fasttext-sum	0.786	0.786	0.786	0.577	0.609	0.589
im2rec_fasttext-tfidf	0.780	0.780	0.780	0.561	0.629	0.583
im2rec_fasttext-concat	0.780	0.780	0.780	0.561	0.627	0.583

Table A.5: LogisticRegression cuisine classification results overview (test data)

A.4.3 Detailed Results for the Best Embedding

im2rec_joint_null	precision	recall	f1-score	support
Western	0.714	0.842	0.773	2670.0
Eastern	0.410	0.688	0.514	559.0
South-Asian	0.485	0.503	0.494	7285.0
Southern	0.901	0.876	0.888	40322.0
micro avg	0.818	0.818	0.818	50836.0
macro avg	0.627	0.727	0.667	50836.0

Table A.6: LogisticRegression cuisine classification best embedding detailed results (train data).

im2rec_joint_null	precision	recall	f1-score	support
Western	0.633	0.513	0.567	296.0
Eastern	0.354	0.629	0.453	62.0
South-Asian	0.430	0.420	0.425	809.0
Southern	0.879	0.884	0.881	4480.0
micro avg	0.795	0.795	0.795	5647.0
macro avg	0.574	0.611	0.581	5647.0

Table A.7: LogisticRegression cuisine classification best embedding detailed results (test data).

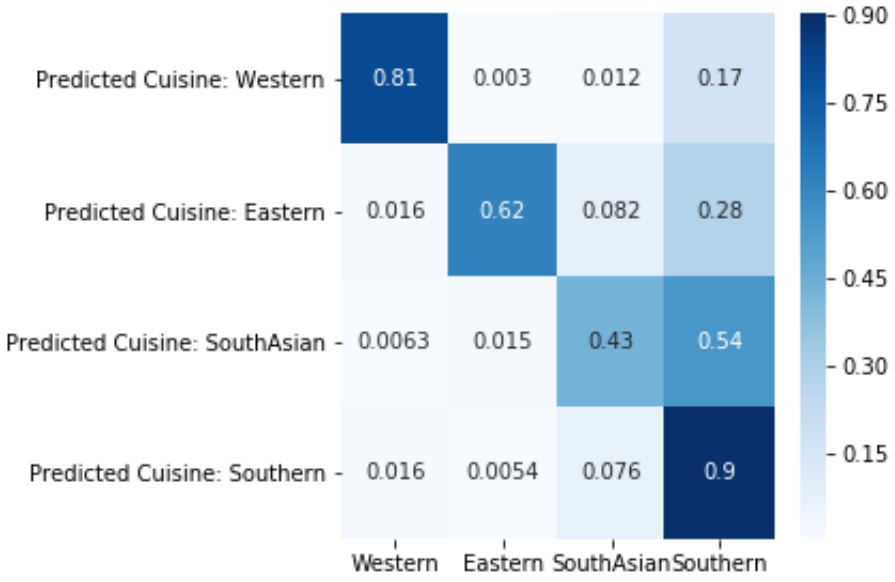

FIGURE A.1: Confusion matrix for the best cuisine embedding (train dataset).

FIGURE A.2: Confusion matrix for the best cuisine embedding (test dataset).

Appendix B

Topic Model Appendix

B.1 Latent Dirichlet Allocation Parameter Benchmarks

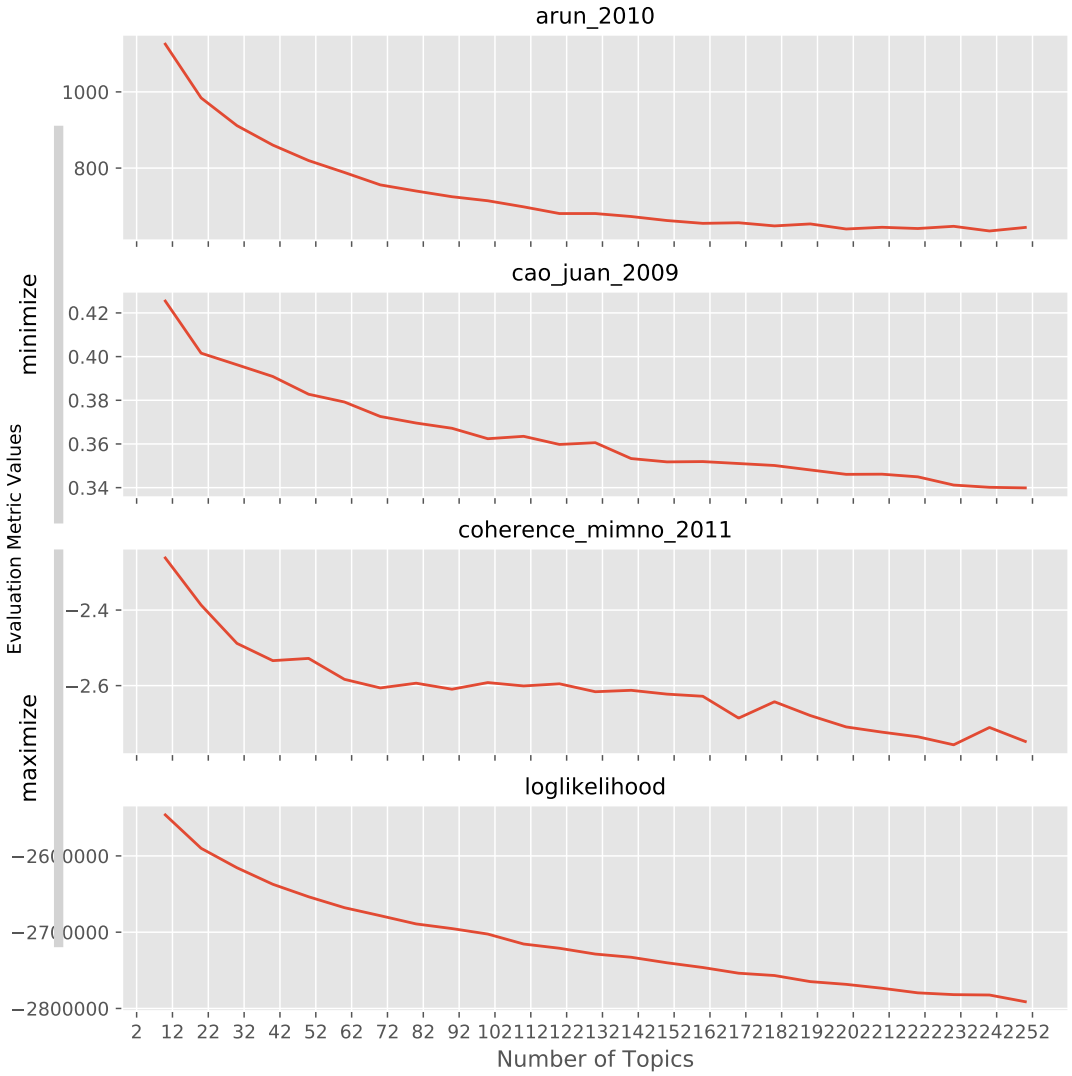

Figure B.1: LDA-Model Metrics (α=1/k, β=0.1 for 2-252 Topics)

Figure B.1 shows the four metrics ”arun_2010”, ”cao_juan_2009”, coherence_mimno_2011 and log-likelihood as described in Chapter 3 for LDA models from two up to 252 topics with the parameters α=1/k and β=0.1 trained with 2.000 maximum iterations.

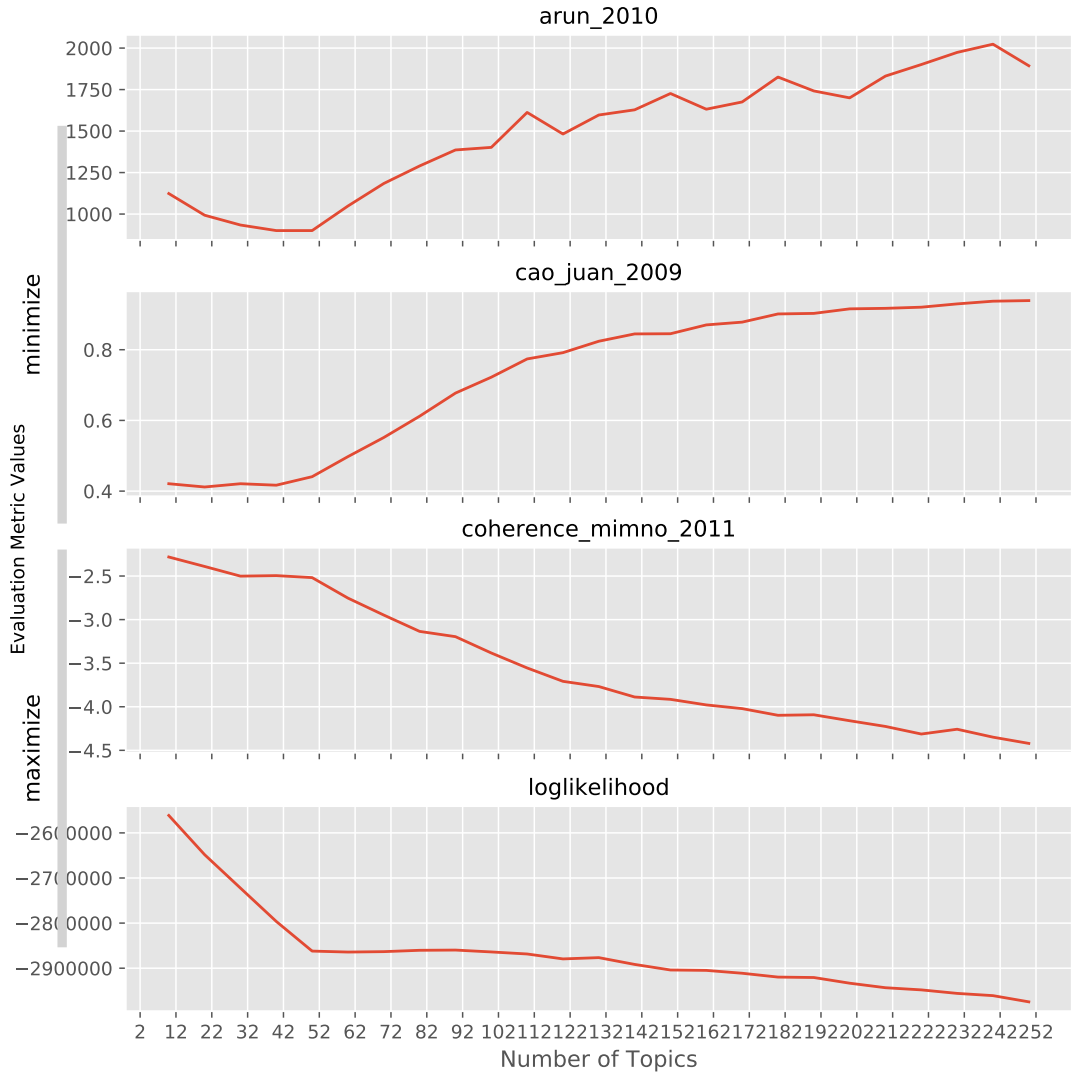

FIGURE B.2: LDA-Model Metrics (α=1/k, β=1/10k for 2-252 Topics)

Figure B.1 shows the four metrics "arun_2010", "cao_juan_2009", coherence_mimno_2011 and log-likelihood as described in Chapter 3 for LDA models from two up to 252 topics with the parameters α=1/k and β=1/10k trained with 2.000 maximum iterations.

Training time for each set of LDA models was around 45 minutes, i.e. in total less than 90 minutes.

B.2 Top Words for each Topic

B.2.1 Topic Model №1: Ten Top Words for each of the 50 Topics

Topic №	Top ten words
1	butter NorthAmericanCUISINE olive_oil garlic chicken_broth mushroom parmesan_cheese cream white_wine SouthernEuropeanCUISINE
2	NorthAmericanCUISINE wheat egg butter vanilla cocoa cane_molasses milk lard walnut
3	onion black_pepper carrot garlic beef thyme WesternEuropeanCUISINE NorthAmericanCUISINE butter olive_oil
4	cayenne cilantro lime_juice LatinAmericanCUISINE onion garlic tomato olive_oil NorthAmericanCUISINE avocado
5	egg wheat butter milk almond vanilla SouthernEuropeanCUISINE WesternEuropeanCUISINE cream cinnamon
6	NorthAmericanCUISINE wheat egg cinnamon vanilla vegetable_oil butter cane_molasses walnut oat
7	garlic SoutheastAsianCUISINE cayenne fish lime_juice cilantro vegetable_oil ginger rice soy_sauce
8	cayenne onion tomato LatinAmericanCUISINE garlic cheddar_cheese beef corn cumin cheese
9	wheat NorthAmericanCUISINE yeast egg butter milk vegetable_oil whole_grain_wheat_flour honey seed
10	vinegar cane_molasses NorthAmericanCUISINE onion tamarind black_pepper mustard garlic vegetable_oil egg
11	olive_oil garlic SouthernEuropeanCUISINE parsley white_wine tomato bell_pepper onion black_pepper fish
12	onion cayenne garlic tomato NorthAmericanCUISINE beef cumin black_pepper bell_pepper LatinAmericanCUISINE
13	NorthAmericanCUISINE egg vegetable_oil garlic cream bread onion butter cream_cheese pepper
14	egg NorthAmericanCUISINE milk cocoa wheat cream butter vanilla coffee almond
15	NorthAmericanCUISINE vegetable_oil vinegar egg onion mustard celery cucumber pepper green_bell_pepper
16	NorthAmericanCUISINE lemon orange_juice lime orange pineapple lemon_juice strawberry rum banana
17	cumin coriander turmeric fenugreek pepper SouthAsianCUISINE onion vegetable_oil garlic cayenne
18	NorthAmericanCUISINE wheat milk egg butter cocoa vanilla cream cream_cheese yeast
19	EastAsianCUISINE garlic scallion soy_sauce cayenne black_pepper sesame_oil ginger soybean rice
20	olive_oil vinegar NorthAmericanCUISINE black_pepper shallot mustard garlic lemon_juice red_wine WesternEuropeanCUISINE
21	garlic tomato olive_oil basil SouthernEuropeanCUISINE parmesan_cheese onion mozzarella_cheese macaroni oregano
22	NorthAmericanCUISINE vegetable_oil vinegar egg celery chicken lettuce apple onion lemon_juice
23	vegetable_oil egg onion mustard parsley cream chive NorthAmericanCUISINE cucumber wheat
24	NorthAmericanCUISINE egg bread onion butter cheddar_cheese milk pepper vegetable_oil ham
25	basil garlic NorthAmericanCUISINE olive_oil oregano tomato rosemary thyme onion pepper
26	NorthAmericanCUISINE lemon_juice egg black_pepper vegetable_oil butter lemon dill parsley bread
27	NorthAmericanCUISINE butter egg wheat milk vanilla cane_molasses cream cinnamon pecan
28	olive_oil garlic onion cumin lemon_juice MiddleEasternCUISINE black_pepper AfricanCUISINE parsley cayenne
29	wheat NorthAmericanCUISINE egg milk yeast butter corn vegetable_oil buttermilk lard
30	EastAsianCUISINE rice soy_sauce vinegar scallion sesame_oil cayenne vegetable_oil soybean vegetable
31	NorthAmericanCUISINE onion mushroom chicken butter milk pepper cheddar_cheese cream celery
32	NorthAmericanCUISINE wheat egg butter milk cream vanilla lemon_juice strawberry cream_cheese
33	soy_sauce garlic ginger NorthAmericanCUISINE scallion vegetable_oil starch chicken rice vinegar
34	NorthAmericanCUISINE cream wheat pineapple gelatin milk egg vanilla cream_cheese butter
35	milk butter wheat NorthAmericanCUISINE egg cheddar_cheese cream cheese black_pepper parmesan_cheese
36	NorthAmericanCUISINE wheat cinnamon egg butter nutmeg milk ginger cane_molasses vanilla
37	vinegar tomato corn garlic celery_oil NorthAmericanCUISINE onion cane_molasses tamarind mustard
38	NorthAmericanCUISINE butter potato onion milk cream pepper carrot celery chicken_broth
39	NorthAmericanCUISINE vinegar cane_molasses garlic onion soy_sauce mustard black_pepper honey pepper
40	wheat NorthAmericanCUISINE butter egg cinnamon milk lard cane_molasses lemon_juice apple
41	NorthAmericanCUISINE onion garlic cayenne green_bell_pepper rice tomato pepper black_pepper bell_pepper
42	olive_oil garlic macaroni SouthernEuropeanCUISINE parmesan_cheese tomato basil NorthAmericanCUISINE parsley cheese
43	wheat NorthAmericanCUISINE egg butter cinnamon cane_molasses vanilla raisin milk walnut
44	NorthAmericanCUISINE butter vanilla wheat cocoa egg milk peanut_butter cane_molasses corn
45	NorthAmericanCUISINE onion sage black_pepper celery thyme butter rosemary chicken_broth marjoram
46	NorthAmericanCUISINE cinnamon orange_juice butter cane_molasses apple cranberry orange ginger lemon_juice
47	cayenne LatinAmericanCUISINE onion garlic tomato corn cilantro cumin cheese vegetable_oil
48	olive_oil garlic NorthAmericanCUISINE vinegar tomato onion bell_pepper lemon_juice parsley oregano
49	onion NorthAmericanCUISINE carrot tomato garlic celery olive_oil potato pepper bean
50	onion NorthAmericanCUISINE pepper vinegar potato beef black_pepper butter cabbage vegetable_oil

Table B.1: Topic model №1 top ten words for each of the 50 topics

B.2.2 Topic Model №2: Ten Top Words for each of the 50 Topics

Topic №	Top ten words
1	butter NorthAmericanCUISINE olive_oil garlic chicken_broth parmesan_cheese cream white_wine mushroom SouthernEuropeanCUISINE
2	NorthAmericanCUISINE wheat egg milk butter cocoa vanilla cream coffee vegetable_oil
3	NorthAmericanCUISINE lemon lime orange_juice orange pineapple lime_juice rum lemon_juice honey
4	NorthAmericanCUISINE butter WesternEuropeanCUISINE cream bread onion cream_cheese rice egg olive
5	NorthAmericanCUISINE wheat egg butter milk cream vanilla lemon_juice lemon cream_cheese
6	NorthAmericanCUISINE egg wheat cinnamon butter milk nutmeg cane_molasses vanilla ginger
7	cumin turmeric coriander pepper fenugreek onion garlic SouthAsianCUISINE vegetable_oil cayenne
8	cayenne onion LatinAmericanCUISINE garlic tomato cumin corn cheese cheddar_cheese bell_pepper
9	NorthAmericanCUISINE wheat egg cinnamon butter vanilla walnut cane_molasses vegetable_oil raisin
10	NorthAmericanCUISINE onion thyme celery sage black_pepper butter rosemary chicken_broth bread
11	olive_oil garlic SouthernEuropeanCUISINE tomato parsley onion bell_pepper white_wine fish black_pepper
12	NorthAmericanCUISINE egg vegetable_oil garlic cream bread onion cream_cheese cheddar_cheese bacon
13	olive_oil garlic lemon_juice NorthAmericanCUISINE tomato onion SouthernEuropeanCUISINE parsley black_pepper mint
14	NorthAmericanCUISINE wheat butter egg milk vanilla cream cocoa cream_cheese coconut
15	NorthAmericanCUISINE vinegar vegetable_oil egg onion mustard cucumber pepper celery cider
16	NorthAmericanCUISINE vegetable_oil egg celery chicken lemon_juice apple pepper lettuce grape
17	olive_oil garlic onion cumin cayenne black_pepper cinnamon AfricanCUISINE bell_pepper MiddleEasternCUISINE
18	NorthAmericanCUISINE butter cocoa vanilla wheat milk peanut_butter egg cane_molasses cream
19	EastAsianCUISINE soy_sauce garlic scallion cayenne sesame_oil rice black_pepper soybean ginger
20	soy_sauce garlic ginger scallion NorthAmericanCUISINE vegetable_oil rice EastAsianCUISINE starch vinegar
21	vinegar cane_molasses onion NorthAmericanCUISINE tamarind garlic beef black_pepper pepper vegetable_oil
22	onion NorthAmericanCUISINE carrot tomato garlic beef celery potato pepper bean
23	onion black_pepper garlic carrot butter thyme NorthAmericanCUISINE beef WesternEuropeanCUISINE olive_oil
24	wheat NorthAmericanCUISINE egg milk butter yeast cheddar_cheese vegetable_oil corn onion
25	NorthAmericanCUISINE vinegar olive_oil mustard black_pepper shallot red_wine grape_juice lettuce honey
26	NorthAmericanCUISINE butter egg milk cheddar_cheese wheat bread onion pepper ham
27	NorthAmericanCUISINE cinnamon cane_molasses orange_juice apple orange butter cranberry ginger honey
28	SoutheastAsianCUISINE garlic cayenne fish cilantro lime_juice vegetable_oil rice ginger scallion
29	egg wheat milk cream butter vanilla WesternEuropeanCUISINE cocoa NorthAmericanCUISINE almond
30	NorthAmericanCUISINE lemon_juice butter black_pepper egg parsley lemon olive_oil vegetable_oil garlic
31	NorthAmericanCUISINE butter potato onion milk cream pepper wheat celery chicken_broth
32	vinegar tomato NorthAmericanCUISINE onion garlic corn cane_molasses celery_oil mustard beef
33	NorthAmericanCUISINE soy_sauce garlic honey ginger vinegar onion cane_molasses chicken pork
34	NorthAmericanCUISINE cream pineapple gelatin strawberry banana orange apple cream_cheese cherry
35	wheat NorthAmericanCUISINE egg yeast butter milk vegetable_oil whole_grain_wheat_flour honey seed
36	NorthAmericanCUISINE wheat butter egg milk vanilla pecan cocoa cream cane_molasses
37	NorthAmericanCUISINE black_pepper mustard cinnamon ginger bay lovage vegetable_oil egg butter
38	wheat egg butter almond NorthAmericanCUISINE milk orange cinnamon raisin lemon
39	onion NorthAmericanCUISINE butter potato cabbage seed pepper black_pepper beef WesternEuropeanCUISINE
40	butter wheat egg WesternEuropeanCUISINE milk black_pepper cheese cream bread NorthAmericanCUISINE
41	NorthAmericanCUISINE onion butter mushroom chicken milk pepper cream cheddar_cheese celery
42	NorthAmericanCUISINE olive_oil garlic onion basil bell_pepper oregano tomato black_pepper vinegar
43	NorthAmericanCUISINE wheat butter egg vanilla cocoa cane_molasses milk lard walnut
44	wheat NorthAmericanCUISINE butter egg cinnamon milk cane_molasses apple vanilla lard
45	olive_oil garlic SouthernEuropeanCUISINE tomato macaroni basil parmesan_cheese cheese parsley black_pepper
46	egg vegetable_oil NorthAmericanCUISINE onion mustard parsley chive cream cucumber wheat
47	cayenne LatinAmericanCUISINE cilantro garlic onion tomato lime_juice olive_oil corn NorthAmericanCUISINE
48	NorthAmericanCUISINE butter lemon pepper cayenne egg WesternEuropeanCUISINE dill nut orange
49	garlic tomato basil olive_oil oregano onion NorthAmericanCUISINE parmesan_cheese mozzarella_cheese SouthernEuropeanCUISINE
50	onion NorthAmericanCUISINE garlic cayenne tomato green_bell_pepper pepper bell_pepper black_pepper rice

Table B.2: Topic model №2 top ten words for each of the 50 topics

B.2.3 Topic Model №1 and №2 Distinct Words

	Most distinct words	Least distinct words
1	lemongrass	sturgeon_caviar
2	SoutheastAsianCUISINE	NorthAmericanCUISINE
3	fenugreek	roasted_hazelnut
4	lovage	roasted_pecan
5	gin	roasted_nut
6	celery_oil	jamaican_rum
7	seaweed	muscat_grape
8	galanga	pelargonium
9	wasabi	mate
10	red_bean	angelica
11	sauerkraut	strawberry_jam
12	AfricanCUISINE	pimenta
13	melon	soybean_oil
14	chicory	emmental_cheese
15	rye_flour	carnation
16	endive	laurel
17	turmeric	durian
18	SouthAsianCUISINE	jasmine_tea
19	tequila	balm
20	chinese_cabbage	lilac_flower_oil
21	smoked_salmon	beech
22	thai_pepper	geranium
23	kelp	onion
24	caraway	black_pepper
25	enokidake	garlic
26	katsuobushi	butter
27	EastAsianCUISINE	pepper
28	bitter_orange	vegetable_oil
29	watermelon	bell_pepper
30	mussel	strawberry_juice
31	black_sesame_seed	mutton
32	red_kidney_bean	egg
33	roasted_sesame_seed	roasted_almond
34	kiwi	wheat
35	gruyere_cheese	citrus_peel
36	okra	blackberry_brandy
37	nira	cream
38	lime_peel_oil	sheep_cheese
39	popcorn	spearmint
40	sage	rapeseed
41	mandarin	peppermint_oil
42	coriander	red_algae
43	rye_bread	raw_beef
44	grape	chicken
45	pumpkin	kohlrabi
46	black_mustard_seed_oil	chamomile
47	tamarind	holy_basil
48	potato_chip	violet
49	cereal	pork_liver
50	marjoram	tomato

Table B.3: Topic model №1 distinct words according to Chuang et al. 2012.

	Most distinct words	Least distinct words
1	fenugreek	NorthAmericanCUISINE
2	lovage	malt
3	turmeric	sour_cherry
4	SoutheastAsianCUISINE	eel
5	EastAsianCUISINE	wood
6	lime	oatmeal
7	tamarind	mackerel
8	coriander	shellfish
9	celery_oil	frankfurter
10	lime_juice	gardenia
11	soy_sauce	beef_liver
12	peanut_butter	black_sesame_seed
13	sesame_oil	provolone_cheese
14	pineapple	matsutake
15	cumin	roasted_meat
16	chive	japanese_plum
17	gelatin	artemisia
18	cucumber	lingonberry
19	sage	cabernet_sauvignon_wine
20	cilantro	black_raspberry
21	orange_juice	ouzo
22	AfricanCUISINE	jasmine
23	LatinAmericanCUISINE	kohlrabi
24	strawberry	nectarine
25	cocoa	baked_potato
26	orange	roasted_beef
27	sauerkraut	sunflower_oil
28	caraway	katsuobushi
29	SouthAsianCUISINE	violet
30	cranberry	spearmint
31	grape	mandarin_peel
32	cinnamon	red_algae
33	yeast	raw_beef
34	whole_grain_wheat_flour	mutton
35	bay	coconut_oil
36	mint	rapeseed
37	WesternEuropeanCUISINE	soybean_oil
38	ginger	pimenta
39	SouthernEuropeanCUISINE	pear_brandy
40	shallot	balm
41	lemongrass	laurel
42	apple	roasted_almond
43	rum	sour_milk
44	MiddleEasternCUISINE	muscat_grape
45	mustard	jasmine_tea
46	oat	chayote
47	cheddar_cheese	emmental_cheese
48	fish	pelargonium
49	vanilla	mate
50	roasted_sesame_seed	elderberry

Table B.4: Topic model №2 distinct words according to Chuang et al. 2012.

B.2.4 Topic Model №1 and №2 Salient Words

	Most salient words	Least salient words
1	wheat	sturgeon_caviar
2	egg	NorthAmericanCUISINE
3	milk	roasted_hazelnut
4	vanilla	roasted_pecan
5	butter	roasted_nut
6	olive_oil	jamaican_rum
7	vinegar	muscat_grape
8	NorthAmericanCUISINE	pelargonium
9	tomato	mate
10	garlic	angelica
11	cayenne	strawberry_jam
12	onion	pimenta
13	cane_molasses	soybean_oil
14	cinnamon	emmental_cheese
15	cocoa	carnation
16	cream	laurel
17	EastAsianCUISINE	durian
18	soy_sauce	jasmine_tea
19	LatinAmericanCUISINE	balm
20	vegetable_oil	lilac_flower_oil
21	cumin	beech
22	basil	geranium
23	yeast	onion
24	mustard	black_pepper
25	SouthernEuropeanCUISINE	garlic
26	parsley	butter
27	ginger	pepper
28	corn	vegetable_oil
29	black_pepper	bell_pepper
30	pepper	strawberry_juice
31	celery	mutton
32	lemon_juice	egg
33	parmesan_cheese	roasted_almond
34	cilantro	wheat
35	beef	citrus_peel
36	thyme	blackberry_brandy
37	cheddar_cheese	cream
38	oregano	sheep_cheese
39	rice	spearmint
40	scallion	rapeseed
41	macaroni	peppermint_oil
42	chicken	red_algae
43	chicken_broth	raw_beef
44	potato	chicken
45	tamarind	kohlrabi
46	carrot	chamomile
47	bread	holy_basil
48	coriander	violet
49	cheese	pork_liver
50	nutmeg	tomato

TABLE B.5: Topic model №1 salient words according to ibid.

	Most salient words	Least salient words
1	wheat	NorthAmericanCUISINE
2	egg	malt
3	butter	sour_cherry
4	vanilla	eel
5	NorthAmericanCUISINE	wood
6	milk	oatmeal
7	olive_oil	mackerel
8	onion	shellfish
9	garlic	frankfurter
10	vinegar	gardenia
11	tomato	beef_liver
12	cinnamon	black_sesame_seed
13	cane_molasses	provolone_cheese
14	cocoa	matsutake
15	cayenne	roasted_meat
16	cream	japanese_plum
17	vegetable_oil	artemisia
18	soy_sauce	lingonberry
19	EastAsianCUISINE	cabernet_sauvignon_wine
20	SouthernEuropeanCUISINE	black_raspberry
21	cumin	ouzo
22	black_pepper	jasmine
23	mustard	kohlrabi
24	ginger	nectarine
25	parsley	baked_potato
26	LatinAmericanCUISINE	roasted_beef
27	basil	sunflower_oil
28	pepper	katsuobushi
29	yeast	violet
30	lemon_juice	spearmint
31	celery	mandarin_peel
32	cheddar_cheese	red_algae
33	potato	raw_beef
34	cilantro	mutton
35	parmesan_cheese	coconut_oil
36	chicken	rapeseed
37	tamarind	soybean_oil
38	WesternEuropeanCUISINE	pimenta
39	corn	pear_brandy
40	lemon	balm
41	thyme	laurel
42	scallion	roasted_almond
43	beef	sour_milk
44	carrot	muscat_grape
45	macaroni	jasmine_tea
46	coriander	chayote
47	oregano	emmental_cheese
48	bread	pelargonium
49	chicken_broth	mate
50	apple	elderberry

TABLE B.6: Topic model №2 salient words according to Chuang et al. 2012.

Bibliography

Ahn, Yong-Yeol, Sebastian E. Ahnert, James P. Bagrow, and Albert-László Barabási (Dec. 2011). "Flavor Network and the Principles of Food Pairing". en. In: *Scientific Reports* 1, p. 196. ISSN: 2045-2322. DOI: 10.1038/srep00196. URL: https://www.nature.com/articles/srep00196 (visited on 12/10/2018).

Altosaar, Jaan (Jan. 2017a). *food2vec - Augmented cooking with machine intelligence*. URL: /food2vec-augmented-cooking-machine-intelligence/ (visited on 01/15/2019).

— (2017b). *GitHub.com/altosaar/food2vec*. original-date: 2017-01-17T19:28:52Z. URL: https://github.com/altosaar/food2vec (visited on 01/15/2019).

Ankenbrand, Von Hendrik and Schanghai (2015). "Mehr Nahrung für mehr Menschen: Die Chinesen: zum Kartoffel-Essen verdammt". de. In: ISSN: 0174-4909. URL: https://www.faz.net/1.3387160 (visited on 01/17/2019).

Arun, R., V. Suresh, C. E. Veni Madhavan, and M. N. Narasimha Murthy (2010). "On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations". en. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Mohammed J. Zaki, Jeffrey Xu Yu, B. Ravindran, and Vikram Pudi. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 391–402. ISBN: 978-3-642-13657-3.

BBC News (Jan. 2019). "China's cotton seeds sprout on Moon". en-GB. In: URL: https://www.bbc.com/news/world-asia-china-46873526 (visited on 01/17/2019).

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin (Mar. 2003). "A Neural Probabilistic Language Model". In: *J. Mach. Learn. Res.* 3, pp. 1137–1155. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944966 (visited on 01/21/2019).

Bkkbrad (Feb. 2008). *Latent Dirichlet allocation diagram in plate notation*. URL: https://commons.wikimedia.org/wiki/File:Latent_Dirichlet_allocation.svg (visited on 01/22/2019).

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (Apr. 2017). "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518. arXiv: 1601.00670, pp. 859–877. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.2017.1285773. URL: http://arxiv.org/abs/1601.00670 (visited on 01/16/2019).

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (Mar. 2003). "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3, pp. 993–1022. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944937 (visited on 01/13/2019).

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (July 2016). "Enriching Word Vectors with Subword Information". In: *arXiv:1607.04606 [cs]*. arXiv: 1607.04606. URL: http://arxiv.org/abs/1607.04606 (visited on 01/13/2019).

Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool (2014). "Food-101 – Mining Discriminative Components with Random Forests". en. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Lecture Notes in Computer Science. Springer International Publishing, pp. 446–461. ISBN: 978-3-319-10599-4.

Bostan, Laura Ana Maria (July 2017). "Ingredient-Driven Recipe Generation Using Neural and Distributional Models". MA thesis. Trento, Italy: University of Trento.

Bump, Philip (2019). *Analysis | President Trump's extravagant, $3,000, 300-sandwich celebration of Clemson University*. en. URL: https://www.washingtonpost.com/politics/2019/01/15/president-trumps-extravagant-sandwich-celebration-clemson-university/ (visited on 01/17/2019).

Cao, Juan, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang (Mar. 2009). "A Density-based Method for Adaptive LDA Model Selection". In: *Neurocomput.* 72.7-9, pp. 1775–

1781. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2008.06.011. URL: http://dx.doi.org/10.1016/j.neucom.2008.06.011 (visited on 01/13/2019).

Carvalho, Micael, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord (Apr. 2018). "Cross-Modal Retrieval in the Cooking Context: Learning Semantic Text-Image Embeddings". In: *arXiv:1804.11146 [cs]*. arXiv: 1804.11146. URL: http://arxiv.org/abs/1804.11146 (visited on 12/14/2018).

Chang, Jonathan, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei (2009). "Reading Tea Leaves: How Humans Interpret Topic Models". In: *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*. NIPS'09. USA: Curran Associates Inc., pp. 288–296. ISBN: 978-1-61567-911-9. URL: http://dl.acm.org/citation.cfm?id=2984093.2984126 (visited on 01/22/2019).

Chang, Minsuk, Leonore V. Guillain, Hyeungshik Jung, Vivian M. Hare, Juho Kim, and Maneesh Agrawala (2018). "RecipeScape: An Interactive Tool for Analyzing Cooking Instructions at Scale". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: ACM, 451:1–451:12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174025. URL: http://doi.acm.org/10.1145/3173574.3174025 (visited on 12/14/2018).

Cheng, Ling (2016). *GitHub.com/ingcheng99/Flavor-Network*. original-date: 2016-03-21T16:40:14Z. URL: https://github.com/lingcheng99/Flavor-Network (visited on 01/15/2019).

Christopher Olah (2019). *Home - colah's blog*. URL: http://colah.github.io/ (visited on 01/21/2019).

Chuang, Jason, Christopher D. Manning, and Jeffrey Heer (2012). "Termite: Visualization Techniques for Assessing Textual Topic Models". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. New York, NY, USA: ACM, pp. 74–77. ISBN: 978-1-4503-1287-5. DOI: 10.1145/2254556.2254572. URL: http://doi.acm.org/10.1145/2254556.2254572 (visited on 01/16/2019).

Cortes, Corinna and Vladimir Vapnik (Sept. 1995). "Support-Vector Networks". en. In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411 (visited on 01/13/2019).

Crammer, Koby and Yoram Singer (Mar. 2002). "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines". In: *J. Mach. Learn. Res.* 2, pp. 265–292. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944790.944813 (visited on 01/22/2019).

De Clercq, Marlies (2014). *Prediction of ingredient combinations using machine learning techniques*. URL: https://lib.ugent.be/fulltxt/RUG01/002/166/653/RUG01-002166653_2014_0001_AC.pdf.

Despres, Sylvie (May 2014). "Construction d'une ontologie modulaire pour l'univers de la cuisine numérique". In: *IC - 25èmes Journées francophones d'Ingénierie des Connaissances*. Ed. by Catherine Faron-Zucker. Clermont-Ferrand, France, pp. 27–38. URL: https://hal.inria.fr/hal-01010222 (visited on 12/10/2018).

Deuber, Lea (Jan. 2019). "Staatlich verordnete Kartoffeldiät". de. In: *sueddeutsche.de*. ISSN: 0174-4917. URL: https://www.sueddeutsche.de/panorama/china-ernaehrung-kartoffel-1.4279106 (visited on 01/17/2019).

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin (June 2008). "LIBLINEAR: A Library for Large Linear Classification". In: *J. Mach. Learn. Res.* 9, pp. 1871–1874. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1390681.1442794 (visited on 01/21/2019).

Friedman, Jerome H. (Feb. 2002). "Stochastic Gradient Boosting". In: *Comput. Stat. Data Anal.* 38.4, pp. 367–378. ISSN: 0167-9473. DOI: 10.1016/S0167-9473(01)00065-2. URL: http://dx.doi.org/10.1016/S0167-9473(01)00065-2 (visited on 01/13/2019).

Frost, Natasha (2019). *Germany just released an unspeakably ugly coin commemorating currywurst*. en. URL: https://qz.com/1523284/a-german-mint-just-released-a-coin-commemorating-70-years-of-currywurst-and-its-unspeakably-ugly/ (visited on 01/17/2019).

Goldberg, Yoav (Oct. 2015). "A Primer on Neural Network Models for Natural Language Processing". In: *arXiv:1510.00726 [cs]*. arXiv: 1510.00726. URL: http://arxiv.org/abs/1510.00726 (visited on 01/21/2019).

Goldberg, Yoav and Omer Levy (Feb. 2014). "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method". In: *arXiv:1402.3722 [cs, stat]*. arXiv: 1402.3722. URL: http://arxiv.org/abs/1402.3722 (visited on 01/21/2019).

Gomez, Raul, Lluis Gomez, Jaume Gibert, and Dimosthenis Karatzas (Aug. 2018). "Learning to Learn from Web Data through Deep Semantic Embeddings". In: *arXiv:1808.06368 [cs]*. arXiv: 1808.06368. URL: http://arxiv.org/abs/1808.06368 (visited on 12/14/2018).

Griffiths, Thomas L. and Mark Steyvers (Apr. 2004). "Finding scientific topics". en. In: *Proceedings of the National Academy of Sciences* 101.suppl 1, pp. 5228–5235. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0307752101. URL: https://www.pnas.org/content/101/suppl_1/5228 (visited on 01/13/2019).

Harashima, Jun, Michiaki Ariga, Kenta Murata, and Masayuki Ioki (2016). "A Large-scale Recipe and Meal Data Collection as Infrastructure for Food Research". In: *LREC*. Portorož, Slovenia. URL: www.lrec-conf.org/proceedings/lrec2016/pdf/320_Paper.pdf.

Harris, Zellig S. (Aug. 1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162. DOI: 10.1080/00437956.1954.11659520. URL: https://doi.org/10.1080/00437956.1954.11659520.

Hinton, Geoffrey E and Sam T. Roweis (2003). "Stochastic Neighbor Embedding". In: *Advances in Neural Information Processing Systems 15*. Ed. by S. Becker, S. Thrun, and K. Obermayer. MIT Press, pp. 857–864. URL: http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf (visited on 01/22/2019).

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735 (visited on 12/14/2018).

Hotelling, H. (1933). "Analysis of a complex of statistical variables into principal components". In: *Journal of Educational Psychology* 24.6, pp. 417–441. ISSN: 1939-2176(Electronic),0022-0663(Print). DOI: 10.1037/h0071325.

Institute for Genomics and Bioinformatics - Graz University of Technology and Department of Information Design - FH JOANNEUM - Graz University of Applied Sciences (2005). *ProClassify User's Guide - Cross-Validation Explained*. URL: http://genome.tugraz.at/proclassify/help/pages/XV.html (visited on 01/22/2019).

Jordanous, Anna (Sept. 2012). "A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative". en. In: *Cognitive Computation* 4.3, pp. 246–279. ISSN: 1866-9964. DOI: 10.1007/s12559-012-9156-1. URL: https://doi.org/10.1007/s12559-012-9156-1 (visited on 01/14/2019).

Jurafsky, Dan (Aug. 2014). *The secret language of food*. en-GB. URL: https://www.ft.com/content/82f41202-27f3-11e4-ae44-00144feabdc0 (visited on 01/17/2019).

— (Oct. 2015). *The Language of Food: A Linguist Reads the Menu*. English. 1 edition. W. W. Norton & Company. ISBN: 978-0-393-35162-0.

Jurafsky, Daniel and James H. Martin (Sept. 2018). *Speech and Language Processing, 3nd Edition draft*. English. 3nd edition. Upper Saddle River, N.J: Prentice Hall. ISBN: 978-0-13-187321-6. URL: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.

Kalajdziski, S., G. Radevski, I. Ivanoska, K. Trivodaliev, and B. R. Stojkoska (May 2018). "Cuisine classification using recipe's ingredients". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1074–1079. DOI: 10.23919/MIPRO.2018.8400196.

Kazama, Masahiro, Minami Sugimoto, Chizuru Hosokawa, Keisuke Matsushima, Lav R. Varshney, and Yoshiki Ishikawa (2018). "A Neural Network System for Transformation of Regional Cuisine Style". English. In: *Frontiers in ICT* 5. ISSN: 2297-198X. DOI: 10.3389/fict.2018.00014. URL: https://www.frontiersin.org/articles/10.3389/fict.2018.00014/full (visited on 12/10/2018).

Kicherer, Hanna, Marcel Dittrich, Lukas Grebe, Christian Scheible, and Roman Klinger (2017). "What You Use, Not What You Do: Automatic Classification of Recipes". en. In: *Natural Language Processing and Information Systems*. Ed. by Flavius Frasincar, Ashwin Ittoo, Le Minh Nguyen, and Elisabeth Métais. Lecture Notes in Computer Science. Springer International Publishing, pp. 197–209. ISBN: 978-3-319-59569-6. URL: www.romanklinger.de/publications/kicherer2017-nldb.pdf.

Kiddon, Chloé, Luke Zettlemoyer, and Yejin Choi (2016). "Globally Coherent Text Generation with Neural Checklist Models". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 329–339. DOI: 10.18653/v1/D16-1032. URL: http://aclweb.org/anthology/D16-1032 (visited on 12/14/2018).

Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler (June 2015). "Skip-Thought Vectors". In: *arXiv:1506.06726 [cs]*. arXiv: 1506.06726. URL: http://arxiv.org/abs/1506.06726 (visited on 12/10/2018).

klausbatb, user (2019). *r/unitedkingdom - I've downloaded bbc.co.uk/food/ - What now?* en. URL: https://www.reddit.com/r/unitedkingdom/comments/4jq6i2/ive_downloaded_bbccoukfood_what_now/ (visited on 01/15/2019).

Konrad, Markus (Nov. 2017). *Topic Model Evaluation in Python with tmtoolkit*. en-US. URL: https://datascience.blog.wzb.eu/2017/11/09/topic-modeling-evaluation-in-python-with-tmtoolkit/ (visited on 01/16/2019).

— (n.d.). *tmtoolkit 0.72 Docs topicmod.evaluate*. URL: https://www.pydoc.io/pypi/tmtoolkit-0.7.2/autoapi/topicmod/evaluate/index.html?highlight=wallach#topicmod.evaluate.metric_held_out_documents_wallach09.

Kusmierczyk, Tomasz and Kjetil Nørv\a ag (2016a). "Online Food Recipe Title Semantics: Combining Nutrient Facts and Topics". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. New York, NY, USA: ACM, pp. 2013–2016. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983897. URL: http://doi.acm.org/10.1145/2983323.2983897 (visited on 01/14/2019).

Kusmierczyk, Tomasz, Christoph Trattner, and Kjetil Nørva ag (2016b). "Understanding and Predicting Online Food Recipe Production Patterns". In: *Proceedings of the 27th ACM Conference on Hypertext and Social Media*. HT '16. New York, NY, USA: ACM, pp. 243–248. ISBN: 978-1-4503-4247-6. DOI: 10.1145/2914586.2914632. URL: http://doi.acm.org/10.1145/2914586.2914632 (visited on 12/10/2018).

Kusu, Kazuma, Hyuk-In Choi, Tomoya Kambara, Taiki Kinoshita, Takamitsu Shioi, and Kenji Hatano (2017). "Searching Cooking Recipes by Focusing on Common Ingredients". In: *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services*. iiWAS '17. New York, NY, USA: ACM, pp. 95–101. ISBN: 978-1-4503-5299-4. DOI: 10.1145/3151759.3151797. URL: http://doi.acm.org/10.1145/3151759.3151797 (visited on 12/14/2018).

Lazzari, Gianrocco, Yannis Jaquet, Djilani J. Kebaili, Laura Symul, and Marcel Salathé (2018). "FoodRepo: An Open Food Repository of Barcoded Food Products". English. In: *Frontiers in Nutrition* 5. ISSN: 2296-861X. DOI: 10.3389/fnut.2018.00057. URL: https://www.frontiersin.org/articles/10.3389/fnut.2018.00057/full (visited on 01/15/2019).

Le, Quoc and Tomas Mikolov (2014). "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, pp. II–1188–II–1196. URL: http://dl.acm.org/citation.cfm?id=3044805.3045025 (visited on 12/10/2018).

Lenci, Alessandro (2018). "Distributional Models of Word Meaning". In: *Annual Review of Linguistics* 4.1, pp. 151–171. DOI: 10.1146/annurev-linguistics-030514-125254. URL: https://doi.org/10.1146/annurev-linguistics-030514-125254 (visited on 12/10/2018).

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.Nov, pp. 2579–2605. ISSN: ISSN 1533-7928.

URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html (visited on 01/14/2019).

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (July 2008). *Introduction to Information Retrieval by Christopher D. Manning*. en. DOI: 10.1017/CBO9780511809071. URL: /core/books/introduction-to-information-retrieval/669D108D20F556C5C30957D63B5AB65C (visited on 12/10/2018).

Marin, Javier, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba (Oct. 2018). "Recipe1M: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images". In: *arXiv:1810.06553 [cs]*. arXiv: 1810.06553. URL: http://arxiv.org/abs/1810.06553 (visited on 01/14/2019).

Maura Judkis (2019). *Perspective | What it means that Trump served Big Macs in the State Dining Room*. en. URL: https://www.washingtonpost.com/news/voraciously/wp/2019/01/15/what-it-means-that-trump-served-big-macs-in-the-state-dining-room/ (visited on 01/17/2019).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (Jan. 2013). "Efficient Estimation of Word Representations in Vector Space". In: *arXiv:1301.3781 [cs]*. arXiv: 1301.3781. URL: http://arxiv.org/abs/1301.3781 (visited on 01/13/2019).

Mimno, David, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum (2011). "Optimizing Semantic Coherence in Topic Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 262–272. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145462 (visited on 01/13/2019).

Min, Weiqing, Bing-Kun Bao, Shuhuan Mei, Yaohui Zhu, Yong Rui, and Shuqiang Jiang (2018a). "You Are What You Eat: Exploring Rich Recipe Information for Cross-Region Food Analysis". In: *IEEE Transactions on Multimedia* 20, pp. 950–964. DOI: 10.1109/TMM.2017.2759499.

Min, Weiqing, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain (Aug. 2018b). "A Survey on Food Computing". In: *arXiv:1808.07202 [cs]*. arXiv: 1808.07202. URL: http://arxiv.org/abs/1808.07202 (visited on 12/10/2018).

Mouritsen, Ole G., Rachel Edwards-Stuart, Yong-Yeol Ahn, and Sebastian E. Ahnert (2017). "Data-driven Methods for the Study of Food Perception, Preparation, Consumption, and Culture". English. In: *Frontiers in ICT* 4. ISSN: 2297-198X. DOI: 10.3389/fict.2017.00015. URL: https://www.frontiersin.org/articles/10.3389/fict.2017.00015/full (visited on 12/12/2018).

Nezis, Angelos, Haris Papageorgiou, Pavlos Georgiadis, Petr Jiskra, Dimitris Pappas, and Maria Pontiki (2018). "Towards a Fully Personalized Food Recommendation Tool". In: *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*. AVI '18. New York, NY, USA: ACM, 77:1–77:3. ISBN: 978-1-4503-5616-9. DOI: 10.1145/3206505.3206590. URL: http://doi.acm.org/10.1145/3206505.3206590 (visited on 12/14/2018).

Nobumoto, Kensuke, Daiju Kato, Masaki Endo, Masaharu Hirota, and Hiroshi Ishikawa (2017). "Multilingualization of Restaurant Menu by Analogical Description". In: *Proceedings of the 9th Workshop on Multimedia for Cooking and Eating Activities in Conjunction with The 2017 International Joint Conference on Artificial Intelligence*. CEA2017. New York, NY, USA: ACM, pp. 13–18. ISBN: 978-1-4503-5267-3. DOI: 10.1145/3106668.3106671. URL: http://doi.acm.org/10.1145/3106668.3106671 (visited on 12/14/2018).

Parvez, Md Rizwan, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang (May 2018). "Building Language Models for Text with Named Entities". In: *arXiv:1805.04836 [cs]*. arXiv: 1805.04836. URL: http://arxiv.org/abs/1805.04836 (visited on 12/14/2018).

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: http://aclweb.org/anthology/D14-1162 (visited on 01/13/2019).

Press, The Associated (Jan. 2019). "Bayern Fines Ribery for Comments After Gold-Leaf Steak Furor". en-US. In: *The New York Times*. ISSN: 0362-4331. URL: https://www.nytimes.com/aponline/2019/01/06/world/europe/ap-soc-ribery-fined.html (visited on 01/17/2019).

Qef (July 2008). *The logistic sigmoid function*. URL: https://commons.wikimedia.org/wiki/File:Logistic-curve.svg (visited on 01/21/2019).

Rong, Xin (Nov. 2014). "word2vec Parameter Learning Explained". In: *arXiv:1411.2738 [cs]*. arXiv: 1411.2738. URL: http://arxiv.org/abs/1411.2738 (visited on 01/21/2019).

Sajadmanesh, Sina, Sina Jafarzadeh, Seyed Ali Osia, Hamid R. Rabiee, Hamed Haddadi, Yelena Mejova, Mirco Musolesi, Emiliano De Cristofaro, and Gianluca Stringhini (Oct. 2016). "Kissing Cuisines: Exploring Worldwide Culinary Habits on the Web". In: *arXiv:1610.08469 [cs]*. arXiv: 1610.08469. URL: http://arxiv.org/abs/1610.08469 (visited on 12/10/2018).

Salton, G., A. Wong, and C. S. Yang (Nov. 1975). "A Vector Space Model for Automatic Indexing". In: *Commun. ACM* 18.11, pp. 613–620. ISSN: 0001-0782. DOI: 10.1145/361219.361220. URL: http://doi.acm.org/10.1145/361219.361220 (visited on 01/20/2019).

Sauer, Christopher and Alex Haigh (2017). "Cooking up Food Embeddings Understanding Flavors in the Recipe-Ingredient Graph". In:

Schütze, Hinrich (Nov. 1992). "Dimensions of meaning". In: *Supercomputing '92:Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pp. 787–796. DOI: 10.1109/SUPERC.1992.236684.

— (1993). "Word Space". In: *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, pp. 895–902.

— (Mar. 1998). "Automatic Word Sense Discrimination". In: *Comput. Linguist.* 24.1, pp. 97–123. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=972719.972724 (visited on 12/10/2018).

Schütze, Hinrich and Jan O. Pedersen (1995). "Information Retrieval Based on Word Senses". In: *In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175.

scikit-learn (2019a). *Confusion matrix — scikit-learn 0.20.2 documentation*. URL: https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html (visited on 01/16/2019).

— (2019b). *sklearn.metrics.classification_report — scikit-learn 0.20.2 documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html (visited on 01/16/2019).

Shuai (Dec. 2016). *Topic Modeling and t-SNE Visualization*. URL: https://shuaiw.github.io/2016/12/22/topic-modeling-and-tsne-visualzation.html (visited on 01/22/2019).

Sievert, Carson and Kenneth Shirley (2014). "LDAvis: A method for visualizing and interpreting topics". In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 63–70. DOI: 10.3115/v1/W14-3110. URL: http://aclweb.org/anthology/W14-3110 (visited on 01/17/2019).

Snowdon, Kathryn (Jan. 2018). *BBC Food Website To Remain Open, Preserving Thousands Of Recipes Online*. en. URL: https://www.huffingtonpost.co.uk/entry/bbc-food-website-remain-open_uk_5a4f9ff5e4b01e1a4b14b9f5 (visited on 01/15/2019).

Su, Han, Ting-Wei Lin, Cheng-Te Li, Man-Kwan Shan, and Janet Chang (2014). "Automatic Recipe Cuisine Classification by Ingredients". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. UbiComp '14 Adjunct. New York, NY, USA: ACM, pp. 565–570. ISBN: 978-1-4503-3047-3. DOI: 10.1145/2638728.2641335. URL: http://doi.acm.org/10.1145/2638728.2641335 (visited on 12/13/2018).

Teng, Chun-Yuen, Yu-Ru Lin, and Lada A. Adamic (Nov. 2011). "Recipe recommendation using ingredient networks". In: *arXiv:1111.3919 [physics]*. arXiv: 1111.3919. URL: http://arxiv.org/abs/1111.3919 (visited on 12/10/2018).

Trattner, Christoph and David Elsweiler (Nov. 2017a). "Food Recommender Systems: Important Contributions, Challenges and Future Research Directions". In: *arXiv:1711.02760 [cs]*. arXiv: 1711.02760. URL: http://arxiv.org/abs/1711.02760 (visited on 12/10/2018).

— (2017b). "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*. WWW '17. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, pp. 489–498. ISBN: 978-1-4503-4913-0. DOI: 10.1145/3038912.3052573. URL: https://doi.org/10.1145/3038912.3052573 (visited on 12/13/2018).

Wiegand, Michael, Benjamin Roth, and Dietrich Klakow (2012a). "Data-Driven Knowledge Extraction for the Food Domain". In: *KONVENS*.

— (2012b). "Knowledge Acquisition with Natural Language Processing in the Food Domain : Potential and Challenges". In:

Wiegand, Michael, Benjamin Roth, Eva Lasarcyk, Stephanie Köser, and Dietrich Klakow (2012c). "A Gold Standard for Relation Extraction in the Food Domain". In: *LREC*. Istanbul. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/136_Paper.pdf.

Willis, Angelica, Elbert Lin, and Brian Zhang (2017). "Forage : Optimizing Food Use With Machine Learning Generated Recipes". In:

Wittgenstein, Ludwig (1953). *Philosophical investigations*. German and English; added title pages in German. OCLC: 371912.

Yagcioglu, Semih, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis (Sept. 2018). "RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes". In: *arXiv:1809.00812 [cs]*. arXiv: 1809.00812. URL: http://arxiv.org/abs/1809.00812 (visited on 12/10/2018).

Yamakata, Yoko, Shinji Imahori, Yuichi Sugiyama, Shinsuke Mori, and Katsumi Tanaka (2013). "Feature Extraction and Summarization of Recipes Using Flow Graph". en. In: *Social Informatics*. Ed. by Adam Jatowt, Ee-Peng Lim, Ying Ding, Asako Miura, Taro Tezuka, Gaël Dias, Katsumi Tanaka, Andrew Flanagin, and Bing Tian Dai. Lecture Notes in Computer Science. Springer International Publishing, pp. 241–254. ISBN: 978-3-319-03260-3.

Yuan, Guo-Xun, Chia-Hua Ho, and Chih-Jen Lin (2012). "Recent Advances of Large-Scale Linear Classification". In: *Proceedings of the IEEE* 100, pp. 2584–2603. DOI: 10.1109/JPROC.2012.2188013.