What You Use, Not What You Do: Automatic Classification and Similarity Detection of Recipes

Hanna Kicherer[a], Marcel Dittrich[b], Lukas Grebe[b], Christian Scheible[c], Roman Klinger[a]

[a] *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
{roman.klinger,hanna.kicherer}@ims.uni-stuttgart.de
[b] *Chefkoch GmbH, Rheinwerk 3, Joseph-Schumpeter-Allee 33, 53227 Bonn, Germany*
{marcel.dittrich,lukas.grebe}@chefkoch.de
[c] *Trusted Shops GmbH, Subbelrather Straße 15c, 50823 Köln, Germany*
christian.scheible@etrusted.com

Abstract

Social media data is notoriously noisy and unclean. Recipe collections and their manual categorization built by users are no exception. However, a consistent and transparent categorization is vital to users who search for a specific entry. Similarly, curators are faced with the same challenge given a large collection of existing recipes: They first need to understand the data to be able to build a clean system of categories. This paper presents an empirical study using machine learning classifiers (logistic regression and decision trees) for the automatic classification of recipes on the German cooking website Chefkoch.de. The central question we aim at answering is: Which information is necessary to perform well at this task? In particular, we compare features extracted from the free text instructions of the recipe to those taken from the list of ingredients. On a sample of 5,000 recipes with 87 classes, our feature analysis shows that a combination of nouns from the textual description of the recipe with ingredient features performs best in the logistic regression model (48 % F_1). Nouns alone achieve 45 % F_1 and ingredients alone 46 % F_1. However, other word classes do not complement the information from nouns. Decision trees constantly underperform the logistic regression, however, lead to an interpretable model. On a bigger training set of 50,000 instances, the best configuration shows an improvement to 57 % highlighting the importance of a sizeable data set. In addition, we report on the use of these feature vectors for similarity search and ranking of recipes and evaluate on the task of (near) duplicate detection. We show that our method can reduce the manual curation with precision@3 = 0.52.

Keywords: recipe, cooking, food, classification, multi-label, text mining, similarity search

1. Introduction

In 2012, 63.7 % of Germans used the Internet as source of inspiration for cooking [17]. One popular cooking website is *Chefkoch.de*[1], where every user can contribute to a shared database of recipes and discussions. The result of this social network approach is a large data set of diverse and potentially noisy information.

Commonly, a recipe consists of at least three major parts, exemplified in Figure 1: the *list of ingredients*, whose entries consist of an ingredient type, an amount, and a unit; the *cooking instructions* wherein the steps for preparing the dish using the ingredients is described in natural language; and *meta data* which supplies for instance information about the preparation time and difficulty. Each recipe is assigned to a number of categories, for instance of subtypes *regional* (*e.g.*, *Germany*, *Malta*, *USA and Canada*), *seasonal* (*e.g.*, *Christmas*, *spring*, *winter*), or *course* (*e.g.*, *vegetables*, *pork*, *dessert*) (see `http://www.chefkoch.de/rezepte/kategorien/`). When submitting new recipes, both users and curators may not understand the full range and structure of the category system. Thus, each new recipe may introduce additional noise into the database. Therefore, contributors as well as database curators would benefit from automatic support in choosing appropriate categories for a recipe.

Similarly, a contributor might not be able to find a specific recipe and therefore opt for adding it, though it might exist already. This includes additional noise. A method for discovering potentially similar or even equivalent recipes can help in keeping the number of near duplicates low and, on the query side, help in finding variations of a recipe.

We address both tasks in this paper. For the categorization part, we estimate a statistical model of category assignments based on recipes in the Chefkoch.de database. This model will be beneficial for database completion, adjustment, and consolidation of existing recipes and will help users and curators by suggesting categories for a new recipe. Our main contributions are experiments to investigate the performance of the model: (1) We compare logistic regression and decision tree classification models taking into account different types of information from the ingredient list and textual description. In particular, we make use of ontological information to generalize over spe-

[1] `http://www.chefkoch.de` (all URLs in this paper: last accessed on 2017-01-31.)

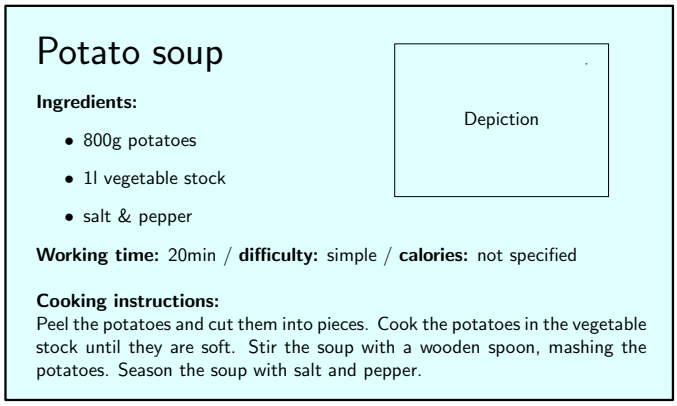

Figure 1: Example recipe.

cific ingredients and we investigate different subtypes of word classes. (2) Our evaluation of different feature sets shows that nouns are more important than verbs and the order of ingredients in the list is only of limited importance for classification. (3) We provide a visualization of the recipes with using dimensionality reduction to contribute to a better understanding of the data. This also highlights which subset of categories are specifically challenging. We work with German data which is characterized by rich morphology, *e.g.*, regarding the variety of plural forms and compounds. However, we do not incorporate any specific handling of German.

To discover similar recipes, potentially duplicates, but also variations of recipes, we propose to use the feature vectors built for the classification task in an unsupervised retrieval setting. We evaluate this method on (near) duplicate detection.

2. Related Work

2.1. Recipe as Subject of Research

Recipes have been the subject of several previous studies. We focus on text-oriented research here (as opposed to for example the classification of image data [1]). Most related to this paper is prior work on recipe classification by Su et al. [30]. They analyzed the correlation between recipe cuisines and ingredients for recipes from *Food.com* (`http://food.com`). They trained support vector machines to predict a single cuisine, using ingredients as features. Overall, they achieved a precision and recall of about 75%.

Naik and Polamreddi [22] performed classification with different models and principle component analysis on *Epicurious* (`http://epicurious.com`) and *Menupan* (`http://menupan.com`) and reached 75 % accuracy. Recent work by Hendrickx et al. [10] predicted wine features from reviews. Min et al. used deep neural networks to learn joint representations of images and ingredients [18]. One of their goals was to classify recipes by cuisine.

Oberlaender and Bostan opted for modelling a recipe in a distributional manner in contrast to using ontologies [24]. Each ingredient was represented as a dense vector, a combination of these vectors represents a full recipe which they used with a long short-term memory neural network (LSTM) to generate recipes.

Wang and Li [33] developed a system to teach cooking, which includes pointing out potential problems that may arise while preparing a dish and offering solutions, based on action or flow graph structures [15, 34] and predicate-argument structures [12, 20, 19]. Based on such graph structures, the similarity as well as specific characteristics of recipes can be calculated [38].

In contrast to the work presented above, we perform multi-class classification while the dominant approach appears to deal only with single-class associations. Our set of classes goes beyond pure cuisine and includes information such as preparation method and course. Lastly, we take into consideration more information about ingredients (like amount and unit type) and the cooking instructions.

2.2. German Recipes in Focus

There is little prior work on German recipe data. Wiegand et al. [36, 35] analyze on *Chefkoch.de* whether a food item can be substituted by another, whether it suits a specific event, and whether it is mentioned as an ingredient of a specific dish. Reiplinger et al. [27] applied distant supervision for the estimation of relation extraction models. Donalies analyzed the use of intentions of language use to name recipes in German [5].

To our knowledge, the work presented here is the first to address multi-class classification of German recipes.

2.3. Domain Knowledge from Ontologies

Next to the automatic analysis of recipe data, previous work attempted to build formal representations of recipes as ontologies. Xie et al. [37] state that such domain knowledge is a prerequisite to model the semantics of a recipe correctly. The *cooking ontology* [23] is such a formalization, specializing in ingredients, condiments, kitchen tools, and movements while cooking and contains lexical variants in Japanese. Other food-related ontologies are for instance the *BBC Food Ontology* (`http://www.bbc.co.uk/ontologies/fo`) and the *LOV Food Ontology* (`http://lov.okfn.org/dataset/lov/vocabs/food`). In this paper, we will make use of *WikiTaaable* [3, 28]. It contains lexical variants for English, French, German, and Spanish and includes a recipe and an ingredient ontology (2875 food items, 540 in German), among other parts. The ontology represents food items hierarchically, and contains their nutritional values and compatibility with dietary restrictions.

2.4. Ingredients as a Core Feature of Recipes

Previous studies have already figured out that ingredients reveal important information about a recipe. Chung et al. estimated the relatedness of an ingredient to a recipe category using frequency measures, working on recipes from the Japanese recipe platform *Rakuten* (`http://recipe.rakuten.co.jp`) [2]. Ozaki et al. [25] extracted characteristic ingredients by cuisine type for recipes from *Cookpad* (`https://cookpad.com/`). They also take cooking actions into consideration. This

is similar to our analysis of ingredients and recipe classes using pointwise mutual information.

In experiments on *Allrecipes* (`http://allrecipes.com`), Kim et al. [13] found that rare ingredients are more important to characterize a recipe. They employed entropy-based measures to set up a similarity network and clustered to group by cuisine. Similarly, Ghewari et al. predicted the geographical origin of recipes on *Yummly* (`http://www.yummly.com`) with 78 % accuracy using ingredient information [7]. Similarly to our work, they ranked features for each cuisine by pointwise mutual information.

To automatically extract relationships between ingredients, Gaillard et al. [6] developed an interactive adaptation knowledge model to substitute ingredients of a given recipe. They extract ingredient choices in recipes (*e.g.*, "500g butter or margarine") from the *WikiTaaable* recipe ontology as their knowledge base. Mota et al. extend this approach by using a food ontology [21]. Sidochi et al. extract substitutes in recipes from *Ajinomoto* (`http://park.ajinomoto.co.jp`) by taking into account process information [29]. Teng et al. [32] consider ingredients replaceable when they regularly co-occur with the same ingredients in other recipes from *Allrecipes*.

Aiming at understanding the internal structure of recipes, Greene [8], Greene and McKaig [9] segment each entry of the ingredients list in recipes from a database of the New York Times into *name*, *unit*, *quantity*, *comment*, and *other* using sequential prediction with linear-chain conditional random fields. Similarly, Jonsson [11] split textual descriptions from *Allrecipes* into *ingredient*, *tool*, *action*, *intermediate product*, *amount*, *unit* and *other* and detect relations between these classes.

In this paper, we combine information extracted from multiple parts of the recipe as well as from external sources such as the ingredient ontology introduced above. To this end, we design features for multi-label classification and for the estimation of recipe similarity, which are in part inspired by the work summarized above. We discuss the features in the following section.

3. Models and Feature Sets

We frame the task of automatically categorizing a recipe as a multi-label classification problem. Each recipe is represented as a high-dimensional feature vector which is the input for multiple binary prediction models, one for each category. The output of each model corresponds to an estimate of the probability of the recipe being associated with this category. Throughout this paper, we report our results with binary logistic regression models [4]. Furthermore, we also briefly discuss results for decision trees [26] which were outperformed by logistic regression models in all configurations using more than one feature type. Our main experiments use logistic regression, a state-of-the-art classifier which can deal with correlations of features. However, these correlations can make it hard to interpret the values of the resulting models. Decision trees therefore complement our analysis as interpretable models, which enable an analysis though they are outperformed by the logistic regression. The inputs to each of these models are different feature sets and their combinations described below.

From the cooking instructions, we extract bag-of-words features without changing case from the textual description (abbreviated as WORDS) as a baseline (changing case does not lead to performance differences). In the example recipe in Figure 1, these are all words from the cooking instructions (*e.g.*, "Peel", "the", "potatoes"). To investigate which word classes are relevant, we use the subsets of VERBS (*e.g.*, "Peel", "cut") and NOUNS (*e.g.*, "potatoes", "pieces", "spoon"). We perform POS tagging with the Stanford Core NLP [16] with the German model.

Based on the ingredient list, we use a variety of features, with the bag of INGREDIENTS ("potatoes", "vegetable stock", "salt & pepper") being the most fundamental one. We introduce generalization by expanding this feature to INGREDIENT CLASSES (IC), adding all parents as defined in the WikiTaaable ontology (adding *vegetable* for *potato*). We encode the order of the list through INGREDIENT RANKS features for the first and second position in the list (IR, e.g., *potatoes@1* and *vegetable-stock@2*).

The feature set UNIT TYPE (UT) adds binary features for each combination of ingredient and unit (*potatoes-weight_unit*, *vegetable_stock-volume_unit*) as an approximation for ingredient amounts. As a motivating example, if flour is specified in kilograms, it is likely to be used to create dough for baking. In contrast, an amount given in table spoons could indicate its use in soup. We restrict this feature to mg, g, kg, ml, cl, dl, l, and spoons, tea-, table spoon, level or heaped.

Similarly to the ingredient rank feature, we assume the ingredient with the highest amount (HIGH. A. INGR., HAI) to be important. Note that this feature requires unit normalization. We normalize amounts of the same type (e.g. kg, g and mg) to the same unit and approximately convert units of different type by using the conversion values of water (e.g. 1kg corresponds to 1l). In the example in Figure 1, the feature *vegetable-stock_highest-amount* holds. The template INGREDIENT NUMBER generates a binary feature for each possible count. We expect it to be of value for low counts which occur more frequently (in the example, the feature *3-ingredients* holds).

All features above consider information from the recipe text or from the ingredient list independently. To combine information from both sources, we introduce the feature sets CONTEXT WORDS (CW), CONTEXT VERBS (CV), and CONTEXT NOUNS (CN). For each ingredient in the list, all occurrences in the recipe text are detected. All combinations of each word, verb, or noun, respectively, with the ingredient in the same sentence form another feature. For instance for verbs, the first sentence of our example yields features "peel-potato" and "cut-potato".

The only feature we extract from the meta-data is PREPARATION TIME (PT) which we represent through stacked bins (in the example, *Preparation > 5min* and *Preparation>15min* hold). We sum preparation time, cooking or baking time, and resting time.

Ingredient Detection. In order to extract the CONTEXT WORDS, CONTEXT VERBS and CONTEXT NOUNS features, occurrences of ingredients in the cooking instructions need to be detected. We

address this issue with a simple rule-based approach with two steps.

In the first step, we perform a term expansion for each ingredient from the ingredient list by separating the main component and additional information (e.g., "Cheese (Emmental)" is separated into "Cheese" and "Emmental") and expanding plural brackets into the singular and plural version of the ingredient (e.g., "Egg(s)" is expanded to "Egg", "Eggs"). We then substitute all terms that are covered by the WikiTaaable ontology with each governing superclass. As an example, for the ingredient "Apple(s), Braeburn" we would get "Apple", "Apples", "Braeburn", "Fruit", "Pomme Fruit".

In the second step, for each of the terms we search for a match among all tokens from the instructions. To cover special cases of ingredient usage, we apply a set of matching rules, including compound splitting (as compound parts are not delimited by spaces in German) and resolution of hanging hyphens (e.g. "apple- and cherry juicy" matches "apple juice").

To measure the performance of ingredient detection, we evaluate the process on 100 randomly chosen cooking instructions which we annotated manually for ingredient occurrences. We obtain a macro-averaged precision of 86% and a macro-averaged recall of 76% which leads to a macro F1-score of 81%. In 9 out of the 100 recipes, all ingredient occurrences were detected correctly.

4. Results

4.1. Experimental Setup for Classification

We use a database dump of 263,854 Chefkoch.de recipes from June 2016. The minimum number of ingredients in any recipe is 1, the maximum 61, the average is 9.98. The overall number of unique ingredients is 3,954. The number of recipes varies across categories (on average 7,825.3, however, the median is only 1,592). The categories on Chefkoch.de are structured hierarchically with up to four levels. Within this hierarchy, recipes associated with any leaf node are assumed to also belong to all parent categories on the path to the root. Thus, we need to predict only association of a recipe to the leaves of the hierarchy. This means that we use "flat classifiers", in contrast to a global approach which takes into account the whole hierarchy in the model ("big-bang") or local classification decisions ("top-down") [31].

This leads to a total of 182 categories, out of which 162 are leaves. There are 7 nodes on the top level (10,43 children on average), 73 nodes on the second level (1,38 children on average), 101 nodes on the third level (0,01 children on average) and one node on the forth level. The leaf nodes are not all in the same height but spread over the second (61 leaves), third (100 leaves) and fourth level (1 leaf).

An excerpt of the hierarchy is shown in Figure 2. The minimum number of categories assignment to a recipe is 0, the maximum 36, the mean is 4.8. The category with the fewest recipes is "Malta", which contains 30 entries. "Baking" is the largest category, containing 67,492 recipes.

For the classification experiments and evaluation, we use a random sample of 20 % (52,771) of the recipes as our test set. As our main goal of this paper is to develop a model which is suitable for database consolidation, we omit all categories which occur fewer than 500 times in this test set (≈ 1 %). After this sampling step, 87 categories remain in the dataset which we use in our experiments.[2] Note that this step does not omit any recipes due to the multi-label classification setting. The logistic regression models use L_2 regularization and a stopping criterion of $\varepsilon = 1.0$. We train the decision trees with a minimum of two recipes in a leaf node and pruned with a confidence factor of 0.5.

As feature vectors grow rapidly for some feature types, especially the feature types extracted from the cooking instructions, the working memory of the available computers were not sufficient to train these models on all available data. Figure 3 shows how feature vectors grow with the increasing number of recipes read from the training set. Therefore, we first compare the results of different feature sets using a training set of 5,000 randomly selected instances. Among these variations, we determine the best model configuration through evaluation on the development set. We then train this model on a larger set of 50,000 instances. Test results in the paper are reported on a test set of 5,000 instances for all models.

4.2. Classification Results

In the following, we will first discuss the differences between feature sets in detail under consideration of the classification method (Section 4.2.1). Further, we provide an analysis of high-level categories (Section 4.2.2) and a visualization of the data (Section 4.3). In addition we perform a more in depth feature analysis (Section 4.4) and error analysis (Section 4.5). Finally, we add an additional experiment in which we perform similarity search based on the feature-based respresentation of the recipes (Section 4.6).

4.2.1. Comparison of Feature Sets

Table 1 shows macro-averaged F_1 over all categories (we do not report accuracy values due to the unbalancedness of the data) with different feature sets for decision trees and logistic regression classifiers. In the following, we focus our discussion on the logistic regression models.

Considering only information from the instruction text, we find that the model using all WORDS yields 46 % F_1. Using only NOUNS performs comparably well, albeit at a loss of precision compensated for by higher recall. In contrast, VERBS in isolation lead to a drop by 14 percentage points. Information about *entities* involved in the preparation process is much more important than information about *activities*.

The INGREDIENTS feature alone yields an F_1 of 43 %, which is comparable to the instructions-based results above. Most other features in this group (IC, IR, HAI, UT) perform relatively poor.

[2] Listed at `http://www.ims.uni-stuttgart.de/data/recipe-categorization`

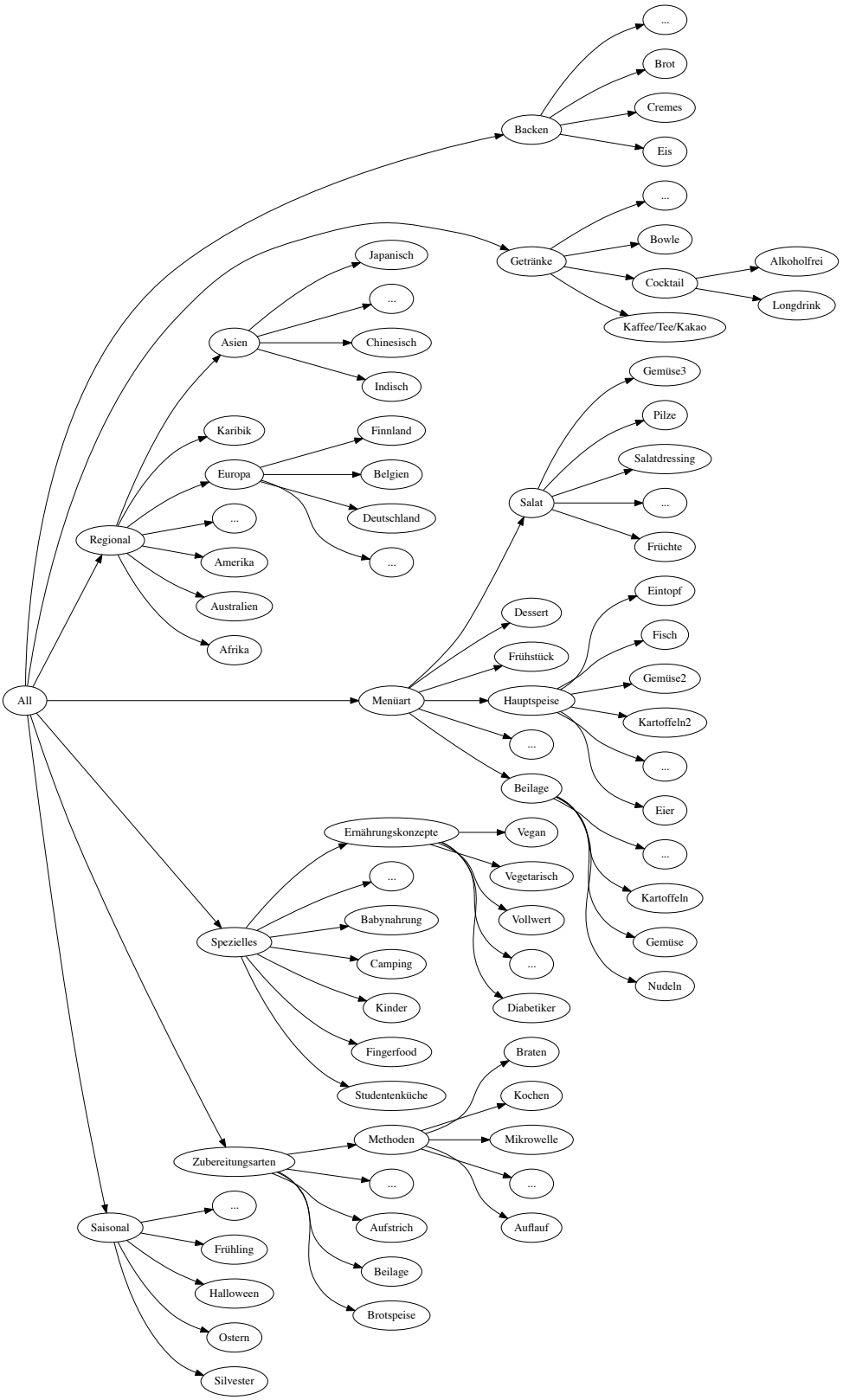

Figure 2: Excerpt of the label hierarchy at Chefkoch.de. The complete hierarchy is available at `https://www.chefkoch.de/rezepte/kategorien/`.

		Logistic Regression			Decision Trees		
Feature combination	Features	P	R	F_1	P	R	F_1
WORDS (Baseline)	19,942	**63**	36	**46**	50	30	38
NOUNS	11,176	58	**37**	45	**52**	**33**	**40**
VERBS	3,580	46	25	32	40	22	28
INGREDIENTS	1,448	**58**	**34**	**43**	52	**34**	**41**
INGR. CLASSES (IC)	61	23	19	21	52	19	27
INGR. RANKS (IR)	1,302	42	21	28	58	21	30
HIGH. A. INGR. (HAI)	633	14	24	18	**62**	15	24
UNIT TYPE (UT)	1,548	24	30	27	53	25	34
INGR. NUMBER (IN)	32	10	0	0	41	0	0
PREP. TIME (PT)	9	3	2	5	0	0	0
CONTEXT WORDS (CW)	194,042	**41**	**25**	**31**	40	**24**	**30**
CONTEXT NOUNS (CN)	71,311	39	24	30	**45**	23	**30**
CONTEXT VERBS (CV)	41,269	29	24	26	41	19	26
INGR & IC	1,509	60	34	44	52	34	41
INGR & IR	2,750	58	35	44	52	33	40
INGR & UT	2,996	56	35	43	52	33	40
INGR & HAI	2,081	58	35	44	52	33	41
INGR & WORDS	21,390	**64**	39	**48**	**53**	33	40
INGR & NOUNS	12,624	61	**40**	**48**	**53**	**35**	**42**
INGR & VERBS	5,028	58	38	46	52	34	41
INGR & CW	195,490	**63**	27	38	42	**27**	32
INGR & CN	72,759	62	**29**	**39**	**48**	**27**	**34**
INGR & CV	42,717	60	**29**	**39**	47	26	33
INGR & IC & IR	2,813	**58**	**36**	44	**52**	**33**	**40**
INGR & IR & HAI	3,391	**58**	**36**	**45**	**52**	32	**40**
INGR & IC & IR & HAI	3,452	**58**	**37**	**45**	**52**	**33**	**40**

Table 1: Precision, recall and F1 measures in percent for recipe classification with 5,000 training instances and different feature combinations for logistic regression models and decision trees. The best results in each column for each feature group are highlighted in bold.

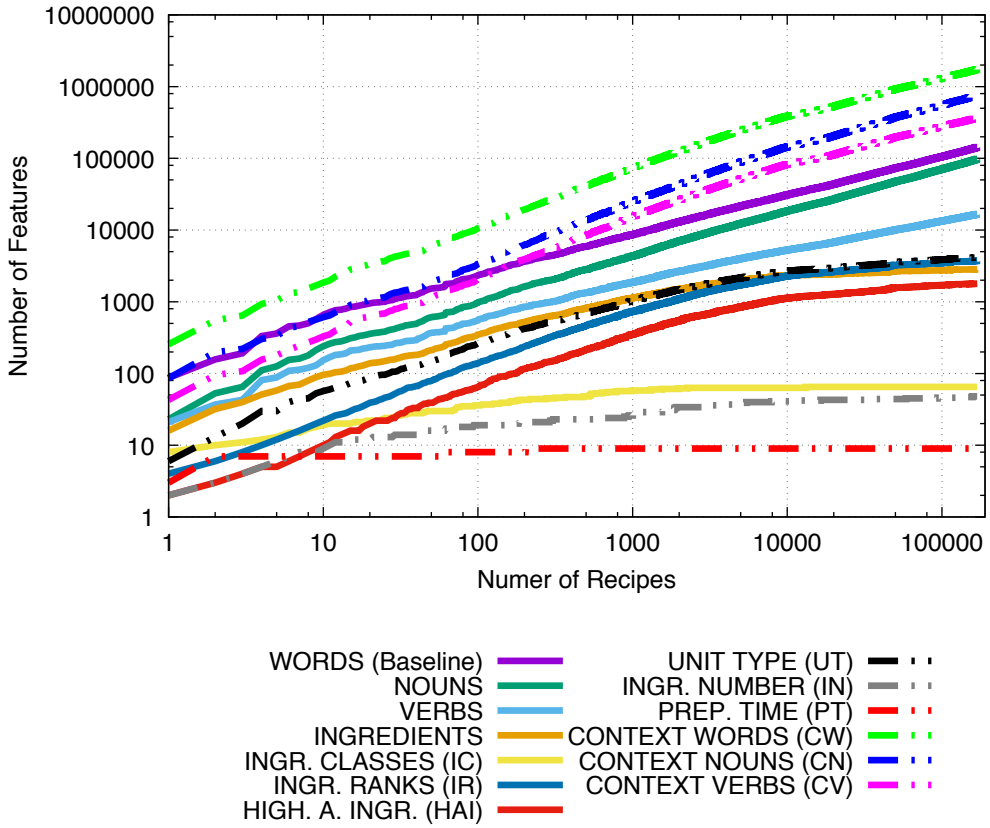

Figure 3: Feature vector growth dependent on the number of read recipes. Note the logarithmic scale of the axes.

IN and PREPARATION TIME provides no useful signal. Using CONTEXT WORDS, CONTEXT NOUNS, and CONTEXT VERBS instead of their standalone counterparts leads to losses of up to 15 percentage points. We suspect that the main reason is sparsity due to large feature set sizes.

First, note that any combination of the INGREDIENTS feature with other features based either on the instructions or the list of ingredients yields an improvement. Conversely, combinations with the CONTEXT WORDS, CONTEXT NOUNS, and CONTEXT VERBS lead to drops, which is another indicator for sparsity – the CONTEXT WORDS features vastly outnumber the INGREDIENTS features.

Combinations of more than two feature sets do not lead to further improvements. Our overall best model makes use of INGREDIENTS and NOUNS, performing at 48 % F_1. In order to determine whether performance improves with the availability of a larger training set, we re-run the experiment for this setting with a sample of 50,000 training instances. We find a considerable improvement in precision (67 %), recall (49 %), and F_1 (57 %).

Turning to decision trees, we find similar patterns for the different features results as we did for the logistic regression models: INGR. NUMBER and PREP. TIME provide no useful signals, WORDS, NOUNS and VERBS perform better than CONTEXT WORDS, CONTEXT NOUNS and CONTEXT VERBS. This holds both in the single feature condition and in combination with the INGREDIENTS feature. Combinations of more than two feature do not lead to any improvement anymore.

In direct comparison, we see that logistic regression models outperform decision trees for all conditions with combined features (lower section of the table). However, for the single features INGREDIENT CLASSES, INGREDIENT RANKS, HIGH. A. INGR. and UNIT TYPE the decision trees reveal better results, for INGREDIENT NUMBER, CONTEXT NOUNS and CONTEXT VERBS the F_1 scores are on par in the majority of cases.

Unlike logistic regression models, decision trees can be easily visualized, as they consist of a set of rules over features of the instance to be classified. Figure 4 shows part of the decision tree model for the category "baking", trained with the INGREDIENTS feature.

4.2.2. Comparison of Categories

The comparison of macro-F_1 estimates above provides only a coarse analysis as it summarizes over a total of 87 categories. As we are unable to provide a full results listing due to space constraints, we highlight those categories where our logistic regression model performs best and worst, respectively, in Table 2. For this comparison, we make use of the results using the training set of size 50,000 introduced above. This analysis is based on the best-performing model, INGREDIENTS and NOUNS.

The category for which our model performs best is "baking"

Top Category	P	R	F_1
Backen (baking)	89	88	88
Pasta & Nudel (pasta)	88	87	87
Kuchen (cake)	85	85	85
Brot/Brötchen (bread)	87	78	82
Kekse & Plätzchen (cookies)	86	78	82
Fisch (fish)	84	76	80
Rind (beef)	81	78	80
Torten	82	73	77
Vegetarisch (vegetarian)	77	76	76
Dessert (dessert)	80	71	75
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
Bottom Category	**P**	**R**	**F_1**
Kalorienarm (low-calorie)	36	11	17
Resteverwertung (leftover meals)	36	11	17
Dünsten (steaming)	32	10	15
Studentenküche (students' cuisine)	35	10	15
Camping (camping)	33	9	14
Spezial (Special)	22	10	14
Beilage (side dish)	28	8	13
Frankreich (France)	32	6	11
Raffiniert & preiswert (clever & cheap)	22	4	7
Geheimrezepte (secret recipes)	09	01	2

Table 2: The 10 best (top) and 10 worst (bottom) categories with INGREDIENTS and NOUNS feature combination.

with an F_1 measure of 88 %. A large amount of the remaining top 10 categories are defined by certain ingredients (henceforth *defining ingredients*) of the dish, such as "pasta", "beef", or "fish". In contrast, the 10 categories where performance is worst mostly center around abstract ideas or processes. For instance, the most difficult category is "secret recipes" with only 2 % F_1. This and other categories such as "cheap & clever", "camping", or "students' cuisine" require world knowledge beyond what can be learned from the recipes alone. Overall, among the 87 categories considered in this experiment, 41 categories have an F_1 above 50 %. The remaining 46 categories score lower. Table 3 reports the mean results for leaves in specific subtrees in the hierarchy. The model performs best for leaves under the inner node "baking & desserts" – which is the largest category by recipe count – with an F_1 of 79 %. Next is "course" with 65 % F_1. "regional" (22 % F_1) is the most challenging category, followed by "special" (42 % F_1).

Overall, we find that there are certain types of categories, in particular those that have defining ingredients, where our model performs particularly well. This results suggest that more complex features may be necessary in order to fully capture more abstract ideas such as the purpose or cultural origin of a dish.

4.3. Visualization

One hypothesis as to why some categories are more difficult to predict than others is that the conceptual definition and distinction between them is unclear. To investigate this in more detail, we visualize randomly selected subsets of 5,000 recipes by projecting the feature matrix with INGREDIENTS features into two dimensions via t-SNE [14]. Figure 5 shows plots of the resulting spaces for six different root categories. Each point represents a recipe of a specific leaf category. "Overlap" denotes recipes that belong to more than one leaf. We find that some categories seem to be comparably easy to separate from others after projection, for example "dessert" in "course", or "baking" in "preparation methods". Other categories have recipes located as single cluster but are subsumed by other categories,

Category Class	**Prec.**	**Rec.**	**F1**
Backen & Süßspeisen (Baking & Desserts)	83	75	79
Menärt (Course)	71	60	65
Zubereitungsarten (Preparation Methods)	68	53	60
Saisonal (Seasonal)	57	34	43
Spezielles (Special)	58	33	42
Regional (Regional)	39	16	22

Table 3: Macro evaluation scores for feature combination INGREDIENTS, INGR. CLASSES, INGR. RANKS and HIGH. A. INGR. over all categories of a superclass of the hierarchy with the logistic regression classifier.

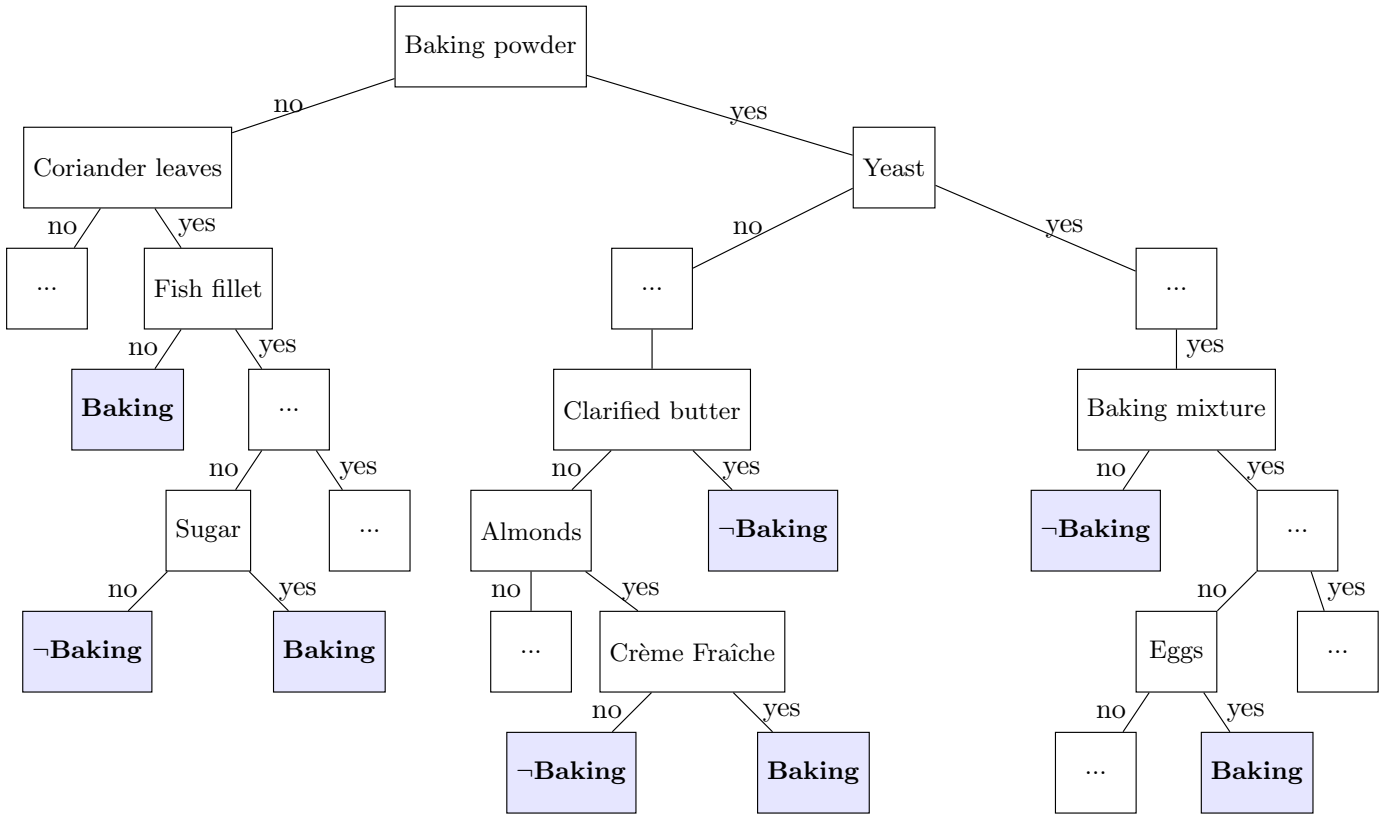

Figure 4: Visualization of an excerpt of the decision tree for category "baking", trained with the INGREDIENTS feature.

e.g., "cookies" in "baking & desserts" and "Christmas" in "seasonal".

Another phenomenon is that the recipes of a category are spread across the whole plot but with varying density (like "quick and easy" in "special"). Some categories do not form clusters, such as "cake". This pattern is particularly noticeable for "special" where most categories (except for "quick and easy") are indistinguishable. This result suggests that some categories are more difficult to distinguish than others. This may be caused either by inaccurate category definitions or by inadequate feature representations.

4.4. Feature Analysis

To understand the problem structure of the classification task, we generate lists of features ranked by pointwise mutual information. The complete list of most relevant features is available as download[3]. Here, we limit ourselves to an exemplifying discussion of the categories "pork" and "vegetarian". Interestingly, the ingredient "pork" does not appear in the list of typical features for the category "pork". This is in line with previous results on wine reviews [10]. In contrast, the most relevant features are "yellow pepper", "fried pepper", "orange mustard", and "barbecue sausages". For other categories with defining ingredients such as "eggs", we see a similar pattern: The defining ingredient is often not among the most typical features. This is presumably because eggs occur in many dishes and are therefore not precise enough to distinguish egg recipes from non-egg recipes. Putting eggs into focus happens by cooccurrence with other specific ingredients.

Most atypical recipes in the "pork" category are "vanilla sugar", "strawberries", "powdered sugar" and "raspberries", all of which match our gustatory intuition. As the most atypical features for "vegetarian" recipes, we find fish and meat ingredients, whereas the list of most typical features are mainly vegetables (*e.g.*, tomatoes, flour, green spelt grain, rice cream, falafel).

4.5. Error Analysis

For a qualitative error analysis, we pick the category "Pasta". We identified three prominent classes of false positives: first, recipes containing noodles as their main ingredient (*e.g.* "Italienischer Pastasalat" (Italian pasta salad) or "Sommerspaghetti" (summer spaghetti)); second, recipes where noodles are a side dish (*e.g.* "Schweinelendchen in Käsesoße" (pork loin in cheese sauce)); third, recipes for pasta dough (which are often not labeled as *pasta* in the database). Note that the first and third case are arguably annotation errors caused by the lack of annotation guidelines. Thus, error analysis could be used to consolidate the recipe database either manually or semi-automatically. Conversely, the second case may be caused by features not being expressive enough. For example, the feature "Nudeln" (noodles) is only a moderately good indicator for pasta dishes as it

[3] `http://www.ims.uni-stuttgart.de/data/recipe-categorization`

Figure 5: Visualization of categories based on INGREDIENTS with t-SNE.

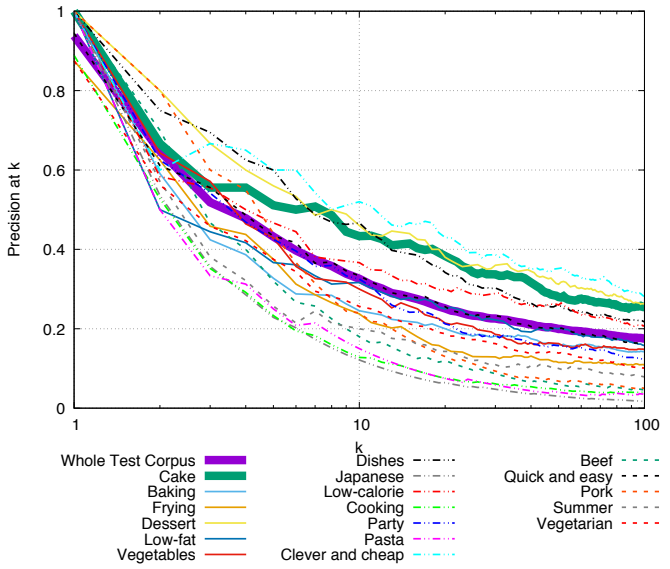

Figure 6: Precision@k for different values of k (log axis) for the similarity search experiment for categories with more than five results in our evaluation sample.

does not capture the importance of the ingredient. This points towards a need for more sophisticated features.

The set of false negative "Pasta" dishes consists mainly of sauces that are served with pasta but do not contain any noodles themselves (*e.g.* "Bolognese sauce"). Another frequent cause of false negatives are again underspecified annotation guidelines. For instance, "Ravioli mit zweierlei Füllung" (ravioli with twofold filling) is an example of a recipe for dough annotated as "Pasta". However, the features for this recipe are either too basic (e.g., "Teig" (dough)) or too specific (e.g., "Teigrädchen" (dough circles)) to be good predictors for this category on their own.

4.6. Similarity Search

In addition to the supervised classification setup described in Section 4, we evaluate the viability of the feature sets in an unsupervised setting. Our intended use case is similarity search, *i.e.*, the retrieval of related recipes for a reference recipe. To accomplish this, we use cosine similarity over the vector space spanned by the best feature set for classification: INGREDIENTS and NOUNS.

For the evaluation in the setting of discovering similar, potentially duplicate recipes, we randomly sample 81 query recipes from the full data set (described in Section 4.1). For each query recipe, we calculate its cosine similarity to every other recipe and rank them by similarity. We annotate each pair of query and candidate for relevance, *i.e.*, that the candidate is similar to the query in the sense that it is a (near) duplicate. Based on this annotation, we calculate the averaged precision@k curve for $1 \leq k \leq 100$. Please note, however, that we did not calculate inter-annotator agreement and did not perform multiple annotations for each query-recipe pair. The annotators were three authors of this paper.

Figure 6 shows precision@k over our manually annotated test set. We find that precision is high at the top of the ranking and then degrades rapidly for lower-ranked items. At the top ranked position, we achieve a precision@1 of 0.94. Precision@5 is at 0.43, while Precision@10 is at 0.35. Precision@100 is 0.18. These results show that the similarity prediction is particularly useful when a small set of related recipes is to be retrieved. Given a duplicate detection task as an instance of similarity search, these results show that most duplicates are indeed in the top results of the ranked list. The decreasing value for precision@k is not necessarily an indicator for inferiority of our method. Instead, it shows that the average number of similar recipes for a query is limited.

This is backed by differences for different categories in our evaluation data. For instance, query recipes from the category "Cakes" show a higher result (amongst others). Here, the results show a higher precision@k curve, for instance with precision@5 of 0.51.

We observe differences when comparing the similarity search experiment to the classification experiment. Only similarity search for the categories "cake" and "dessert", which performed well in the classification experiment, scored better than average. The categories "baking", "pasta", "clever and cheap", "beef" and "vegetarian", which were also amongst the 10 best performing categories for classification, yielded scores worse than average. However, the category "clever and cheap", that performed second worst in the classification experiment is the best performing category for similarity search (among the examined categories).

5. Discussion, Conclusion & Future Work

In this paper, we have shown that logistic regression can classify recipes on the Chefkoch.de database with up to 57 % F_1. The decision tree classifier underperforms but allows for gaining insight in the structure of recipes and the influence of ingredients on the category, as shown on the example of the category "Baking". Feature analysis revealed that ingredients alone are nearly as good an indicator as the recipe description. Information from both sources complements each other.

We expected the combination of verbs with ingredients to be superior to other word classes and features from the classes separately. Surprisingly, our best model contradicts our intuition: Nouns are more important for classification than verbs. Combining verbs and ingredients even causes a drop in performance, presumably due to data sparsity with the resulting large feature set and overfitting to the comparably small number of training instances. We conclude that nouns are more important than activities in the description. Our error analysis revealed that many classification mistakes arise due to inconsistencies in the dataset. This suggests the applicability of our model to curate the database as well as to support users in finding appropriate categories.

Visualization of our recipe feature spaces highlights the difficulty of the task. For some categories, classification is comparably simple, while for others it remains challenging. This is, at least partially, due to the selection of our feature sets

– for instance the visualization based on ingredients suggests that subcategories of baking and desserts are difficult to distinguish. However, features which take into account the process of preparation may be able to measure the difference between, for instance, tortes and cake.

For future work, we will investigate the use of our statistical model for supporting manual database curation. After correcting part of the database, we can retrain the model to spot new inconsistencies. This leads to an iterative cleaning process. To address challenges presented by classes centered around culture (*e.g.*, purpose or origin of a dish), we could either make use of more external resources, such as databases of the origin of ingredients, or attempt a hierarchical classification approach, first determining the location of a dish on a higher level (*e.g.*, "Asia") before moving towards a more granular set of classes (e.g., detecting "South East Asia", then finally "Thailand"). For further feature engineering, we suggest two routes: On the one hand, we can enrich the textual description through structured information extraction; this includes more sophisticated grounding to ontological concepts and semantic role labeling. On the other hand, we suggest to develop embeddings of both ingredients and activities into a joint vector space. These will enable generalization over different substitutes and preparation procedures. Such an approach might also be helpful to learn what differentiates a defining ingredient from others. Another route of future work is the use of structured learning approaches to also make use of relations between different categories. Methods to be employed will include probabilistic graphical models.

[1] Cadene, R., 2015. Deep learning and image classification on a medium dataset of cooking recipes. Online: http://remicadene.com/uploads/summer-internship-2015.pdf.

[2] Chung, Y., 2012. Finding food entity relationships using user-generated data in recipe service. In: CIKM.

[3] Cordier, A., Lieber, J., Molli, P., Nauer, E., Skaf-Molli, H., Toussaint, Y., 2009. WIKITAAABLE: A semantic wiki as a blackboard for a textual case-based reasoning system. In: SemWiki 2009, ESWC.

[4] Cox, D. R., 1958. The regression analysis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological), 215–242.

[5] Donalies, E., 2017. Himmel und Erde – wie wir Gerichte benennen und warum wir das tun. Sprachreport 33 (3), 4–6.

[6] Gaillard, E., Nauer, E., Lefevre, M., Cordier, A., 2012. Extracting generic cooking adaptation knowledge for the TAAABLE case-based reasoning system. In: Cooking with computers workshop, ECAI.

[7] Ghewari, R., Raiyani, S., 2015. Predicting cuisine from ingredients. Online: https://cseweb.ucsd.edu/~jmcauley/cse255/reports/fa15/029.pdf.

[8] Greene, E., 2015. The New York Times Open Blog: Extracting structured data from recipes using conditional random fields. Online: https://open.blogs.nytimes.com/2015/04/09/extracting-structured-data-from-recipes-using-conditional-random-fields/, accessed: 2016-06-20.

[9] Greene, E., McKaig, A., 2016. The New York Times Open Blog: Our tagged ingredients data is now on GitHub. Online: https://open.blogs.nytimes.com/2016/04/27/structured-ingredients-data-tagging/, accessed: 2016-01-31.

[10] Hendrickx, I., Lefever, E., Croijmans, I., Majid, A., van den Bosch, A., August 2016. Very quaffable and great fun: Applying nlp to wine reviews. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Berlin, Germany, pp. 306–312.
URL http://anthology.aclweb.org/P16-2050

[11] Jonsson, E., 2015. Semantic word classification and temporal dependency detection on cooking recipes. Thesis, Linköpings universitet.

[12] Kiddon, C., Ponnuraj, G. T., Zettlemoyer, L., Choi, Y., 2015. Mise en place: Unsupervised interpretation of instructional recipes. In: EMNLP.

[13] Kim, S.-D., Lee, Y.-J., Kang, S. H., Cho, H.-G., Yoon, S.-M., 2015. Constructing cookery network based on ingredient entropy measure. Indian Journal of Science and Technology 8 (23).

[14] Maaten, L. v. d., Hinton, G., 2008. Visualizing data using t-SNE. Journal of Machine Learning Research 9, 1–48.

[15] Maeta, H., Sasada, T., Mori, S., 2014. A framework for recipe text interpretation. In: UbiComp Adjunct.

[16] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit. In: ACL Demo.

[17] Media, T. F., 2012. Wenn Sie kochen, woher beziehen Sie Anregungen für Ihre Gerichte? Online: https://de.statista.com/statistik/daten/studie/305898/umfrage/umfrage-in-deutschland-zu-den-rezept-quellen-fuer-selbstgekochte-gerichte/, (in German).

[18] Min, W., Jiang, S., Sang, J., Wang, H., Liu, X., Herranz, L., May 2017. Being a supercook: Joint food attributes and multimodal content modeling for recipe retrieval and exploration. IEEE Transactions on Multimedia 19 (5), 1100–1113.

[19] Mori, S., Maeta, H., Yamakata, Y., Sasada, T., 2014. Flow graph corpus from recipe texts. In: LREC.

[20] Mori, S., Sasada, T., Yamakata, Y., Yoshino, K., 2012. A machine learning approach to recipe text processing. In: Cooking with Computers workshop, ECAI.

[21] Mota, S. G., Agudo, B. D., 2012. ACook: Recipe adaptation using ontologies, case-based reasoning systems and knowledge discovery. In: Cooking with Computers workshop, ECAI.

[22] Naik, J., Polamreddi, V., 2015. Cuisine classification and recipe generation. Online: http://cs229.stanford.edu/proj2015/233_report.pdf.

[23] Nanba, H., Doi, Y., Tsujita, M., Takezawa, T., Sumiya, K., 2014. Construction of a cooking ontology from cooking recipes and patents. In: UbiComp Adjunct.

[24] Oberländer, J., Bostan, L., 2015. Distributional gastronomics. In: ESSLLI 2015 Student Session. Barcelona, Spain, pp. 196–203.

[25] Ozaki, T., Gao, X., Mizutani, M., March 2017. Extraction of characteristic sets of ingredients and cooking actions on cuisine type. In: 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA). pp. 509–513.

[26] Quinlan, J. R., 1993. C4.5: Programming for machine learning. Morgan Kauffmann, 38.

[27] Reiplinger, M., Wiegand, M., Klakow, D., 2014. Relation extraction for the food domain without labeled training data–is distant supervision the best solution? In: NLP. pp. 345–357.

[28] Ribeiro, R., Batista, F., Pardal, J. P., Mamede, N. J., Pinto, H. S., 2006. Cooking an ontology. In: AIMSA. pp. 213–221.

[29] Shidochi, Y., Takahashi, T., Ide, I., Murase, H., 2009. Finding replaceable materials in cooking recipe texts considering characteristic cooking actions. In: Workshop on Multimedia for Cooking and Eating Activities. CEA.

[30] Su, H., Lin, T.-W., Li, C.-T., Shan, M.-K., Chang, J., 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. ACM, New York, NY, USA, pp. 565–570.
URL http://doi.acm.org/10.1145/2638728.2641335

[31] Sun, A., Lim, E.-P., 2001. Hierarchical text classification and evaluation. In: Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE, pp. 521–528.

[32] Teng, C.-Y., Lin, Y.-R., Adamic, L. A., 2012. Recipe recommendation using ingredient networks. In: WebSci.

[33] Wang, L., Li, Q., 2007. A personalized recipe database system with user-centered adaptation and tutoring support. In: SIGMOD2007 Ph.D. Workshop on Innovative Database Research.

[34] Wang, L., Li, Q., Li, N., Dong, G., Yang, Y., 2008. Substructure similarity measurement in chinese recipes. In: WWW.

[35] Wiegand, M., Roth, B., Klakow, D., 2012. Data-driven knowledge extraction for the food domain. In: KONVENS. pp. 21–29.

[36] Wiegand, M., Roth, B., Klakow, D., 2012. Web-based relation extraction for the food domain. In: NLDB. pp. 222–227.

[37] Xie, H., Yu, L., Li, Q., 2010. A hybrid semantic item model for recipe search by example. In: IEEE International Symposium on Multimedia.

[38] Yamakata, Y., Imahori, S., Sugiyama, Y., Mori, S., Tanaka, K., 2013. Feature extraction and summarization of recipes using flow graph. In: Social Informatics.