What You Use, Not What You Do: Automatic Classification of Recipes

Hanna Kicherer[1], Marcel Dittrich[2], Lukas Grebe[2], Christian Scheible[1], Roman Klinger[1]

[1] Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
[2] Chefkoch GmbH, Rheinwerk 3, Joseph-Schumpeter-Allee 33, 53227 Bonn, Germany
{roman.klinger,hanna.kicherer,christian.scheible}@ims.uni-stuttgart.de
{marcel.dittrich,lukas.grebe}@chefkoch.de

Abstract. Social media data is notoriously noisy and unclean. Recipe collections built by users are no exception, particularly when it comes to cataloging them. However, 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 on the automatic classification of recipes on the German cooking website Chefkoch. 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 (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. 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.

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

1 Introduction

In 2012, 63.7 % of Germans used the Internet as source of inspiration for cooking [1]. One popular cooking website is *Chefkoch*[3], where every user can contribute to a common 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

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

The final publication is available at Springer via https://doi.org/10.1007/978-3-319-59569-6_22

Potato soup

Ingredients:

- 800g potatoes
- 1l vegetable stock
- salt & pepper

Working time: 20min / **difficulty:** simple / **calories:** not specified

Cooking instructions:
Peel the potatoes and cut them into pieces. Cook the potatoes in the vegetable stock until they are soft. Stir the soup with a wooden spoon, mashing the potatoes. Season the soup with salt and pepper.

Fig. 1. Example recipe

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 would benefit from automatic support in choosing appropriate categories for a recipe.

In this paper, we estimate a statistical model of category assignments based on recipes in the Chefkoch 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 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 specific 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.

2 Related Work

Recipes have been the subject of several previous studies. We focus on text-oriented research here, a counter-example is classification on image data [2]. Most related to this paper is prior work on recipe classification on the Japanese recipe platform *Rakuten* (http://recipe.rakuten.co.jp) by Chung et al. [3]. They estimated the relatedness of an ingredient to a recipe category using frequency measures. In experiments on *Allrecipes* (http://allrecipes.com), Kim et al. [4] 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 [5]. Similarly to our work, they ranked features for each cuisine by pointwise mutual information. Naik et al. [6] 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. [7] predicted wine features from reviews.

Next to the automatic analysis of recipe data, previous work attempted to build formal representations of recipes as ontologies. Xie et al. [8] state that such domain knowledge is a prerequisite to model the semantics of a recipe correctly. The *cooking ontology* [9] 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* [10,11]. 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.

To automatically extract relationships between ingredients, Gaillard et al. [12] 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. [13] extend this approach by using a food ontology. Sidochi et al. [14] extract substitutes in recipes from *Ajinomoto* (http://park.ajinomoto.co.jp) by taking into account process information. Teng et al. [15] 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 et al. [16,17] 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 [18] split textual descriptions from *Allrecipes* into *ingredient*, *tool*, *action*, *intermediate product*, *amount*, *unit* and *other* and detect relations between these classes.

Wang et al. [19] 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 [20, 21] and predicate-argument structures [22–24]. Based on such graph structures, the similarity as well as specific characteristics of recipes can be calculated [25].

Few works exists on German recipe data. Wiegand et al. [26, 27] analyze on *Chefkoch* 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. [28] applied distant supervision for the estimation of relation extraction models.

Features in our multi-label classification approach are inspired by these previous approaches. We discuss these in the following.

3 Feature Sets for Recipe Classification

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 [29] which outperformed decision trees [30], in all configurations using more than one feature. 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 [31] 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). 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".

As the only feature from the meta data, we add the PREPARATION TIME (PT) as stacked bins (in the example, *Preparation>5min* and *Preparation>15min* hold). We sum preparation time, cooking or baking time, and resting time.

4 Results

4.1 Experimental Setup

We use a database dump of 263,854 Chefkoch 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 are structured hierarchically with up to four levels. Recipes are associated with leaf categories and then automatically belong to all parent categories. This leads to a total of 182 categories, out of which 162 are leaves. 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.

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 %). This leads to 87 categories we work with[4]. Note that with this step, we do not omit any recipes due to the multi-label classification setting. The comparison of feature sets and their impact is performed on a training set of 5,000 instances. The best model configuration is then trained on 50,000 and tested on the same test set for further analyses.

4.2 Classification Results

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. The left column shows the individual contributions of each feature set. 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*.

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

Table 1. Precision, recall and F1 measures in percent for recipe classification with 5000 training instances and different feature combinations. The left table shows feature classes in isolation. The right table shows combinations of feature classes.

Feature combination	Features	P	R	F_1
WORDS (Baseline)	19,942	63	36	46
NOUNS	11,176	58	37	45
VERBS	3,580	46	25	32
INGREDIENTS	1,448	58	34	43
INGR. CLASSES (IC)	61	23	19	21
INGR. RANKS (IR)	1,302	42	21	28
HIGH. A. INGR. (HAI)	633	14	24	18
UNIT TYPE (UT)	1,548	24	30	27
INGR. NUMBER (IN)	32	10	00	00
PREP. TIME (PT)	9	03	02	05
CONTEXT WORDS (CW)	194,042	41	25	31
CONTEXT NOUNS (CN)	71,311	39	24	30
CONTEXT VERBS (CV)	41,269	29	24	26

Feature combination	Features	P	R	F_1
INGR & IC	1,509	60	34	44
INGR & IR	2,750	58	35	44
INGR & UT	2,996	56	35	43
INGR & HAI	2,081	58	35	44
INGR & WORDS	21,390	64	39	48
INGR & NOUNS	12,624	61	40	48
INGR & VERBS	5,028	58	38	46
INGR & CW	195,490	63	27	38
INGR & CN	72,759	62	29	39
INGR & CV	42,717	60	29	39
INGR & IC & IR	2,813	58	36	44
INGR & IR & HAI	3,391	58	36	45
INGR & IC & IR & HAI	3,452	58	37	45

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. 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.

The right column of Table 1 shows combinations of feature sets. 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 %), a recall (49 %), and F_1 (57 %).

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 model performs best and worst, respectively, in Table 2. For this comparison, we make use of the results using the training set

Table 2. The 10 best (left) and 10 worst (right) categories with INGREDIENTS and NOUNS feature combination.

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

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" 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 [32]. Figure 4.3 shows plots of the resulting spaces for six different root categories. Each point represents a recipe of a specific leaf category.

Table 3. Macro evaluation scores for feature combination *Ingredients*, *Ingredient Classes*, *Ingredient Ranks* and *Highest Amount Ingredient* over all categories of a superclass of the hierarchy.

Category Class	Precision	Recall	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

"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, *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[5]. 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 [7]. 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).

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

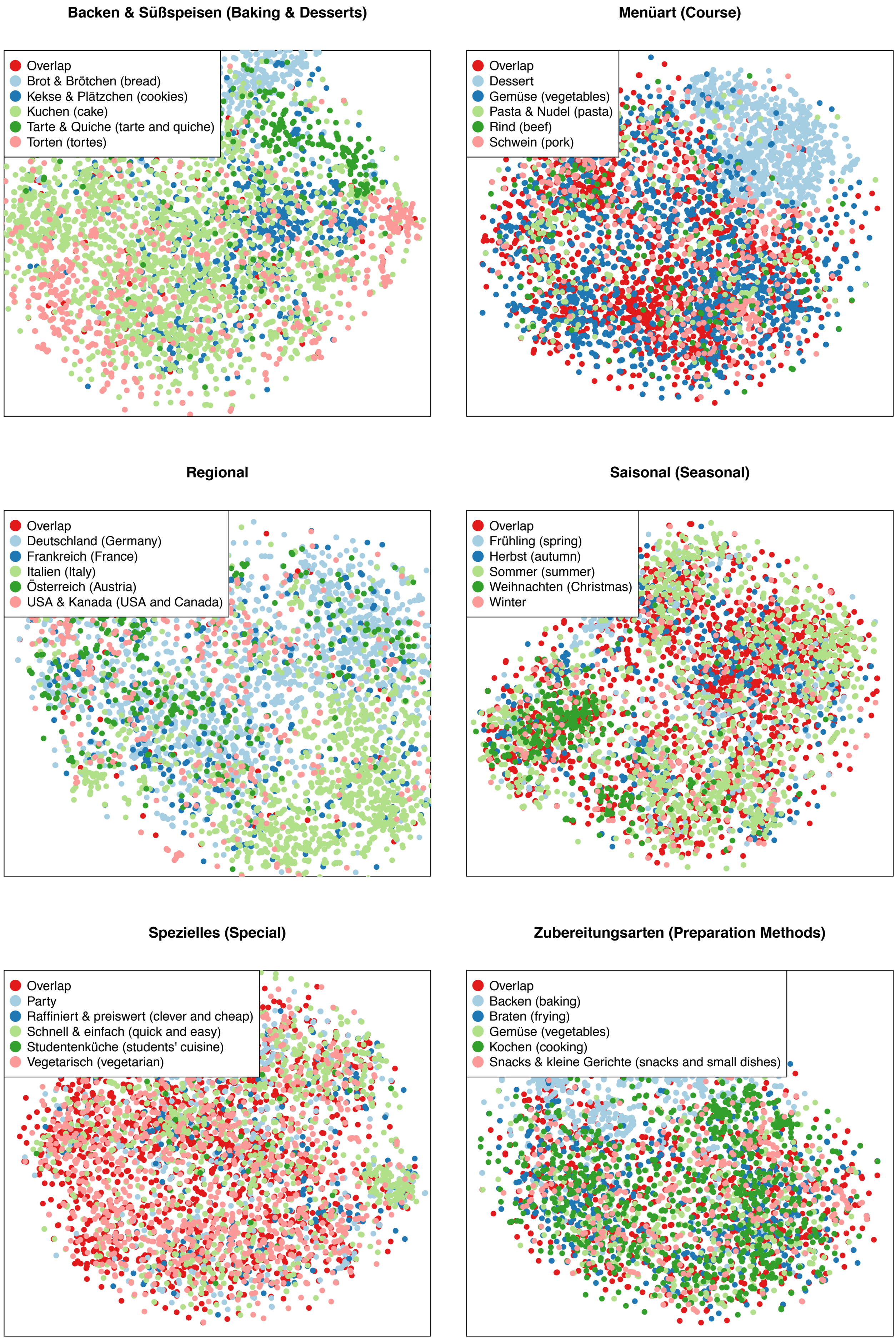

Fig. 2. Visualization of categories based on INGREDIENTS with t-SNE.

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 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.

5 Discussion, Conclusion & Future Work

In this paper, we have shown that logistic regression can classify recipes on the Chefkoch database with up to 57 % F_1. Feature analysis revealed that ingredients alone are nearly as good an indicator as the recipe description. Information from both sources complement 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. 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.

References

1. Media, T.F.: 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/ (2012) (in German).
2. Cadene, R.: Deep learning and image classification on a medium dataset of cooking recipes. Online: http://remicadene.com/uploads/summer-internship-2015.pdf (2015)
3. Chung, Y.: Finding food entity relationships using user-generated data in recipe service. In: CIKM. (2012)
4. Kim, S.D., Lee, Y.J., Kang, S.H., Cho, H.G., Yoon, S.M.: Constructing cookery network based on ingredient entropy measure. Indian Journal of Science and Technology **8**(23) (2015)
5. Ghewari, R., Raiyani, S.: Predicting cuisine from ingredients. Online: https://cseweb.ucsd.edu/~jmcauley/cse255/reports/fa15/029.pdf (2015)
6. Naik, J., Polamreddi, V.: Cuisine classification and recipe generation. Online: http://cs229.stanford.edu/proj2015/233_report.pdf (2015)
7. Hendrickx, I., Lefever, E., Croijmans, I., Majid, A., van den Bosch, A.: 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), Berlin, Germany, Association for Computational Linguistics (August 2016) 306–312
8. Xie, H., Yu, L., Li, Q.: A hybrid semantic item model for recipe search by example. In: IEEE International Symposium on Multimedia. (2010)
9. Nanba, H., Doi, Y., Tsujita, M., Takezawa, T., Sumiya, K.: Construction of a cooking ontology from cooking recipes and patents. In: UbiComp Adjunct. (2014)
10. Ribeiro, R., Batista, F., Pardal, J.P., Mamede, N.J., Pinto, H.S.: Cooking an ontology. In: AIMSA. (2006) 213–221
11. Cordier, A., Lieber, J., Molli, P., Nauer, E., Skaf-Molli, H., Toussaint, Y.: WIKITAAABLE: A semantic wiki as a blackboard for a textual case-based reasoning system. In: SemWiki 2009, ESWC. (2009)

12. Gaillard, E., Nauer, E., Lefevre, M., Cordier, A.: Extracting generic cooking adaptation knowledge for the TAAABLE case-based reasoning system. In: Cooking with computers workshop, ECAI. (2012)
13. Mota, S.G., Agudo, B.D.: ACook: Recipe adaptation using ontologies, case-based reasoning systems and knowledge discovery. In: Cooking with Computers workshop, ECAI. (2012)
14. Shidochi, Y., Takahashi, T., Ide, I., Murase, H.: Finding replaceable materials in cooking recipe texts considering characteristic cooking actions. In: Workshop on Multimedia for Cooking and Eating Activities. CEA (2009)
15. Teng, C.Y., Lin, Y.R., Adamic, L.A.: Recipe recommendation using ingredient networks. In: WebSci. (2012)
16. Greene, E.: 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/ (2015) Accessed: 2016-06-20.
17. Greene, E., McKaig, A.: 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/ (2016) Accessed: 2016-01-31.
18. Jonsson, E.: Semantic word classification and temporal dependency detection on cooking recipes (2015) Thesis, Linköpings universitet.
19. Wang, L., Li, Q.: A personalized recipe database system with user-centered adaptation and tutoring support. In: SIGMOD2007 Ph.D. Workshop on Innovative Database Research. (2007)
20. Wang, L., Li, Q., Li, N., Dong, G., Yang, Y.: Substructure similarity measurement in chinese recipes. In: WWW. (2008)
21. Maeta, H., Sasada, T., Mori, S.: A framework for recipe text interpretation. In: UbiComp Adjunct. (2014)
22. Kiddon, C., Ponnuraj, G.T., Zettlemoyer, L., Choi, Y.: Mise en place: Unsupervised interpretation of instructional recipes. In: EMNLP. (2015)
23. Mori, S., Sasada, T., Yamakata, Y., Yoshino, K.: A machine learning approach to recipe text processing. In: Cooking with Computers workshop, ECAI. (2012)
24. Mori, S., Maeta, H., Yamakata, Y., Sasada, T.: Flow graph corpus from recipe texts. In: LREC. (2014)
25. Yamakata, Y., Imahori, S., Sugiyama, Y., Mori, S., Tanaka, K.: Feature extraction and summarization of recipes using flow graph. In: Social Informatics. (2013)
26. Wiegand, M., Roth, B., Klakow, D.: Web-based relation extraction for the food domain. In: NLDB. (2012) 222–227
27. Wiegand, M., Roth, B., Klakow, D.: Data-driven knowledge extraction for the food domain. In: KONVENS. (2012) 21–29
28. Reiplinger, M., Wiegand, M., Klakow, D.: Relation extraction for the food domain without labeled training data–is distant supervision the best solution? In: NLP. (2014) 345–357
29. Cox, D.R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological) (1958) 215–242
30. Quinlan, J.R.: C4.5: Programming for machine learning. Morgan Kauffmann (1993) 38
31. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL Demo. (2014)
32. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9** (2008) 1–48