Do Good Recipes Need Butter? Predicting User Ratings of Online Recipes

Ning Yu
University of Kentucky
Lexington, KY, USA
ning.yu@uky.edu

Desislava Zhekova
CIS, University of Munich
München, Germany
desi@cis.uni-muenchen.de

Can Liu
Indiana University
Bloomington, IN, USA
liucan@indiana.edu

Sandra Kübler
Indiana University
Bloomington, IN, USA
skuebler@indiana.edu

Abstract

In this work, we investigated the automatic prediction of user ratings for recipes. Information including the ingredients, the instructions, and the reviews from Epicurious were fed into a machine learner, a multi-class support vector machine, to examine how reliable they are when predicting recipe ratings. Our results show that information from the reviews results in the most reliable predictions: we reached an accuracy of 62%. The problem is difficult, partly because of the skewing of the ratings: most recipes are rated with 3 or 4 out of 4 forks.

1 Introduction

Exchanging recipes over the internet has become popular over the last decade. There are numerous sites that allow us to upload our own recipes, to search for and to download others, as well as to rate and review them. Such sites aggregate invaluable information, not only in terms of providing recipes, but also in providing information about cultural preferences with regard to food. For example, the site Epicurious[1] has more than 1 100 recipes for *chili*, but only 24 of those are *low sodium*. The site also presents 174 recipes for *muffins*, out of which 21 do not contain any dairy products. Such facts give us a first indication that Americans may eat more salty chilis than low-sodium ones, if we assume that the users of the site have a similar distribution as the American population. Additionally, a closer look reveals that out of all muffin recipes, only 11.5% have the highest rating of four forks[2], while among the non-dairy muffins, the percentage of four-fork ratings is 23.8%. From this, we could conclude that Americans like non-dairy muffins better than the ones containing dairy (keeping in mind the small size of the sample).

In this paper, we investigate whether we can predict user ratings for individual recipes, given the ingredients, the instructions, the reviews, or a combination of these features. If these experiments are successful, we can draw conclusions 1) about which ingredients good recipes include, 2) whether quantities of ingredients or 3) specific steps in the instructions have an influence on the ratings, and 4) whether we can detect reliable clues in the user reviews that allow us to deduce how much users like a recipe.

Questions 1) and 2) exploit intuitions and knowledge such as the inference that since fat is a flavor carrier, larger amounts of fat would generally increase taste and subsequently ratings. However, this does not carry across categories: having a larger amount of sugar in a cookie recipe may increase ratings while having the larger amount of sugar in a pot roast may have the opposite effect.

Question 3) is based on the assumption that easier recipes may be rated higher than more difficult ones. Difficulty may concern the number of steps in a recipe or involving certain techniques, such as melting chocolate in a double boiler rather than in a normal pot or measuring the temperature of melted sugar with a candy thermometer.

Question 4) is an extension of *sentiment analysis*, which investigates methods to classify users' attitudes towards a product (or other entities). In our case, the product is the recipe in question. However, the situation is complicated by the fact that we do not classify individual reviews or even sentences into positive or negative sentiment, but rather attempt to classify a recipe based on all reviews, which may be contradictory. The reviews for one randomly selected muffin recipe, for example, show ratings that range from 1 fork to 4.

The remainder of the paper is structured as follows: We will present related work in section 2. Then, we will discuss the data set and the questions that we will tackle in more detail in section 3 and the experimental setup in section 4. In section 5, we will present and discuss the results, and in section 6, we will conclude our findings and discuss future work.

2 Related Work

The history of applying computer technology to support cooking activities goes back to 1986 when CHEF, a case-based machine planner was created to generate new cooking plans from experience (i.e., old plans) [Hammond, 1986]. Later, various interactive cooking support systems including CounterActive [Ju *et al.*, 2001], eyeCook [Bradbury *et al.*, 2003], and Smart Kitchen [Hashimoto and Mori, 2008] were proposed. Recent developments focus on health driven cooking support systems [Karikome and Fujii, 2010; Kamieth *et al.*, 2011; Wagner *et al.*, 2011]. All these smart kitchen systems require heavy domain knowledge, which nowadays can

[1] http://www.epicurious.com

[2] Ratings in Epicurious are represented with 0 to 4 forks, including the intermediate values 1.5, 2.5 and 3.5.

be generated via crowd sourcing, e.g., online recipe sharing. In the last decade, researchers have studied online recipes for making recommendations to meet personal preferences [Ueda *et al.*, 2011] or health concerns [Freyne and Berkovsky, 2010; Mino and Kobayashi, 2009]. Such studies often focus on a specific cuisine or type of food, e.g., cookies, and need to build user models [Sobecki *et al.*, 2006]. Different from these recommendation systems, our work is content-driven and is interested in understanding the overall recipe preferences from all users. Nevertheless, recipe features used in such systems are applicable to our work.

Ingredients are the most commonly studied features for recipe recommendations. [Freyne and Berkovsky, 2010] considered all ingredients to be equally weighted within a recipe and aggregates the ratings of ingredients to predict the recipe rating. They found that this simple break down and construction approach worked better than a user-driven approach that takes user rating into consideration. While ingredient ratings are often not available, their experience shows the value of ingredients for predicting user ratings. Till today, most studies have treated ingredients equally [Forbes and Zhu, 2011; Teng *et al.*, 2012] and used them as binary features. [Zhang *et al.*, 2008] manually grouped ingredients into three levels of importance and ingredients that the researchers considered most important have the highest weight. To the best of our knowledge, this work is the first to use the actual quantities of ingredients within a recipe as feature values.

Cooking methods are another type of features that has been used for recipe recommendation. In the past, these features were either created manually [van Pinxteren *et al.*, 2011] or mined from existing knowledge bases [Teng *et al.*, 2012], which may be due to the noise in instruction text. Our work proposes a simple linguistic approach to directly extracting cooking methods and other features from instructions.

Although user reviews are the basis for recipes ratings, they have been used to identify refinements (e.g., reducing sugar in a recipe) [Druck and Pang, 2012; Teng *et al.*, 2012], but not to rate recipes. Our work uses sentiment analysis on recipe reviews and compares review features with content-based features in terms of their effectiveness for predicting ratings.

More recipe features can affect user ratings, and they are often used together. [van Pinxteren *et al.*, 2011] manually developed 55 features (e.g., soup, French cuisine), which covered 13 recipe characteristics (e.g., meal type, preparation time, preparation technique) for pasta. [Teng *et al.*, 2012] applied ingredient features, ingredient co-occurrence and substitution network features, and primary recipe features such as cooking methods, preparation time, and nutrition information to predict which recipe has the higher rating between a pair of similar recipes. They found that ingredient network features and nutrition features were most effective in their machine learning experiments, with accuracies of 75% and 78.6% respectively. When working with various features, [Freyne and Berkovsky, 2010; van Pinxteren *et al.*, 2011] used the weighted average to determine feature preferences, and [Forbes and Zhu, 2011] used a more sophisticated matrix factorization approach.

3 How to Predict User Ratings

In this section, we will first describe the data sets, and then the research questions that we are investigating in this paper.

3.1 Data Set

We developed a web crawler to scrape recipes with their information from the Epicurious site. Overall, we extracted more than 28 000 recipes. Then, we excluded all recipes for which we did not have the information to extract the features interesting to us (e.g., there are recipes that received no reviews and hence no rating information), which reduced the data set considerably, to 10 146 recipes. To avoid overwhelming the classifier by highly reviewed recipes, we did not include more than 10 reviews per recipe. For all recipes with more than 10 reviews, we took a random sample of 10 reviews out of the total number of reviews. We also only used full ratings, i.e., 3.5 forks are rounded down to the 3-fork class, based on the observation that users are generous when rating recipes. After this step, the data has the following distribution of classes/forks across all examples:

- 1 fork :112 examples
- 2 forks: 795 examples
- 3 forks: 5 670 examples
- 4 forks: 3 569 examples

This shows that we are dealing with a well known problem in machine learning: our data is heavily skewed towards higher rankings.

For every recipe, we extracted and prepared features in the following four groups:

- The overall rating (our gold standard classification) in terms of forks
- Metadata
  1. whether there is a picture
  2. whether there is wine pairing suggestion
  3. whether there is a quick meal label
  4. whether there is a healthy meal label
  5. whether it appears in the Epicurious menu
  6. the type of the recipe (e.g., Alcoholic, Salad)
  7. the type of cuisine (e.g., African, Italian)
  8. the recipe's dietary condition (e.g., Healthy, Vegan)
  9. number of available metadata items
  10. number of cuisine types associated with the recipe
  11. number of dietary conditions covered by the recipe
  12. the gender of the contributor
- Ingredients
  1. the ingredients used in the recipe that occur in ≥ 4 recipes
  2. the quantities of the ingredients
  3. the number of ingredients
  4. the main ingredients of the recipe (a separate group provided by Epicurious for each recipe)
- Instructions

1. the cooking steps/methods (e.g., chop, boil)
2. number of major steps (e.g., prepare the dough)
3. number of cooking steps

- Reviews
  1. a list of indicative uni-, bi-, and trigrams
  2. a list of best TF/IDF indicative uni-, bi-, and trigrams

Most features are represented by a binary value that indicates the presence or absence of each of the categories possible for this feature. For example, for each of the existing types of the recipe (e.g., Alcoholic, Salad, Sauce) we include a separate feature/value pair. Quantitative features, such as the number of ingredients, are represented with a scaled-down value ranging from 0 to 1 by dividing each feature value by the maximum number of features.

From the list of ingredients of a recipe, we extracted the ingredients and reduced them to their main nouns, noun compounds, or noun phrases. For example, the ingredient "3 tablespoons unsalted butter" is reduced to "butter", "1 Granny Smith apple, peeled, cored, and finely chopped (1 1/2 cups)" is reduced to the compound "Granny Smith apple". This was carried out based on a part-of-speech (POS) analysis of the ingredients. POS tagging is a well researched technique from Natural Language Processing that assigns a word class label to every word. Thus, both "Granny" and "Smith" would be assigned the label NNP, for proper noun, while apple is assigned NN, for common noun. The word "unsalted" would be assigned the label JJ for adjective. Excluding text within paraphrases, we kept all nouns (common and proper) and noun phrases that consist of one adjective modifier and one single noun head (e.g., "black pepper"). To avoid sparse features, we also excluded adjectives if the adjective falls into a manually created stopword list, including such words as "fresh". In order to use ingredient quantities as the values for ingredient features, we converted all the volume measures to tablespoon (tbsp). Thus, if one recipe needs "1 cup of sugar" while the other recipe needs "2 tablespoons of sugar", both instances will be assigned the feature "sugar_tbsp" and the values will be "16" and "2" respectively. We identified around 9 000 ingredient-measurement combinations and kept the ones that occur in at least 4 recipes to ensure that the features generalize well. Thus, we acquire a collection of 2 406 ingredient-measure features for our experiments.

For POS tagging, we used the Stanford POS tagger [Toutanova *et al.*, 2003][3], with the best performing model. However, since this POS tagger model was trained on the Wall Street Journal section of the Penn Treebank [Marcus *et al.*, 1993], we were using it out of domain, thus increasing the error rate. As a consequence, many verbs from the instructions are mistagged as nouns since they often occur as imperatives at the beginning of the sentence, and they occur more often as nouns in the Penn Treebank. For example, in the instruction "Drain over a rack.", the first word is mistagged as a noun instead of as a verb in the base form (VB).

From the instructions, we extracted all verb clusters and normalize them to the last verb. I.e., we extracted "set" from the sentence "You might want to set a timer." All verbs in past tense (e.g., "boiled", "chopped") and verbs that occur in less than 3 recipes were removed. Mistagged verbs were also manually removed. At the end, we had a total of 340 actions (e.g., "add", "roast", "stir").

From the reviews, we extracted n-grams (sequences of n words) that have discriminating capability between ratings. We consider uni-, bi-, and trigrams, based on words and on POS tags. E.g., "absolutely amazing", "very disappointing" etc. are covered by the bigram "RB JJ". "everyone loved" is based on "NN VBD". The most discriminating n-grams are the ones that achieved the highest term frequency/inverse document frequency (TF/IDF) scores. Since TF/IDF reflects how import a word is to a document, the higher the TF/IDF of a word, the more discriminating power it has. In calculating the TF/IDF values, we considered every review a document. TF/IDF values ranged from 6 to approximately 20 000. We then restricted the n-grams to the set of those that have a TF/IDF of at least 5 000.

3.2 Research Questions

In this work, we used a machine learning approach to investigate four main questions: 1) Can we reliably predict user ratings from the set of ingredients and their quantities? If this experiment is successful, we can conclude that there are specific ingredients or combinations of them that have an influence on how much users like the recipe. 2) Can we reliably predict user ratings from the instructions? In other words, are there certain instructions or combinations of them that influence user ratings, potentially because they are more complicated or difficult to execute? 3) Can we reliably predict user ratings from the set of reviews for the recipe? We assume that there is a close connection between the two, but the connection may be obscured by the fact that more people submitted ratings but only a small subset of these also provided reviews, which can vary considerably per recipe. Note that we approached all these questions indirectly, not by looking at statistical information directly but rather by employing classification. This has the advantage that we can use a successful classifier to predict the success for a novel recipe. This leads us to our last question: 4) Which types of information do we need in order to obtain a reliable rating classifier?

In order to investigate the first question concerning the ingredients and their quantities, we conducted the following experiments:

- INGREDIENT: Here, we used the ingredients and their quantities as well as the number of ingredients, i.e. features 1 through 3 from the Ingredients features in section 3.1. Ingredients are in the form of the ingredient plus the measurement for the quantity, e.g. sugar_tbsp; the quantities serve as values for their corresponding ingredient.
- INGR:NOQUANT: Here, we used the ingredients as presence features, i.e., they take binary values (present/not present) plus the number of ingredients.
- INGR:ADDMAIN: Here we used all features from INGREDIENT and also add the main ingredients, i.e., features 4 from the Ingredients features in section 3.1.

[3] http://nlp.stanford.edu/software/tagger.shtml

In order to investigate the second question, we conducted the following experiments:

- INSTRUCT: We used all features from the Instructions features in section 3.1.

In order to investigate question 3, we compare the following experiments:

- REVIEW:ALL: Here we used the full n-grams (without filter) from the Reviews group in section 3.1 (i.e. features 1-3).
- REVIEW:FILTER: In this setting, we only used the list of best TF/IDF indicative n-grams. (i.e. features 4-6).

The final question is concerned with finding the best classifier. We decided to experiment with the full set of features and with classifiers in which one of the feature sets is excluded:

- ALLFS: This setting includes all collected features.
- ALL:NOMETA: In this setting, all metadata features were removed.
- ALL:NOINGREDIENT: All features that represent the ingredients were excluded.
- ALL:NOINSTRUCT: All features that represent the instructions were excluded.
- ALL:NOREVIEW: In this setting, the set of all review features was excluded.

4 Experimental Setup and Evaluation

For classification, we used Support Vector Machines (SVMs) in the implementation of SVM^{light}. In particular, we employed the multi-class version SVM^{multiclass} [Crammer and Singer, 2002][4]. SVM is a machine learning classification approach which uses a function (called a kernel) to map example instances onto a linearly separable space.

All experiments reported here were performed with default parameter settings. In preliminary experiments, we observed that a higher value of the c parameter (i.e., the trade-off between training errors and margin) yields better results than its default value. However, it also resulted in unmanageable training time due to sheer volume of the data. In order to avoid benign data splitting, we used 10-fold cross-validation and report all results averaged over the 10 runs. As baseline, we used the setting in which the class with the highest number of recipes (the three-fork rating) is used to label all instances. We report accuracy for each evaluation setting as well as precision (P), recall (R), and F-scores (F) per class within the distinct settings. After feature extraction and composition of the feature vectors, we achieved a collection of 5 241 feature/value pairs that represent the information listed in four separate groups in section 3.1.

5 Experiments

5.1 How Important Are Ingredients and Their Quantities?

In order to investigate how important ingredients and their quantities are for user ratings, we carried out a classification experiment, in which we used the features from the ingredients (see section 3.1) and their quantities (INGREDIENT), and another one without quantities (INGR:NOQUANT). The baseline is shown first, and the results for the experiments with ingredients are shown in the second part of table 1.

The results show that we have a very competitive baseline: Just by choosing the class 3-forks for every recipe, we reached an accuracy of 56%. This is due to the heavy skewing in the distribution of the user ratings. When we used ingredient information, we reached a considerably lower accuracy of 50%. Almost none of the recipes were correctly classified as disliked recipes (1- and 2-forks). By leaving out the quantities (INGR:NOQUANT), we reached a marginally higher accuracy of 51%, gaining mostly in recall for the 3-fork rating. From these results, we can conclude that neither the list of ingredients nor the combination of these ingredients and their quantities are good predictors of ratings, and thus, it is unlikely that specific ingredients are directly related to the rating. Additionally, to answer the question in the title: While butter is a flavor carrier, it is the most frequent ingredient across all four ratings. Thus, butter is not a good discriminative feature.

In the next setting (INGR:ADDMAIN), we added the list of main ingredients as listed on the recipe page. This results in a higher accuracy of 55%, which is close to the baseline. However, it is interesting to see that we thus predicted a very similar distribution to the baseline: All recipes are grouped in the 3- and 4-fork categories, with a preference for the majority class of 3 forks. The results show that for the 3-forks rating, we still have a high recall (87%). This means, that the additional main ingredients help increase accuracy but at the expense of the lower ratings, meaning that the main ingredients are typical for 3-forked recipes, but not for poorly or highest rated recipes.

Sampling of ratings: Since we have a heavily skewed distribution of possible classes, we decided to apply up- and down-sampling to reach more balanced ratios of training instances. In downsampling, we restricted the number of training instances for the classes with more examples to match the number of instances in the least represented class (1 fork). In upsampling, the instances from lower classes in the training set are duplicated until they reach a more balanced ratio. For both methods, we tested different ratios. However, all these experiments resulted in a decrease in accuracy, so we used the full training set for the remaining experiments.

5.2 Can Instructions Predict Ratings?

For this question, we looked at the INSTRUCT setting in the third part of table 1. The results show that this setting is again close to the baseline and the INGR:ADDMAIN setting: We reached an accuracy of 56%, and all ratings are in the 3- and 4-fork category. From this, we can conclude that there is no direct interaction between the instructions and the distribution of ratings.

5.3 Can Reviews Predict Ratings?

For this question, we looked at the REVIEW:ALL and the REVIEW:FILTER settings in the fourth part of table 1. REVIEW:ALL uses the full list of n-grams from the reviews

[4] http://svmlight.joachims.org/svm_multiclass.html

		1-fork			2-fork			3-fork			4-fork			Acc.
		P	R	F	P	R	F	P	R	F	P	R	F	
1	BASELINE	0	0	0	0	0	0	0.56	1.00	**0.72**	0	0	0	0.56
2	INGREDIENT	0.02	0.05	0.03	0.10	0.03	0.05	0.57	0.71	0.63	0.39	0.28	0.32	0.50
	INGR:NOQUANT	0.03	0.05	0.03	0.14	0.02	0.04	0.57	0.74	0.64	0.40	0.28	0.33	0.51
	INGR:ADDMAIN	0	0	0	0.05	0	0	0.57	0.87	0.69	0.43	0.18	0.26	0.55
3	INSTRUCT	0	0	0	0	0	0	0.56	0.97	0.71	0.47	0.05	0.10	0.56
4	REVIEW:ALL	0.17	0.15	0.15	0.37	0.08	0.13	0.64	0.77	0.70	0.60	0.51	0.55	**0.62**
	REVIEW:FILTER	0.14	0.21	0.17	0.26	0.25	0.25	0.64	0.68	0.66	0.58	0.52	0.55	0.59
5	ALLFS	0.14	0.17	0.15	0.27	0.30	0.28	0.66	0.66	0.66	0.58	0.56	**0.57**	0.59
	ALL:NOMETA	0.18	0.21	0.19	0.28	0.29	0.29	0.66	0.68	0.67	0.59	0.55	**0.57**	0.60
	ALL:NOINGREDIENT	0.18	0.24	**0.21**	0.33	0.32	**0.32**	0.65	0.67	0.66	0.58	0.55	0.57	0.59
	ALL:NOINSTRUCT	0.14	0.18	0.15	0.27	0.30	0.28	0.66	0.67	0.66	0.58	0.55	**0.57**	0.59
	ALL:NOREVIEW	0	0	0	0.10	0	0	0.59	0.84	0.69	0.50	0.29	0.37	0.57

Table 1: The experimental results in all evaluation settings; P=precision, R=recall, F=F-score.

while for REVIEW:FILTER, we use the TF/IDF filtered n-grams. The results show that the REVIEW:ALL setting reaches the highest results with an accuracy of 62% and the best distribution over all ratings. When we filtered the n-grams, we reached higher recall and F-scores for the disliked categories: The F-score increases from 15% to 17% for the 1-fork rating and from 13% to 25% for 2 forks. However, this is offset by a loss in recall in the 3-fork rating, leading to a decrease in overall accuracy. This means that the filtered n-gram features lose their ability to distinguish 3-fork ratings to a certain extent. In the future, we will experiment with other feature selection methods.

5.4 Which Types of Information are Necessary?

For this question, we investigated whether we can obtain better ratings when we integrate information from different types of information about the recipe. Here, we looked at an experiment that used all available features, ALLFS, then we reduced the feature set by each group of features. The results are shown in the last part of table 1. Surprisingly, using all features results in an accuracy of 59%, which is higher than the baseline, but it is also considerably lower that the accuracy reached by using only review features.

The experiments where we removed groups of features show that only one group of features is detrimental to the overall results: the metadata features. The removal of this group results in a minimal improvement of accuracy over all features. This is not surprising since Epicurious attempts to promote all recipes. However, we will need a closer look to see whether individual features from this group may be useful. As expected, the removal of review features leads to the lowest accuracy (57%) in this group of experiments.

Overall, we can conclude that reviews provide the most reliable information for predicting ratings for recipes. However, if we aggregate all reviews into one decision, as we have done here, this means that the lower ratings are dispreferred by the classifier. From these results, we conclude that we need more information and more reliable features for the classifier.

6 Conclusion and Future Work

In this work, we investigated the prediction of recipes from the Epicurious site. We extracted information about the following categories: metadata, ingredients, instructions, and reviews. Our experiment showed that using only information from reviews results in the highest accuracy. However, this is a difficult problem because of the skewed distribution of the ratings. When using all n-grams from the reviews, we obtain an accuracy of 62%.

This work is still at the beginning. We are planning to improve the approach in three different areas: First, we will investigate our current features in more detail. Some features can be cleaned up further, and the groups of features we investigated may have been too coarse. I.e., looking at individual features may help us identify predictive features.

Second, we need to improve the approach to sentiment analysis with respect to the reviews. We are planning to integrate a separate classifier for individual reviews, so that we can include information about the distribution of the reviews into the recipe classifier. Here, we need to investigate which features are important for both classifiers in order to gain optimal results. We are also planning to explore the use of both general and corpus-based sentiment lexicons, e.g., Wilson's subjectivity terms [Wilson *et al.*, 2003] and a frequency lexicon containing patterns that could capture intentionally misspelled words (e.g., "luv," "hizzarious") [Yang *et al.*, 2008].

Third, we are planning to improve the Natural Language component and perform domain adaptation along the lines of [Kübler and Baucom, 2011]. We are also planning to integrate a dependency parser, MaltParser [Nivre *et al.*, 2007] into the system. The dependency parses provide syntactic information in form of dependencies (directed arcs) between pairs of words. The dependencies also carry grammatical information, which allows us to detect "who does what to whom". Additionally, they can also provide information with regard to the scope of negation. Negation has to be dealt with in sentiment analysis to ensure that "not good" is treated as a negative expression. However, more complex cases of negation such as " I don't think, the recipe is good." are often not handled correctly.

References

[Bradbury *et al.*, 2003] Jeremy S. Bradbury, Jeffrey S. Shell, and Craig B. Knowles. Hands on cooking: Towards an

attentive kitchen. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, pages 996–997, Fort Lauderdale, FL, 2003.

[Crammer and Singer, 2002] Koby Crammer and Yoram Singer. On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines. *Journal of Machine Learning Research*, 2:265–292, 2002.

[Druck and Pang, 2012] Gregory Druck and Bo Pang. Spice it up? Mining refinements to online instructions from user generated content. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 545–553, Jeju, South Korea, 2012.

[Forbes and Zhu, 2011] Peter Forbes and Mu Zhu. Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 261–264, Hong Kong, China, 2011.

[Freyne and Berkovsky, 2010] Jill Freyne and Shlomo Berkovsky. Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, pages 321–324, Santa Monica, CA, 2010.

[Hammond, 1986] Kristian J. Hammond. CHEF: A model of case-based planning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 267–271, Philadelphia, PA, 1986.

[Hashimoto and Mori, 2008] Atsushi Hashimoto and Naoyuki Mori. Smart kitchen: A user centric cooking support system. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based System*, pages 848–854, Malaga, Spain, 2008.

[Ju *et al.*, 2001] Wendy Ju, Rebecca Hurwitz, Tilke Judd, and Bonny Lee. CounterActive: An interactive cookbook for the kitchen counter. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 269–270, Seattle, WA, 2001.

[Kamieth *et al.*, 2011] Felix Kamieth, Andreas Braun, and Christian Schlehuber. Adaptive implicit interaction for healthy nutrition and food intake supervision. In *Proceedings of the 14th International Conference on Human-Computer Interaction*, pages 205–212, Orlando, FL, 2011.

[Karikome and Fujii, 2010] Shihono Karikome and Atsushi Fujii. A system for supporting dietary habits: planning menus and visualizing nutritional intake balance. In *Proceedings of the 4th International Conference on Ubiquitous Information Management and Communication*, pages 56:1–56:6, Siem Reap, Cambodia, 2010.

[Kübler and Baucom, 2011] Sandra Kübler and Eric Baucom. Fast domain adaptation for part of speech tagging for dialogues. In *Proceedings of the International Conference on Recent Advances in NLP*, Hissar, Bulgaria, 2011.

[Marcus *et al.*, 1993] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[Mino and Kobayashi, 2009] Yoko Mino and Ichiro Kobayashi. Recipe recommendation for a diet considering a user's schedule and the balance of nourishment. In *IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 3, pages 383–387, Shanghai, China, 2009.

[Nivre *et al.*, 2007] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[Sobecki *et al.*, 2006] Janusz Sobecki, Emilia Babiak, and Marta Slanina. Application of hybrid recommendation in web-based cooking assistant. In *Proceedings of the 10th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pages 797–804, Bournemouth, UK, 2006.

[Teng *et al.*, 2012] Chun-Yuen Teng, Yu-Ru Lin, and Lada Adamic. Recipe recommendation using ingredient networks. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 298–307, Evanston, IL, 2012.

[Toutanova *et al.*, 2003] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259, Edmonton, Canada, 2003.

[Ueda *et al.*, 2011] Mayumi Ueda, Mari Takahata, and Shinsuke Nakajima. User's food preference extraction for personalized cooking recipe recommendation. In *Proceedings of the Second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation*, Bonn, Germany, 2011.

[van Pinxteren *et al.*, 2011] Youri van Pinxteren, Gijs Geleijnse, and Paul Kamsteeg. Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, pages 105–114, Palo Alto, CA, 2011.

[Wagner *et al.*, 2011] Juergen Wagner, Gijs Geleijnse, and Aart van Halteren. Guidance and support for healthy food preparation in an augmented kitchen. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, pages 47–50, Palo Alto, CA, 2011.

[Wilson *et al.*, 2003] T. Wilson, D. R. Pierce, and J. Wiebe. Identifying opinionated sentences. In *Proceedings of HLT-NAACL*, pages 33–34, Edmonton, Canada, 2003.

[Yang *et al.*, 2008] Kiduk Yang, Ning Yu, and Hui Zhang. WIDIT in TREC2007 blog track: Combining lexicon-based methods to detect opinionated blogs. In *Proceedings of the 16th Text Retrieval Conference*, Gaithersburg, MD, 2008.

[Zhang *et al.*, 2008] Qian Zhang, Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Back to the future: knowledge light case base cookery. In *European Conference on Case-Based Reasoning*, pages 239–248, Trier, Germany, 2008.