Automated Cuisine Classification of Recipes

Steven Jorgensen
Michigan State University
220 Trowbridge Rd.
East Lansing, MI 48824
jorgen72@msu.edu

Thrilok Nagamalla
Michigan State University
220 Trowbridge Rd.
East Lansing, MI 48824
nagamall@msu.edu

Abstract

Document classification processes have become very popular in many domains as they are useful in segregating corpuses for further use. Being able to classify cooking recipes into correct cuisine type could be very useful to deal with large untagged dataset, classify a new recipe etc. In this project four different classifiers are used to classify food recipes into their cuisine types and seven features are explored to find which are most useful for recipe classification. The recipes are scraped from www.allrecipies.com and Python and R programming languages are used to implement the classifying algorithms.

1 Introduction

Recipe classification for a huge dataset can be very useful and informative Many websites host different recipes uploaded by people, but the sheer volume of recipes uploaded can be difficult to sort through if unlabeled. Users may also write their own recipes and not be sure what kind of cuisine best fits, which may cause them to not label their recipe, or worse they may misclassify it. This project will focus on choosing effective classifiers and features for these types of tasks, as well as determining which words provide the most information when classifying.

Being able to choose the best feature type and classifier that can effectively deduce cuisine type will help to automate the process. This will also give insight into each feature type strengths and weaknesses. Another application of automating classification would be to help users label uncategorized recipes in their possession. This would save the user time and effort in determining the category for themselves, as well as potentially preventing the user from accidently misclassifying the recipe.

2 Related Work

Others have done work like this. Classification of recipes into respective cuisines based on ingredients has been done by [1] using techniques like associative classification and support vector machines. In their approach, the set of ingredients from recipes were extracted and used as their training corpus. They then used features such as the number of signature ingredients in a dish and the presence of ingredients to better improve their classifier's results.

Another approach, done by [2] was to use a graph-like model and try to match substructures among the graphs to find similar recipes. In their approach, the authors used a filtering algorithm to prune out dissimilar cooking graphs from the recipe. They then took this subgraph and further simplified it by combining the actions and ingredients. From there a cooking graph index was made and the recipes were measured based on similarity.

A different type of approach, done by [3], modeled recipes using a variety of features like taste, smell, and texture. In this approach, the authors took the cooking procedures, taste, smell, and other cooking features, then created a cooking features graph for each recipe. These cooking features were divided into larger classes such as *eating fea-*

tures and *nutrition features*. The authors then took these classes and used them to create a HSI model of each recipe. The user of this model can then use other recipes as examples when querying the model for similar recipes.

In [4], the authors used the ingredients of recipes to classify recipes by cuisine type. The authors used SVM, Perceptron, and PCA as their classifiers. Once the recipes were classified, the authors took a list of cooking actions generated from WordNet and attempted to used random subsets of ingredients from a cuisine to generate new recipes of that cuisine type.

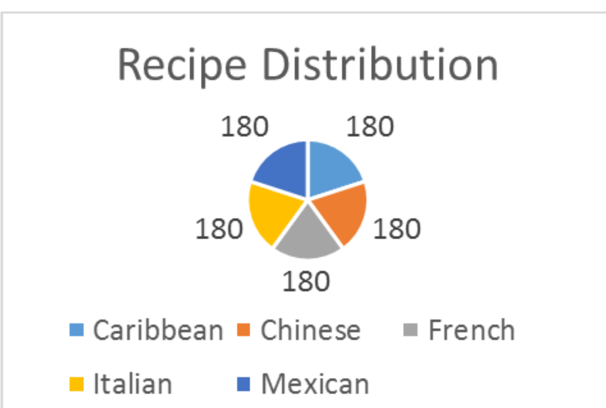
In this project, both the ingredients and cooking instructions are extracted from recipes to use in the corpus. Tested and untested features are then used to classify the recipes, and are evaluated based on their performance. The words with the most mutual information from each cuisine are also examined.

3 Methods

This project's implementation details can be found in our online repository using the following link: https://github.com/stevenjson/CuisineClassifying.

3.1 Data

Data used in this study was collected from www.allrecipes.com using the python tool Beautiful Soup 4. A total of 900 recipes were used which comprised of 180 from 5 different cuisine type; Caribbean, Chinese, French, Italian, and Mexican. From there the data was tagged using the NLTK POS tagger.



This information was used to extract seven different features from the corpus. The features studied are: Unigrams, Bigrams, Nouns, Verbs, Nouns and Verbs, Ingredients, Cooking Actions. Four different classifiers have been used to study the features effectiveness: Naïve Bayes, Recursive Partitioning, K-nearest neighbors (KNN) and Support Vector Machine (SVM). To generate the Cooking Actions feature, a list of verbs was downloaded from WordNet and added to by the authors. A recipe from the corpus looks like this:

> 3 cups bread flour 1 1/4 cups boiling water 2 tablespoons vegetable oil salt and pepper to taste 1 bunch green onions , finely chopped 2 teaspoons vegetable oil | Use a fork to mix flour and boiling water in a large bowl . Knead dough into a ball . Cover bowl with plastic wrap ; let dough rest for 30 to 60 minutes . Evenly divide dough into 16 pieces . Roll each piece into a 1/4 inch thick circle . Brush each circle with oil , season with salt and pepper , and sprinkle with about 1 teaspoon of green onions . Roll up , cigar style , and pinch open ends together to form a circle . Roll each circle flat to 1/4 inch . Heat 2 teaspoons oil in a large skillet . Fry cakes until golden brown , about 2 minutes on each side .

Six-fold cross validation was performed to obtain an informative accuracy. For this project's purposes, each fold has 750 training recipes and 150 for testing. In both the training and testing datasets the cuisine recipes are in equal proportions.

3.2 Classifiers

The package "Rpart" was used in "R" programming language to implement the Recursive partitioning algorithm. The tree was limited to maximum of 20 nodes. The decision tree learned by recursive partitioning algorithm is given in appendix A.

The tree has 15 nodes and 16 leaves. The nodes are the predictor variables: "soy," "Parmesan," "tortilla," "chili," "wok," "cumin," "Oil," "Pasta," "lime," "Butter," "cheese," "egg," "wine," "salsa," and "garlic." Fourteen of these are ingredients and one is a utensil. Even though only one utensil has been used in the tree since it is the 5th node in the tree it has a high partitioning information and hence utensils can be very important feature for classification. The word "soy" is used as the first partitioning node which split the entire data set into two: "Chinese," and "Italian". Out of 150 Chinese recipes 103 of them contained the word "soy" while none of the Italian recipes had the word. The second node "Parmesan" is very informative to split the data further into "Italian" and "Mexican", none of the remaining 145 Mexican recipes have this word while 59 of 150 Italian recipes it.

The Naïve Bayes classifier was implemented using the Scikit Learn python package. First the data was run through a word count vectorizer. Next the data was run through a term frequency – inverse document frequency vectorizer to weight the terms appropriately before classifying. The data was then run through the package's multinomial Naïve Bayes implementation. All the default settings were used for this classifier.

The SVM classifier was also implemented using the Scikit Learn python package. The data was also run through a word count vectorizer as well as a term frequency – inverse document frequency vectorizer. The data was then run through the package's SGDClassifier, which is a text based SVM implementation. The loss function was set to "log", the penalty was set to 'l2' which is the standard regularizer for linear SVM models. The smoothing constant used was 0.001 and the random seed was set to 42.

The KNN classifier was implemented in python by the authors. The classifier took the Levenshtein distance between the recipe it was trying to classify and every recipe in the training corpus. The classifier would then select the K closest recipes from the corpus. Using the selected recipes, the most frequent cuisine type would be determined and be predicted as the test recipe's cuisine type. This was by far the slowest of the classifiers used.

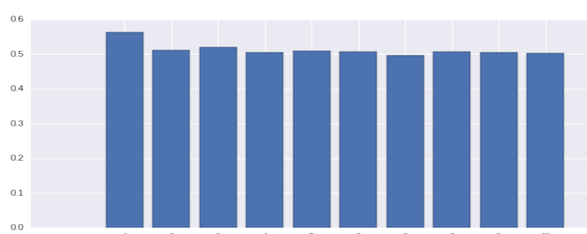


Fig 1. Graph of overall accuracy for KNN for k = 1 to 10 for unigrams.

Fig 1 shows the accuracy of the KNN classifier as k increases. The algorithm tends to do best when k is between 1 and 5. Lower k-values work better because the algorithm can find a few similar recipes. When the k-value increases, more noise is introduced into the system since after the few similar recipes most recipes are roughly the same distance away from the test recipe.

4 Results

The baseline accuracy for our experiments is 20%. This is due to the uniform nature of our testing corpus. If a classifier were to always guess one of the five cuisines, it would get by chance 20% of the recipe classifications correct.

Recursive partitioning did best when Unigrams were used as the feature with 70.12% overall accuracy, Naive Bayes and Support Vector Machine did better when other features were used.

Feature	NB	Rpart	KNN	SVM
Caribbean	67.22	58.89	52.78	65.56
Chinese	87.78	80.55	66.11	81.67
French	63.33	77.78	48.33	68.89
Mexican	56.67	76.67	57.22	52.22
Italian	67.22	56.67	57.22	63.33
Overall	68.44	70.12	56.33	63.33

Table 1. Accuracies with unigrams as features

Table 1 shows the accuracies obtained using Unigrams as feature. All the unique words in a recipe are used as predictor variables. While Naïve Bayes and SVM did equally well Rpart did the best. Overall, the classifiers have been most successful in classifying Chinese recipes.

Feature	NB	Rpart	KNN	SVM
Caribbean	75.56	73.34	-	72.78
Chinese	91.11	75.56	-	91.67
French	77.22	62.23	-	76.67
Mexican	67.22	64.45	-	66.67
Italian	70.56	38.34	-	70.00
Overall	76.33	62.78	-	75.56

Table 2. Accuracies with bigrams as features

Table 2 shows that both Naïve Bayes and SVM perform well with the bigrams feature, with 76.33% accuracy and 75.56% accuracy respectively. It is observed that the Chinese recipes are classified higher for all three classifiers. Bigrams were not used with the KNN classifier due to time constraints, which is a result of the computational complexity of the classifier.

Feature	NB	Rpart	KNN	SVM
Caribbean	73.89	64.44	56.67	71.11
Chinese	87.78	80.56	66.67	86.67
French	71.67	67.23	50.00	71.67
Mexican	61.67	56.12	56.67	59.44
Italian	63.33	57.23	49.44	61.67
Overall	71.67	65.12	55.89	70.11

Table 3. Accuracies with nouns as features

Table 3 shows the performance of the nouns feature. Nouns contained all the ingredients, utensil, cookware, and any other possible objects in the recipe. Overall the nouns feature performs quite well, and even manages to outpace the unigrams feature. Looking at the data, Naïve Bayes and SVM perform the best with this feature; with Naïve Bays performing best with a 71.67% accuracy and SVM a close second with a 70.11% accuracy. It is also observed that the Chinese recipes have the highest rate of accuracy across all classifiers.

Feature	NB	Rpart	KNN	SVM
Caribbean	47.22	50.55	41.67	51.67
Chinese	96.44	67.23	46.11	62.22
French	65.56	36.12	41.11	66.67
Mexican	48.89	42.23	39.44	45.00
Italian	52.22	41.12	49.44	49.44
Overall	56.67	47.45	43.56	55.00

Table 4. Accuracies with verbs as features

Table 4 shows the performance of the verbs feature. Verbs contained all cooking verbs, as well as placement verbs such as “set” and linking verbs such as “is”. This feature set does rather poorly when compared to others. From the data, it is observed that the Naïve Bayes and SVM classifiers perform the best with this feature; with Naïve Bays performing best with a 56.67% accuracy and SVM at 55.00% accuracy. Again, the Chinese recipes have the highest percent accuracy across classifiers.

Feature	NB	Rpart	KNN	SVM
Caribbean	72.78	62.78	50.00	70.00
Chinese	87.22	81.12	66.11	86.67
French	72.22	62.78	52.22	72.78
Mexican	59.44	54.45	53.33	56.67
Italian	63.33	58.89	57.78	62.22
Overall	71.00	64.00	55.89	69.67

Table 5. Accuracies with Nouns and verbs as features

Table 5 exhibits the capabilities of both the noun and verb features together. This feature does slightly worse than the nouns feature, but overall still performs well. Again, it is discerned that Naïve Bayes and SVM perform the best and are within a couple percent accuracy of each other. The Chinese cuisine type does better than the other cuisine types.

Feature	NB	Rpart	KNN	SVM
Caribbean	64.44	62.78	52.22	66.11
Chinese	81.11	75.00	61.67	80.56
French	64.44	68.34	54.44	62.78
Mexican	53.33	63.89	58.33	52.78
Italian	62.22	56.67	52.22	61.67
Overall	65.11	65.34	55.78	64.78

Table 6. Accuracies with ingredients as features

In Table 6, we evaluate the capabilities of the ingredients feature. The ingredients feature is a subset of the nouns feature. It performs rather well, with the Rpart classifier doing the best overall, and the Naïve Bays and SVM classifiers close behind. Chinese recipes end up doing the best across all classifiers here as well.

Feature	NB	Rpart	KNN	SVM
Caribbean	42.78	56.66	43.89	40.00
Chinese	58.33	23.33	43.89	55.00
French	45.00	46.67	41.11	48.89
Mexican	35.56	53.34	41.67	37.67
Italian	32.78	56.67	38.89	36.67
Overall	42.89	43.83	41.89	43.67

Table 7. Accuracies with cooking actions as features

Table 7 displays the results of the cooking actions feature. This feature is a subset of the verbs, consisting of 160+ cooking verbs. This feature does terribly, scoring even lower than the verbs feature. Rpart ends up with the highest overall accuracy, with Naïve Bayes and SVM within a percent.

Feature	NB	Rpart	KNN	SVM	Average
Unigram	68.44	70.12	56.33	66.33	65.31
Bigram	76.33	62.78	-	75.56	71.56
Nouns	71.67	65.12	55.89	70.11	65.70
Verbs	56.67	47.45	43.56	55.00	50.67
Nouns and verbs	71.00	65.12	55.89	69.67	65.42
Ingredients	65.11	65.34	55.78	64.78	62.75
Cooking Actions	42.89	43.89	41.89	43.67	43.08

Table 8. Accuracies of all features for each classifier

It is observed in Table 8 that the bigram feature did the best, whereas the cooking actions feature did the worst. It is also noted that both the nouns feature and the nouns and verbs features performed better than just standard unigrams.

Caribbean	Chinese	French	Italian	Mexican
Lime	Soy	Butter	Cheese	Cumin
Juice	Wok	Bake	Parmesan	Tortilla
Blender	Sesame	Oven	Grate	Corn
Allspice	Ginger	Preheat	Olive	Chili
Cinnamon	Cornstarch	Melted	Basil	Cilantro

Table 9. Top 5 words from each cuisine with the most pointwise mutual information.

Caribbean	Chinese	French	Italian	Mexican
0.12796	0.34231	0.10102	0.26592	0.15228
0.06165	0.19141	0.08550	0.18796	0.13971
0.05395	0.17942	0.08136	0.12646	0.11690
0.04780	0.15770	0.06122	0.11605	0.11063
0.04169	0.13759	0.06090	0.11347	0.09189

Table 10. Bits of information provided by each word in Table 9.

From Table 9 and Table 10, we can see that overall Chinese has the most distinguishing top 5 words. Caribbean words have the least mutual information out of all the cuisine types.

Caribbean	Chinese	French	Italian	Mexican
Salt	Mince	Bake	Grate	Shred
Refrigerate	Beat	Melt	Salt	Blend
Grill	Fry	Slice	Spread	Chop
Leave	Boil	Sprinkle	Taste	Roll
Scoop	Mix	Beat	Arrange	Puree

Table 11. Top 5 cooking actions from each cuisine with the most pointwise mutual information.

Caribbean	Chinese	French	Italian	Mexican
0.02759	0.02929	0.09141	0.12646	0.08488
0.01627	0.02710	0.06090	0.09895	0.03125
0.00999	0.02632	0.03440	0.02897	0.02273
0.00920	0.02573	0.01626	0.01365	0.01662
0.00906	0.02409	0.01574	0.01235	0.01264

Table 12. Bits of information provided by each word in Table 11.

From Table 11 and Table 12, we can see that overall there isn't a clear cuisine that does better than the rest. Caribbean words have the least mutual information overall out of all the cuisine types again. Another notable observance is that overall the mutual information in Table 12 is lower than in Table 10.

5 Evaluation

Using Table 8, the bigram feature is shown to outperforms all the other features. This is most likely since the order that things occur in is at least moderately important when determining cuisine classification. Bigrams may also help get more information from ingredients that have more than one word in the name, such as soy sauce.

The accuracies obtained using only verbs as features set has been very low (Table 4). This seems to indicate that all the recipes have a very good overlap of verbs used and hence less useful for classification. This is further confirmed by looking at Table 11 and Table 12. Not only are there verbs that share multiple cuisine categories, such as salt, none of these words have much mutual information when compared to Table 9 and Table 10.

Cooking actions, being a subset of verbs, had the least accuracy overall. This is unsurprising since the cooking actions is a subset of verbs. As other verbs are removed, bits of information are lost which causes the classifiers to perform worse. Due to this large overlap, verbs are not that useful for classification.

When the accuracies from Table 5 (where nouns and verbs are used as features) are compared with accuracies in Table 3 (where only nouns are used as features), we can see that the verbs have detrimental effect on the performance of classifiers. This is most likely due to added noise from the verbs. Since the verbs give so little information (Table 12) compared to nouns (Table 10), it confuses the classifiers, causing the slight decrease in performance.

Ingredients, when used alone as the feature set did not perform that well when compared to the study done with only nouns, despite their large presence in Table 9. This suggests that other nouns, such as utensils and cookware, provide important information to the classifiers, leading to an overall higher accuracy.

Most of the Tables in section 4 have Chinese recipes as the best performing cuisine type for each classifier. This is corroborated by Table 9 and 10, which show that the top 5 words most distinguishing in Chinese recipes give more information than the top 5 words of any other cuisine. This suggests that Chinese cuisine is much more distinct than the other cuisines used in the study. Since both French and Italian are European and Caribbean and Mexican are both Latin American, this is most likely the case.

6 Conclusion

Out of the features this project covers, bigrams are significantly better in terms of overall accuracy than any of the other features. The most probable reason for this is because bigrams capture not only the words themselves, but also pairs of words. This is important when describing ingredients, as many ingredients have names containing multiple words.

Another interesting feature in this project ended up being nouns. Nouns did much better than unigrams in terms of accuracy. Nouns were also much of the most distinguishing words in the corpus. Even more interesting was that when the ingredients were separated from the other nouns, they had a worse overall accuracy. This implies that ingredients are not the only necessary components to determine cuisine type; utensils and cookware play an important role in classification as well.

A third interesting discovery was that verbs don't help much in determining the cuisine of a recipe. Many of the verbs in both the verbs and cooking actions features overlapped heavily between cuisines. Even the most distinguishing words provided little information, and often overlapped themselves. When combined with the nouns feature, the verbs hurt the overall accuracy slightly.

There are many more possible avenues to explore with this project. For example, the temporal location of the words may give better insight into the cuisine of a given recipe. Exploring the sematic relations between the verbs and nouns of the recipe may also offer more information. A third improvement that could be made is more preprocessing of the data. Currently the data is only split up into words and punctuation. Adding stemming may help the system learn more from the verbs in the dataset. Removing stop words may also help to get rid of noise in the data. One final improvement that could be made is adding more data to the dataset. This project includes a file gatherer and web scraper that should make it easier to add more data to the corpus for future experiments using the recipe dataset.

Many avenues exist for the expansion of this project as well. Currently, only 5 cuisine types are used. Extending the number of cuisines that can be accurately classified would allow recipes to be sorted better. Another improvement to expand the project would be to enable the classifiers to sort the recipes by dish type (such as main dish, side dish, etc.). This would allow the data to be sorted even further, and help the cuisine classifying by using dish type as a feature of the recipe.

References

[1] Su, Han, et al.” Automatic recipe cuisine classification by ingredients.” Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. ACM, 2014.

[2] Wang, Liping, et al.” Substructure similarity measurement in Chinese recipes.” Proceedings of the 17th international conference on World Wide Web. ACM, 2008.

[3] Xie, Haoran, Lijuan Yu, and Qing Li.”A Hybrid Semantic Item Model for Recipe Search by Example.” Multimedia (ISM), 2010 IEEE International Symposium on. IEEE, 2010.

[4] Juhi Naik, Vinaya Polamreddi “Cuisine Classification and Recipe Generation” http://cs229.stanford.edu/proj2015/233_report.pdf