Identifying Cuisines From Ingredients

Sridhar Srinivasasubramanian Brajesh Kushwaha Vishal Parekh

***Abstract*—In this report we describe our model to predict the cuisine based on the ingredients. Such a model has many practical implications. One of them is to predict the cuisine based on the analysis of the picture of the item. Another is to suggest users food items based on the items that they have had previously. We used classification algorithms to group together the ingredients that belong to the same cuisine and using it to predict the cuisine. We end the paper with a discussion of the observations and lessons learnt from the project.**

***Keywords*—Data mining, K-means, Random forests, Cuisines, Ingredients,Bag of Words**

I. INTRODUCTION

We daily come across many ingredients that are used to make delicious recipes. All these recipes are part of a certain cuisine. These certain style of cooking and ingredients identify the cuisine. Each of the ingredients tells a lot about the type of recipe or cuisine that is been cooked. Ingredients signify the cultural traits and the geographical proximity of its origin of the ingredients.

Certain ingredients give more information of the type of cuisine that it is part of then others. For example, *garam masala* is more certainly indicative of the cuisine being Indian. Another example being that a recipe with *enchiladas* as an ingredient will most likely be Mexican. More importantly the ingredients in combination can be used to almost certainly to predict the cuisine. But some of the ingredients like salt and water barely carry any significance because each of these ingredients are almost part of many of the cuisines. Hence we would like these ingredients to have the least weight.

The practical implications of such models are many. The first example is that of images. In the age of social network revolution where people upload many images of food items that they consume. The heat maps and colors of the pictures can be used to predict the ingredients. Then based upon the ingredients the prediction can be made of the type of cuisine. This knowledge of a particular user can help to suggest other food items and restaurants that they may find similar cuisines. Our model focuses on the later half of the problem i.e. we just accept the ingredients and predict the cuisine that these ingredients would be part of.

In the first section, we analyze the previous work that has been done in this specific field of data mining. More specifically we analyze the research that has been carried out in this particular problem. Finally adding the impact each of the material had on our take of the model we developed.

The second section we perform an exploratory analyzes of the data. This data is provided by a Kaggle competition by the name of “Whats Cooking?”. This competition's problem is to identify the cuisine based on the recipe's ingredients. The training data provides the ingredients and the cuisine it belongs to. The next step of this section is to analyze the data. We analyze number of ingredients in a cuisine, top ingredients overall, top ingredients per cuisine, spread of ingredients across cuisines etc. This analysis helped us to efficiently develop the model.

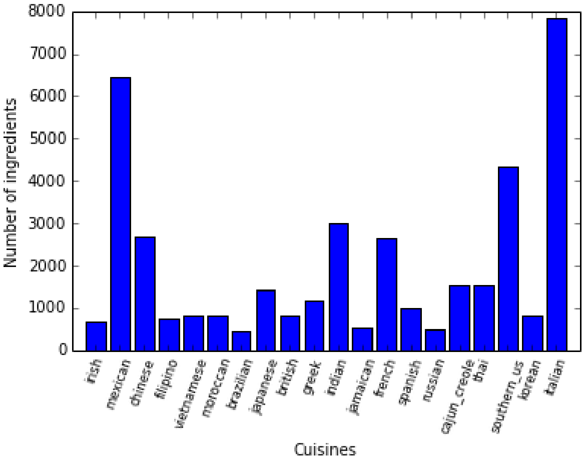

Fig. 1. Histogram showing the count of recipes for each cuisine in the training set

In the third section we describe our model in detail. We describe how we performed dimensionality reduction on the data, next we clustered the data and finally used the clustered data as a feature vector to identify the cuisine. We used to k-means for clustering the data and then used random forests on the clustered data.

In the fourth section we show the results of our model. We have got 76% percentage accuracy on the test data which was provided in Kaggle. We then describe the approaches we tried but those that did not yield in better results.

In the final two sections we conclude the paper and describe the things that we learnt from the Assignment and the overall methodology used. We then describe the future work that we will like to perform.

II. RELATED WORK

We used the tutorial “Bag of Word”[1] provided by Kaggle for the competition. This tutorial uses Natural Language Processing techniques and Word2vec algorithm to get distributed word vectors. Word2vec [2] was developed by Google for bag of words and skip-gram architectures for computing vector representations of words. The k-means clustering algorithm[3] is used for clustering.

Finally we used random forests[4]. A random forest is a meta estimator that fits a number of decision tree classifiers

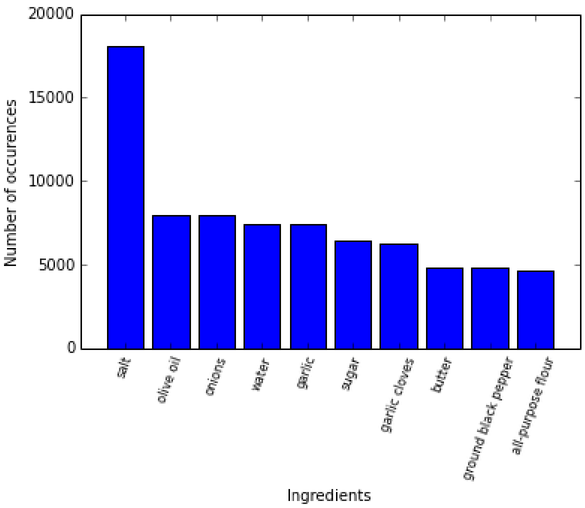

Fig. 2. Histogram showing the counts of occurrences of top ingredients

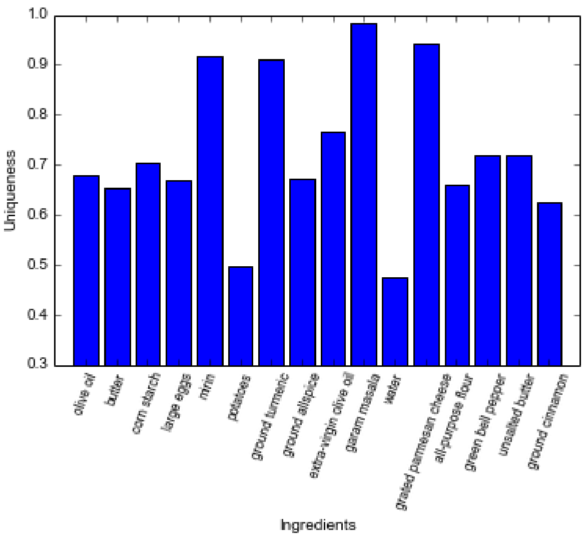

Fig. 3. Uniqueness measure of ingredient to particular cuisines

TABLE I.
TOP INGREDIENTS PER CUISINE

Cuisine	Ingredient	Count
Italian	salt	0.4407
	olive oil	0.3969
	garlic cloves	0.2066
	grated parmesan cheese	0.2016
	garlic	0.1877
Mexican	salt	0.4225
	onions	0.2319
	ground cumin	0.2091
	garlic	0.2046
	olive oil	0.2002
Southern US	salt	0.5301
	butter	0.2905
	all-purpose flour	0.2829
	sugar	0.2440
	large eggs	0.1727
Chinese	soy sauce	0.5099
	sesame oil	0.3423
	salt	0.3393
	corn starch	0.3389
	sugar	0.3083
Indian	salt	0.6440
	onions	0.3979
	garam masala	0.2870
	water	0.2731
	ground turmeric	0.2424

on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. We choose random forests because there is no smooth decision boundary in the data provided. We used the sklearn library[5] to perform the random forest classifier.

III. Exploratory Analysis

The dataset used for this project was available as part of the Kaggle challenge. The dataset comprised of 39774 recipes with their constituent ingredients tagged against the cuisine they belong to. Like any other recipe book, the constituent ingredients in the data-set were represented by their common/popular name which many times would be multi-word descriptions. A major challenge was to find the catch words/phrases from the generic adjectives added to the ingredients(eg. dried, crushed, etc.). To understand the data, we started of by analysing the cuisine and ingredient distribution across the training set.

To get an overall view of the data, we started by plotting the histogram of recipes per cuisine in the training set to observe the distribution of the training data. Figure 1 shows the plotted histogram.

Next we tried to find the popular ingredients in each of the cuisines. This gives an idea of the ingredients that form an integral part of the cuisine. Table 1 shows the normalized occurrences ingredient (Percentage occurrence of an ingredient in a cuisine) in popular cuisines. As seen in Table 1, Salt defines Italian cuisine but it tops the chart for almost every other cuisine as well. Hence, salt does not help in identifying the cuisine rather the exotic ingredients like garam masala (for Indian). This shows us that simply identifying a popular ingredient may not be enough. We need to find the popular ingredients in each cuisine that are rarely found in others. The next approach tries to do that.

For identifying how unique an ingredient is to a cuisine, we needed to measure the spread of an ingredient. So, we formulated a metric 'uniqueness' which takes into account the distribution of normalized occurrences of an ingredient across cuisines. For example, if salt occurs in 5 different cuisines with a normalized counts of 0.9, 0.8, 0.7, 0.6, 0.5, it's 'uniqueness' will be the ratio of sum of top twenty percentile of the occurrences and sum of all normalized occurrences. Here, the 'uniqueness' score will be 0.2571. Whereas another ingredient which has a normalized count of 0.9, 0.1, 0.05, 0.02, 0.01 will be 0.8333. Fig. 3. shows the distribution of the 'uniqueness' measure amongst the popular ingredients listed in Table 1.

IV. Model

For this predictive task, we chose to do clustering for dimensionality reduction and random forests for training the data as it can separate non-linear decision boundaries across cuisines. We found 6714 unique ingredients in then training set, which also including repetitions due to adjectives and catch words. A naive way of representing a feature vector would be to have a boolean variable for each of the 6714 ingredients whether it is present or not. It is not very efficient because the feature vector will occupy a lot of memory and also sparse.

We also needed to somehow group similar ingredients together, so that the model can be trained considering them equivalently.

A. Transformation

We need to translate each of the ingredient into Gaussian space using contextual cues to group similar ingredients. We used the Word2vec library which is an implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. We trained our word2vec model by passing individual recipes as sentences. Each of our input words can now be represented as a 300 dimensional vector on which clustering algorithms could be applied for dimensionality reduction.

B. Clustering

The Gaussian mapping, in the step above, maps each word of the ingredient into a feature vector of size 300. These data points are now clustered together based on these feature vectors. We used k-means clustering algorithm to achieve the same. These clusters will then be used to create a new feature vector to be used for the final classification (in the next step) using random forest. The number of clusters has a high impact on the accuracy of classification and rightly so. As shown in Fig. 4, we obtain nearly 750 as an optimal number of clusters after which the accuracy of prediction starts dropping.

Table 2 shows some sample clusters obtained based on the K-means clustering of the word vectors. Some of the exotic ingredients of Mexican, Italian, French and Indian can be seen to be correctly separated out.

The accuracy falls on having too low number of clusters because it leads to very generic clusters which loses the specific relationship among the ingredients for example, with a cluster size of 20, we get Chilies and Chardonnay together. Chardonnay is very exotic to French cuisine whereas Chilies is generic to all. Ideally Chilies should be part of a non-exotic cluster had there been enough clusters. Increasing the number of clusters beyond a certain point leads to very sparse feature vectors which demands a large number of training samples for random forest to be effective.

To understand the effectiveness of the clustering, let us compute the cluster correlation for clustering with $k = 30$. Figure 5 shows few very high correlations e.g., cluster (11 & 21) and (13 & 22). Cluster 11 consists of Indian ingredients

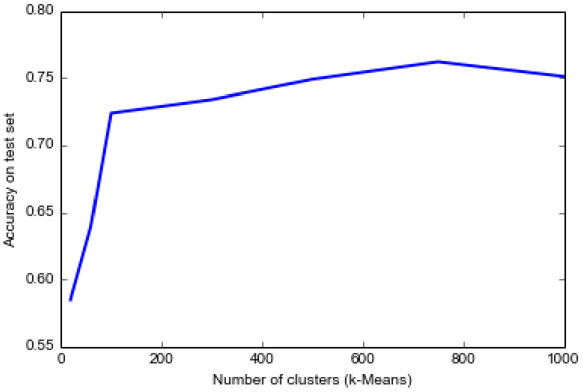

Fig. 4. Effect on accuracy w.r.t number of clusters in k-Means

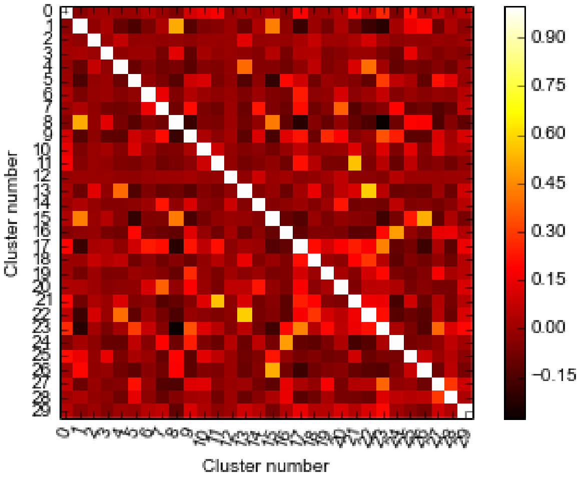

Fig. 5. Heat map showing cluster correlation for #clusters = 30

TABLE II.
Sample clusters obtained from K-Means

Cluster	Ingredients
Mexican	jack, pinto, Jack, salsa, non-fat, picante, enchilada, verde, tortillas, tortilla, Mexican, pico, guacamole, refried, El, blend, colby, Paso, monterey, Monterey, gallo, taco, chips'
Italian	bucatini, fava, parsley, cavatelli, olive, guanciale, gemelli, linguine, spaghettini, tagliatelle, lasagne, orecchiette, rigate, freshly, pappardelle, spaghetti, risotto, parmagiano
French	Johnsonville, Tomatoes, Sausage, dog, Virgin, Wish-Bone, Canola, Diced, pasta, Broth, capers, Pompeian, Andouille, Crystal, Flour, Swanson, Red, cube, bouillon, Garden, Gourmet, bun
Indian	jeera, garam, amchur, yoghurt, curds, methi, paneer, kasuri, fenugreek, ghee

like jaggery, basmati, masala while cluster 21 consists of other set of Indian ingredients like rajma, chutney, naan, etc. Similary cluster 13 and 22 consist of Chinese ingredients showing very high correlation. This shows that the clustering technique is effective in group similar ingredients together. The low cluster size (= 30) leads to poor correlation in other clusters but due to high density of a higher number of cluster correlation map, we are showing a scaled down version.

C. Supervised Learning

Once we have obtained cluster assignments for each ingredient, our feature vector is an array of size number of clusters and each element in the array (a[c]) is the count of number of times an ingredient in the recipe belongs to a cluster 'c'. The target variable is the different cuisine categories. Random Forests are an ensemble learning method that construct multiple decision trees during training and output the mean prediction of the individual forests. Random Forest Classifier was used for training as opposed to using multi-class logistic regression as there is a non-linear decision boundary (presence/absence of a cluster can have a huge impact on the prediction).

V. Results

We tried several different approaches with the above pipeline.

Our preliminary approach was to train word2vec by grouping all ingredients by their cuisine. So, we had 20 different bag of words for each cuisine. Our assumption was that, each of the ingredient in the same bag would get very close Gaussian distance and hence can be clustered using K-means easily. But this didn't yield us good results as there were very few datapoints and there were a lot of repetitions of the ingredients. This yielded a meagre 56 % accuracy.

The next idea was to not group all the recipes of a cuisine together rather than group only the ingredients of a recipe together thereby, giving us number of groups equal to the total number of recipes. This was expected to yield better results than the previous approach. The earlier approach of grouping all the recipes together leads to a false sense of correlation between two ingredients of different recipes (but same cuisine). This poor correlation extrapolated over thousands of ingredients across cuisines leads to very poor and generic grouping. This method with 20 clusters for k-means (section IV-B) improved the accuracy to 59%. Increasing the number of clusters to 500 gave an accuracy of 72.4%.

As we wanted to group multi-word ingredients as per the root word, the approach we took was to split the string and train the word2vec neural network. Now, common root words are slowly pulled towards a common cluster as they might be common across multiple cuisines. Ingredients which are specific to one particular cuisine cluster together. This gave us a more generic solution which performed well with the unseen data. This approach gave us an accuracy of 76.2 %.

In order to prune the data further, we tried removing ingredients which are too common across all cuisines based on a set threshold of 'uniqueness' score. But that ended up removing more ingredients than we anticipated and our accuracy dropped to 74 %. More work needs to be done in identifying exotic ingredients and mitigating the effect of common ingredients.

TABLE III.
Evaluation of various methodologies

Methodology	Accuracy (%)
Word2vec ingredients grouped by cuisine	56
Word2vec ingredients grouped by recipes	72.24
Word2vec ingredients grouped by recipe and split multiword	76.2
Pruning with threshold uniqueness score	74

Table 3 shows the comparison in accuracies of the methodologies.

VI. Conclusion

Our approach to the problem of identifying the cuisines based on the ingredients has been broken into three parts - we have taken the data set from Kaggle and used it train our word2vec library. We initially transformed our data into Gaussian space, then we clustered the data using k-means and finally applied Supervised learning using random forests to classify.

Our preliminary accuracy on the test set in the Kaggle set was only 56%. But then increasing the number of clusters, split the multi word ingredients and other small tweaks led us to a final accuracy of 76.2%. We can improve our results with supervised learning in the first few steps of the model rather than unsupervised learning. The accuracy of word2vec also depends on the number of training examples. Also, determining the optimal number of clusters depending on the pruning of data remains to be done. It remains to be seen if we will have better looking clusters with more recipe training examples.

VII. Future Work

Our model was based on clustering using word2vec library based on the patterns it sees in the recipes. If we could somehow associate some cuisine information whilst training and enhance it with additional information, we can get better word vectors in the Gaussian space which leads to better looking predictions. We also need to do further investigation on different supervised learning approaches.

A second thing that we would like to work would be to use our model to predict the cuisine based on the image of the food item. The ingredients could be found based on image analysis and then based on the ingredients the cuisine could be predicted. The final piece would be to suggest the user restaurants based on the images/cuisines.

References

[1] https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words

[2] https://code.google.com/p/word2vec/

[3] https://en.wikipedia.org/wiki/K-means_clustering

[4] http://link.springer.com/article/10.1023