# Classification of Cuisines from Sequentially Structured Recipes

Tript Sharma
*Department of Mechanical Engineering*
*Delhi Technological University*
Delhi, India
triptsharma22@gmail.com

Utkarsh Upadhyay
*Department of Electronics Engineering*
*Jamia Millia Islamia University*
Delhi, India
utkarshdhy@gmail.com

Ganesh Bagler
*Department of Computational Biology*
*Indraprastha Institute of Information Technology Delhi (IIIT-Delhi)*
bagler@iiitd.ac.in



***Abstract*—Cultures across the world are distinguished by the idiosyncratic patterns in their cuisines. These cuisines are characterized in terms of their substructures such as ingredients, cooking processes and utensils. A complex fusion of these substructures intrinsic to a region defines the identity of a cuisine. Accurate classification of cuisines based on their culinary features is an outstanding problem and has hitherto been attempted to solve by accounting for ingredients of a recipe as features. Previous studies have attempted cuisine classification by using unstructured recipes without accounting for details of cooking techniques. In reality, the cooking processes/techniques and their order are highly significant for the recipe's structure and hence for its classification. In this article, we have implemented a range of classification techniques by accounting for this information on the RecipeDB dataset containing sequential data on recipes. The state-of-the-art RoBERTa model presented the highest accuracy of 73.30% among a range of classification models from Logistic Regression and Naive Bayes to LSTMs and Transformers.**

***Index Terms*—Recurrent Neural Networks, Transformers, Classification, Sequential Recipes**

## I. Introduction

Cuisines represent the culinary imprint of cultures. The structure of cuisines is shaped by composition of their recipes. Increasing availability of data of cuisines has led to data-driven explorations of cuisines such as food pairing, culinary fingerprinting and cuisine classification. Classification of cuisines is an interesting problem with applications for recipe recommendation and generation of novel recipes. Hitherto, cuisine classification has been attempted using ingredients of recipes as a feature. These approaches have overlooked key factors such as cooking techniques and their order which clearly form a key aspect of recipes.

We account for the loss of these information by including these details for the cuisine classification problem. As opposed to treating recipes as an itemset, we propose a methodology that views the problem as a Text Classification (TC) problem. TC refers to annotating text with one category or other based on content words and their collocations. In this article, we propose different architectures for cuisine classification on sequential datasets.

## II. Related Work

Availability of detailed recipe data has evoked the interest in recipe recommendation and generation. In the past, classification of cuisines has been attempted on the basis of various factors such as time, ethnicity and place by creating six different feature stylistic sets from the data document [1]. Han Su et. al. [2] have worked on cuisine identification by using ingredients used in recipes as a basis. By identifying ingredients as features, they could provide insights on cuisine similarity. A personalised cuisine recommendation system based on user's preferences has also been proposed [3] where user's preferences are derived from their browsing activities.

Support Vector Machines [4] and several other machine learning techniques have also been implemented towards generation of a cuisine. Recently, a study on classification of cuisine on the basis of the recipe's ingredients [5] suggested a detailed relation between a recipe and its ingredients.

In this article, we propose that, beyond ingredients, even the processes and utensils involved in cooking a recipe and their order of occurrence can provide significant insights into the cuisine. We have used RecipeDB (site) [6] dataset. And to test our hypothesis, we perform classification on the dataset using several machine learning techniques, neural networks and transformers.

## III. Dataset

RecipeDB was used as the source of structured data on recipes for the analysis. The dataset contains 118,071 recipes obtained from sources like AllRecipes, Epicurious Food Network, and TarlaDalal. The dataset consists of 26 cuisines as shown in Table II. Moreover, it contains an aggregation of 20280 unique ingredients, 256 unique processes and 69 unique utensils. Sample dataset of RecipeDB can be seen in Table I. Our analysis involves the following substructures of cooking recipes pertaining to traditional recipes, namely, recipes, ingredients, processes and utensils.

RecipeDB consists of ingredients, processes and utensils mined from unstructured recipe scraped from the above mentioned resources. The substructures for the recipes are mined in a sequential fashion depending upon the order in which they are used in preparing the dish. The dataset is highly sparse with a sparsity ratio of 99.50%. Out of the 20,400 distinct ingredients obtained, 11738 occur at most in one recipe such as 'lasagna noodle wheat', while 'add' appeared 1,88,004 number of times. The corresponding cumulative frequency table for the

TABLE I
SAMPLE DATASET FROM RECIPDB

| Recipe ID | Continent | Cuisine | Recipe |
|---|---|---|---|
| 2610 | African | Middle Eastern | ['water', 'red lentil', 'rom tomato', ,'smooth', 'stir', 'heat'] |
| 3957 | Asian | Southeast Asian | ['olive oil', 'onion', 'garlic', 'ginger', , 'stir', 'add', 'cook ', 'season', 'garnish', 'pot'] |
| 4153 | Asian | Indian Subcontinent | ['coconut milk', 'milk', 'white sugar', 'basmati rice', , 'stir', 'cook', 'saucepan', 'bowl'] |
| 79897 | Latin American | Mexican | ['beef', 'chunky salsa', 'mushroom', 'garlic', , 'heat', 'simmer', 'serve', 'skillet'] |
| 138976 | European | Deutschland | ['oven buttermilk biscuit', 'onion', 'cream', ..., 'spread', 'sprinkle', 'bake', 'pan'] |
| 149191 | North American | Canadian | ['raisin', 'fig', 'water', 'date', 'butter', , 'chill', 'cut', 'bowl', 'processor', 'pan'] |

TABLE II
DATASET INFORMATION

| Cuisine | Number of Recipes | Cuisine | Number of Recipes |
|---|---|---|---|
| Australian | 5823 | Japanese | 2041 |
| Belgian | 1060 | Korean | 668 |
| Canadian | 6700 | Mexican | 14463 |
| Caribbean | 3026 | Middle Eastern | 3905 |
| Central American | 460 | Northern Africa | 1611 |
| Chinese and Mongolian | 5896 | Rest Africa | 2740 |
| Deutschland | 4323 | Scandinavian | 2811 |
| Eastern European | 2503 | South American | 7176 |
| French | 6381 | Southeast Asian | 1940 |
| Greek | 4185 | Spanish and Portuguese | 2844 |
| Indian Subcontinent | 6464 | Thai | 2605 |
| Irish | 2532 | UK | 4401 |
| Italian | 16582 | US | 5031 |

TABLE III
FREQUENCY DISTRIBUTION OF FEATURES

| Number of Features | Frequency | Number of Features | Frequency |
|---|---|---|---|
| 304 | >1000 | <2 | 11738 |
| 106 | >5000 | <3 | 14015 |
| 57 | >10000 | <4 | 15002 |
| 43 | >15000 | <5 | 15620 |
| 34 | >20000 | <6 | 16073 |
| 24 | >25000 | <7 | 16394 |
| 19 | >30000 | <8 | 16627 |
| 17 | >35000 | <10 | 17016 |
| 13 | >40000 | <15 | 17314 |
| 12 | >45000 | <20 | 17519 |

number of items shown in Table III, represents the nature of the dataset.

## IV. PREPROCESSING

Before performing cuisine classification on RecipeDB data, preprocessing was implemented on structured and sequential lists of ingredients, processes and utensils. Furthermore, the digits or symbols were omitted from the items to only keep words, thereby reducing the noise in this highly sparse dataset. The preprocessing further involved tokenization followed by lemmatization of the dataset, resulting in 20,400 distinct entities.

The data is further processed to conform to the classification model requirements. Since an individual word itself doesn't impart any semantic or syntactic significance to the classification models that require quantified features as inputs, each item was translated to vectors using two techniques, namely, TF-IDF vectorization and word embedding. Depending upon the preprocessing method used, the models employed can be broadly classified into two categories: sequential models and statistical models. If the dataset is sequential, it is evident that sequential models like RNNs work better while for non sequential datasets, models like Logistic Regression, SVM, etc. perform better.

Word embeddings are essentially word representation as vectors such that semantically similar words have similar vectors whereas TF-IDF vectorization method observes the sequence of items as distinct words. Thus, TF-IDF vectors don't preserve the sequential nature of the data. Yet, we used TF-IDF technique because of its weighted function which reduces the effect of high frequency yet less meaningful words and provides a good analytical cause.

## V. CLASSIFICATION

Classifying the recipes region/cuisine-wise, based on the elements involved in cooking a recipe is a major and the most important part of our analysis. The analysis treats recipes either as a sequential or as an unordered set of items. Many state-of-the-art machine learning models with TF-IDF vector inputs, such as Naive Bayes, Logistic Regression, Ensemble models along with boosting and Support Vector Machines, were tested for text classifications. Also, sequence models such as Recurrent Neural Networks and state-of-the-art NLP transformers such as BERT and RoBERTa were tested on our dataset to analyse the 'sequential nature' of the dataset. We will further discuss the implementations of these classifiers in the 'Experiments' section.

### A. Naive Bayes

Naive Bayes (NB) classifier is probabilistic in nature. It is based on the supposition that all the features are independent and autonomous. NB selects the label which maximizes the posterior probability:

$$P(C_k|x) = P(C_k) * P(x|C_k)/P(x) \quad (1)$$

while the naive supposition is:

$$P(x_i|x_{i+1}, , x_n, C_k) = P(X_i|C_k) \quad (2)$$

In spite of the fact that the naive supposition is false most of the time, NB gave extremely competitive results with respect to other classifiers.

### B. Logistic Regression

Logistic Regression (LG) is the most used and a fundamental classifier. LG is also probabilistic in nature. It is based on the following Sigmoidal equation:

$$S(x) = 1/(1 + e^{-x}) \quad (3)$$

LG treats the problem as a generalized linear regression model, which can be expressed as:

$$f(k, i) = \beta_{0,k} + \beta_{1,k}x_{1,i} + \beta_{2,k}x_{2,i} + \cdots + \beta_{M,k}x_{M,i}, \quad (4)$$

Here, for our multi-class classification problem LG is trained on a one-vs-rest scheme. Similar steps were followed in a previous research on a different dataset [5] where LG presented with the best results. This hold true in our cse as well in comparison with other baseline models.

### C. Support Vector Machine

Support Vector Machines (SVMs) have been demonstrated to be having the best performance when working with textual data [4]. For classification, SVMs require translation of a multi-class classification problem to binary classification. For this, the One-vs-All approach was used. Single classifier per class was trained with the training set belonging to that class annotated as positive while the rest of the samples as negative. A strong real-valued confidence score along with a class label, by the base classifiers is required for the decision. SVM then searches for the two best-fit parallel hyperplanes which separates the two classes of data, so that they are farthest from each other.

### D. Random Forest with Boosting

Random forest (RF) is a bagging decision tree approach [7]. When used as a classifier, it might not perform that well when working with a small number of features. But given that our problem is characterized with a large number of features, techniques such as RF with AdaBoost can turn out to be a good text classifier.

### E. RNN (LSTM)

Recurrent Neural Networks follow a temporal or sequential connection between nodes of a layer [8]. Therefore, they are an upgrade to the conventional neural networks which consider mutual independence among the sequential inputs. Furthermore, RNNs contain an internal 'memory' and hence making them suitable for remembering previous inputs. Therefore, the characteristics of RNNs align with our problem.

We employed a state-of-the-art RNN, the Long-Short Term Memory based neural network (LSTM) [9]. LSTMs are more complex than simple RNNs as they involve a cell- and gate-like input, output and forget gate. Using these gates it controls the flow of information through the temporal dimension. It decides whether any piece of information is significant for the broader or immediate goal, or should it be removed from the memory. On account of this significant characteristic, we employed a simple 2-layer LSTM.

### F. Transformer

RNNs go through words in a temporal fashion and if the sequence is long as in case of RecipeDB, the model tends to forget the crucial features of sequentially distant features. In order to overcome this limitation, attention based transformers were developed. Transformers [10] are the NLP models which are used to boost the speed of attention based models by enabling parallelization. They completely eliminate the recurrence with self attention to establish relationship between input and output, thus making them suitable for multiple language processing applications.

We employed BERT-base [11] and RoBERTa [12] models on RecipeDB. Both models perform bidirectional encoding implementing transformers after pre-training with the exception that RoBERTa is trained differently. RoBERTa was trained on longer sequences for more training steps than BERT.

## VI. Experiments

For the purpose of cuisine classification we implemented different machine learning models. We tested the accuracy of the models on the RecipeDB dataset to validate the results obtained. The data was divided into 7:1:2 ratio to obtain training, validation and testing datasets respectively. Therefore out of 1,18,071 recipes training, validation and testing datasets consist of 82,650, 12,021 and 23,380 recipes respectively.

Since recipes were represented as sequences of ingredients, processes and utensils all concatenated together, long sequence were generated. The sequences were pre-processed differently for statistical models and sequential models as described in Section IV. By feeding the features obtained as input to the classifiers mentioned in Section V yielded results shown in Table IV. The corresponding code and relevant files are present in the GitHub repository: https://github.com/cosylabiiit/cuisine-classification.

Among the various statistical models that were implemented for cuisine classification, Logistic Regression performed the best, but with an accuracy of only 57.70%. These models learn the frequency of occurrence of an ingredient or process or utensil to obtain the features unique to a cuisine instead of treating the recipes as an interrelationship among these items. Furthermore, since the dataset is sparse the models couldn't fit better, leading to high bias.

Owing to the temporal relationship among the items in recipes, we observed that sequential models perform better than the statistical models. However, LSTM model gave a lower accuracy than Logistic Regression and Linear kernel SVM used in [5]. The lower accuracy is justified as the model is among the most simplistic models in the recurrent neural

TABLE IV
PERFORMANCE METRICS OF APPLIED MODELS

| Dataset | Performance Metric | LogReg | Naive Bayes | SVM (linear) | Random Forest | LSTM | Transformer | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BERT | RoBERTa |
| RecipeDB | Accuracy | 57.70 | 51.64 | 56.60 | 50.37 | 53.61 | 68.71 | 73.30 |
| | Loss | 1.51 | 7.14 | 2.97 | 2.32 | 1.65 | 0.21 | 0.10 |
| | Precision | 0.56 | 0.50 | 0.54 | 0.48 | 0.53 | 0.58 | 0.67 |
| | Recall | 0.57 | 0.51 | 0.56 | 0.50 | 0.54 | 0.60 | 0.71 |
| | F1 Score | 0.56 | 0.50 | 0.54 | 0.49 | 0.53 | 0.57 | 0.69 |

network class. Furthermore while comparing the LSTM model with Transformers, the sequences are treated differently i.e. LSTMs consider left to right sequence order unlike the bi-directional check in Transformers. Moreover, despite having better memory logic than vanilla RNNs, LSTMs are limited by the number of words in the sequence which further reduces the accuracy.

The limitations of LSTMs have been overcome in Transformers as explained earlier which resulted in the optimal accuracy of 73.3% and a loss of 0.10 on the RecipeDB dataset. Hence, the model is able to predict the class with least errors on minimum number of datasets among the models tested. The model presents a high average precision, recall and F1 score values representing its ability for cuisine classification.

## VII. CONCLUSIONS

This article investigates different approaches for cuisine classification as a synthesis of ingredients, processes and utensils inherent to a cuisine. It also examines the effect of temporal relationships among the features to fingerprint the worldwide cuisines with the state-of-the-art RoBERTa model giving optimal results for the problem. Thus, we present a strategy to treat recipes as chains of events that are similar for a region and simultaneously contrasting from others to some extent to enable classification. Further, this articles has raised issues that can help optimise the results in different computational contexts such as recipe generation and recipe recommendation.

Apart from this, the article also raises some new research questions relating to cuisine classification. While our analysis considered for the sequential nature of recipes, the relationship among the three substructures remains unaccounted. Moreover, what features aid or hinder the classification of a recipe which could help one to uniquely distinguish between the cuisines? While maintaining the sequential nature of the recipes, redundant features were not removed. Hence, future analysis needs to identify the effect induced by these features on the classification accuracy of the models. Furthermore the imbalance among the classes affects the cuisine prediction accuracy of the classifiers. This can be reduced by ignoring the low frequency classes but would lead to a limited exploration of the world cuisines. This trade-off presents as a dilemma in this analysis.

We believe this article adds another dimension to the existing body of research on cuisine classification. This is of value for cuisine classification of unknown recipes and also aids in identifying salient features intrinsic to a cuisine.

## VIII. ACKNOWLEDGEMENT

G.B. thanks the Indraprastha Institute of Information Technology (IIIT-Delhi) for providing computational facilities and support. T.S and U.U. are Research Interns in Dr. Bagler's lab (Complex Systems Laboratory) at the Center for Computational Biology and thank IIIT-Delhi for the support.

## REFERENCES

[1] Y. HaCohen-Kerner, H. Beck, E. Yehudai, M. Rosenstein, and D. Mughaz, "Cuisine: Classification using stylistic feature sets and/or name-based feature sets," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 8, pp. 1644–1657, 2010.

[2] H. Su, T.-W. Lin, C.-T. Li, M.-K. Shan, and J. Chang, "Automatic recipe cuisine classification by ingredients," in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: adjunct publication*. ACM, 2014, pp. 565–570.

[3] M. Ueda, M. Takahata, and S. Nakajima, "Recipe recommendation method based on users food preferences," in *Proceedings of the IADIS International Conference on e-Society*, 2011, pp. 591–594.

[4] J. Weston and C. Watkins, "Multi-class support vector machines," Citeseer, Tech. Rep., 1998.

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

[6] D. Batra, N. Diwan, U. Upadhyay, J. S. Kalra, T. Sharma, A. K. Sharma, D. Khanna, J. S. Marwah, S. Kalathil, N. Singh *et al.*, "Recipedb: A resource for exploring recipes," *Available at SSRN 3482237*, 2019.

[7] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[8] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[9] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.