e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:12/December-2022 Impact Factor- 6.752 www.irjmets.com

HAN TO PREDICT CUISINE FROM RECIPES INGREDIENTS

Amir Ali*1, Anna Wróblewska*2

*1,2Faculty Of Mathematics And Information Science Warsaw University Of Technology, Warsaw, Poland.

DOI : https://www.doi.org/10.56726/IRJMETS32662

ABSTRACT

Cuisine is an important aspect of a recipe, as it influences the flavors and techniques used in the preparation of a dish. Predicting cuisine from recipe ingredients is a problem that involves classifying a recipe into one of many possible cuisines based on the ingredients it contains. This is a challenging task because there are many different cuisines in the world, each with its own unique set of ingredients and flavors. To accurately predict the cuisine of a recipe, a model must be able to learn the characteristics of each cuisine and use them to identify the correct cuisine based on the ingredients of the recipe. In this study, we implement different deep learning techniques, such as fastText, TextCNN, TextRNN, TextBiRNN, TextAttBiRNN, and HAN, to the task of accurately predicting cuisine from ingredients. Our results showed that all of the techniques gave a good performance, but HAN achieved the highest accuracy, at 87%.

Keywords: Text Classification, Deep Learning, Cuisine Prediction, HAN, TextAttBiRNN.

I. INTRODUCTION

The cuisine is a term that refers to the traditional foods, cooking styles, and cultural practices of a particular region. Recipes are sets of instructions for preparing a specific dish, and ingredients are the various food items or substances that are used in the recipe. The ingredients listed in a recipe are typically combined and cooked or prepared in a specific way to create the final dish. Different cuisines often have their unique ingredients and flavor profiles and may use specific cooking techniques or methods to prepare dishes [1].

There are many factors that can influence the flavor profile of a recipe, including the geographic location, climate, and cultural traditions of the region. For example, a recipe that includes a combination of cumin, coriander, and turmeric is likely to be of Indian origin, while a recipe featuring oregano, basil, and parsley is more likely to be of Italian or Greek origin.

Predicting the cuisine of a recipe based on its ingredients is a challenging task that requires a deep understanding of the flavors and techniques used in different cuisines around the world. It involves analyzing the unique combinations of spices, herbs, and other ingredients that are characteristic of specific regions and cultures and using this knowledge to accurately classify a recipe into one of many different categories.

In this study, we aim to develop a machine-learning model that can accurately predict the cuisine of a recipe based on its ingredients. To achieve this, we experiment with different classifier techniques, including TextRNN [2], TextCNN [3], fastText [4], TextAttBiRNN [5], and HAN [6]. We collect a dataset of recipes from a popular online recipe platform Yummly [7], pre-process the data, and then train the different classifier models that we mentioned.

Overall, recipe cuisine classification by ingredients is a useful tool for understanding the flavors and ingredients that are popular in different cuisines and can inform menu development and product development decisions in the food industry.

II. LITERATURE REVIEW

There has been limited research found which conducted on the use of ingredients in recipes as a way to classify and predict cuisine. This research has typically involved collecting a dataset of recipes and their corresponding cuisines and then using traditional machine-learning techniques to build a model that can accurately predict the cuisine based on the ingredients in the recipe.

In the paper [8], the authors demonstrate how tree-boosting algorithms can be used to accurately predict the cuisine of a dish based on its ingredients. Using a dataset of cuisine and their ingredients, the authors trained a tree-boosting algorithm and evaluated its performance using metrics such as accuracy. The results showed that the XGBoost was able to accurately identify the cuisine of a dish based on its ingredients, with an overall 80%

accuracy. This work demonstrates the potential of tree-boosting algorithms to be used in the food industry to understand the flavors and ingredients that are most popular in different cuisines.

In the paper [9], the authors investigate the use of support vector machines (SVMs) for predicting the cuisine of a dish based on the ingredients it contains. In this study, data is collected from Yummly [7], and then trained an SVM model on the training data, using the ingredients as the input features and the cuisine as the target variable. The authors found that the SVM model was able to accurately predict the cuisine of a dish based on its ingredients, with an overall accuracy of around 81%. The authors also compared the performance of the SVM model to other machine learning algorithms and found that it outperformed other methods, such as decision trees and k-nearest neighbors. This work highlights the potential of SVMs for predicting cuisine based on ingredients and demonstrates the importance of considering different machine-learning algorithms in this type of analysis.

In the paper [10], the authors analyze the performance of various classification algorithms for predicting the cuisine of a dish based on its ingredients. The authors collected a dataset of dishes and their ingredients and used this data to train and evaluate multiple classification models, including naïve Bayes, logistic regression, Random Forrest, and Linear SVC. The results showed that some models performed better than others, with linear SVC achieving the highest overall accuracy of around 79%. The authors also explored the impact of different preprocessing techniques and feature selection methods on the model's performance. This work provides a comprehensive analysis of the effectiveness of various classification algorithms for predicting cuisine based on ingredients and highlights the importance of carefully selecting and preprocessing the data to achieve the best results.

In the paper [11], the authors propose the use of association classification to automatically classify the cuisine of a recipe based on its ingredients. Association classification is a machine learning technique that identifies patterns and relationships between different items in a dataset. In this case, the authors used association classification to analyze the ingredients of a recipe and predict its cuisine. The authors also compared the performance of the association classification model to other machine learning techniques, including SVM and SVM+SVD400. This work demonstrates the potential of association classification as a tool for automating the process of classifying recipes based on their ingredients.

Most of the studies used traditional machine learning models, which have some limitations when it comes to predicting cuisine based on ingredients. Some of these limitations include:

1. Sensitivity to input data: Traditional machine learning models can be sensitive to the quality and structure of the input data and may produce poor results if the data is biased, incomplete, or poorly formatted.
2. Overfitting: Traditional machine learning models can sometimes "overfit" the training data, meaning they perform well on the training set but poorly on unseen data. This can be a problem when trying to make predictions about new dishes or cuisines.
3. Limited interpretability: Traditional machine learning models can be difficult to interpret, making it difficult to understand how the model arrived at its predictions. This can be a problem when trying to identify specific ingredients or combinations of ingredients that are indicative of a particular cuisine.
4. Lack of domain knowledge: Traditional machine learning models do not incorporate domain knowledge about the characteristics and flavors of different cuisines, which can limit their ability to make accurate predictions.

Overall, traditional machine learning models can be effective tools for predicting cuisine based on ingredients, but they may not always provide the most accurate or interpretable results.

III. DATA COLLECTION

We collected the cuisine-based ingredients dataset from Yummly [7], which is one of the most popular recipe-sharing websites. The Yummly website allows users to search for and share recipes from a variety of cuisines around the world.

We chose to collect the dataset from Yummly because it is a well-known and trusted source for recipe information, and it has a large database of recipes from various cuisines. This made it easy for us to gather a comprehensive list of ingredients used in different cuisines.

For our dataset, we focused specifically on ingredients used in different cuisines. This means that the dataset includes a list of ingredients that are commonly used in a particular type of cuisine, such as Italian or Chinese.

The dataset contains 39774 rows and 3 attributes which are id, cuisine, and Ingredients; below the table, you can see the information of these following attributes.

Table 1: Attribute Information of cuisine-based ingredients dataset

Attributes	Information
Id	The id attribute is a unique identifier for each recipe in the dataset. It is a numerical value that helps to distinguish one recipe from another.
Cuisine	Cuisine refers to the style of cooking and the types of foods that are prepared in a particular region or culture. This dataset has 20 unique cuisines, which could be Italian, Chinese, Mexican, etc.
Ingredients	The ingredients attribute is a list of ingredients that are used in the recipe. This includes items such as spices, herbs, and other food items that are necessary for preparing the dish.

IV. DATA PREPROCESSING

In the data preprocessing step, the ingredients attribute is first converted to lowercase to ensure that the model does not treat words with the same spelling but different cases as different words. This can be useful because some models may be sensitive to the case of the words and treat "Salt" and "salt" as two different words, leading to poor performance.

After that, we applied stemmed operation to reduce inflected words to their word stem, which helped reduce the number of unique words in the vocabulary and make the model more robust.

One-hot encoding is applied after stemming. The one_hot function from the tensorflow.keras.preprocessing.text [13] module used to perform this encoding. It takes in the text data and a vocabulary size, which is the maximum number of unique words to consider in the encoding. In this case, the vocabulary size is set to 5000. Then we applied the pad_sequences function from the tensorflow.keras.preprocessing.sequence [14] module to ensure that all the encoded sequences have the same length. This will be useful when we are training a deep learning model because as input sequences with different lengths can be difficult to process. The pad_sequences function takes in a list of sequences and the desired sequence length and pads or truncates the sequences to that length. In this case, the desired sequence length is set to 40.

The target attribute is label encoded using the label encoder function from scikit-learn to ensure that the model can understand and process the categorical data.

Finally, the dataset is split into a training set and a test set in an 80/20 ratio. The training set is used to train the model, while the test set is used to evaluate the model.

V. SYSTEM ARCHITECTURE

In order to correctly classify the cuisine from the ingredients, we experiment with different deep learning models that we explained below. And then, we compared the performance of these models on the task of classifying cuisine from ingredients. This process involves training each model on the training set, evaluating its performance on the test set, and comparing the results

fastText

In the process of predicting cuisine from ingredients using fastText, a look-up table is first used to convert bags of ngrams (sequences of n consecutive words) into word representations. These word representations are then averaged to create a text representation, which is a hidden variable that captures the meaning of the text. The text representation is fed to a linear classifier, which uses it to predict the probability of the text belonging to one of the predefined classes (in our case, the classes would be the different cuisines). The classifier uses the softmax function to compute the probability distribution over the classes, and the class with the highest probability is chosen as the prediction.

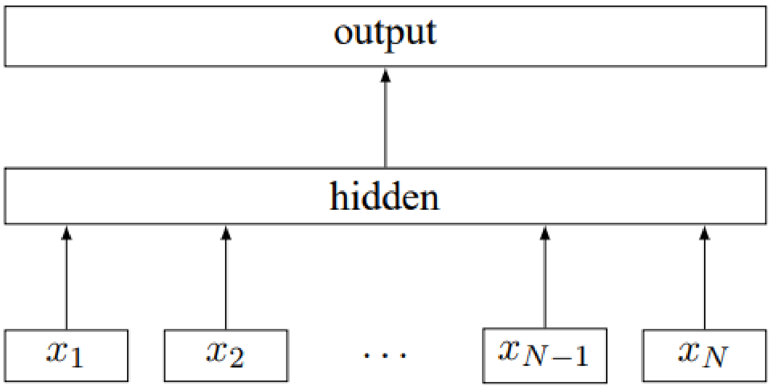

Figure 1: Model architecture of fastText for a sentence with N ngram features x1, . . . , xN . The features are embedded and averaged to form the hidden variable [15]

TextCNN

In the process of predicting cuisine from ingredients using Text Convolutional Neural Network (TextCNN), the first step is to represent the sentence with static and non-static channels. The static channel captures the meaning of the words in the sentence, while the non-static channel captures the context in which the words are used. Next, the representation is convolved with multiple filter widths and feature maps to extract features from the text. Max-over-time pooling is then used to reduce the dimensionality of the representation. Finally, a fully connected layer with dropout is used to make the prediction, and the softmax function is used to compute the probability distribution over the predefined classes (in our case, the classes would be the different cuisines).

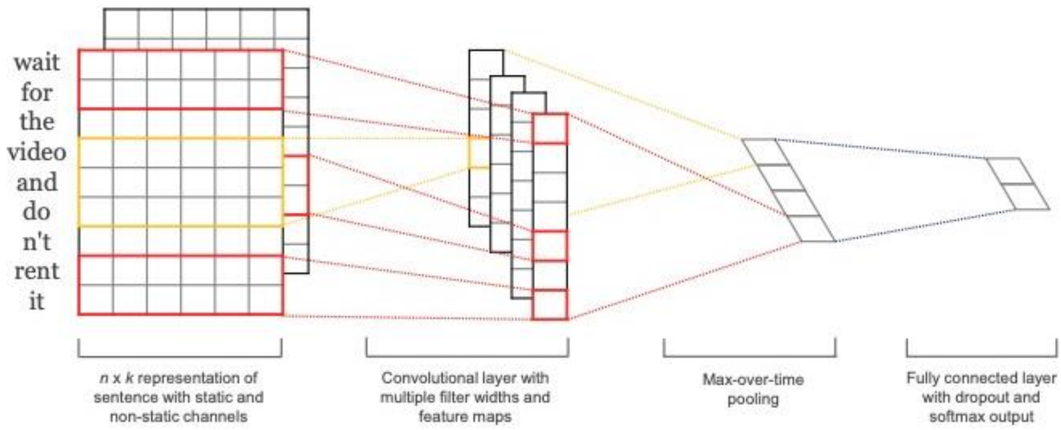

Figure 2: Model architecture with two channels for an example sentence [16]

TextRNN

In the process of predicting cuisine from ingredients using Text Recurrent Neural Network (TextRNN), the input text is first processed and transformed into a numerical representation, such as a sequence of word embeddings. This representation is then fed into the TextRNN model, which consists of one or more layers of recurrent cells. The recurrent cells process the input sequence one element at a time, using the hidden state from the previous element to inform the prediction for the current element. This allows the TextRNN to capture long-term dependencies and patterns in the data. After the input sequence has been processed, the final hidden state is used to make the prediction, which is typically done using a fully connected layer with a softmax output. The softmax function is used to compute the probability distribution over the predefined classes (in our case, the classes would be the different cuisines).

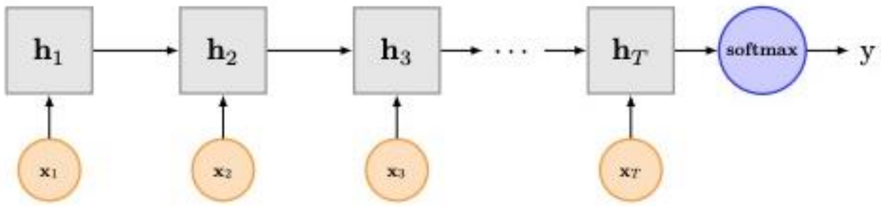

Figure 3: Recurrent Neural Network for Classification [17]

TextBiRNN

In the process of predicting cuisine from ingredients using Text bidirectional RNN, the first step is to represent the sentence as a sequence of word embeddings. These word embeddings capture the meaning of the words in the sentence and are learned from the data during training. The word embeddings are then fed into a bidirectional RNN, which processes the sequence in both forward and backward directions. This allows the model to capture the context of the words in the sentence and use it to make the prediction. The output of the bidirectional RNN is then fed into a fully connected layer with dropout, and the softmax function is used to compute the probability distribution over the predefined classes (in our case, the classes would be the different cuisines).

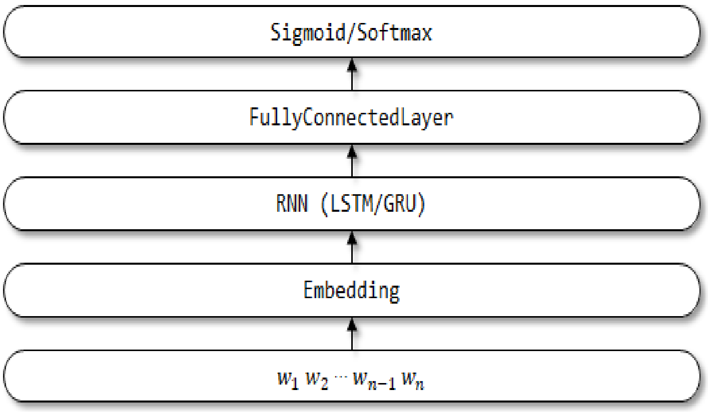

Figure 4: Network structure of TextBiRNN [18]

TextAttBiRNN

In the process of predicting cuisine from ingredients using Text Attention bidirectional RNN, the model first encodes the input text into a sequence of hidden states using a bi-directional RNN. The hidden states capture the contextual information of the words in the text. Next, an attention mechanism is applied to the hidden states to weight the importance of each word in the input text. The weighted hidden states are then passed through a fully connected layer to make the prediction.

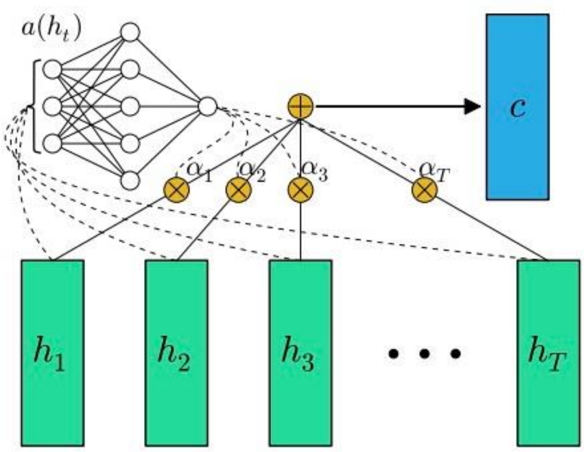

Figure 5: Network structure of TextAttBiRNN [19]

HAN

In the process of predicting cuisine from ingredients using a Hierarchical Attention Network(HAN), the model first encodes the ingredients into a fixed-length representation using word embeddings and a bi-directional GRU (Gated Recurrent Unit) layer. The GRU layer is a type of recurrent neural network that is used to process sequential data, such as text. The encoded ingredients are then passed through an attention layer, which weights the importance of each word in the ingredients based on its relevance to the task of predicting the cuisine. Finally, the weighted sum of the encoded ingredients is passed through a fully connected layer with a softmax output to predict the probability distribution over the predefined classes (in this case, the classes would be the different cuisines). The class with the highest probability is chosen as the prediction.

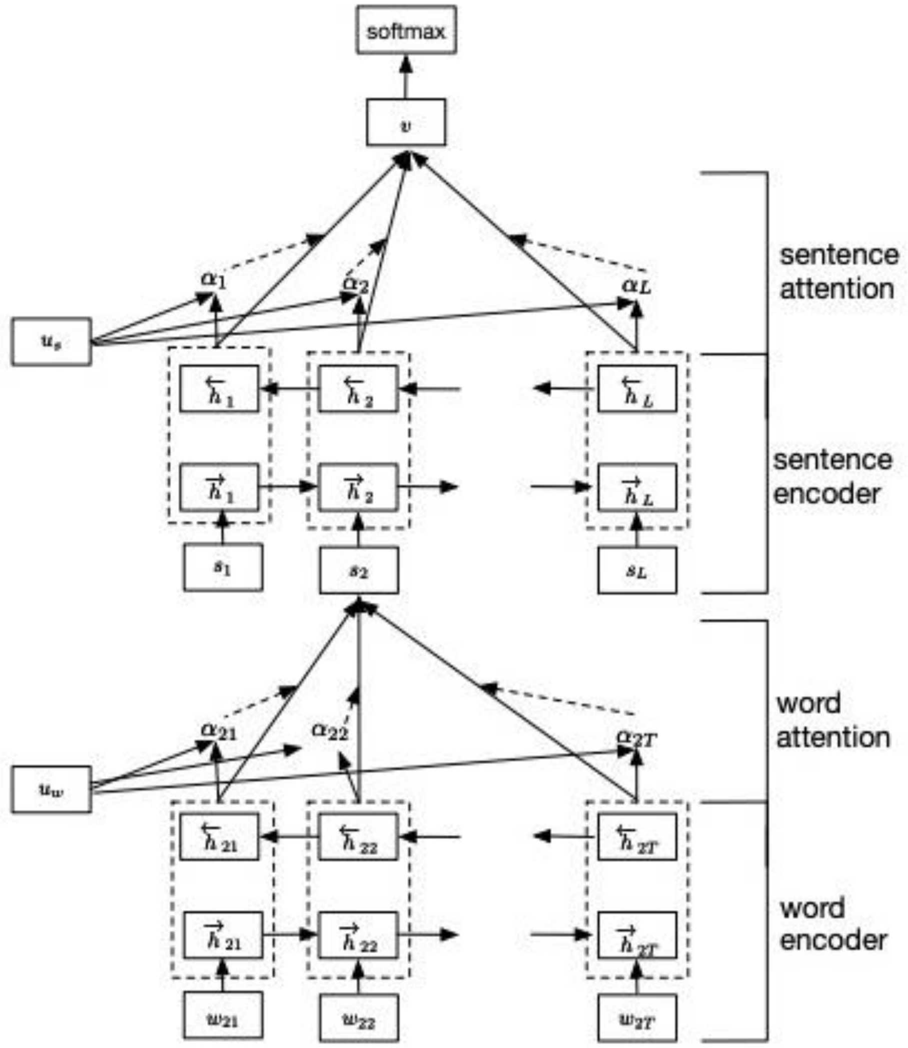

Figure 6: Hierarchical Attention Network [20]

VI. RESULT & DISCUSSION

Here is a discussion of the results of implementing six different techniques (fastText, TextCNN, TextRNN, TextBiRNN, TextAttBiRNN, and HAN) to predict cuisine from ingredients:

Overall, it appears that all of the techniques gave good results, as they were able to accurately classify the cuisine from the ingredients. However, HAN (Hierarchical Attention Network) seemed to perform the best, with an accuracy of 87%. This suggests that HAN is particularly well-suited for this task, possibly because it is able to capture the hierarchical structure of the ingredients and weight the importance of each word based on its relevance to the task of predicting the cuisine.

Below you can you the results:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:12/December-2022 Impact Factor- 6.752 www.irjmets.com

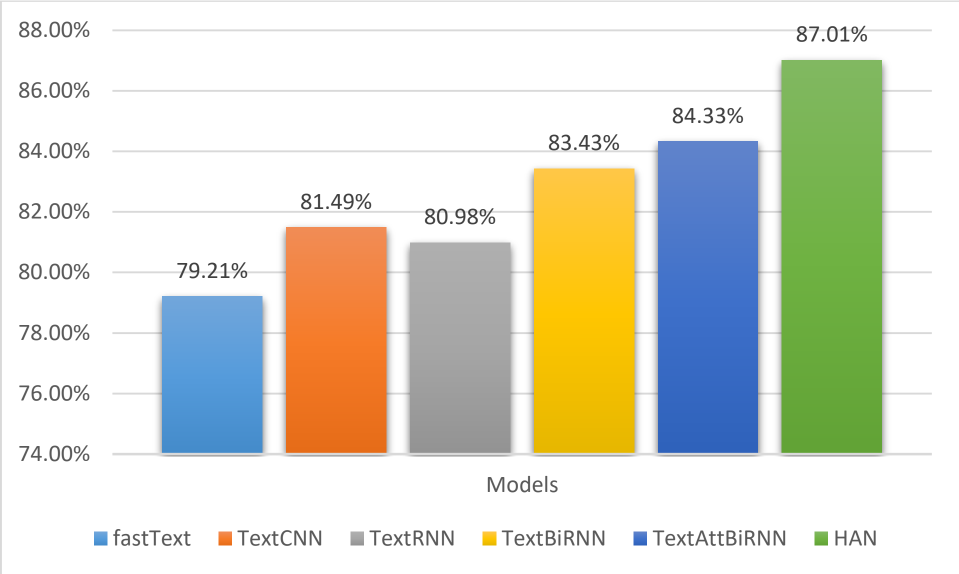

Figure 7: Experimental Results

VII. CONCLUSION

In conclusion, the experimental results demonstrate the effectiveness of deep learning models for natural language processing tasks, such as text classification, and highlight the importance of carefully evaluating the performance of different techniques on the specific dataset. And HAN proved to be the best model for accurately predicting cuisine from ingredients and may be a good choice for similar tasks.

VIII. FUTURE WORK

In future work, applying different data preprocessing techniques might lead to improved results.

IX. REFERENCES

[1] Zhang, M. M. (2011). Identifying the cuisine of a plate of food. Univ. of California at San Diego, La Jolla, CA, USA, Tech. Rep. CSE, 190.

[2] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923..

[3] Solovyeva, E., & Abdullah, A. (2021). Binary and Multiclass Text Classification by Means of Separable Convolutional Neural Network. Inventions, 6(4), 70.

[4] Mahoney, M. V. (2000, May). Fast Text Compression with Neural Networks. In FLAIRS conference (pp. 230-234).

[5] Du, C., & Huang, L. (2018). Text classification research with attention-based recurrent neural networks. International Journal of Computers Communications & Control, 13(1), 50-61.

[6] Pappas, N., & Popescu-Belis, A. (2017). Multilingual hierarchical attention networks for document classification. arXiv preprint arXiv:1707.00896.

[7] Malladi, Rajavardhan. "Recipe search engine using Yummly API." (2016).

[8] Kumar, R. R. V., Kumar, M. A., & Soman, K. P. (2016). Cuisine prediction based on ingredients using tree boosting algorithms. Indian Journal of Science and Technology, 9(45), 12.

[9] Ghewari, R., & Raiyani, S. (2015). Predicting cuisine from ingredients. University of California San Diego.

[10] Jayaraman, S., Choudhury, T., & Kumar, P. (2017, August). Analysis of classification models based on cuisine prediction using machine learning. In 2017 international conference on smart technologies for smart nation (SmartTechCon) (pp. 1485-1490). IEEE.

[11] Su, H., Lin, T. W., Li, C. T., Shan, M. K., & Chang, J. (2014, September). Automatic recipe cuisine classification by ingredients. In Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: adjunct publication (pp. 565-570).

[12] Okada, S., Ohzeki, M., & Taguchi, S. (2019). Efficient partition of integer optimization problems with

one-hot encoding. Scientific reports, 9(1), 1-12.

[13] Module:Tf.keras.preprocessing.text.tokenizer; tensorflow V2.11.0 (no date) TensorFlow. Available at: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer(Accessed: August 28, 2022).

[14] Module: Tf.keras.preprocessing.sequence; tensorflow V2.11.0 (no date) TensorFlow. Available at: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/sequence(Accessed:August28, 2022).

[15] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[16] Chen, Y. (2015). Convolutional neural network for sentence classification (Master's thesis, University of Waterloo).

[17] Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101.

[18] Li, Qiujia, Ness, Preben, Ragni, Anton, and Mark Gales. "Bi-Directional Lattice Recurrent Neural Networks for Confidence Estimation." arXiv, (2018). Accessed December 30, 2022. https://doi.org/10.48550/arXiv.1810.13024.

[19] Raffel, C., & Ellis, D. P. (2015). Feed-forward networks with attention can solve some long-term memory problems. arXiv preprint arXiv:1512.08756.

[20] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics.