International Journal of Information Technology (IJIT)
Volume 4, Issue 01, January-June 2023, pp. 18-30 Article ID: IJIT_4_01_003
Available online at https://iaeme.com/Home/issue/IJIT?Volume=4&Issue=1
Journal ID: 4573-3410, DOI: https://doi.org/10.17605/OSF.IO/GZAY2

CONTEXTUAL RECOMMENDATION SYSTEM FOR LOCAL BUSINESSES

Mayukh Maitra
Department of Computer Science, Stony Brook University,
New York, USA

Surabhi Sinha
Department of Computer Science, University of Southern California,
Los Angeles, USA

ABSTRACT

Going through several reviews could be laborious when this has to be done for multiple restaurants. One could instead read a graphical representation of what is great at the restaurant. Currently on Yelp, the food recommendations are only based on the total number of mentions of the food item in the reviews. Higher mentions, irrespective of the context, get an up-vote toward recommended items. Including context from reviews and tips could greatly improve the list of recommended items.

In this project, we combine Named Entity Recognition and Sentiment Analysis of reviews. Based on the sentiment of the reviews we aim to suggest the best dishes of a restaurant or the best restaurant offering a dish. We have leveraged various feature engineering methods to produce state-of-the-art results. We established that if chosen, the appropriate feature vectors can significantly improve the classification performance. Fine-tuning BERT and bi-directional LSTM are producing better results than the machine learning models and if trained for more epochs can eventually prove to be the best classifier models.

Keywords: Contextual Recommendation, Named Entity Recognition, BERT, LSTM, Count Vectors, TF-IDF

Cite this Article: Mayukh Maitra and Surabhi Sinha, Contextual Recommendation System for Local Businesses, *International Journal of Information Technology (IJIT)*, 4(1), 2023, pp. 18-30.
https://iaeme.com/Home/issue/IJIT?Volume=4&Issue=1

I. INTRODUCTION

Yelp provides consumers to read and share reviews on local businesses. It is however time-consuming to read through all the reviews on a restaurant, particularly if one were to review multiple restaurants. In contrast, an effective way to review is by reading a gist of the *good* and *bad* reviews of the restaurant. Currently, on Yelp, the food recommendations are based only on the total number of mentions of food items in the reviews. As a result, higher mentions, irrespective of the context, get an upvote toward recommended items.

Asuncion et al. proposed an NLP-based recommendation system that used collaborative filtering on ordinal rating data such as star ratings [1]. Huang et al. showed that analyzing insights from user reviews helps in improving users' visual search interface experiences [2]. Since Yelp's rating algorithm determines the overall rating without taking into account the reviewers' special characteristics such as people's diverse rating tendencies to give constant good or bad reviews, Yen et al. used machine learning techniques to bridge the gap between the personalized rating tendency and the overall rating statistics [3].

In this work, we were able to come up with a methodology of combining named entity recognition with sentiment analysis that deals with the shortcomings of the current systems. We have classified whether an item from a particular business is \textit{good} or \textit{bad} as per user reviews and generated recommendations taking into account the reviews and tips given by the user. Also, we have successfully employed various feature engineering methods which have significantly improved the classification of reviews. Binary classification of reviews into *good* or *bad* can reveal the benefits of extending to new sources of user preferences

The output of our model is a CSV report which contains the restaurants with their top 10 recommended dishes (sample Figure 1). This report can be leveraged both by businesses and users to improve their experience. Named entity recognition combined with sentiment analysis can significantly improve user preference analysis and improved recommendation targeting.

restaurant	top_rated_dishes
submerge-sandwich-los-angeles	[Pickles I Lemon Herb Chicken Sandwich I BTLA...
reggies-deli-and-cafe-los-angeles	[BLT Sandwich I Latte I Potato Salad I Cobb S...

Figure 1: Sample recommendation output

II. DATASET

Yelp is currently the most widely used restaurant and merchant information software across the United States. However, Yelp only provides users with aggregated metrics, such as overall ratings, and displays a limited number of reviews out of the entire set. For our project, we scraped Yelp data using the Fusion API provided by Yelp. The Yelp Fusion API allows us to get the best local content and user reviews from millions of businesses across 32 countries. In our scenario, we scraped data for the Los Angeles area and retrieved details for 419 restaurants. The dataset extracted comprised 3 individual datasets, each corresponding to restaurant information, menu items information, and review information respectively. Figure 2 gives us a high-level overview of the data statistics.

Dataset details	
Restaurant details	419 unique restaurants across LA
Menu items	30,942 menu items
Reviews	250,091 unique reviews

Figure 2: Dataset Statistics

From the above 3 files, we used the below fields for our analysis:

1. From the restaurants' information file, we have used the menu item, restaurant id, and restaurant name for our analysis. Mapping menu items and restaurants were performed using these files.
2. Similarly from the menu items file we used the menu item and restaurant id to recommend dishes and map restaurants with the menu items.
3. From the reviews file, we utilized the review text field for sentiment analysis and menu item extraction. We also utilized the rating field in this file for labeling purposes.

III. METHODOLOGY

In this project, we have combined named entity recognition along with sentiment analysis. Hence, we can broadly divide our methodology into two segments, one concerned with entity recognition and the other concerned with sentiment analysis. Figure 3 gives us a high-level overview of our workflow.

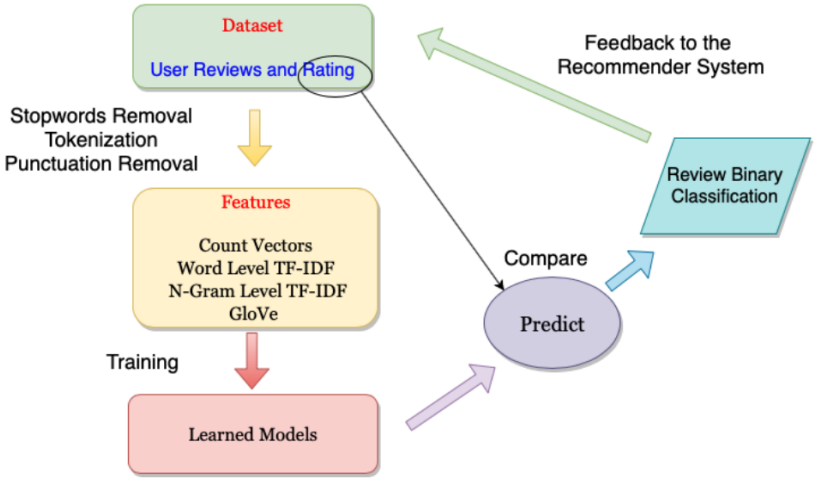

Figure 3: High-level methodology workflow

JJJ. NAMED ENTITY RECOGNITION

We have implemented a matching algorithm to extract menu items from review text. We performed direct matching as well as partial matching in order to extract menu items from reviews. Also, we restricted our algorithm to only retrieve at most two menu items from a single review text.

Before performing matching, we pruned the data by removing stop words, removing punctuation any non-English characters. We also stemmed and did lemmatization on the text.

For exact matching, we extracted the restaurant id for which the review was written, using which we extracted the menu items for that restaurant and then performed a direct look-up of those menu items in the review text to extract the menu item.

In order to perform partial matching we used the *fuzzywuzzy* package provided by Python. *FuzzyWuzzy* is a *Levenshtein* Distance based string-matching library. The Levenshtein distance between two strings a,b can be represented by $lev_{a,b}(|a|,|b|)$ where $lev_{a,b}(i,j)$ is achieved by the equation below:

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 4: Levenshtein distance function

Figure 5 below is a sample record after extracting the menu item from the review text.

menu_item	text
Lemon Herb Chicken Sandwich	The lemon herb chicken sandwich What is in it...
Tuna Sandwich	Such a great little gem Was in LA on a busine...

Figure 5: Named entity recognition sample

KKK. SENTIMENT ANALYSIS

In this work we have combined feature engineering with numerous machine learning and recurrent neural networks in order to classify the sentiment of reviews.

1. *Feature Engineering*
    a. For feature engineering we implemented the below mentioned strategies to transform the raw text into feature vectors or embeddings.
        i. Count Vectors
        ii. TF-IDF Vectors
            1. Word level
            2. N-Gram level
            3. Character level
        iii. Glove embeddings as features

Figure 6 depicts a conceptual model that is used for the implementation of GloVe.

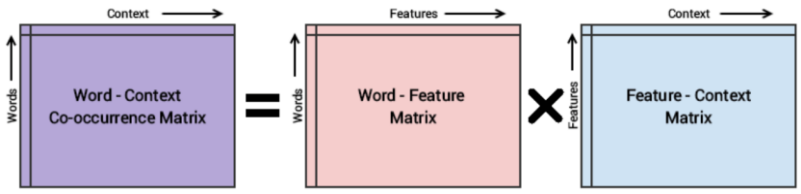

Figure 6: Conceptual model for the GloVe model's implementation

2. *Machine learning models* - For experimentation, we implemented three classical Machine learning algorithms namely, linear classifier based logistic regression, bagging concept based random forest, and XGBoost as a boosting model.
3. *Neural network models* - The first model we implemented was that of a **bi-directional LSTM**. As mentioned by (Zhou, et al.) [5] in their research, text classification significantly improved upon introducing Bidirectional LSTM with two-dimensional max pooling. In this paper [4], the authors discussed their work on combining RNN LSTM and GRU with complaint classification word embeddings for both single and bi-directions. We took inspiration from this to implement our Bidirectional LSTM model.

For the bidirectional LSTM, we have an embedding layer and instead of loading random weight, we have loaded the weights from our glove embeddings.

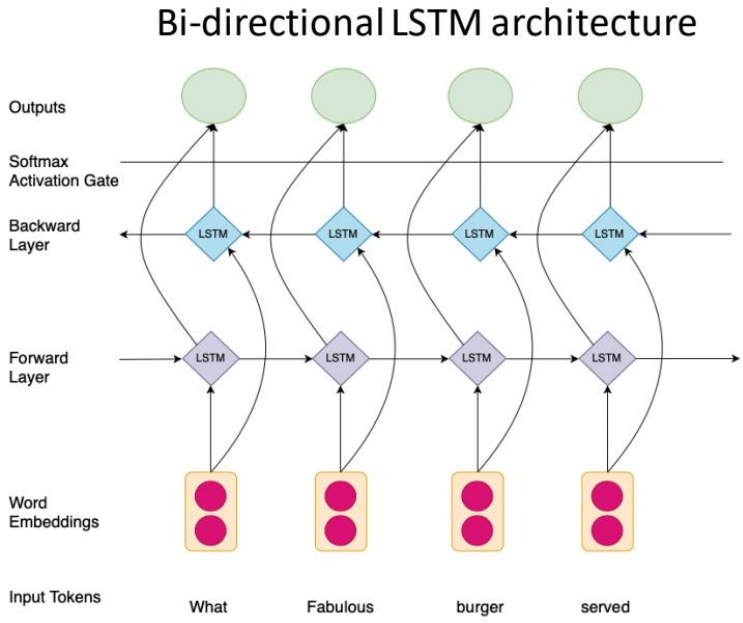

Figure 7: LSTM model architecture

The next model we implemented was that of **BERT**. The paper [6] has introduced a language representation model called Bidirectional Encoder Representation from Transformers, popularly known as BERT. This model introduced a novel concept that enables pre-training deep bidirectional representations from text which are not even labeled. It achieves so by joint conditioning on the left context as well as the right context in all the layers. This gives the BERT model the additional power of being fine-tuned. This has a wide range of applications like binary and multi-label text classification, question-answering models, and so on. According to the experiments performed by the authors of [6], they were able to achieve a GLUE score of 80.50\%.

For our scenario, we have even fine-tuned our BERT model. We plug in task-specific inputs and the corresponding outputs into BERT, following an end-to-end fine-tuning of parameters. During input, we can say that the first and second sentences are either paraphrased pairs or we can say them to be pairs of a hypothesis and its premise. They can also be part of QA and pairs of a question and passage or they can also be degenerate pairs.

For the output, these token embeddings or representations are processed through the output layer where various tasks such as sequence tagging or QA are performed. For further tasks such as sentiment analysis or classification, the CLS representation gets processed through the output layer. Fine-tuning is relatively inexpensive when compared to pre-training. Another benefit of BERT is that it can be used as an ELMo-style feature extractor for nearly as good results without fine-tuning.

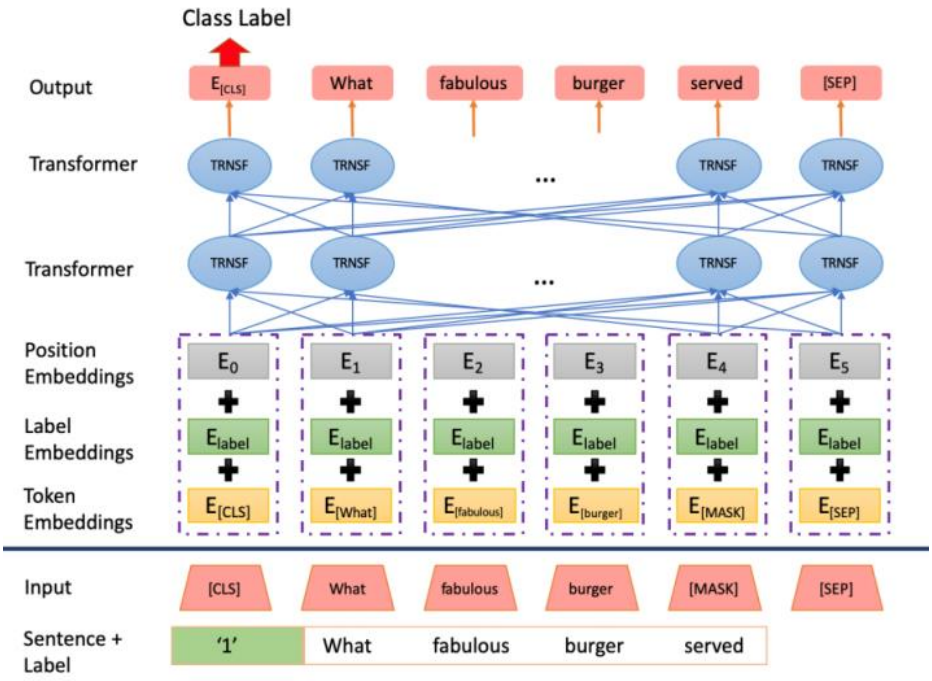

Figure 8: BERT model architecture

Figure 9 depicts our model setups for all our experiments mentioned above.

Training and testing set:
Data attributes: review_text, labels
Total reviews: 139183, Training set: 80% of total reviews
Testing set: 20% of total reviews

Training data format for classifiers:

- count vectorizer object for count vectors: analyzer='word', token_pattern=r'\w{1,}'
- word level tf-idf: analyzer='word', token_pattern=r'\w{1,}', max_features=5000
- N-gram level tf-idf: analyzer='word', token_pattern=r'\w{1,}', ngram_range=(2,3), max_features=100
- Char-level tf-idf: analyzer='char', token_pattern=r'\w{1,}', ngram_range=(2,3), max_features=100

BERT setup:
Bert pre-trained model : 'bert-base-cased'
The maximum total input sequence length after WordPiece tokenization: 128
training batch size = 24, evaluation batch size = 8
learning rate = 2e-5, epochs = 1
output mode = 'classification'

Bi-directional LSTM setup:
Maximum number of words = 1000
maximum sequence length = 100
Embedding dimension = 100
dropout=0.1, recurrent_dropout = 0.1
loss='binary_crossentropy', optimizer='adam'
Activation: relu and sigmoid
epochs=2, batch_size=128

Figure 9: Experiment setup details

4. *Data labeling* - Due to a large amount of data, manual annotation of sentiment for the reviews was not feasible. Hence, we used the review rating field in the reviews data file to label our data. The rating ranged from 1-5. For all reviews with a rating of below 3, we labeled them as 0 or negative sentiment, and all reviews with a rating of 3 or more were labeled as 1 or positive sentiment.

IV. RESULTS

Below are the results for each scenario. Table 1 depicts the accuracy performance of the various machine learning models with respect to the different feature engineering methods. Logistic regression on count vector features performed the best in terms of accuracy.

Table 2 depicts the f1 score performance for the various machine learning models with respect to the different feature engineering methods. Random forest is giving the most consistent and best f1 score.

	XGBoost	Random forest	Logistic Reg.
Count Vectors	94.1%	96.05%	**97.4%**
Word Level	93.4%	96.13%	93.5%
Ngram Level	93.2%	96.04%	93.3%
Character level	93.3%	96.06%	93.2%

TABLE I
ACCURACY PERFORMANCE FOR MODELS USING DIFFERENT FEATURES

	XGBoost	Random forest	Logistic Reg.
Count Vectors	0.96%	**0.98%**	0.98
Word Level	0.96%	0.98%	0.96%
Ngram Level	0.96%	0.98%	0.96%
Character level	0.96%	0.98%	0.96%

TABLE II
F1 SCORE FOR MODELS USING DIFFERENT FEATURES

Table 3 depicts the ROC score performance for the various machine learning models with respect to the different feature engineering methods. The closer the ROC curve is to the upper left corner or the higher the score, the higher the overall accuracy of the test. Random forest is giving the best performance on the ROC score metric.

Table 4 depicts the mean squared error performance for the various machine learning models with respect to the different feature engineering methods. Logistic regression is giving the lowest error for count vector features.

	XGBoost	Random forest	Logistic Reg.
Count Vectors	0.90%	**0.97%**	0.91
Word Level	0.92%	0.95%	0.76%
Ngram Level	0.88%	0.94%	0.80%
Character level	0.82%	0.94%	0.71%

TABLE III
ROC SCORE FOR MODELS USING DIFFERENT FEATURES

	XGBoost	Random forest	Logistic Reg.
Count Vectors	0.058%	0.034%	**0.025**
Word Level	0.065%	0.035%	0.064%
Ngram Level	0.067%	0.037%	0.066%
Character level	0.066%	0.036%	0.067%

TABLE IV
MSE SCORE FOR MODELS USING DIFFERENT FEATURES

We evaluate the calibration performance in terms of the Brier score, reported in the legend of figures 10, 11, 12, and 13 above wherein the lower the Brier score is, the better it is considered to be. One can observe here that logistic regression is well calibrated on count vector features while random forest and XGBoost perform badly.

However, Random forest is better calibrated on the rest of the features. However, redundant features lead to poorer Brier scores as they violate the assumption of feature independence resulting in an overfitted classifier, which is indicated by the typical transposed-sigmoid curve.

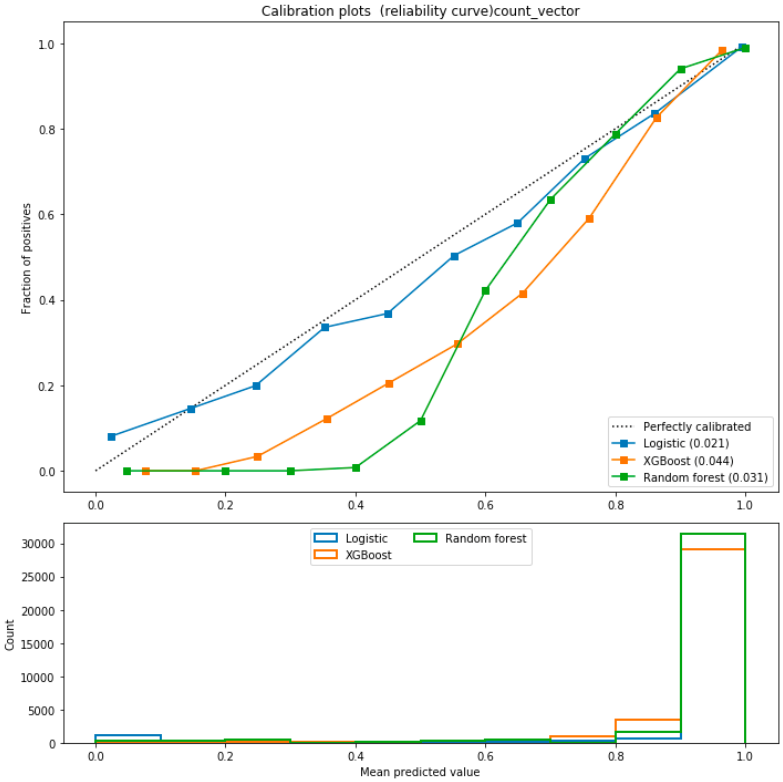

Figure 10: Calibration plot for count vector features

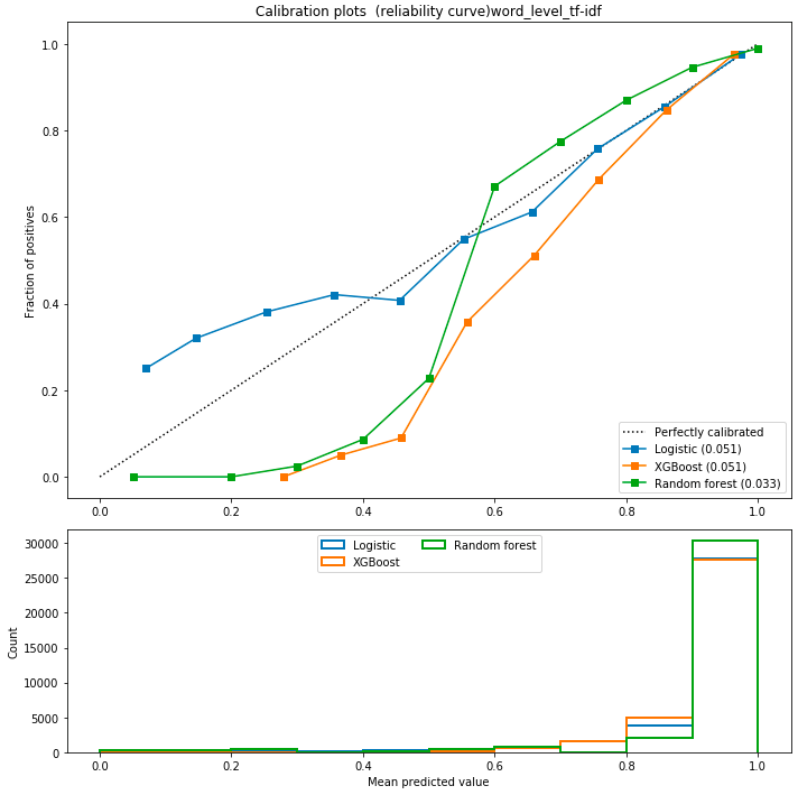

Figure 11: Calibration plot for word level features

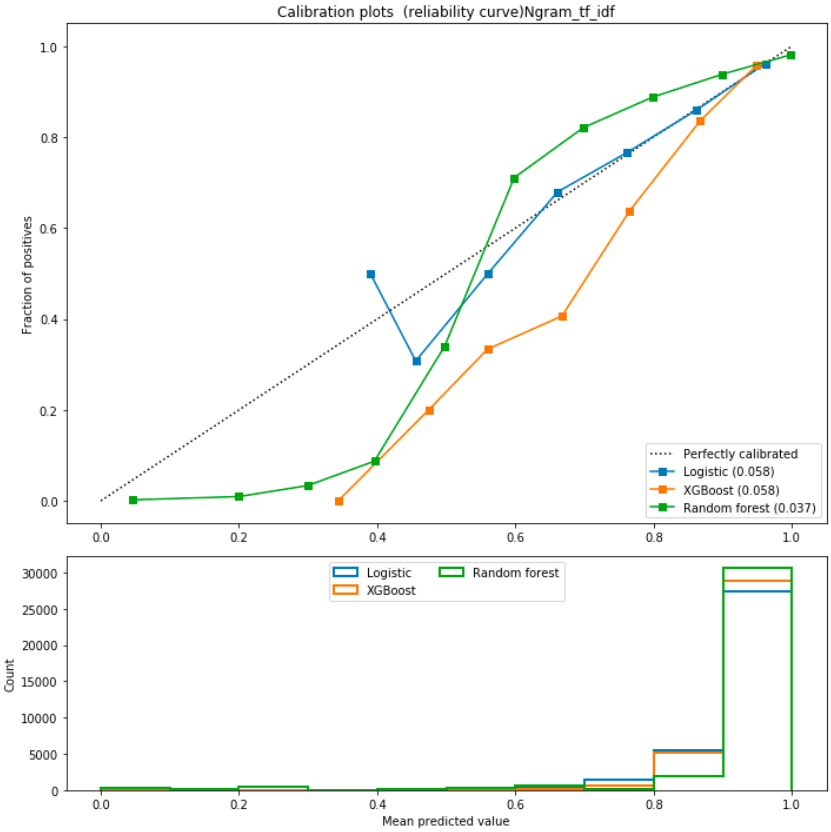

Figure 12: Calibration plot for ngram level features

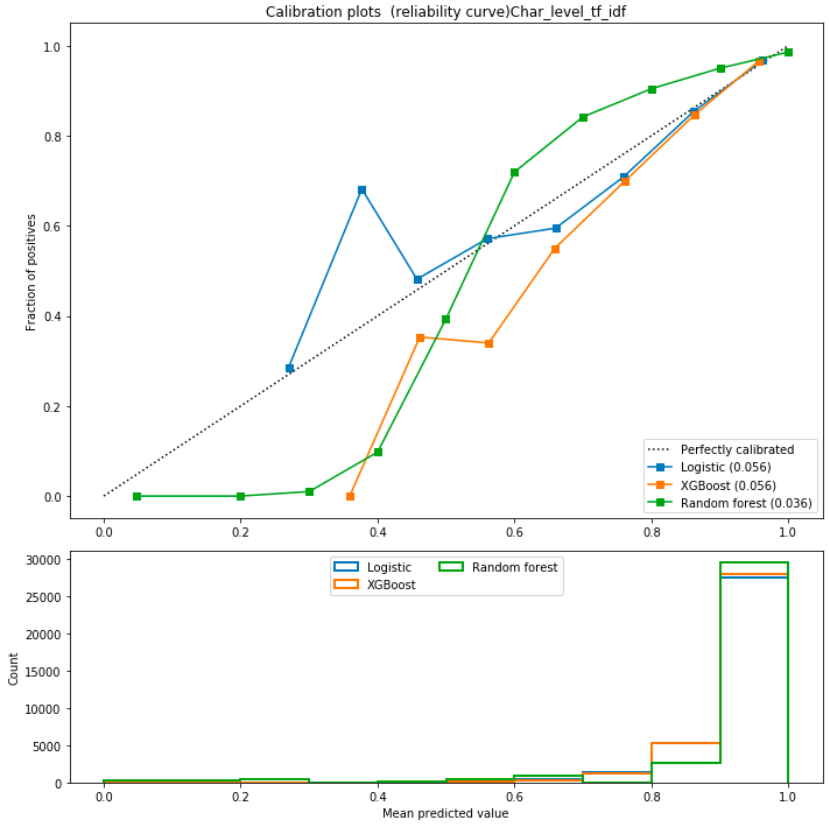

Figure 13: Calibration plot for char level features

The LSTM model was trained for 10 epochs and performed very well with a high accuracy of 98.5% and very low mean squared error of 0.014.

Accuracy	F1-score	ROC score	MSE score
98.5%	0.99	0.96	0.014

TABLE V
EVALUATION PERFORMANCE FOR LSTM

The BERT model was trained for 1 epoch and performed reasonably well with a high accuracy of 96.7% and a very low evaluation loss of 0.088. Also, the Matthews correlation coefficient (MCC score) is closer to 1 implying that it is a pretty good binary classifier.

The BERT if trained for more epochs, can definitely give a much improved performance.

Accuracy	F1-score	MCC score	Eval loss
96.7%	0.98	0.088	0.72

TABLE VI
EVALUATION PERFORMANCE FOR BERT

JJJ. CASES WHERE LSTM FAILED

1. **Example 1**: *The place is nice very chic and great atmosphere*. The server was pleasant and overall good customer service The reason for *Not rating 5 stars* was the food It was okay *nothing to rave about* I had a tomato bisque soup that just tastes like a can of tomato the menu *not too flexible* for those that have dietary restrictions too much pork on the menu and I am allergic to pork and beef I ate there late night so maybe the menu was simplified with limited options But I will say go check it out is a really nice place and I will go back to hang out but *not to eat*.
   In the example above, the user rated the restaurant a 4. However, his review had mixed reactions. The user made positive comments about the atmosphere, however, he made negative reviews about the food. There were a lot of *negation* words in the review, which prompted the LSTM to predict this review as *negative* whereas originally it had a positive sentiment, given the rating was 4.
2. **Example 2**: Decided to try this place out because of how *popular* it has become on Instagram I didn't see what all the hype was about If you are looking for your typical asado al pastor chicken tacos then *do not come to guisados*. The steak picado merely has any flavor chicharron was fat served in a tortilla and the chorizo was drenched in oil The only *good thing* about this place was theri tortillas Tasted like handmade Other than that Would not recommend this place to anyone Similar to example 1, this review also had mixed sentiment. Although the general tone was of disappointment, it also had positive comments. This mixed sentiment prompted the LSTM model to predict this as positive, although it was labeled as a negative sentiment.
   We think, introducing negative sampling and equally balanced training, along with training the model for more epochs can help reduce these mistakes in classification.

KKK. RECOMMENDATIONS

We used the LSTM model to recommend dishes as it performed the best out of all the models. We used the sentiment of the reviews along with the menu items extracted during named entity recognition to generate a CSV report containing the top 10 favored dishes by restaurants. Figure 14 below shows the top 10 dishes from the most favored restaurant.

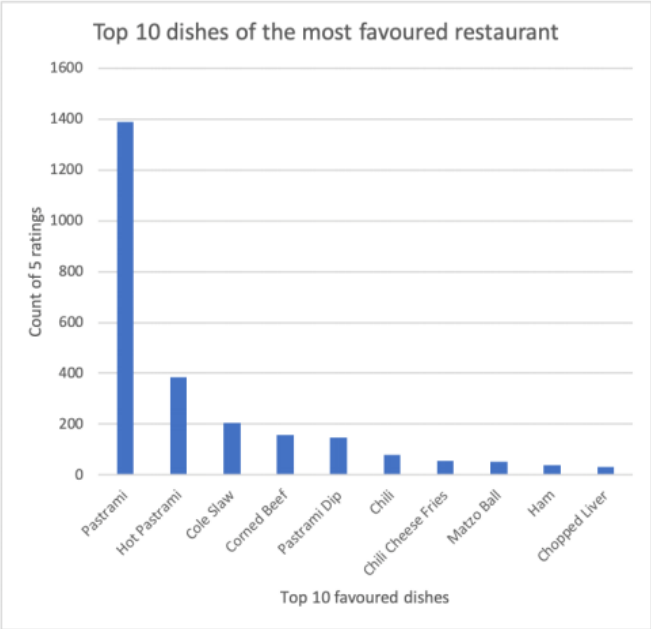

Figure 14: Top 10 dishes by restaurant

In order to generate the recommendations for the most favored dishes, the dishes were scored as (total number of positive reviews \- total number of negative reviews). The dishes with the highest scores were recommended for that restaurant. Similarly to get the most favored restaurants, we summed up the scores for each dish of that restaurant and recommended the restaurants having the maximum scores. Figure 15 below is a word cloud for the top 20 restaurants.

Figure 15: Top 20 restaurants

Figure 16 shows the top 10 restaurants having the maximum number of positively reviewed food items.

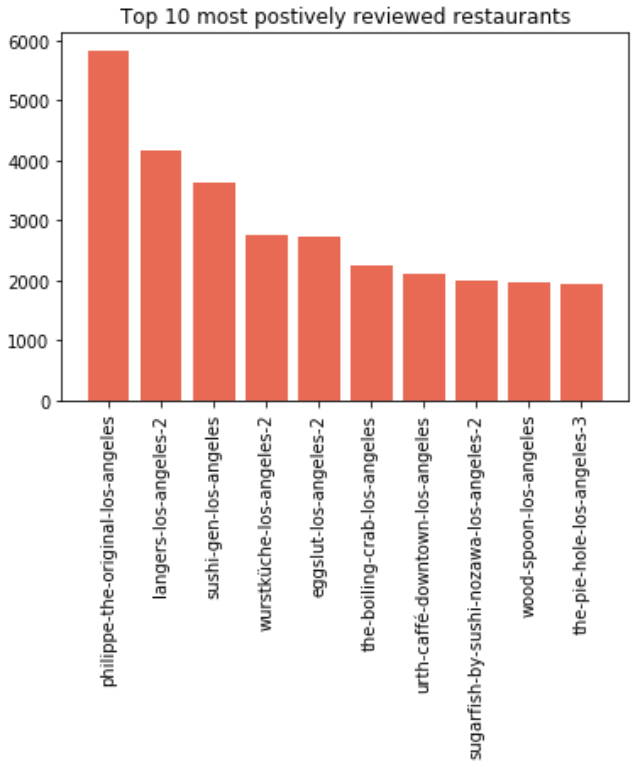

Figure 16: Top 10 restaurants with the most positively reviewed food items

V. CHALLENGES

While working on this problem we did encounter some challenges.

1. Data labeling: currently we have labeled the data based on ratings received, however, human-judged annotation would serve as a better labeling method.
2. Matching algorithm to perform named-entity recognition takes ~6 hours to run. This needs to be improved upon to deal with the larger datasets.
3. Dataset suffers from asymmetrical distribution for positive and negative reviews, with the positive ones significantly overpowering the latter. Would need to perform negative sampling for better results in case we don't retrieve a more symmetrical dataset.

VI. CONCLUSION

In this work we are able to establish some important findings such as performing feature engineering proved to improve the performance of binary classifiers. Also, our ideology of combining named-entity recognition and sentiment analysis can help Yelp improve its existing recommendation system significantly as it will be entirely context based.

We observed that LSTM is currently outperforming all other models and is also producing state-of-the-art results. With the help of entity recognition and combining it with sentiment analysis we're successfully able to identify the most preferred restaurants, most preferred dishes along with the most preferred dishes by the restaurant.

If chosen, the appropriate feature vectors can significantly improve the classification performance. One important aspect to be noted is the performance of BERT. Fine-tuning BERT can produce better results than recurrent neural networks and if trained for more epochs can eventually prove to be the best classifier model.

Thus, we can say that named-entity recognition combined with sentiment analysis can significantly improve user preference analysis and improved recommendation targeting.

Improving sentiment analysis is still a problem to be perfected and future experiments with ELMo embeddings can prove to perform better than Glove.

REFERENCES

[1] Kouvaris, Peter; Pirogova, Ekaterina; Sanadhya, Hari; Asuncion, Albert; and Rajagopal, Arun (2018) "Text Enhanced Recommendation System Model Based on Yelp Reviews," SMU Data Science Review: Vol. 1: No. 3, Article 8

[2] Jeff Huang, Oren Etzioni, Luke Zettlemoyer, Kevin Clark, and Christian Lee. 2012. Revminer: An extractive interface for navigating reviews on a smartphone. In Proceedings of UIST 2012

[3] Q. Yan, J. Ji, and H. Li, “Rating the Raters: Bias Analysis on Yelp Reviews for Improved Star Rating System”, Project report 334, Stanford University, 2015

[4] Text Classification Improved by Integrating Bidirectional {LSTM} with Two-dimensional Max Pooling, Zhou, Peng, Qi, Zhenyu, Zheng, Suncong, Xu Jiaming, Bao Hongyun and Xu Bo, Proceedings of {COLING} 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016

[5] Recurrent Neural Network with Word Embedding for Complaint Classification, Panuwat Assawinjaipetch, Virach Sornlertlamvanich, Kiyoaki Shirai, Sanparith Marukata, 2016

[6] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova; arXiv:1810.04805v2 [cs.CL] 24 May 2019

Citation: Mayukh Maitra and Surabhi Sinha, Contextual Recommendation System for Local Businesses, International Journal of Information Technology (IJIT), 4(1), 2023, pp. 18-30

https://doi.org/10.17605/OSF.IO/GZAY2

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJIT/VOLUME_4_ISSUE_1/IJIT_4_01_003.pdf

editor@iaeme.com