TABLE OF CONTENTS

1) Text Category Classification 03

1.1) Dataset Description 03

1.1.1) Dataset Information and Structure 03

1.1.2) Data Processing Pipeline 03- 04

1.2) Text Classification Algorithm 04

1.2.1) Bert Models Information Table .... 04 - 05

1.3) Sentences Categories Classification 06

1.3.1) Dataset Consideration 06

1.3.2) Data crucial Issues and Concerns 07

1.4) Fine Tuning Bert for Multi Class Sentence Classification 08

1.4.1) Bert-Base-cased 09

1.4.2) Bert-large-cased 10

1.4.3) Roberta-Base 11

1.4.4) Roberta-Base 12

1.4.5) Albert-Base-cased 13

1.4.6) Albert-large-cased 14

1.4.7) Conclusion 15

1.5) Data Balancing Techniques 16

1.5.1) Stratification 16

1.5.2) Ensemble Techniques 17

1.5.2.1) Under Sampling .... 17 - 19

1.5.2.2) Over Sampling .... 20 - 22

1.5.2.3) Class Weight Balancing .... 23 - 25

1.5.2.4) Smote .... 26 - 30

1.5.3) DL based Data Enhancement 31
1.5.3.1) Data Augmentation .... 31 - 37
1.5.3.2) Data Enriching using CHATGPT... 38 - 40
1.6) Conclusion 41

1.7) Data Balancing Techniques Comparison and Remarks ....42 – 44

1.8) Sentence Aspects Classification 45
1.8.1) Data Consideration ....45 – 47
1.8.1.1) Data Visualization 47– 48
1.8.2) Out Strategy 49
1.8.3) Stratification Problem 50
1.8.3.1) Visualization after Stratification ....51 – 52
1.8.4) Fine Tuning Bert for Aspects on Original Dataset ....52 – 54
1.8.5) Proposed Techniques ....56 – 57
1.8.6) Data Enhancement Using ML Models 58
1.8.6.1) Limitations and Concerns ....59 – 60
1.8.6.2) Visualization of Enhanced Dataset ....61 – 64
1.8.6.3) Fine Tuning on Combined Dataset ....65 – 66
1.8.6.4) Results and Conclusion ....66 – 67
1.8.6.5) Accuracy Comparison 67
1.8.7) Customized Data Revision Tool 69

1.9) Merging Aspects and Categories Model 70
1.9.1) Independent Training 70-71
1.9.2) Interdependent Training 71- 81

1.10) Future Work 82

1.11) References.. 83

TEXT CLASSIFICATION ML MODELS RESEARCH REPORT

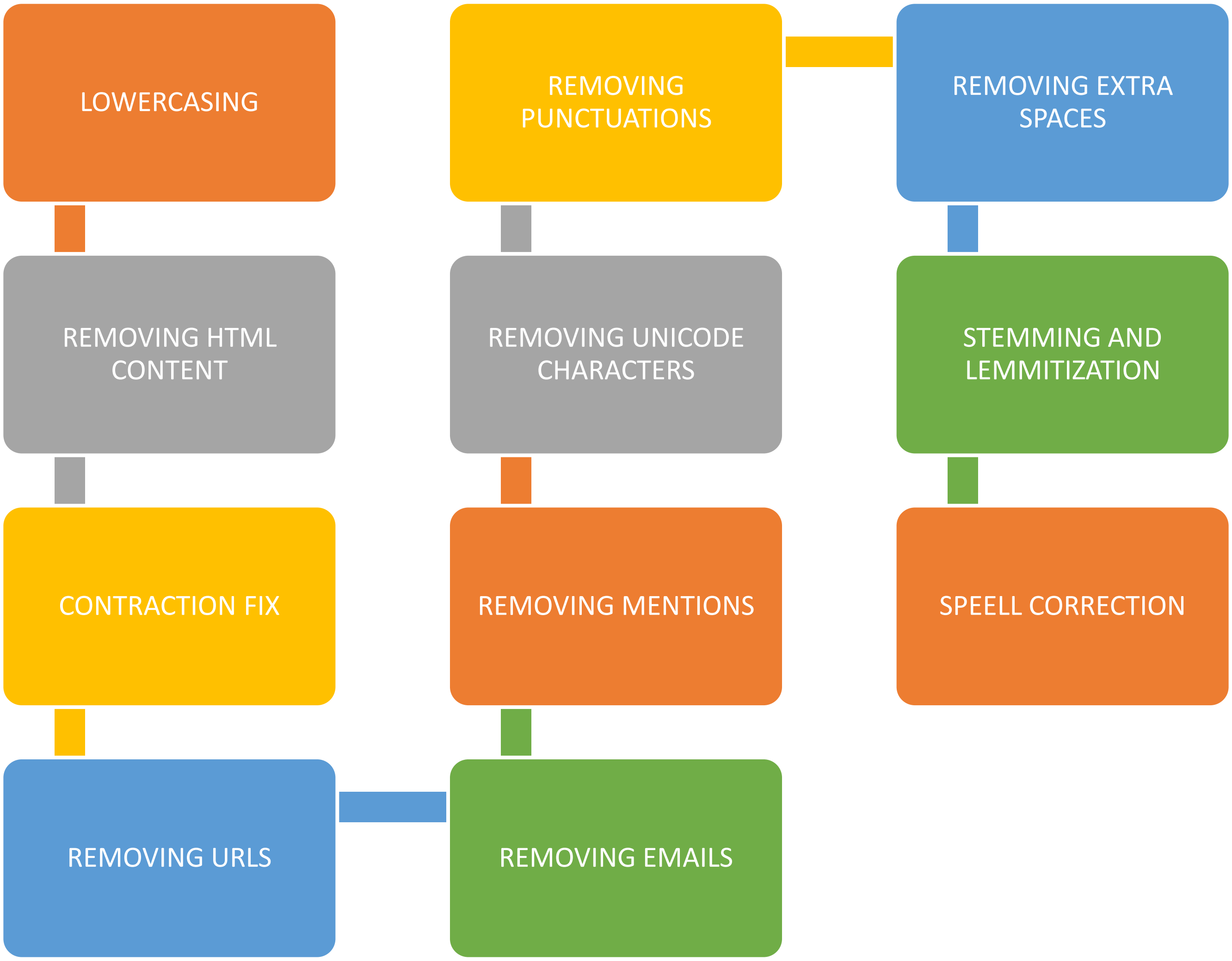
DATA SET

Name	Type	Tables	Entries	Remarks
Annotated dataset	SQLite	Reviews Sentences Aspects	256 in Reviews 1022 in Sentences 1509 in Aspects	Data is annotated manually using Custom Annotation Tool by Analyst.

TABLE COLUMNS

Table	Columns	Remarks
Reviews	Id, Review	Scrapped from the Internet in the form of paragraphs
Sentences	Id, Sentence, Review Id	Sentence based Tokenized Statements that every review may consist of multiple sentences
Aspects	Id, Entity, Start, End, Category, Aspect, Sentiment, Opinion Term, Sentence Id	Every sentence must belongs to some category and may have more then one aspects that are specified through any opinion term and can be sentimentally classified into positive neutral and negative.

DATA PROCESSING PIPELINE



TEXT CLASSIFICATION ALGORITHM

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep learning language model developed by Google. It is widely considered to be one of the most advanced models for natural language processing tasks such as text classification, question answering, and language generation.

BERT is a transformer-based architecture that is trained on a large corpus of text data, which enables it to capture context-dependent relationships between words in a sentence. Unlike traditional language models that use only a left-to-right or right-to-left approach, BERT uses a bidirectional approach, which allows it to consider the context of words in both directions. This makes BERT better equipped to understand the meaning of words in context and perform more accurately on tasks like sentiment analysis or named entity recognition.

CONSIDERED BERT CLASSIFIERS

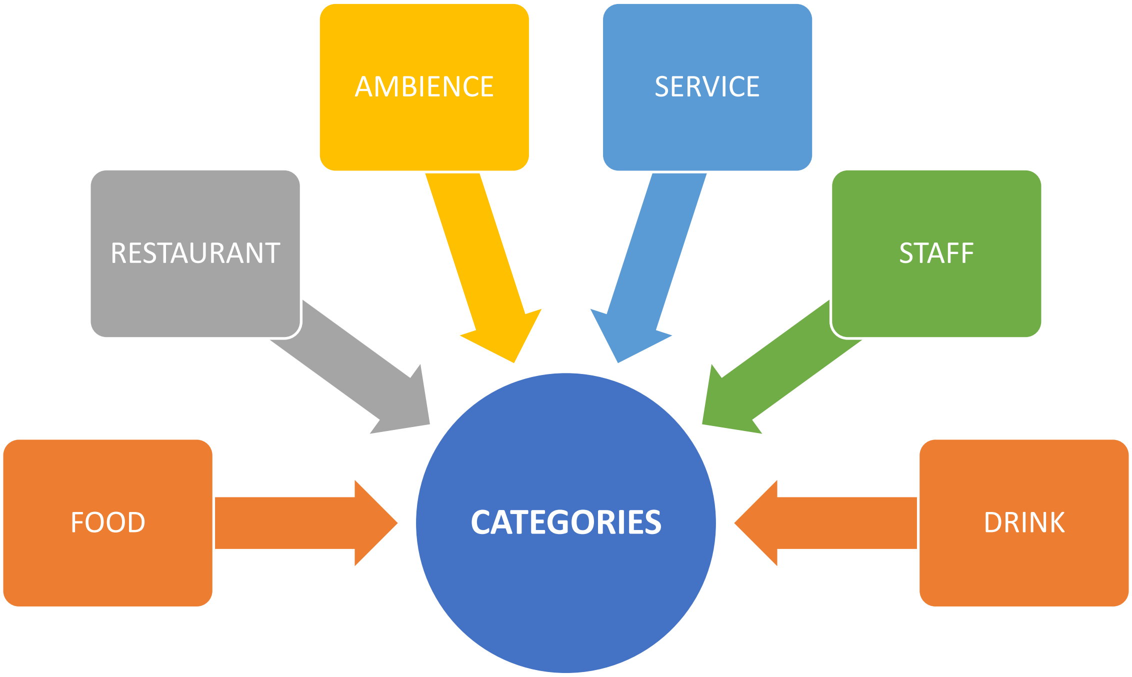
BERT Model	Accuracy	Pretrained Dataset	Pretraining Time	Hidden Units
1) BERT-Base	94.9	Wikipedia + Books Corpus	4 Days	768
2) BERT-Large	96.0	Wikipedia + Books Corpus	4 Days	1024
3) BERT-Base (Multilingual)	93.2	Wikipedia (104 Languages)	4 Days	768
4) Roberta-Base	95.5	Web Text	1 Month	768
5) Roberta-Large	96.5	Web Text	1 Month	1024
6) ALBERT-Base	95.0	N/A	N/A	128
7) ALBERT-Large	96.0	N/A	N/A	512

SENTENCE CATEGORIES CLASSIFICATION

CLASSIFICATION TYPE

Multiple Class Classification

OUTPUT CLASSES



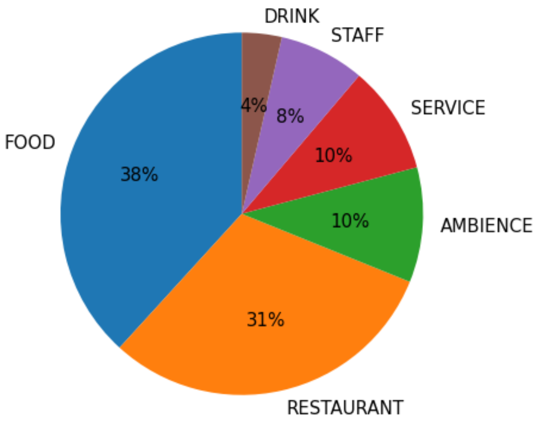
DATA CONSIDERATION

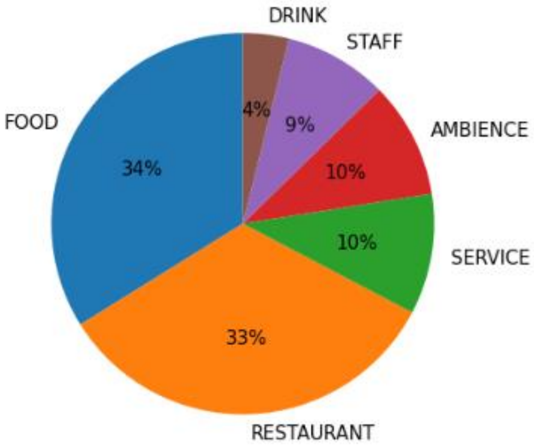
Sentences and Aspects Tables are inner joined to enlist the sentences with its corresponding categories.

Tables considered are Sentences with joined Aspects.

Columns considered are Sentence ID, Sentence, Category

DATA AGAINST CATEGORIES

<table>
<tr><th>Given Dataset</th><th>After Removing Duplication</th></tr>
<tr>
<td>
<table>
<tr><th>Category</th><th>Data Entries</th></tr>
<tr><td>Food</td><td>570</td></tr>
<tr><td>Restaurant</td><td>457</td></tr>
<tr><td>Ambience</td><td>153</td></tr>
<tr><td>Service</td><td>144</td></tr>
<tr><td>Staff</td><td>114</td></tr>
<tr><td>Drink</td><td>53</td></tr>
</table>


</td>
<td>
<table>
<tr><th>Category</th><th>Data Entries</th></tr>
<tr><td>Food</td><td>437</td></tr>
<tr><td>Restaurant</td><td>431</td></tr>
<tr><td>Ambience</td><td>132</td></tr>
<tr><td>Service</td><td>127</td></tr>
<tr><td>Staff</td><td>113</td></tr>
<tr><td>Drink</td><td>50</td></tr>
</table>


</td>
</tr>
</table>

ISSUES AND CONCERNS

- Imbalanced Data against categories
- Lack of Quality Data
- Lack of richness
- Out of context entries
- Incorrect Entries
- Lack of Quantity

FINE TUNING BERT FOR CLASSIFICATION (MULTI CLASS CLASSIFICATION) BEFORE DATA BALANCING TECHNIQUES

CONFIGURATION

- Batch size=12 and 01
- Learning Rate = 1e-5
- Epochs=15
- Warm up steps=0
- No of batches= data Samples / batch size = 1096/12 = 91.5 >> 92 for batch size 12 and 1096 batches for batch size 01
- Training Steps= length (data Loader for Training Samples) *epochs= 92*15=1380

BERT-BASE-UNCASED

CODE LINK

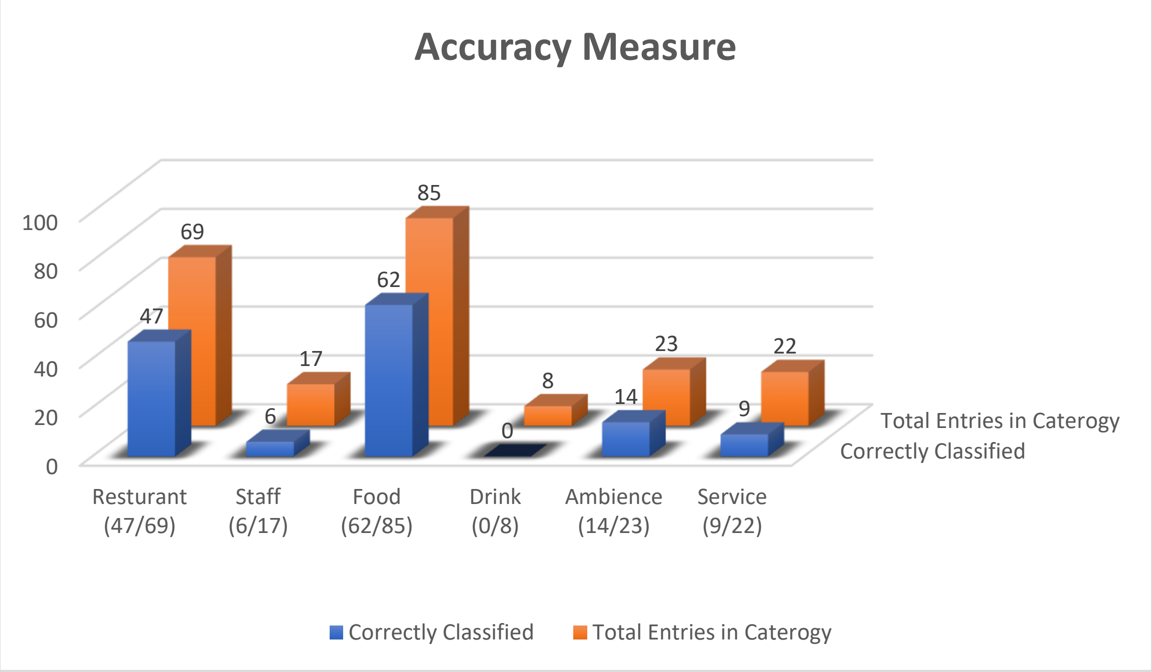
https://colab.research.google.com/drive/1TvEShOTlNWLYEob6aA7HTa6fhQQJSieC?usp=share_link

RESULTS

- BATCH SIZE: 12
    - After Epoch 15 (Maximum Accuracy Acquired) at Epoch 4
    - Training loss: 0.4369886934335502
    - Validation loss: 1.458361318236903
    - F1 Score (Weighted): 0.5963481988670429

- BATCH SIZE: 01
    - After Epoch 15 (Maximum Accuracy Acquired) at Epoch 3
    - Training loss: 1.0521408873451241
    - Validation loss: 2.121864684272753
    - F1 Score (Weighted): 0.6131897081390338

ACCURACY PER CLASS



BERT-LARGE-UNCASED

CODE LINK

https://colab.research.google.com/drive/1QjeiUZqmtOM3PjSCy162kEyz2xWcAYea?usp=share_link

RESULTS

BATCH SIZE: 01

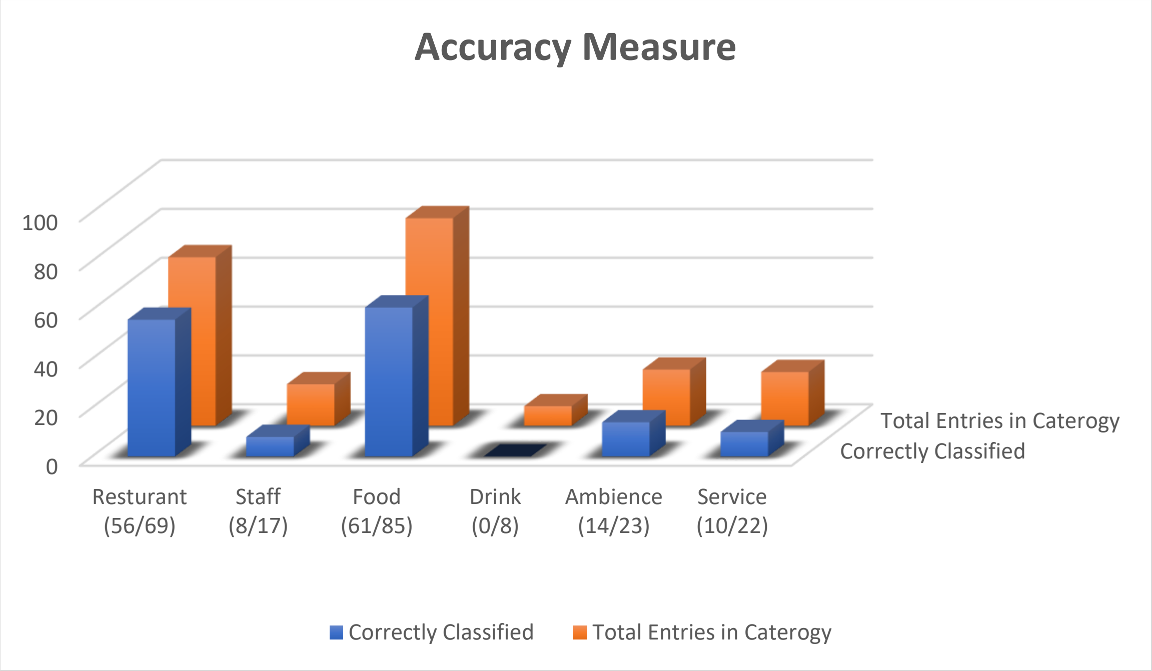
After Epoch 15 (Maximum Accuracy Acquired) at Epoch 02

Training loss: 1.6033821348233213

Validation loss: 1.553178131846445

F1 Score (Weighted): 0.240646605793634 56

ACCURACY PER CLASS



ROBERTA-BASE

CODE LINK

https://colab.research.google.com/drive/1WOsd1bS9yz4hiaaryvu7H8588IQWJAw9?usp=share_link

RESULTS

BATCH SIZE: 01

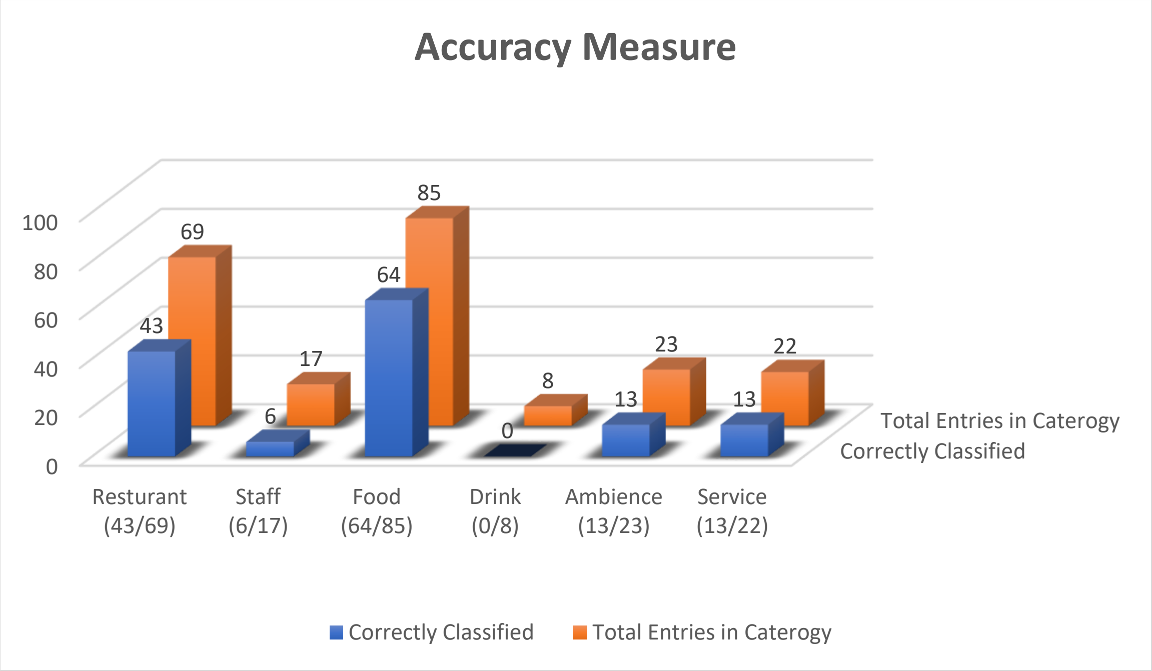
After Epoch 15 (Maximum Accuracy Acquired) at Epoch 02

Training loss: 1.373696972035686

Validation loss: 1.9879762841693653

F1 Score (Weighted): 0.6191816324544386

ACCURACY PER CLASS



ROBERTA-LARGE

CODE LINK

https://colab.research.google.com/drive/1DcZSIn3BK0YV4W2ZoUisEm673u4bCkpQ?usp=share_link

RESULTS

BATCH SIZE: 01

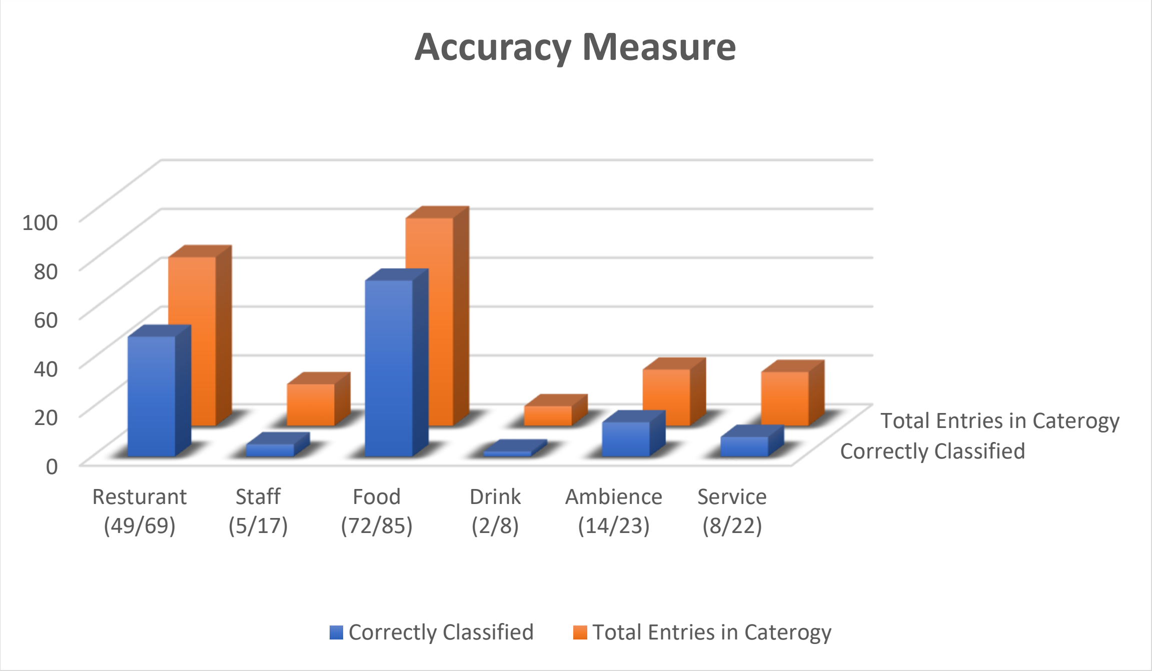
After Epoch 15 (Maximum Accuracy Acquired) at Epoch 03

Training loss: 0.9720405809823108

Validation loss: 1.5274807653970908

F1 Score (Weighted): 0.6602888153934032

ACCURACY PER CLASS



ALBERT-BASE

CODE LINK

https://colab.research.google.com/drive/1sXELzXAx8KSsMWPCWMTNDiP3-QVUdoi7?usp=share_link

RESULTS

BATCH SIZE: 01

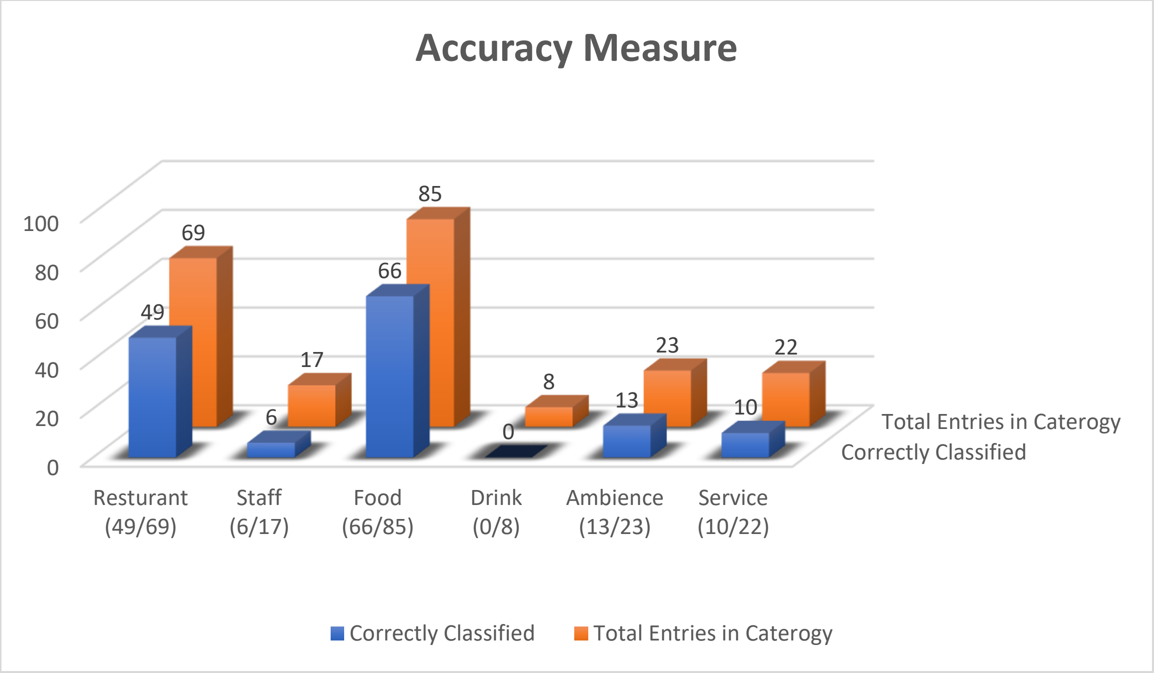
After Epoch 15 (Maximum Accuracy Acquired) at Epoch 04

Training loss: 1.201343481968261

Validation loss: 1.892008155167267

F1 Score (Weighted): 0.6432800582354155

ACCURACY PER CLASS



ALBERT-LARGE

CODE

https://colab.research.google.com/drive/1QjeiUZqmtOM3PjSCy162kEyz2xWcAYea?usp=share_link

RESULTS

BATCH SIZE: 01

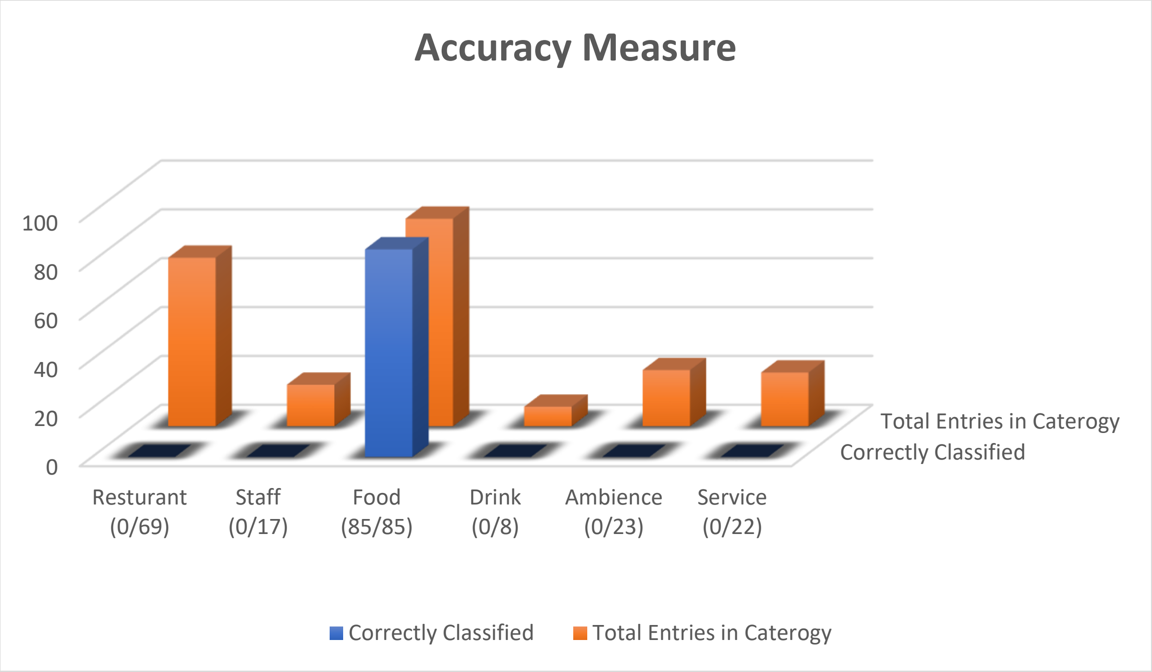
After Epoch 15 (Maximum Accuracy Acquired) at Epoch 01

Training loss: 1.29240462030005

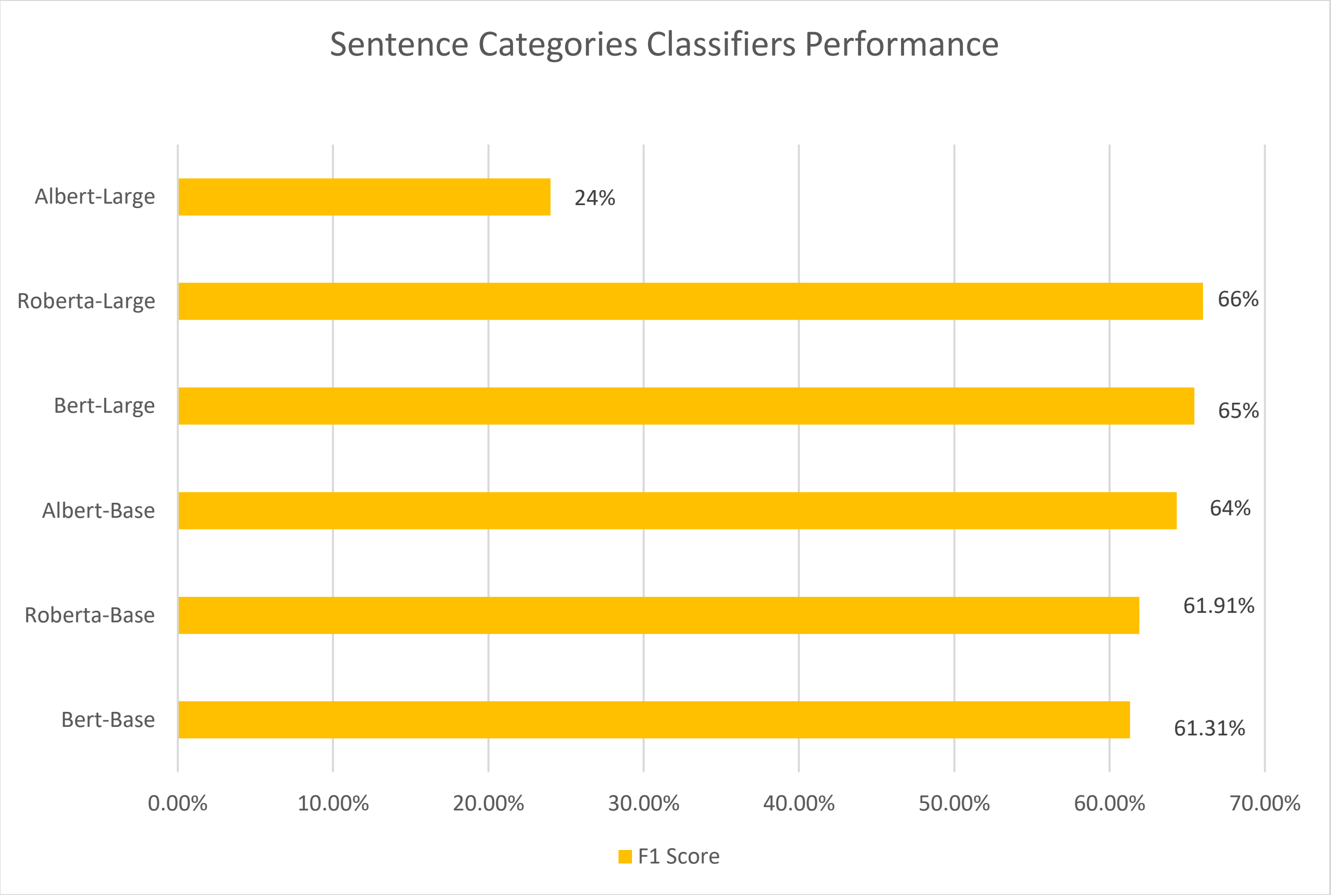
Validation loss: 1.2159249637957796

F1 Score (Weighted): 0.6543694001991041

ACCURACY PER CLASS



ALL BERT MODELS AVERAGE F1 ACCURACY SCORES



OBSERVATION AND CONCLUSION

Experience the cutting-edge accuracy of BERT models with **ROBERTA-LARGE** leading the pack! With a stunning 66.02% accuracy, **ROBERTA-LARGE** outperforms all other pre-trained BERT classifiers, making it the obvious choice for our data balancing efforts.

Not only does **ROBERTA-LARGE** excel in overall accuracy, but it also shines in classifying drinks with a remarkable 20% success rate- a feat that no other model could match.

However, while we celebrate the success of **ROBERTA-LARGE,** we also acknowledge the need to address the imbalance in the data, which has resulted in low classification rates for staff, ambience, and service categories. From deeper analysis we can conclude that our data needs more richness and balance we might take that in account.

PROPOSED DATA BALANCING AND ENHANCING TECHNIQUES

STRATIFICATION

ENSEMBLE TECHNIQUES

- Under Sampling
- Over Sampling
- Class Weight Adjustment / Cost Sensitive Learning
- SMOTE (Synthetic Minority Over-sampling Technique).
- Bagging

MULTIPLE MODELS TO ENRICH DATA

- (Generative Adversarial Networks)
- Variational Autoencoder (VAE)
- Data Balancing and Enhancement using Chat GPT
- Data Augmentation using NLP
  - Contextual and non-Contextual Augmentation

STRATIFICATION

Stratified sampling is a method that is used to ensure that the distribution of important variables in the sample is similar to the distribution in the population. This is particularly useful in the context of train-test split and data balancing in machine learning.

In train-test split, stratified sampling is used to divide the data into training and test sets in a way that the proportion of important variables is maintained in both sets. For example, if the target variable in a binary classification problem has an imbalanced distribution (e.g. 70% of the samples belong to one class and 30% to another), stratified sampling ensures that the imbalance is maintained in both the training and test sets, so that the model can be tested on a representative sample of the population.

IMPACT ON ACCURACY

Stratification of data during the train test split seems to have not significant impact on the accuracy of all BERT Classifiers that we Fine-tuned earlier.

ENSEMBLE TECHNIQUES

UNDER SAMPLING

Under-sampling is a method to balance an imbalanced dataset by reducing the size of the majority class. This is done to prevent the model from being biased towards the majority class, which can occur when training on imbalanced data.

CODE LINK:

https://colab.research.google.com/drive/1DcZSIn3BK0YV4W2ZoUisEm673u4bCkpQ?usp=share_link

UNDER SAMPLING PROCEDURE

a. Finding the Class with Least most occurrence in our Case Drink Class is the Least most occurring category having only 50 entries.
b. Iterating over all the 6 Categories and check if the entries corresponding to that class are greater then 50 i-e Least class entries.
c. (Only resample if the class has more instances than the minority class size)
d. Sample 50 items randomly from each class so to equalize the proportion of each category in dataset.

DATA AFTER UNDER SAMPLING

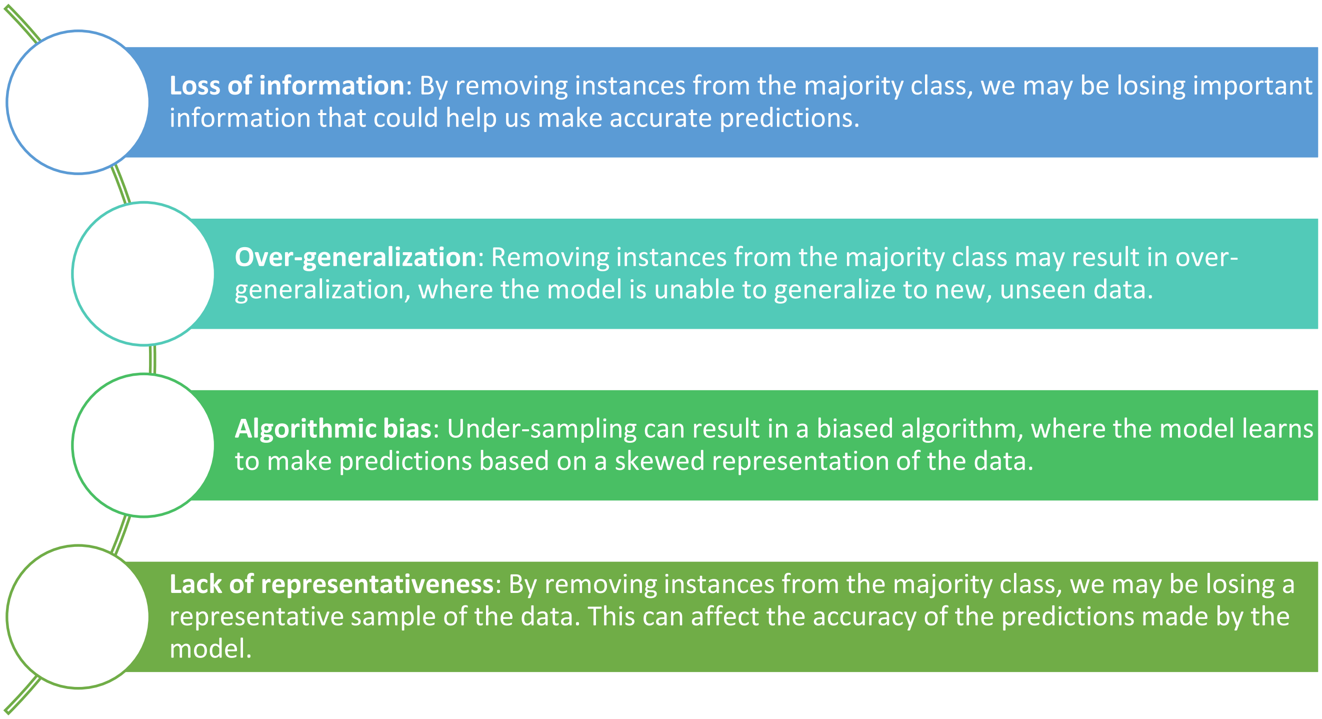
Before Under Sampling: 1290

After Under Sampling: 300

Every Category has randomly sampled entries same as no of entries against least most accrued category i-e 50 * 6 =300

UNDER SAMPLING TRADEOFFS

Under-sampling has several trade-offs:



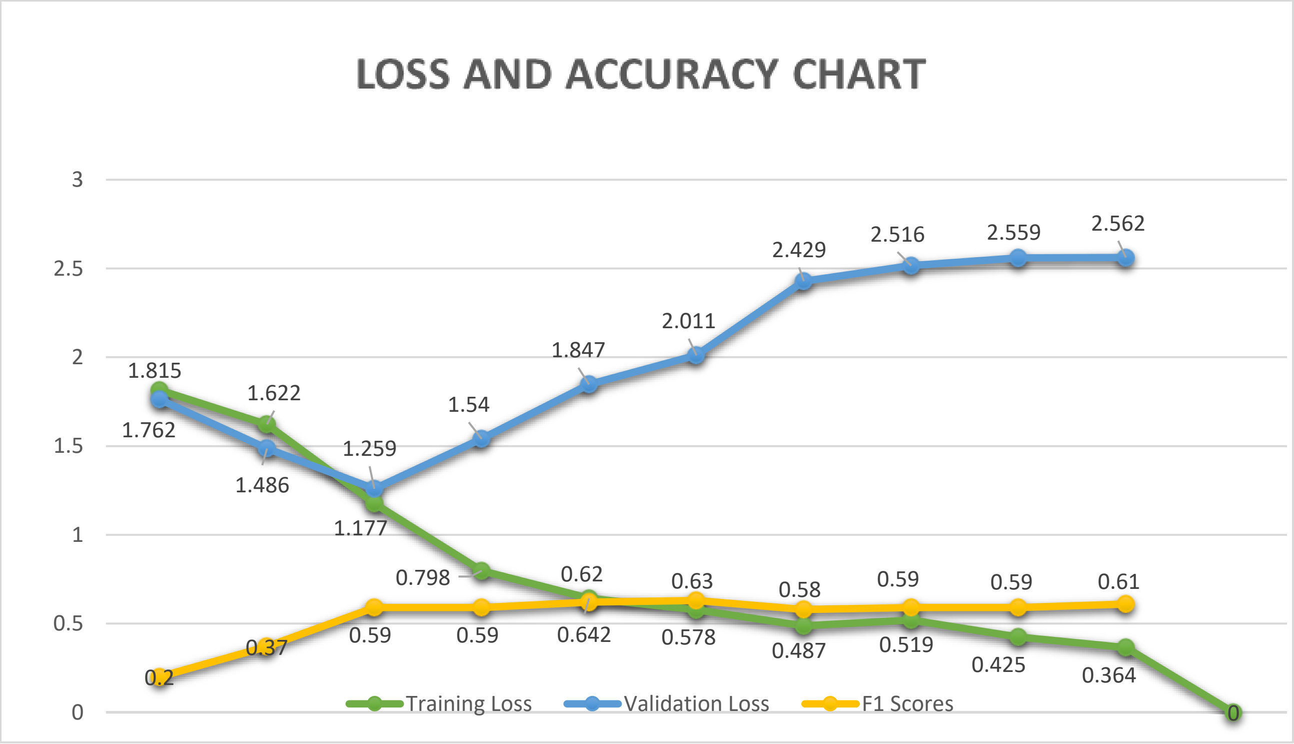
Loss of information: By removing instances from the majority class, we may be losing important information that could help us make accurate predictions.

Over-generalization: Removing instances from the majority class may result in over-generalization, where the model is unable to generalize to new, unseen data.

Algorithmic bias: Under-sampling can result in a biased algorithm, where the model learns to make predictions based on a skewed representation of the data.

Lack of representativeness: By removing instances from the majority class, we may be losing a representative sample of the data. This can affect the accuracy of the predictions made by the model.

ROBERTA-LARGE ACCURACY SCORE AFTER UNDER SAMPLING



HIGHEST F1 SCORES

At Epoch 6

Training loss: 0.5784616832224722

Validation loss: 2.0114593934946847

F1 Score (Weighted): 0.6308027484498072

CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0	0.571	0.500	0.533	8
1	0.778	0.875	0.824	8
2	0.636	0.875	0.737	8
3	0.375	0.375	0.375	8
4	0.375	0.375	0.375	8
5	0.600	0.375	0.462	8

Accuracy			0.562	48
Macro Avg	0.556	0.562	0.551	48
Weighted Avg	0.556	0.562	0.551	48

CONCLUSION

Under sampling can have some drawbacks when it comes to the accuracy and performance of a model. In our case one of the major concerns with under sampling is over generalization. This because the amount of data is reduced to such an extent that the model is unable to learn the underlying patterns and relationships within the data. In our case, the original dataset has just 1290 entries, which is already a relatively small dataset, and when it is under sampled to just 300 entries, the lack of information results in over generalization. This led decrease in accuracy and performance, as the model will not have enough data to make accurate predictions.

Another trade-off with under sampling is the loss of diversity in the data. Since the majority classes are the ones being reduced, the data can become very homogeneous, with the minority classes dominating the dataset. This result in the model not being able to learn the unique characteristics of each class and make accurate predictions.

OVER SAMPLING

oversampling refers to the technique of increasing the size of the minority class in a dataset by replicating its samples. This is often used to address the issue of imbalanced class distributions, where one class has significantly more samples than another class.

CODE LINK:

https://drive.google.com/file/d/1zsWIWt3cAX9EnR4ERhJqOvcxB4Kncw7V/view?usp=share_link

OVER SAMPLING PROCEDURE

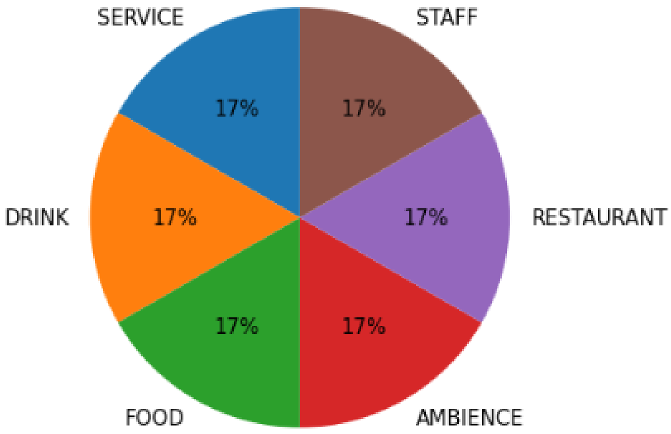
a. Finding the Class with most occurrence in our Case Food Class is the most Least occurring category having only 437 entries.
b. Iterating over all the 6 Categories and check if the entries corresponding to that class are less then 437 i-e most accrued class entries.
c. (Only resample if the class has less instances than the minority class size)
d. Introduce duplicates items randomly from each class so to equalize the proportion of each category in dataset.

DATA AFTER OVER SAMPLING

Before Over Sampling : 1290

After Over Sampling : 2622

Every Category has randomly introduced duplicates entries same as no of entries against most accrued category i-e 437 * 6 =2622



OVER SAMPLING TRADEOFFS

over-sampling has several trade-offs:

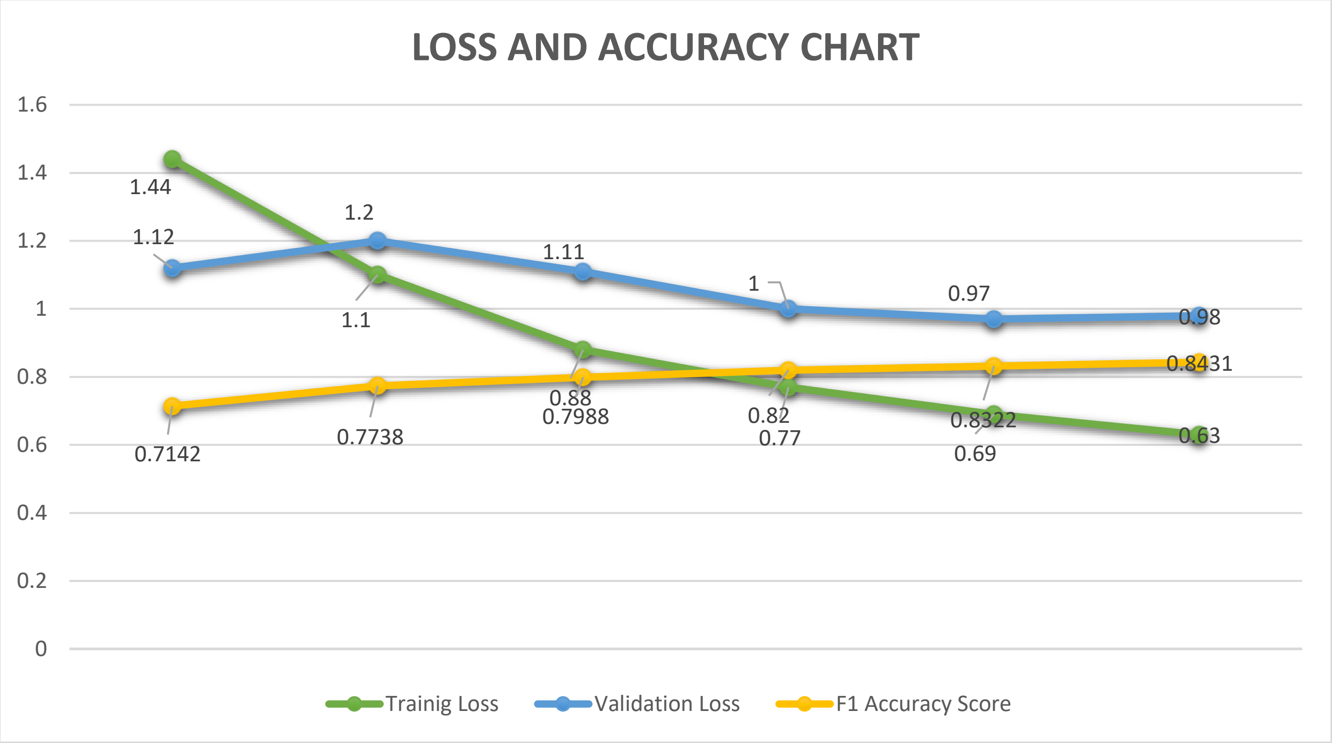
Overfitting: By increasing the size of the minority class, oversampling can lead to overfitting, where the model becomes too closely tied to the training data. This can result in poor generalization performance and reduced accuracy on new, unseen data.

Computational Complexity: Oversampling increases the size of the dataset, which can result in increased computational time and memory usage during the training process.

Artificial Inflation of Importance: By artificially increasing the size of the minority class, oversampling can give the impression that the minority class is more important than it actually is, which can result in biased predictions.

Loss of Diversity: By replicating samples in the minority class, oversampling can result in the loss of diversity within that class. This can result in a model that is too focused on a particular subset of the minority class, and not general enough to make accurate predictions on a diverse range of samples.

ROBERTA-LARGE ACCURACY SCORE AFTER OVER SAMPLING



HIGHEST F1 SCORES

Epoch 4

Training loss: 0.5703232053944705

Validation loss: 0.9031832036538483

F1 Score (Weighted): 0.8435815704500024

CLASSIFICATION REPORT

Class	Precision	Recall	F1 Score	Support
0	0.698	0.706	0.702	85
1	0.816	0.930	0.870	86
2	0.938	0.718	0.813	85
3	0.809	0.847	0.828	85
4	0.899	0.826	0.861	85
5	0.802	0.895	0.846	86

Accuracy			0.821	513
Macro Avg	0.827	0.820	0.820	513
Weighted Avg	0.827	0.821	0.820	513

CONCLUSION

Our findings indicate that the over-sampling technique significantly improved accuracy, reaching a high of 84. 35% (75% + accuracy in all individual classes). However, this came at the cost of duplicated data within the training and testing sets. This makes the model highly dependent on the training data and less able to generalize to new, unseen data. While the model performed well on the training dataset, it may not have the same level of accuracy when applied to real-world data or sentences. Additionally, over-sampling greatly increased the amount of data, leading to a 103% increase in the size of the dataset and a corresponding increase in training time.

CLASS WEIGHTS ADJUSTMENTS

Class weight adjustments can help balance the influence of different classes in a multi-class classification problem. The idea is to give more weight to under-represented classes, so that the model pays more attention to them during training.

CODE LINK:

https://drive.google.com/file/d/1GzjEh7LhS0JxxwSLJFAdPOK0RDWA234w/view?usp=share_link

CLASS WEIGHTS ADJUSTMENTS PROCEDURE

Import the compute_class_weight function from Sklearn's utils module.

Calculate the class weights for the given dataset labels by passing the "balanced" option as the first argument and the unique class labels as the second argument, and the dataset labels as the third argument.

Create a dictionary of class labels and their corresponding weights using the zip function.

Modify the loss function used in the model to consider the class weights. This can be done by passing the class weights as the "weight" argument to the loss function.

Use this modified loss function in the model training and validation process to ensure that the model takes into account the class weights and performs balanced training and evaluation.

DATA AFTER CLASS WEIGHTS ADJUSTMENTS

Before Class Weight Adjustments: 1290

As class weight adjustment do not have thing to do with data enhancing it just only works to make the model cost effective in learning.

BALANCED CLASS WEIGHTS ADJUSTMENTS

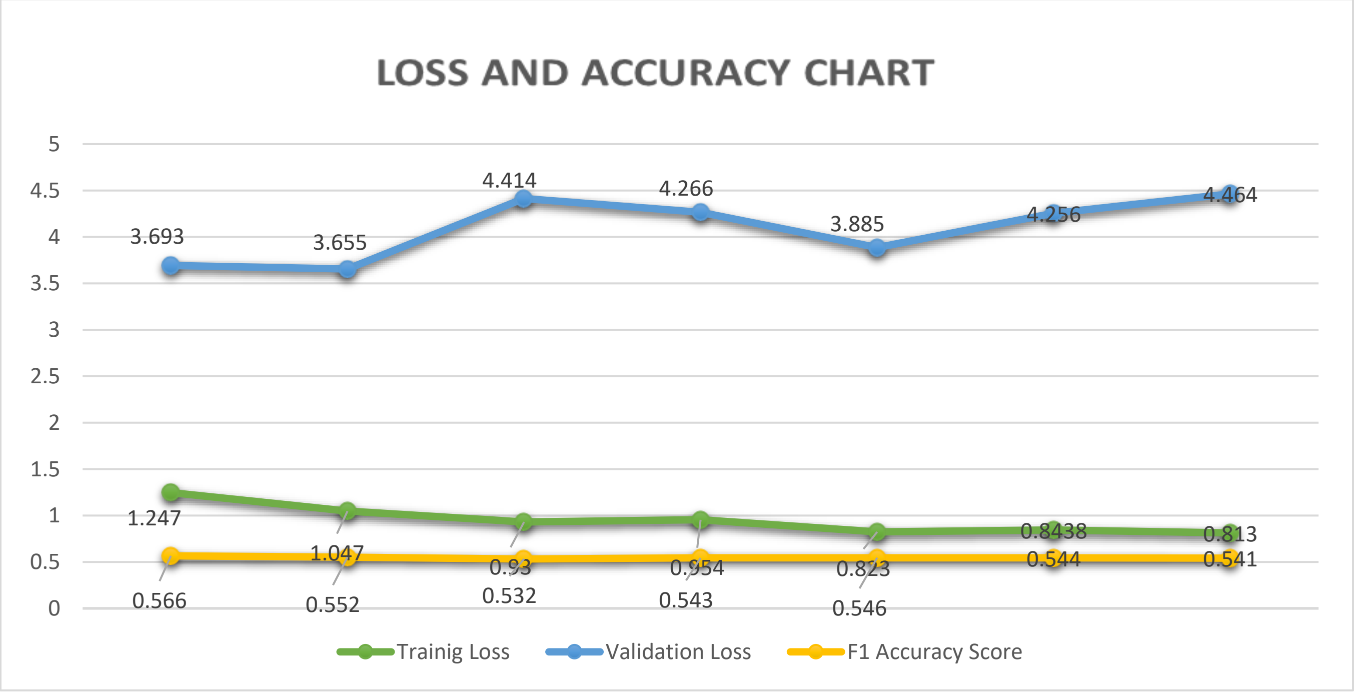
Category	Label	Count	Balanced Weight
Restaurant	0	431	0.4988399071925754
Staff	1	113	1.9026548672566372
Food	2	437	0.4919908466819222,
Drink	3	50	4.388679245283019
Ambience	4	127	1.6929133858267718
Service	5	132	1.628787878787879

CLASS WEIGHTS ADJUSTMENTS TRADEOFFS

Class weights Adjustments has several trade-offs:

- **Over-fitting**: The model may over-fit to the class with the highest weight, causing poor generalization performance.
- **Imbalanced distribution**: If the class distribution is highly imbalanced, the model may still produce poor results even after adjusting the class weights.
- **Unreliable results**: The model may produce unreliable results if the weight adjustments are not properly calculated and applied.
- **Model Complexity**: Adjusting class weights can increase the complexity of the model and cause it to require more training data and computational resources.
- **Biased results:** If the class weight adjustments are not representative of the true class distribution, the model may produce biased results.

ROBERTA-LARGE ACCURACY SCORE AFTER CLASS WEIGHTS ADJUSTMENTS



HIGHEST F1 SCORES

Epoch 2

Training loss: 1.2430305113250248

Validation loss: 3.698126528398752

F1 Score (Weighted): 0.5661334609281204

CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0	0.712	0.730	0.720	65
1	0.643	0.350	0.420	17
2	0.623	0.538	0.570	66
3	0.264	0.577	0.368	7
4	0.345	0.423	0.387	19
5	0.367	0.402	0.384	20
Accuracy			0.562	194
Macro avg	0.478	0.507	0.478	192
Weighted avg	0.583	0.569	0.563	194

CONCLUSION

The results suggest that class weight balancing didn’t produce effective results, as the accuracy and performance of the model dropped up to 56%. This is likely due to the highly imbalanced proportion of data, with the majority and minority classes contributing to approximately 33% and 3.84% of the dataset (difference of 30%), respectively. But we cannot deny the fact that class weight Adjustment did a great job in Drink Category Classification that was the minority class. But overall imbalanced proportions result in unreliable outcomes and increase the complexity and training time of the model.

SMOTE (SYNTHETIC MINORITY OVER SAMPLING TECHNIQUES)

In machine learning, synthetic samples refer to artificially generated samples that are created using statistical models. The main purpose of synthetic samples is to increase the size of the minority class in imbalanced datasets. SMOTE (Synthetic Minority Over-sampling Technique) is a popular technique for generating synthetic samples to balance the dataset. SMOTE generates synthetic samples for the minority class by interpolating between the feature values of existing minority class instances. The synthetic samples are then added to the original dataset to make the minority class distribution similar to the majority class.

CODE LINK:

https://drive.google.com/file/d/1ve3YyUYn54lKcUgKOikB2VVo2ukvlTLR/view?usp=share_link

SMOTE PROCEDURE

Convert all the sentences to numeric form (transform only if the input features are in other format, then numeric or vectorized form) using various methods such as

- **One-hot encoding:** You can convert the textual data into one-hot encoded format, which will convert each unique word or n-gram into a separate feature and assign a 1 or 0 to each feature based on the presence of the word or n-gram in the text.
- **Term Frequency-Inverse Document Frequency (TF-IDF):** Another common method is to use TF-IDF, which will represent the words in the text by their frequency of occurrence, and then adjust the frequency based on how rare the word is in the entire corpus of documents.
- **Word Embeddings:** Another option is to use pre-trained word embeddings, such as Word2Vec or GloVe, which can convert the words in the text into dense vectors of fixed size.
- **Sentence Embeddings:** Another option is to use pre-trained sentence embeddings, such as Sentence-BERT, which can convert a sentence or a document into a dense vector of fixed size.

After that we import the smote function from imblearn.over_sampling and pass the arguments such as sampling strategy in our case we used auto to detect automatically which class to oversample then use fit function to generate the synthetic samples.

Revert or back inverse transform those function from vectors to sentences and feed data to the model.

PROBLEM

In our scenario we used Sentence Embeddings but it does not worked as sentence embedding method could only transform the sentences in to vectorized form and does not have the decoding method to back transform. it only returns the duplicates of the original sentences and does not consider the synthetic sentences so its not worthy to use. Beside this all other methods such as Word Embedding Method ,Count Vectorizer does the same because smote is actually for numerical data not for textual data (word and Sentence Embeddings are for textual data but they do not provide the back transform method to convert the synthetically generated data back to sentences as their embeddings varies from the original sentences embeddings due because the SMOTE uses the any statistical Model to predict the Neighbors for the original samples) . so we an declare that there isn't any to convert the synthetically generated samples embedding back to the sentences only the synthetically sentences that have the same embedding to original sentences in mapping are processed rest of the samples are ignored.

ASSUMPTION

We will be assuming the duplicated sentences for the further processing we can also say that the SMOTE is now works same as the Over sampling.

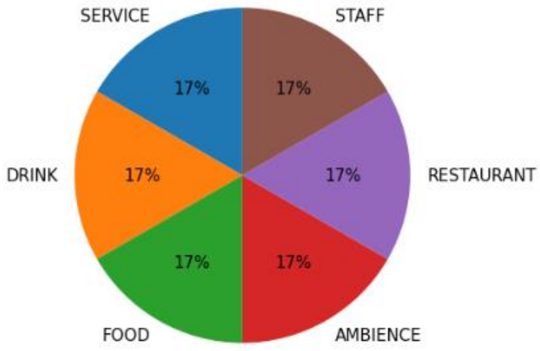
DATA AFTER SMOTE APPLICATION

Before Smote :1290

After Smote: 2622

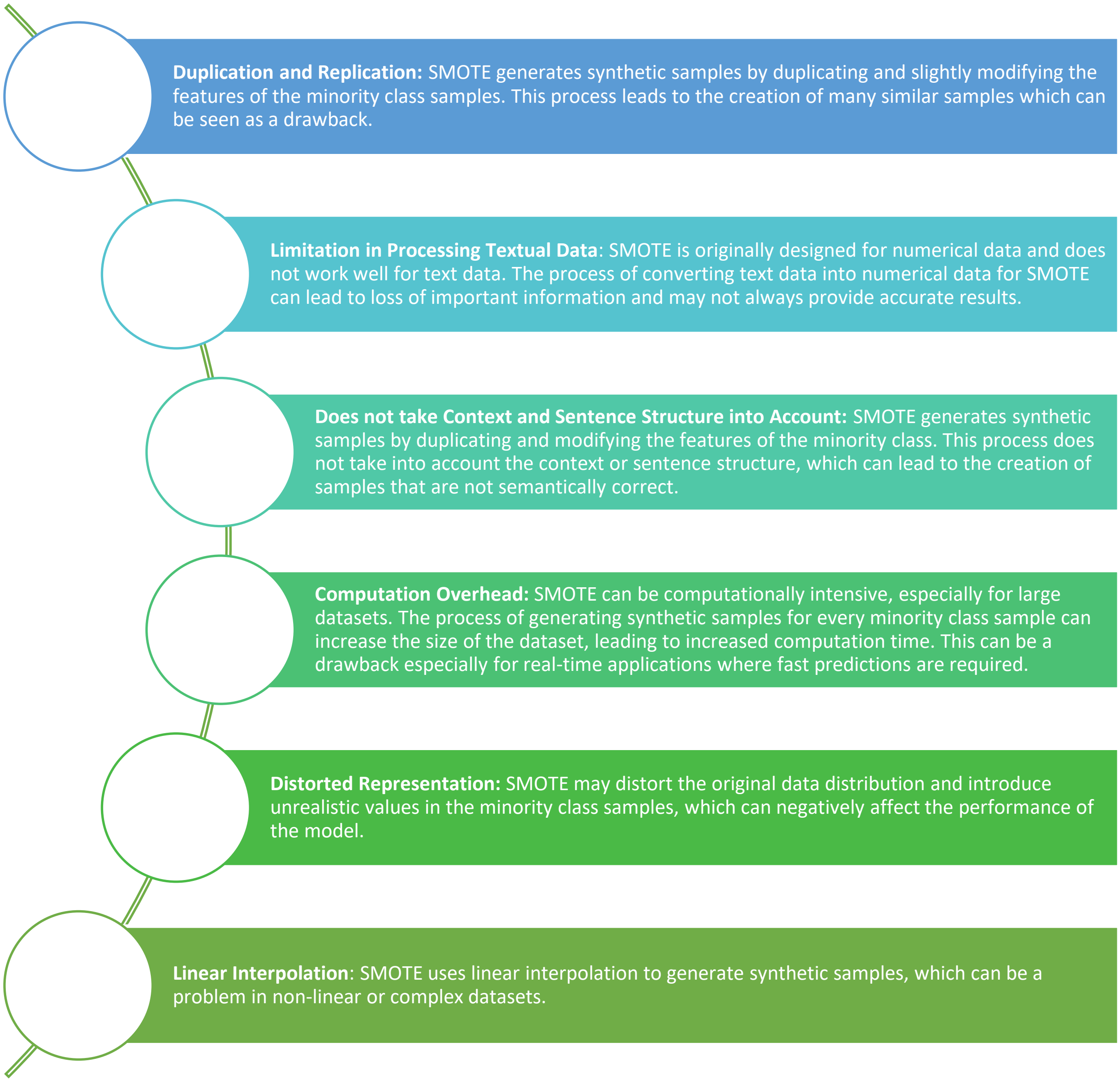
Every category now has been introduced with synthetic entries that may or may not be the duplicates. As we choose auto sampling strategy that it uses the procedure of over sampling to match the minority categories number same as majority no but they may not be replicated and introduced through statistical algorithms may be K nearest neighbors.

So that the total size now is i-e 437 * 6 =2622



SMOTE TRADEOFFS

SMOTE, which is a Synthetic Minority Over-sampling Technique, has several trade-offs when used for balancing imbalanced datasets.



Duplication and Replication: SMOTE generates synthetic samples by duplicating and slightly modifying the features of the minority class samples. This process leads to the creation of many similar samples which can be seen as a drawback.

Limitation in Processing Textual Data: SMOTE is originally designed for numerical data and does not work well for text data. The process of converting text data into numerical data for SMOTE can lead to loss of important information and may not always provide accurate results.

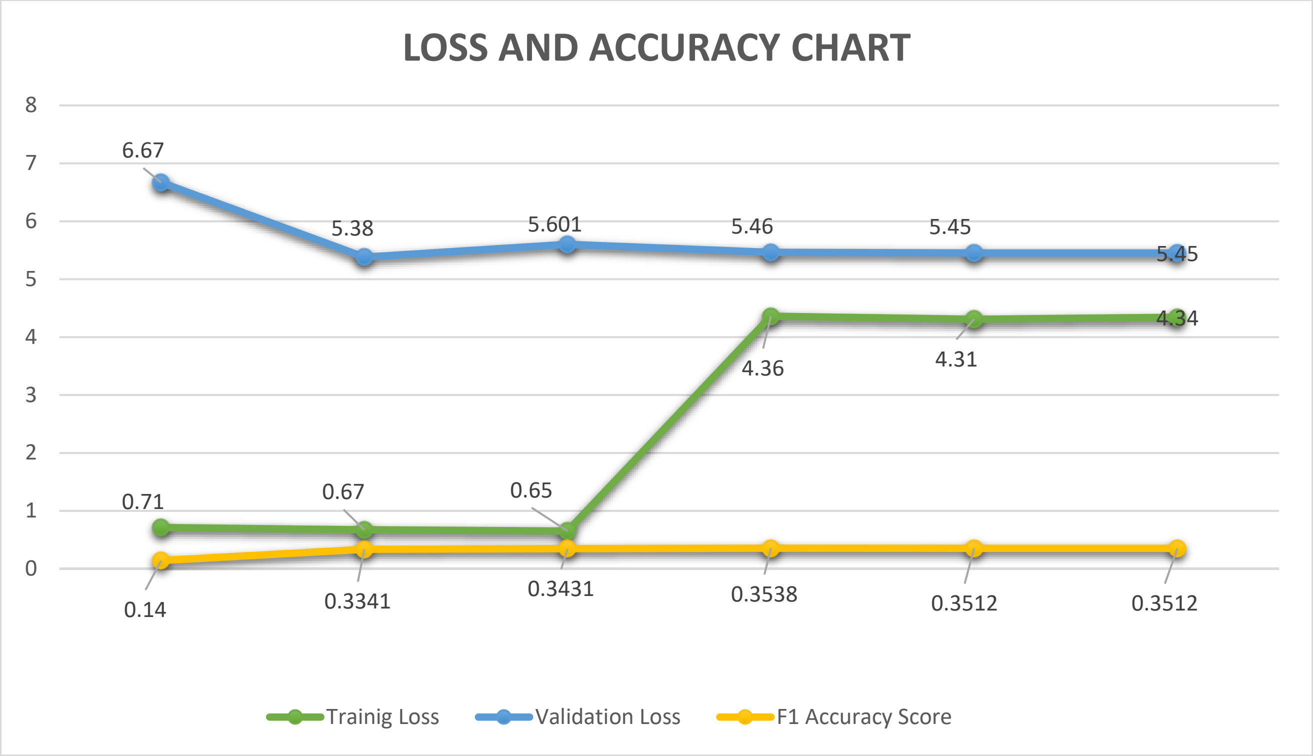
Does not take Context and Sentence Structure into Account: SMOTE generates synthetic samples by duplicating and modifying the features of the minority class. This process does not take into account the context or sentence structure, which can lead to the creation of samples that are not semantically correct.

Computation Overhead: SMOTE can be computationally intensive, especially for large datasets. The process of generating synthetic samples for every minority class sample can increase the size of the dataset, leading to increased computation time. This can be a drawback especially for real-time applications where fast predictions are required.

Distorted Representation: SMOTE may distort the original data distribution and introduce unrealistic values in the minority class samples, which can negatively affect the performance of the model.

Linear Interpolation: SMOTE uses linear interpolation to generate synthetic samples, which can be a problem in non-linear or complex datasets.

ROBERTA-LARGE ACCURACY SCORE AFTER SMOTE



HIGHEST F1 SCORES

Epoch 7

Training loss: 4.348405997801333

Validation loss: 5.456291054010342

F1 Score (Weighted): 0.3512489253318297

CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0	0.500	0.015	0.030	65
1	0.000	0.000	0.000	66
2	0.703	0.394	0.505	66
3	0.000	0.000	0.000	65
4	0.000	0.000	0.000	66

5	0.181	0.970	0.305	66
Accuracy			0.231	394
Macro avg	0.231	0.230	0.140	394
Weighted avg	0.230	0.230	0.141	394

CONCLUSION

We can conclude that by using Smote it introduces the synthetic samples to balance the distribution but it produces samples that make no sense in context and sentence structures thus make the model more prone towards unreliable results and drastically reduce accuracy because the smote works better for the numerical data or numeric input features by using k nearest neighbors it may not be significantly viable to use for textual data because the textual data is more complex by taking context and sentence structure in account. It may not be the worthy solution because it also introduces duplicates and broken sentences with no meaning at all. Regarding the problem that we discussed in the procedure section moreover, despite removing empty rows and feeding clean data to the model, it was observed that using Smote was not a worthy solution for the problem.

ML MODELS TO ENHANCE DATA

NLP AUGMENTATION

Data augmentation is a technique used to increase the size of a dataset in NLP by synthesizing new data from existing data. This is done by applying transformations to the existing data that preserve its meaning but result in new, distinct data points. Some common data augmentation techniques for NLP include:

DATA AUGMENTATION TECHNIQUES

Replace n number words with its synonyms (word embeddings that are close to those words) to obtain a sentence with the same meaning but with different words sub techniques that are used in contextual word embeddings to enhance the relevant dataset are given below:

- **Synonym replacement or Substitution**: Replace words in a sentence with their synonyms to generate new sentences.
- **Random insertion**: Add random words to a sentence to generate new sentences.
- **Random deletion**: Remove words from a sentence to generate new sentences.
- **Random swap**: Swap two words in a sentence to generate new sentences.
- **Random perturbation**: Perturb the words in a sentence by adding, removing, or swapping words to generate new sentences.

These techniques can help to balance the distribution of classes in a dataset and improve the generalization of NLP models by exposing them to a wider range of language structures and variations.

NLP LIBRARY FOR DATA AUGMENTATION

NLPAUG library for nlp data augmentation specifically designed for nlp tasks. It contains various data augmentation techniques specifically designed for nlp tasks such as text substitution, synonym replacement, random deletion, random insertion, random swap, back translation, etc.

Using **NLPAUG**, we can easily perform data augmentation on our nlp dataset and increase the size and diversity of your training data, which can improve the performance of our nlp models. To use **NLPAUG**, we first need to install it and then import the relevant modules in our code. After that, we can use the various augmentation functions provided by the library to perform data augmentation on your text data.

CODE

https://drive.google.com/file/d/1diqD6j3DGjbWzpAzUfGjDXJV7k3NTMkk/view?usp=share_link

https://drive.google.com/file/d/1fhMaiGfFEKK7if_xlZYT-wzicqBJWvyJ/view?usp=share_link

AUGMENTED DATASET LINK

https://docs.google.com/spreadsheets/d/1dNxT5cXciCe6cUTHWaHe6yrHssLs4rfP/edit?usp=share_link&ouid=109317857182659250370&rtpof=true&sd=true

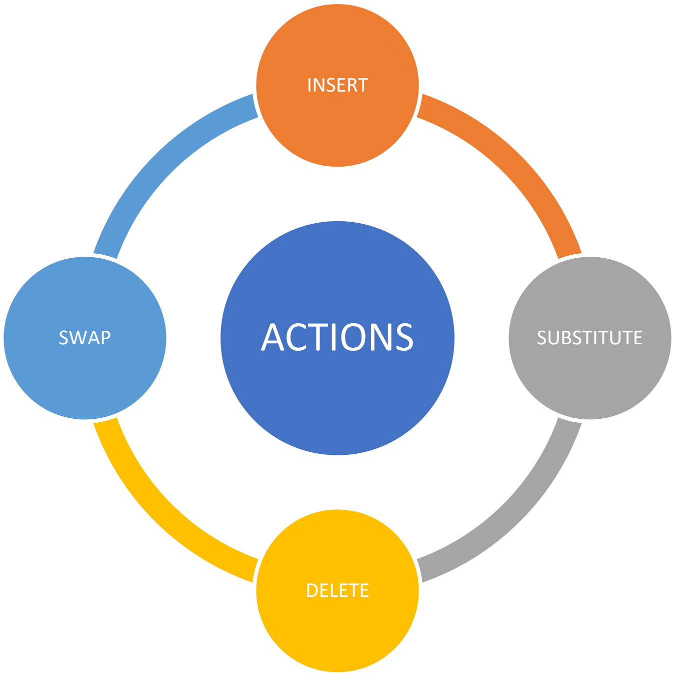
AUGMNETATION TYPES

NLPAUG offers three types of augmentation:

- Character level augmentation
- Word level augmentation
- Sentence level augmentation

CONTEXTUAL WORD EMBEDDING

ACTIONS:



AUGMENTORS:

1. **Synonym**: Augmenter that applies semantic meaning based on textual input.
2. **Antonym:** Augmenter that applies semantic meaning based on textual input.
3. **Random:** Augmenter that applies random word operation to textual input.
4. **Spelling:** Augmenter that applies spelling error simulation to textual input.
5. **Split:** Augmenter that apply word splitting operation to textual input.
6. **Flow Augmentation:** In this type of augmentation, we can make use of multiple augmenters at once. Sequential and sometimes pipelines are used to connect augmenters to make use of many augmentations. A single text can be sent through multiple augmenters to yield a wide range of data.
7. **Word2Vec:** Augmenter used for non-contextual word embeddings.
8. **Sequential:** To add as many augmenters to this flow as you wish, and Sequential will execute them one by one. You can, for example, combine ContextualWordEmbsAug and WordEmbsAug.
9. **Sometimes:** The pipeline can pick a different set of augmenters every time.

OUR STRATEGY

The NLP Augmentation technique that we will be using for data enrichment is the **word level augmentation.**

This is because our dataset consists of individual sentence statements extracted from reviews and their corresponding categorization. Tokenizing the reviews into sentence statements has already provided us with the necessary level of granularity, making it unnecessary to utilize sentence-level augmentation. Additionally, character-level augmentation may not be the best choice for our purposes as it involves random shuffling, insertion, and deletion of characters, which could potentially lead to misspelling of words and negatively impact the quality of the data. By using word level augmentation, we can make more informed changes to the data while preserving the overall meaning and coherence of the sentences.

We will be taking a comprehensive approach to data enhancement by utilizing all the **advanced augmentation techniques provided by the NLPAUG library with contextual word level embedding as well non contextual embeddings** like word2vec through **SomeTimes Augmenter** (because we want to get as much more diversity in data hence using the combination of Flow and Word2vec Augmenter with all possible Actions).Our ultimate goal is to create a single, diverse and abundant data set that will enable us to train our model to its fullest potential.

MODELS USED BY AUGMENTORS

We are using following NLP models.

- **BERT-LARGE-CASED** in **Contextual Word Embeddings**
- **GLOVE** in **Non-Contextual Word Embedding** such as Word to Vec

DATA AFTER NLP AUGMENTATION

We will be using Class weights percentage to balance the data that it to make augmented data same as the number required to balance the distribution of each class in the data and then after that we will be preprocessing the whole data.

IMPORTANT CONSIDERATIONS

From data analysis and avoiding overfitting we will stick to populate every class with somehow more or less than 600 Sentences (small percentage variation for each class participation to improve generalization)

(Minority class have more augmented sentences to balance itself it the majority class)

TABLE ESSENTIALS

- In our case given class participation = 600
- Each sentences requires to generate = no of required Augmented Sentences / count of given Sentences
- No of Augmented Sentences to create = given class participation – count of given sentences
- Condition = Check if the full /partial set of sentences is required to match the specified class size

Category	Required Augmented Sentences	Condition	Sampling Methodology	Selected Samples Size for Augmentation	No of Sentences to be Augmented from given Single Sentence. (floor on float)	Introduced Augmented Sentences	Total Sentences
Food	163	Partial	Random	Randomly selected Samples	1 from each	148	585
Restaurant	169	Partial	Random	Randomly selected Samples	1 from each	191	622
Service	468	Full	Sequential	Count of All Sentences	4 from each	528	660
Ambience	473	Full	Sequential	Count of All Sentences	4 from each	496	635
Staff	487	Full	Sequential	Count of All Sentences	5 from each	565	678
Drink	550	Full	Sequential	Count of All Sentences	11 from each	550	600
						Total Augmented Sentences	2490
						Data Before Augmentation	1290
						Date After Augmentation	3780
						Data After Post processing	3751

PARTICIPATION PERCENTAGE

Category	Label	Count	Balanced Weight
Restaurant	0	612	16%
Staff	1	676	18%
Food	2	575	15.5%
Drink	3	600	16%
Ambience	4	634	17%
Service	5	654	17.5%

AUGMENTATION EXECUTION TIME

It takes around 4.5 hours to generate the required augmented dataset.

HEIGHEST F1 SCORE

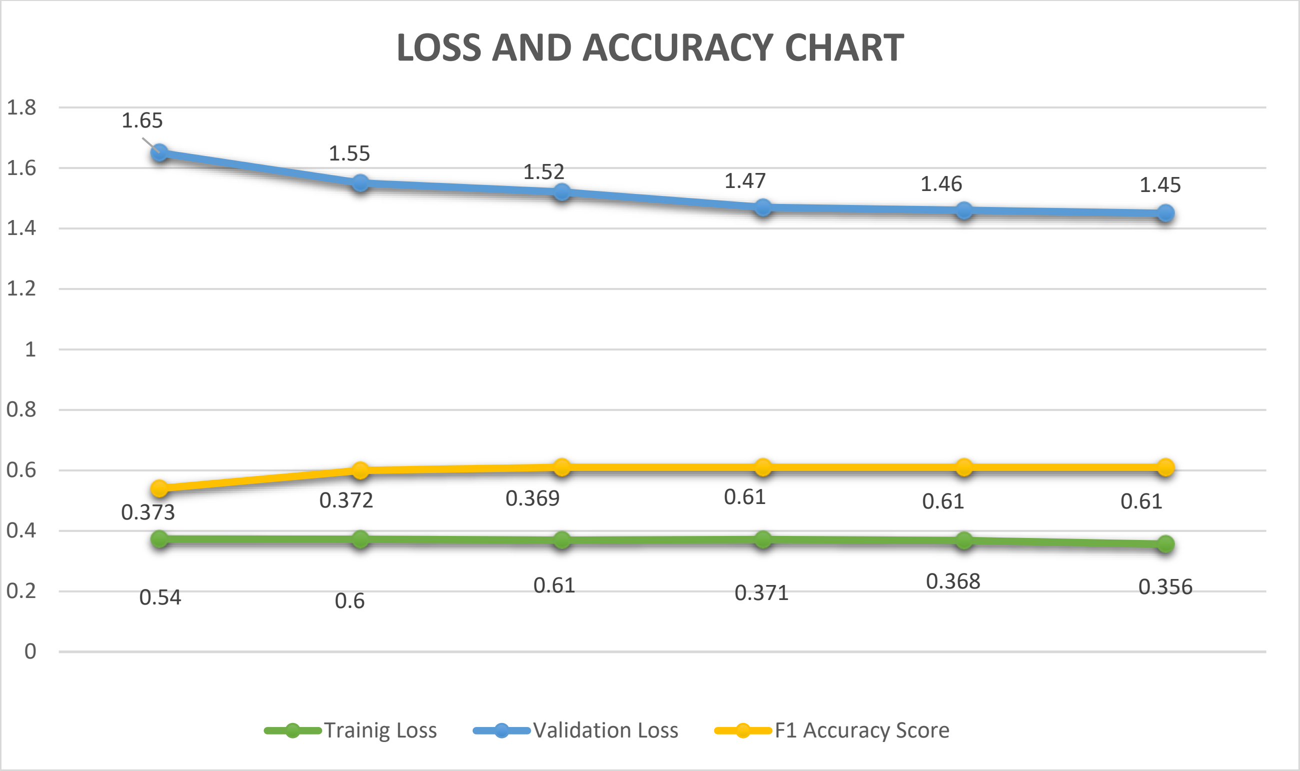
Epoch 5

Training loss: 0.36828010452019616

Validation loss: 1.4565438011162717

F1 Score (Weighted): 0.6162385228508068

ROBERTA ACCURACY AFTER NLP AUGMENTATION



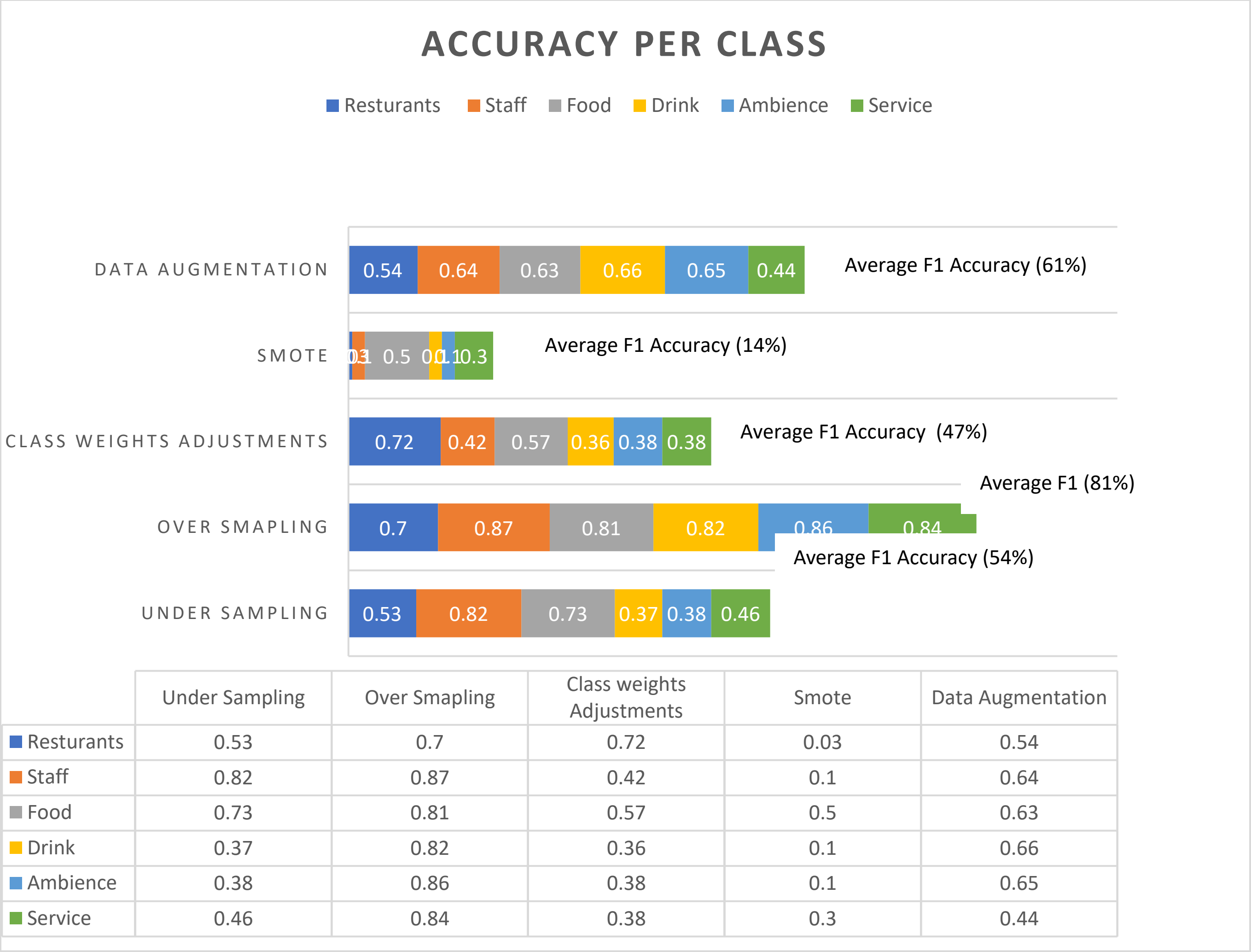
CLASSIFICATION REPORT

Label	Precision	Recall	F1-Score	Support
0	0.549	0.633	0.588	98
1	0.644	0.637	0.640	102
2	0.636	0.609	0.622	92
3	0.663	0.705	0.684	95
4	0.659	0.667	0.663	90
5	0.551	0.442	0.490	86
Accuracy			0.618	563
Macro avg	0.617	0.615	0.615	563
Weighted avg	0.618	0.618	0.616	563

CONCLUSION

Based on our findings, we can confidently state that the **ROBERT** model trained on augmented sentences has achieved an accuracy of 61%. However, it falls short in generalizing the training data to effectively validate the test data set. This is due to the nature of the augmented data set, which balances categories but sacrifices linguistic meaning. The augmented sentences were generated by randomly mutating individual sentences from each category from original dataset and then labeling them to be classified under the same category, even if the mutated sentence may not be semantically appropriate for that category (dataset link Attached). As a result, Sentences in the data set has become heavily misclassified, because the data is not supervised only substituted or mutated sentences are generated without supervision. Thus, augmenting data in this manner may not be a suitable approach as it leads to overfitting and incorrect predictions on unseen data.

DATA BALANCING TECHNIQUES AND ACCURACY COMPARISON



	Under Sampling	Over Smapling	Class weights Adjustments	Smote	Data Augmentation
■ Resturants	0.53	0.7	0.72	0.03	0.54
■ Staff	0.82	0.87	0.42	0.1	0.64
■ Food	0.73	0.81	0.57	0.5	0.63
■ Drink	0.37	0.82	0.36	0.1	0.66
■ Ambience	0.38	0.86	0.38	0.1	0.65
■ Service	0.46	0.84	0.38	0.3	0.44

DATA ENHANCEMENT USING CHAT GPT

MULTILABEL/MULTILABLE CATEGORIES CLASSIFCATION

CHATGPT DATA LINK

https://docs.google.com/spreadsheets/d/1Sb0FQsg_AoFJRmDa59UlWALZpeQf6cCV/edit?usp=share_link&ouid=109317857182659250370&rtpof=true&sd=true

PROCSSED DATA LINK

https://drive.google.com/file/d/1TwLTsWnsTUXC-jFXRM3D8h_yusJ_noGX/view?usp=share_link

DATA PROCESSING CODE

https://colab.research.google.com/drive/1EAcNmRNGOIyF-WVyGyY9YNU3H9oyzbXA?usp=share_link

CODE LINK

https://drive.google.com/file/d/1YybenAg3reymZaOWyRe2mTeN59dS3OsO/view?usp=share_link

BERT MODEL

BERT-BASE because of its high simplicity (pre trained model accuracy of up to 95 and it also performs well for our trained Aspects Model).

CLASSIFICATION TYPE

Multi Label Multi Class Classification

CONSIDERED DATASET

Combined data from our manually created data and CHATGPT augmented dataset in which every sentence has associated more one or more than one category from considered 6 categories and the data got processed in the form of one hot encoding to be inputted to BERT for multi class multilabel classification.

PROBLEM WITH CHAT GPT DATA DATASET

- Duplication/Replication
- Limitation in Dataset Size because after some time its repeats the sentences
- Data Collection is Time Consuming (requesting manually to ChatGPT for data generation)
- Requires basic preprocessing for text cleaning.

DATASET INFORMATION

Original Manually recorded Dataset=1491 reduced to 1290 after removing duplication

CHATGPT Augmented Dataset= 2439 reduced to 2419 after removing duplications

COMBINED DATASET

Combined Dataset=1290 + 2419 =3709 entries in Dataset

DATA AFTER REARRANGMENT

As our Chat GPT dataset as well as original data set was created in such a way that they both have multiple entries of single sentences against each specified category that sentence was classified in to process that we then make custom algorithm to convert that into the data set that will have a sentence without repetition and list of associated categories correspondingly. So, the entries after the process:

Original Manually recorded Dataset=1011

CHATGPT Augmented Dataset=1252

Combined Dataset=2263

DATA BALANCING

Data Collected was already balanced based on the collection query to CHATGPT. It requires trail and check loop to balance the least participation classes data.

BERT-BASE TRAINING AND VALIDATION LOSS

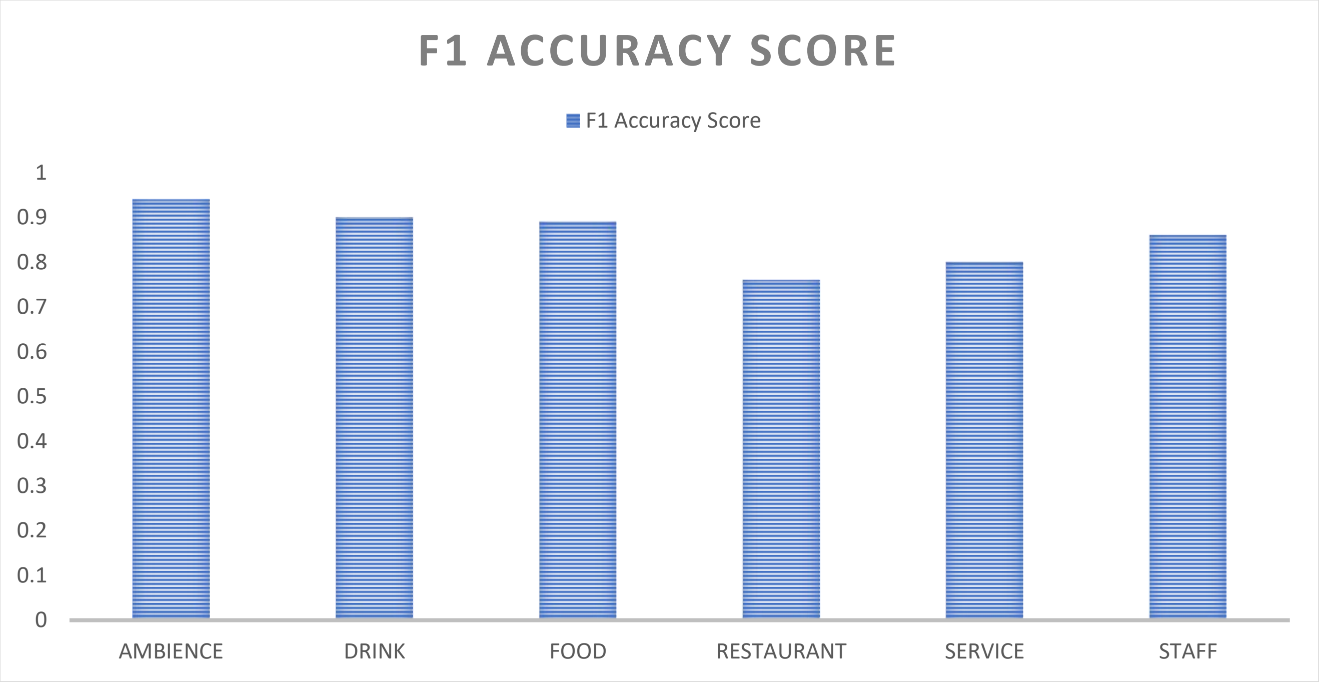
Training Loss=0.0405, Validation Loss=0.209 of the Best Checkpoint with Least Monitored Validation Loss.

EVALUTION COSINE SIMILIARITY

Cosine Similarity: Tensor (0.8312)

AVERAGE F1 SCORE FOR CLASSES

Average F1 Score maintained is of 0.85.



CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
Ambience	0.93	0.95	0.94	82
Drink	0.88	0.93	0.90	70
Food	0.86	0.93	0.89	122
Restaurant	0.73	0.79	0.76	130
Service	0.69	0.96	0.80	90
Staff	0.79	0.95	0.86	77
Micro avg	0.80	0.91	0.85	571
macro avg	0.81	0.92	0.86	571
weighted avg	0.81	0.91	0.85	571
samples avg	0.83	0.91	0.84	571

CONCLUSION

Previously we incorporated most the data balancing techniques with multiclass classification of sentence categories but that wasn’t worth, the accuracy and performance got dropped but coming up with the idea of incorporating qualitative and quantitative data enhancement techniques from Chat GPT and altering the classification problem from multiclass to multiclass/multilabel, the accuracy and performance of our model significantly improved to 85%. This level of improvement is crucial for deploying the model in real-world applications as it enhances its validity for unseen data. With the added flexibility in sentence category classification, our trained model is now more appropriate for deployment in various domains, making it a valuable asset for real-world applications. Overall, the improvements made in the model's accuracy and performance will greatly benefit its deployment in real-world settings.

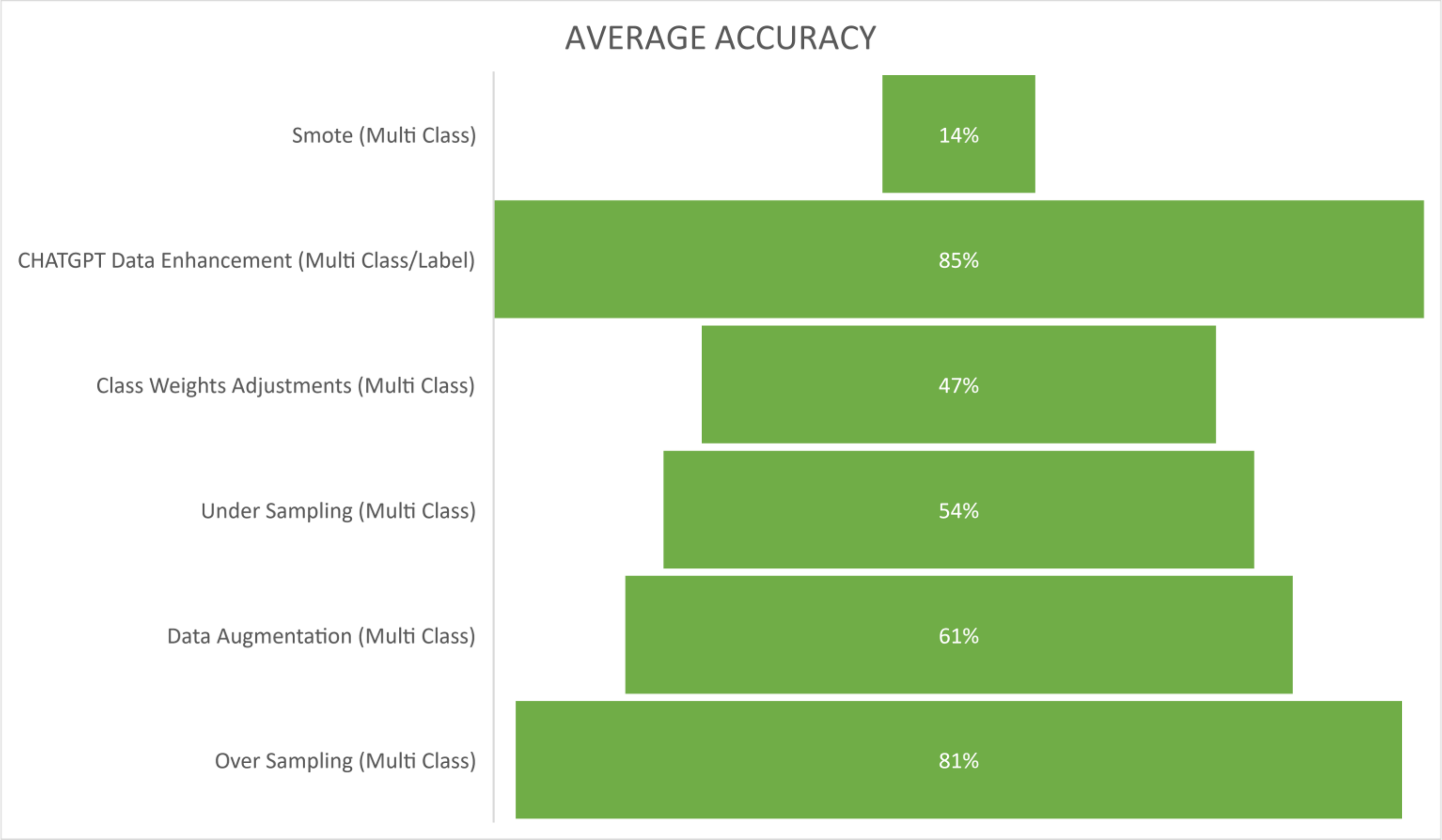
TECHNIQUES AND REMARKS

Techniques	Classification	Performance	Remarks on Trained Model
Under Sampling	Multi Class	Under Fitted	Reduced Data cause the model to under fit Model become too simple and does not capture the underlying patterns in the data, resulting in a poor fit on both the training and test sets. The model is unable to generalize well to new data and has high bias.
Over Sampling	Multi Class	Over Fitted	Over Sampling did maintain Highest accuracy but with the cost of duplication and

			over fitting the overfitted model over generalized and fits too much to the training data results in compromised performance to unseen data.
Class Weight Adjustment	Multi Class	Over Fitted	Class weight Adjustments got over generalized to the category drink with the highest preference weight and under fits towards other that dropped its overall performance.
Smote	Multi Class	Over Fitted	Smote being the model with least accuracy got dropped in performance due to the synthetically generated sentences that breaks out the sentences structure and generated the sentences with no contextual or semantic meaning at all because smote works well for numeric data. So the model trained on biased data returned unreliable results as expected.
Data Augmentation	Multi Class	Over Fitted	Although the dataset was balanced but model isn't able to

			perform well to the unseen data model did learn the under pattern but miss understood the appropriate classification because for the used augmented data set that contains the miss classified sentences due to unsupervised data preparation irrespective of the semantics through augmenters.
CHAT GPT data Enhancement	Multi Class Multi Label	Balanced	Chat GPT generated dataset with combined original dataset after pre processing has proved significant impact on performance of the model as we enhanced the dataset in both Qualitative and Quantitative dimension thus make our model perfect for unseen data.

COMPARISON CHART



SENTENCE ASPECTS CLASSIFICATION

CLASSIFICATION TYPE

Multiple Class Multiple Label Classification

DATA CONSIDERATION

Sentences and Aspects Tables are inner joined to enlist the sentences with its corresponding Aspects.

Tables considered are Sentences with joined Aspects.

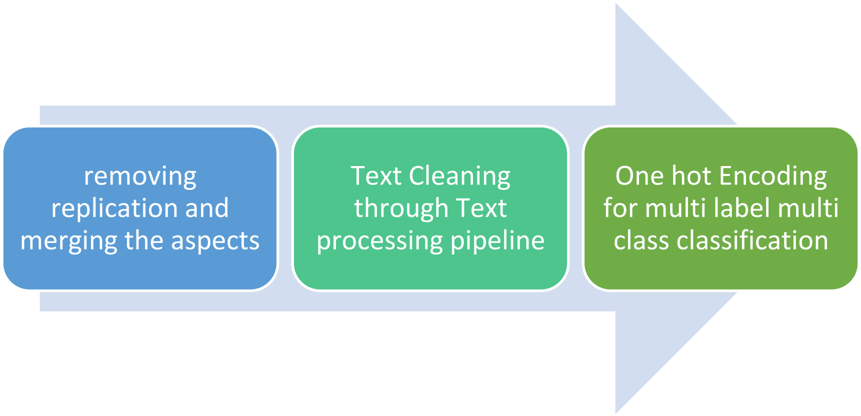
Columns considered are Sentence ID, Sentence, Aspects

ORIGINAL DATA LIMITATIONS

- Lack of Richness against most of Aspects
- Lack of Quality data
- Random and non-contextual Entries
- Lack of Quantity
- Imbalanced Data with the majority and minority class difference of about 90%.

DATA PREPARATION AND ONE HOT ENCODING

Problem with original Data is that every sentence has been replicated against each of its corresponding aspects, so the data needs to be preprocessed in following ways.



removing replication and merging the aspects

Text Cleaning through Text processing pipeline

One hot Encoding for multi label multi class classification

OUTPUT CLASSES OR ASPECTS

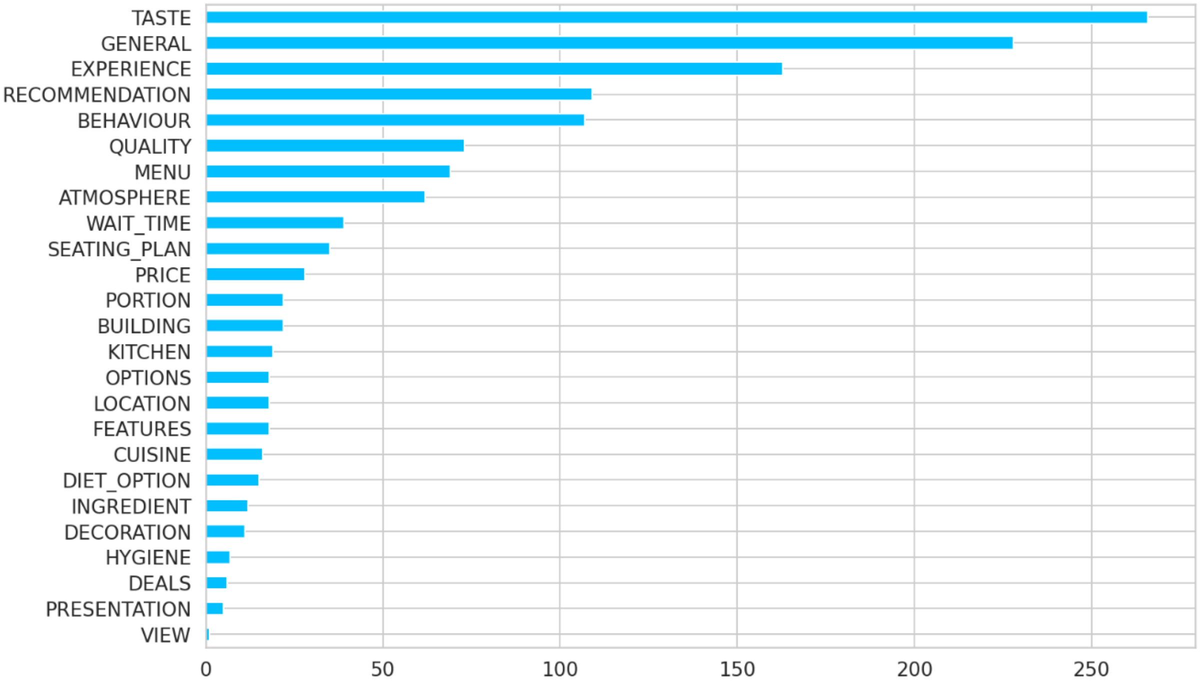
About 25 Aspects are considered that every sentence may have multiple aspects associated with it.

1. ATMOSPHERE
2. BEHAVIOUR
3. BUILDING
4. CUISINE
5. DEALS
6. DECORATION
7. DIET_OPTION
8. EXPERIENCE
9. FEATURES
10. GENERAL
11. HYGIENE
12. INGREDIENT
13. KITCHEN
14. LOCATION
15. MENU
16. OPTIONS
17. PORTION
18. PRESENTATION
19. PRICE
20. QUALITY
21. RECOMMENDATION
22. SEATING_PLAN
23. TASTE
24. VIEW
25. WAIT_TIME

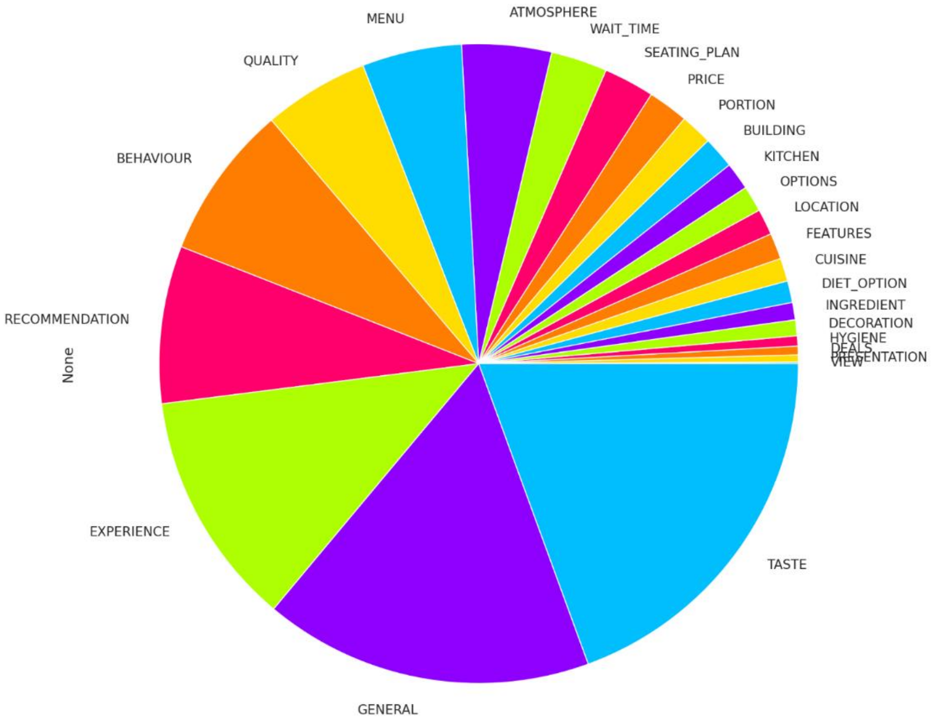
DATA PARTICIPATION VISULIZATION

ASPECTS	DATA COUNT
VIEW	1
PRESENTATION	5
DEALS	6
HYGIENE	7
DECORATION	11
INGREDIENT	12
DIET_OPTION	15
CUISINE	16
FEATURES	18
LOCATION	18
OPTIONS	18
KITCHEN	19
BUILDING	22
PORTION	22
PRICE	28
SEATING_PLAN	35
WAIT_TIME	39
ATMOSPHERE	62
MENU	69
QUALITY	73
BEHAVIOUR	107
RECOMMENDATION	109
EXPERIENCE	163
GENERAL	228
TASTE	266

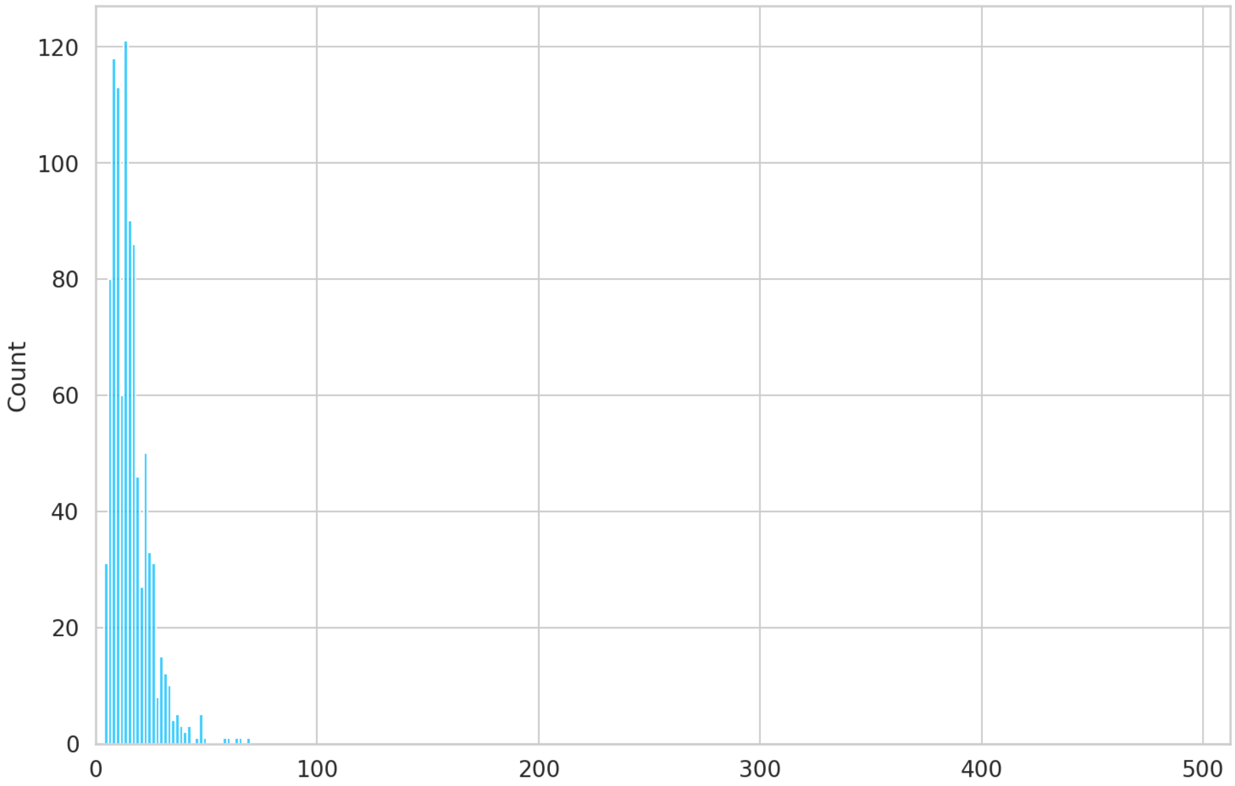
BAR PLOT



PIE PLOT



TOKENS AGAINST SENTENCES



OUR STRATEGY

As from the previous research we have conclude that Bert-base-cased performed well for multi label multi class classification problem and in contrasts Roberta large done great job in multi class classification as for aspects classification is concerned its is justified that we will be going with Bert-base-cased for training as it is multi label multi class classification.

FINE TURNING BERT MULTI-LABEL MULTI-CLASS CLASSIFICATION

Aspects extraction from the sentences being the most important part of the project we will be performing the classification using Bert for the sentence aspects classification which may be explicit or implicitly mentioned in the sentence.

CODE LINK FOR MODEL TRAINED ON ORIGINAL DATA

https://colab.research.google.com/drive/1Nd-Y3o7EBt3tUC-3INwivZNqzQTVrJXf?usp=share_link

CONSIDERED MODEL

Bert-base-cased because of its high accuracy in multi class multi label classification.

STRATIFICATION PROBLEM

As the original data is heavily imbalanced with least most accruing class with count1 and majority class count with around 266 upon train test split to maintain the original proportion of aspects in both train and validation dataset we might be using stratification but following problems occurred which are:

1. As the splitting demands even data but the minority class has the entry with 1 count only.
2. Secondly the most important issue is the stratification of multi label multi class classification dataset because its not that easy as for multi class classification stratification is much important for our data set to evenly split the train and validation data frame with equal proportion.

SOLUTION

1. Add one or more entries for the minority class to make the data even.
2. From Reference to the medium.com proposed solution and other solution for multi label multi class data splitting with stratification. But the appropriate solutions are to use ITERATIVE_TRAIN_TEST_SPLIT and MULTI_LABEL_STRATIFIED_SUFFLE_SPLIT from scikit multi learn.

ITERATIVE_TRAIN_TEST_SPLIT (Lower Granularity Level towards Stratification)

MULTI_LABEL_STRATIFIED_SUFFLE_SPLIT (Greater Granularity Level towards Stratification)

USED STRATIFICATION TECHNIQUE

MULTI_LABEL_STRATIFIED_SUFFLE_SPLIT Because of its higher granularity level and precision in considering the aspects/ categories with least most occurrence (minority Classes) to have same data distribution proportion in both train and validation dataset for balanced Training using Train Data frame and evaluation using Validation Data frame.

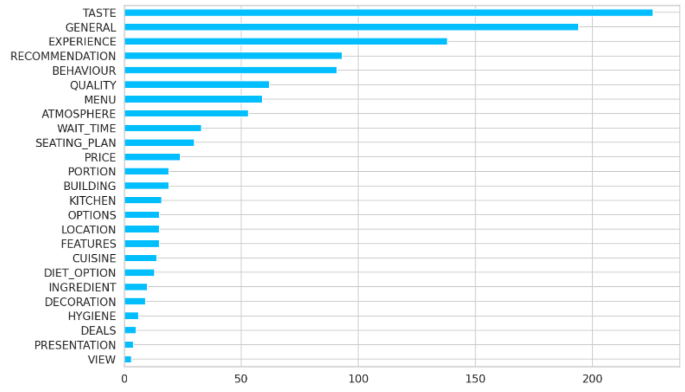
DATA VISUALIZATION AFTER TRAIN TEST SPLIT STRATIFCATION AND RESOLVED ISSUES

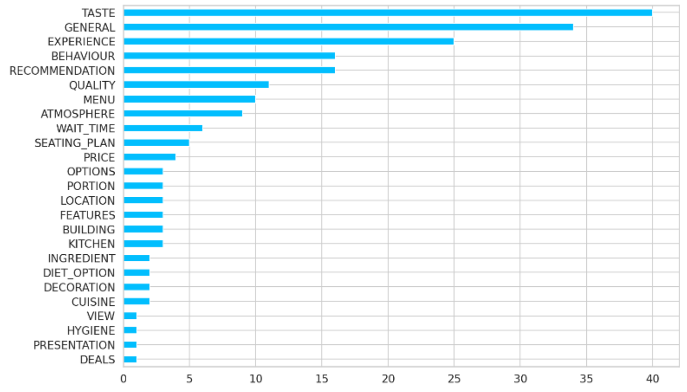
TRAIN DATA FRAME	VALIDATION DATA FRAME
DATA COUNT	

ASPECTS	Number of entries
VIEW	3
PRESENTATION	4
DEALS	5
HYGIENE	6
DECORATION	9
INGREDIENT	10
DIET_OPTION	13
CUISINE	14
FEATURES	15
LOCATION	15
OPTIONS	15
KITCHEN	16
BUILDING	19
PORTION	19
PRICE	24
SEATING_PLAN	30
WAIT_TIME	33
ATMOSPHERE	53
MENU	59
QUALITY	62
BEHAVIOUR	91
RECOMMENDATION	93
EXPERIENCE	138
GENERAL	194
TASTE	226

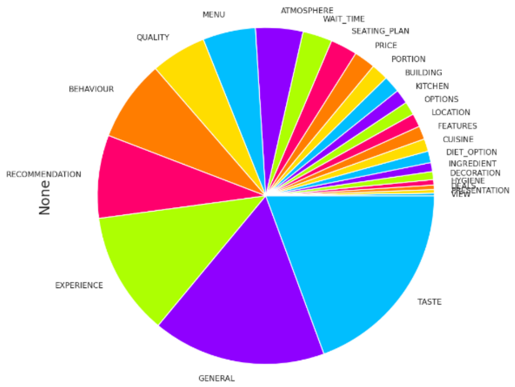
ASPECTS	Count
DEALS	1
PRESENTATION	1
HYGIENE	1
VIEW	1
CUISINE	2
DECORATION	2
DIET_OPTION	2
INGREDIENT	2
KITCHEN	3
BUILDING	3
FEATURES	3
LOCATION	3
PORTION	3
OPTIONS	3
PRICE	4
SEATING_PLAN	5
WAIT_TIME	6
ATMOSPHERE	9
MENU	10
QUALITY	11
RECOMMENDATION	16
BEHAVIOUR	16
EXPERIENCE	25
GENERAL	34
TASTE	40

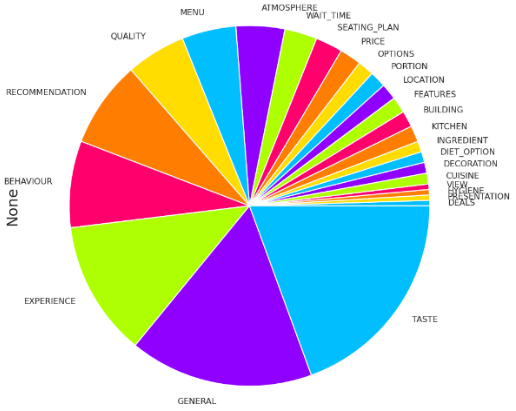
BAR POT





PIE PLOT





MODEL CONFIGURATIONS

N_EPOCHS = 15

BATCH_SIZE = 1

Steps per Epoch =length (Train Data Frame) // BATCH_SIZE

Training Steps = Steps per Epoch * N_EPOCHS

Warmup Steps = Training Steps // 5

ACTIVATON FUNCTION

Sigmoid for treating every individual aspect with independent probability.

WHY SIGMOID

SoftMax is often used for multi-label multi-class classification tasks. However, in our specific case, we will be using sigmoid during the training of all models. This is because our training and predicted tensors can be quite large (up to 25 or 31), and SoftMax may not be the most suitable option. SoftMax normalizes the tensor to make the aggregate sum up to 1, which results in very low participation probabilities for each class. This can lead to confusion in thresholding values. Therefore, using sigmoid to normalize each item in the tensor independently may be a better option.

TRAINING AND VALIDATION LOSS

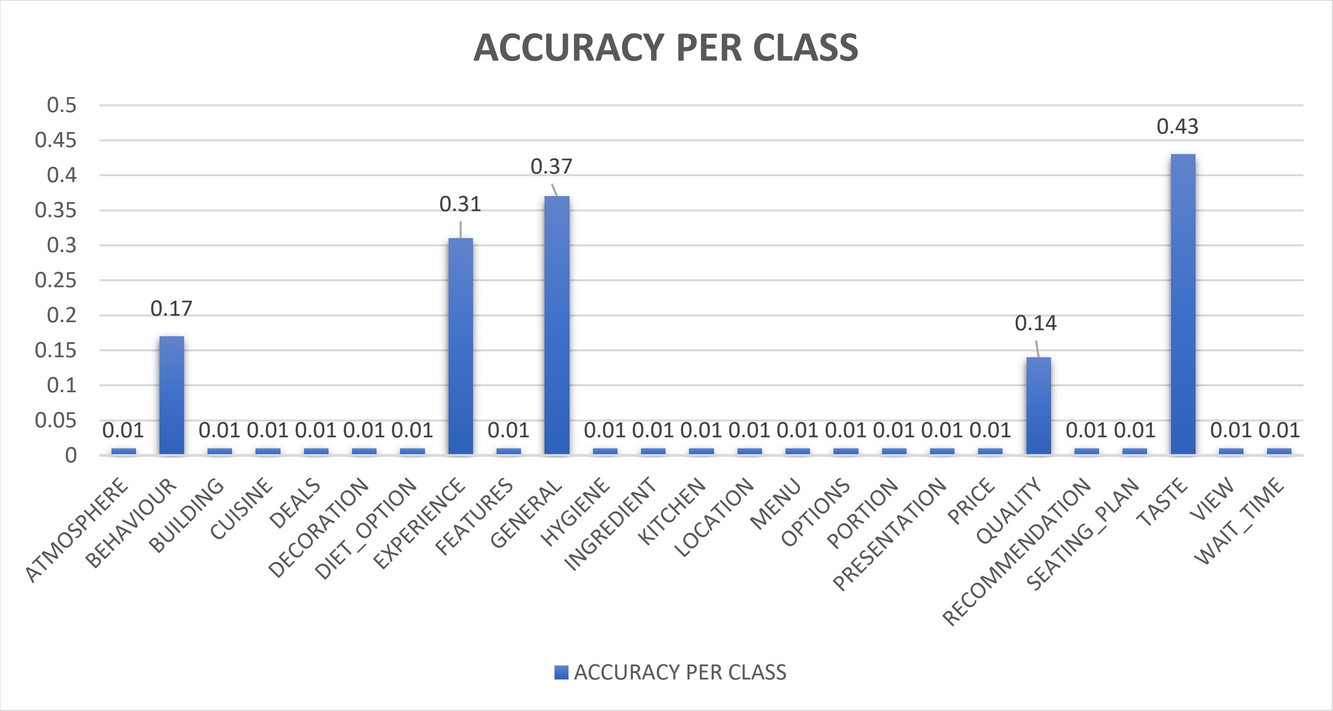
Best Check Point with Validation and Training Loss was at Epoch 7

Training Loss =0.215

Validation Loss =0.250

Average F1 Score = 0.27

ACCURACY PERCENTAGE PER CLASS (ASPECTS)



COSINE SIMILAIRTY

Cosine Similarity : tensor(0.3220) at optimal probability thresholding value of 0.19 I-e 19 %

CLASSIFCATION REPORT

Aspects	Precision	Recall	F1-score	Support
ATMOSPHERE	0.00	0.00	0.00	9
BEHAVIOUR	0.10	0.44	0.17	16
BUILDING	0.00	0.00	0.00	3
CUISINE	0.00	0.00	0.00	2
DEALS	0.00	0.00	0.00	1
DECORATION	0.00	0.00	0.00	2
DIET_OPTION	0.00	0.00	0.00	2
EXPERIENCE	0.18	1.00	0.31	25
FEATURES	0.00	0.00	0.00	3
GENERAL	0.23	1.00	0.37	34
HYGIENE	0.00	0.00	0.00	1
INGREDIENT	0.00	0.00	0.00	2
KITCHEN	0.00	0.00	0.00	3
LOCATION	0.00	0.00	0.00	3
MENU	0.00	0.00	0.00	10
OPTIONS	0.00	0.00	0.00	3
PORTION	0.00	0.00	0.00	3
PRESENTATION	0.00	0.00	0.00	1
PRICE	0.00	0.00	0.00	4
QUALITY	0.07	1.00	0.14	11
RECOMMENDATION	0.00	0.00	0.00	16
SEATING_PLAN	0.00	0.00	0.00	5
TASTE	0.27	1.00	0.43	40
VIEW	0.00	0.00	0.00	1
WAIT_TIME	0.00	0.00	0.00	6

Average Type	precision	recall	f1-score	support
micro avg	0.18	0.57	0.27	206
macro avg	0.03	0.18	0.06	206
weighted avg	0.12	0.57	0.20	206
samples avg	0.18	0.60	0.27	206

CONCULSION

From the results we could conclude that original data that was manually annotated is limited and may be enough to train the Machine learning model for accurate aspects classification such that accuracy dropped up to 32% upon optimal thresholding value. Furthermore, the data got over generalized for the GENERAL aspect the data needs to be revised because the data got legged in both qualitative and quantitative dimensions as well the data is highly imbalanced in the form of individual aspect participation to Data set shown in figure aspects. That also results in limited stratification to the training and validation data. hence it is justified that trained model is worthless to be used for unseen data.

PROPOSED TECHNIQUES CONVINIENCE

from the previous research we now analyzed the performance and accuracy of all data balancing techniques (refer figure DATA BALANCING TECHNIQUES AND ACCURACY COMPARISON) and come up with the effective techniques to be considered for Quality data balancing and enhancing. Data Balancing techniques and their justification for selection is given below.

TECHNIQUES	DOMAIN	SELECTION REMARKS	JUSTIFICATION
Over Sampling	Ensemble Techniques	Not recommended	For Over Sampling the Original Data the Data got introduced with duplicates considering our case if the minority class has 3 entries and in contrast the majority class own up to 300 entries if we consider over sampling we got introduced with 280+ duplicates of 3 entries of minority class to balance with majority class size. Same would be going to happen for other minority classes. Such that the trained

			model got over fitted to the training data and may not be worthy to use for unseen data.
Under Sampling	Ensemble Techniques	Not recommended	Under sampling is concerned with data shrinking up to the least most occurred Aspect for our case the least most category occurrence is about 3 entries for VIEW Aspect that doesn't make sense for training at all. Even if we have the data entries more but due to under sampling, we might make the model under fitted.
Class weight Adjustment	Ensemble Techniques	Not recommended	Class weight adjustment would be preferring the most least occurred aspects such as VIEW,DEALS ,HYGINE etc and may got over generalized for those aspect and further more the data isn't enough for those aspects too so it isn't worth using data balancing techniques but to use data generation techniques.
Smote	Ensemble Techniques	Not recommended	As for the Smote we already discussed the problem for its applicability on Numeric data only.

			That wasn't worth to use for textual Data.
Data Augmentation	Statistical Data Augmentation Model	Not recommended	As per Data Augmentation is concerned the reason behind its rejection is its augmented out of context sentences , augmented sentences doesn't consider the aspect they belong to because the augmenter is independent of that information it only gets the sentence and augment that sentence so because of its wrong sentences as well as its low accuracy and performance Data augmentation isn't worthy to be used.
Chat GPT Data Enrichment	AI Data Generation Model	Recommended	Data generated by the Chat GPT may got duplication but we might not deny the quality of the data, from Chat GPT we already generated the quality data and trained Bert with high accuracy. for aspects data generation and balancing its is the worthy and efficient solution we might peruse.

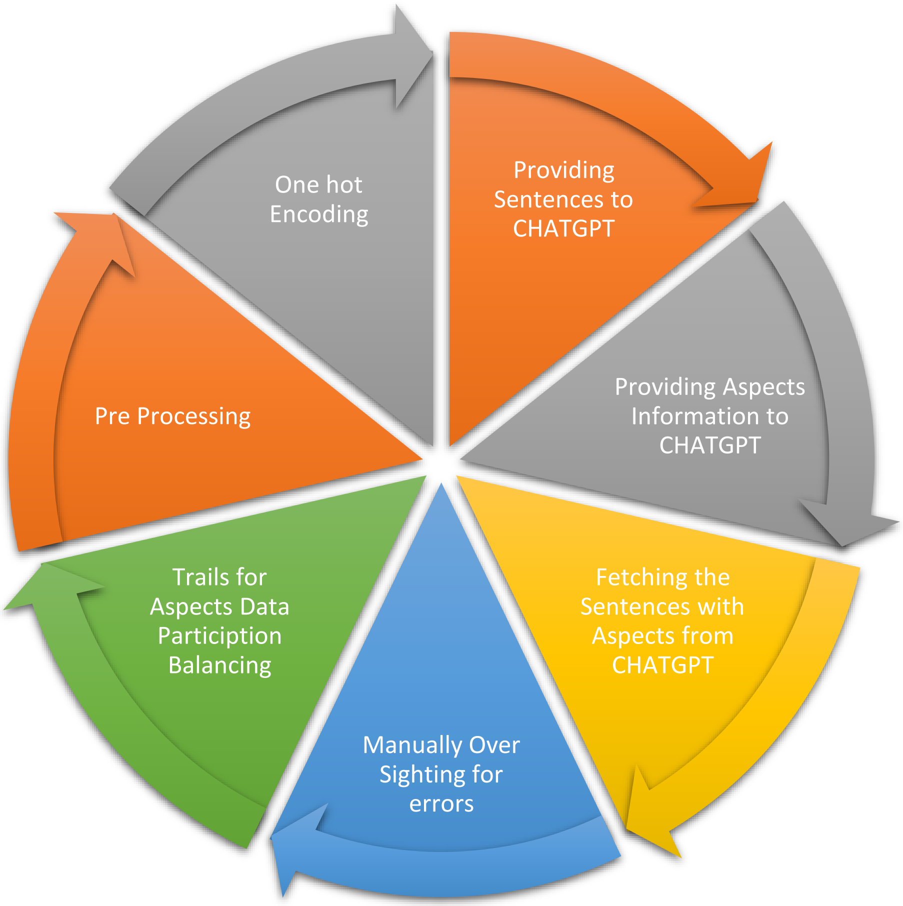
DATA ENHANCMENT USING AI BASED MODELS

MODEL USED FOR DATA GENERATION

CHAT GPT

OUT STRATEGY

As we know that we had already generated the sentences with their categories and preprocessed them accordingly. We will be perusing those sentences for the aspect's extraction from the Chat GPT such that providing the chat GPT with the baccates of already stored sentences data and providing the information of considered aspects to classify the sentences with aspects they might belongs to. This process might be difficult and hard because it involves extensive repetitive cycle as below but worth it.


One hot Encoding
Providing Sentences to CHATGPT
Providing Aspects Information to CHATGPT
Fetching the Sentences with Aspects from CHATGPT
Manually Over Sighting for errors
Trails for Aspects Data Particiption Balancing
Pre Processing

DATA PROBLEMS AND PRE-PROCESSING

Data Generated by the CHATGPT have some draw Backs as following.

- Lot of Duplications and replicated sentences.
- Miss classified Sentences.
- Sentences with other then 25 Aspects that we considered.
- Preprocessing required to remove the unwanted entries and data cleaning steps are required for textual corrections.
- One Hot encoding of updated data.

DATA DESCRIPTION

CHAT GPT GENERATED DATASET LINK

https://docs.google.com/spreadsheets/d/1QcmC2pL69wpp9MrkKsd0T3j4mBi_r6lQ/edit?usp=share_link&ouid=109317857182659250370&rtpof=true&sd=true

COMBINED AND PROCESSED DATA SET LINK

https://drive.google.com/file/d/1yoKdyRh72aad19uf0AXshdoTiChTpXmD/view?usp=share_link

DATASET	COUNT
Original Dataset	1491
Generated Dataset	11979
Combined Dataset	12747

After Rearranging, Merging, Dropping Duplicates Individual replicated Sentences along with their corresponding Aspects.

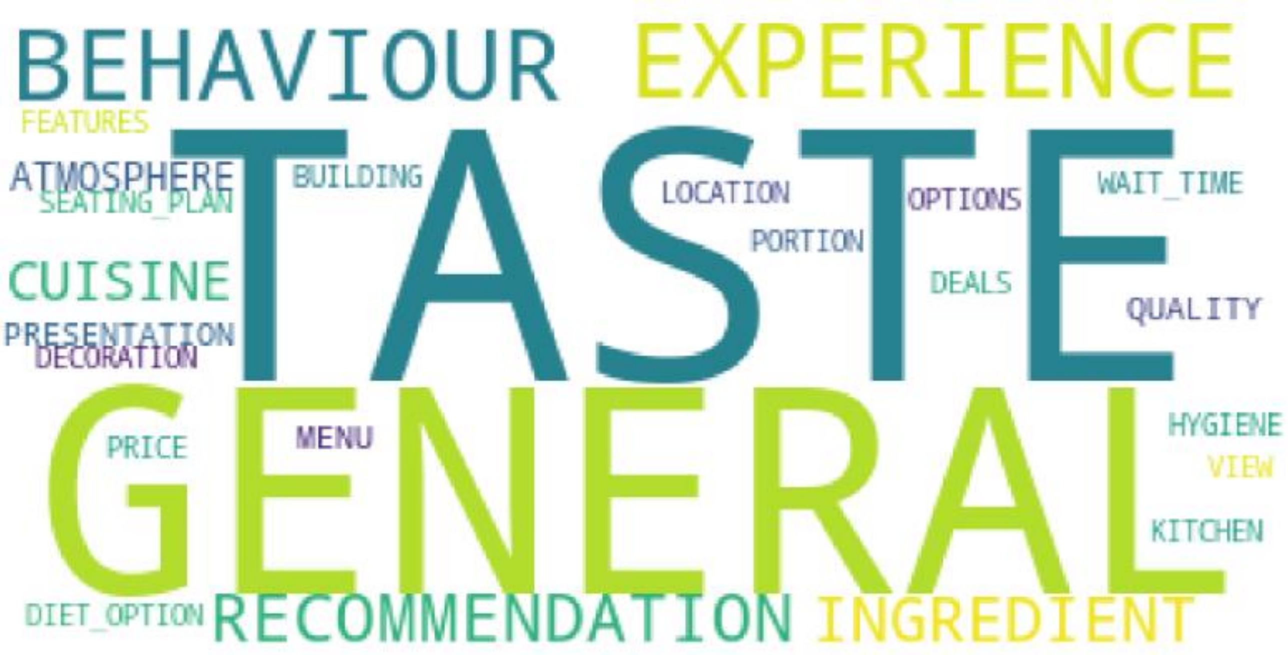
Dataset Count: 8787

CLASS COUNTS

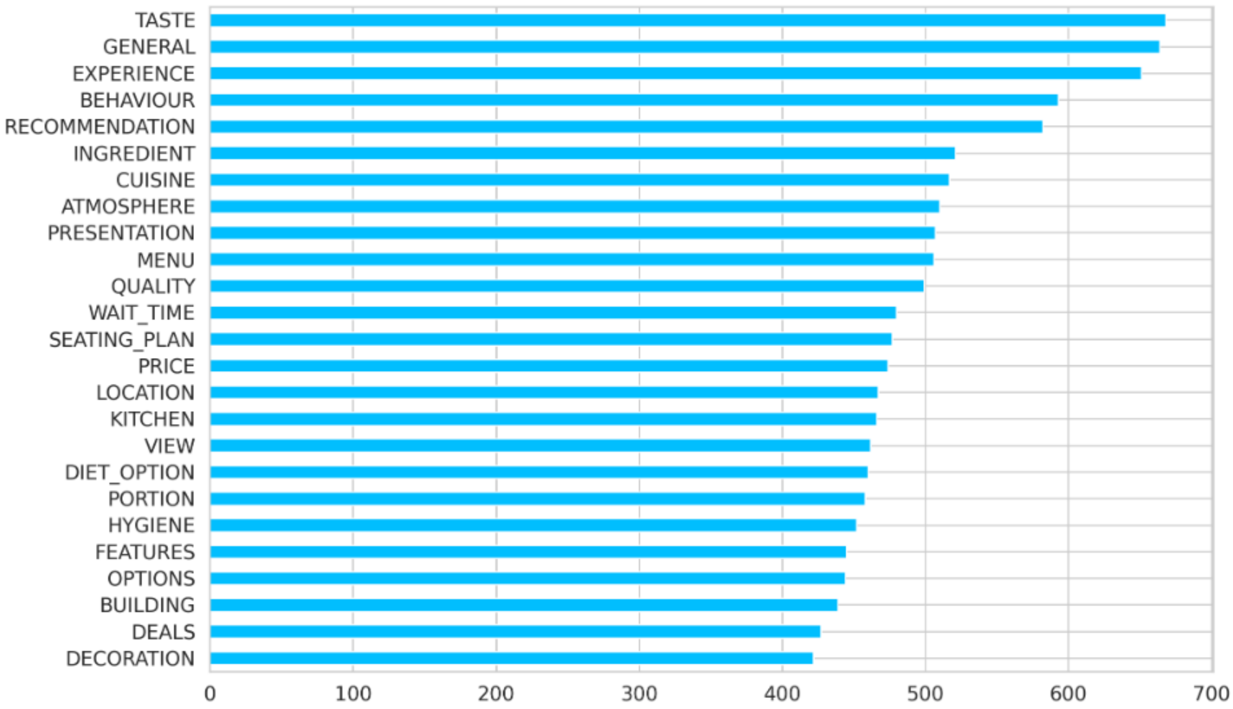
ASPECT	COUNT
DECORATION	422
DEALS	427
BUILDING	439
OPTIONS	444
FEATURES	445
HYGIENE	452
PORTION	458
DIET_OPTION	460
VIEW	462
KITCHEN	466
LOCATION	467
PRICE	474
SEATING_PLAN	477
WAIT_TIME	480
QUALITY	499
MENU	506
PRESENTATION	507
ATMOSPHERE	510
CUISINE	517
INGREDIENT	521
RECOMMENDATION	582
BEHAVIOUR	593
EXPERIENCE	651
GENERAL	664
TASTE	668

DATA VISULIZATION

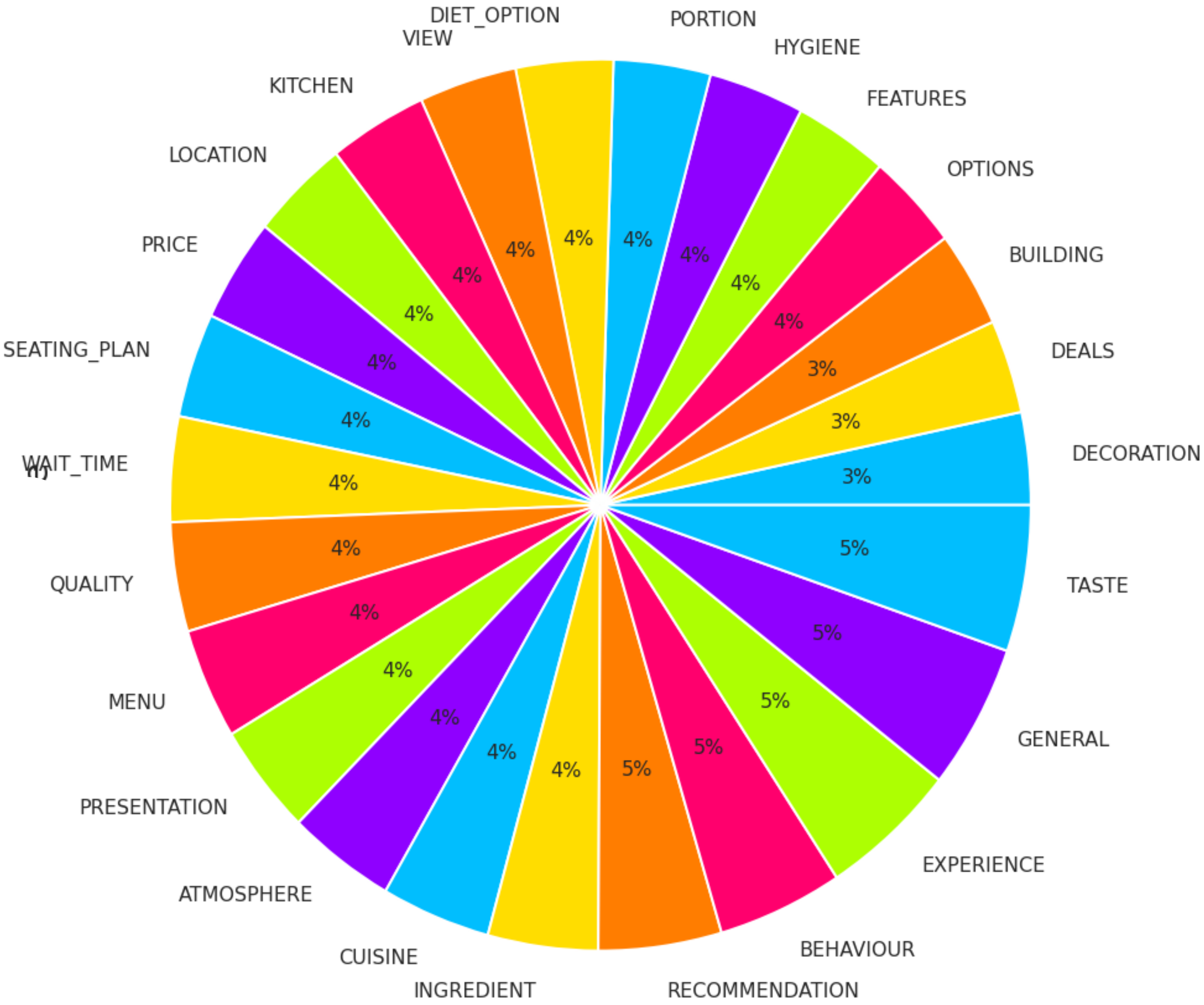
WORD CLOUD



BAR PLOT



PIE PLOT


DIET_OPTION
4%
PORTION
4%
HYGIENE
4%
FEATURES
4%
OPTIONS
4%
BUILDING
3%
DEALS
3%
DECORATION
3%
TASTE
5%
GENERAL
5%
EXPERIENCE
5%
BEHAVIOUR
5%
RECOMMENDATION
5%
INGREDIENT
4%
CUISINE
4%
ATMOSPHERE
4%
PRESENTATION
4%
MENU
4%
QUALITY
4%
(h)
WAIT_TIME
4%
SEATING_PLAN
4%
PRICE
4%
LOCATION
4%
KITCHEN
4%
VIEW
4%

DATA AFTER TRAIN TEST SPLITTING

TRAIN DATA FRAME	VALIDATION DATA FRAME

DATA COUNT

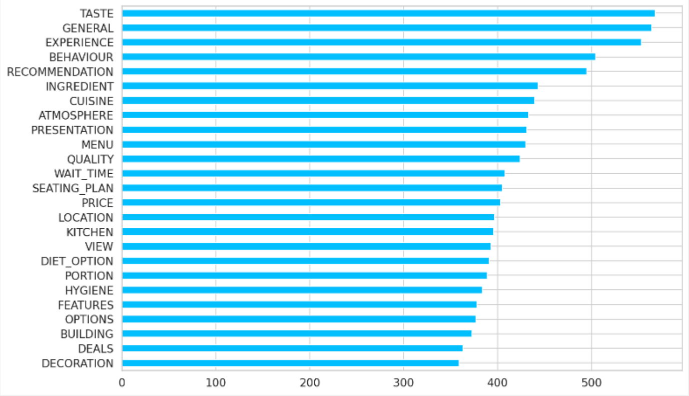
ASPECT	COUNT
DECORATION	359
DEALS	363
BUILDING	373
OPTIONS	377
FEATURES	378
HYGIENE	384
PORTION	389
DIET_OPTION	391
VIEW	393
KITCHEN	396
LOCATION	397
PRICE	403
SEATING_PLAN	405
WAIT_TIME	408
QUALITY	424
MENU	430
PRESENTATION	431
ATMOSPHERE	433
CUISINE	439
INGREDIENT	443
RECOMMENDATION	495

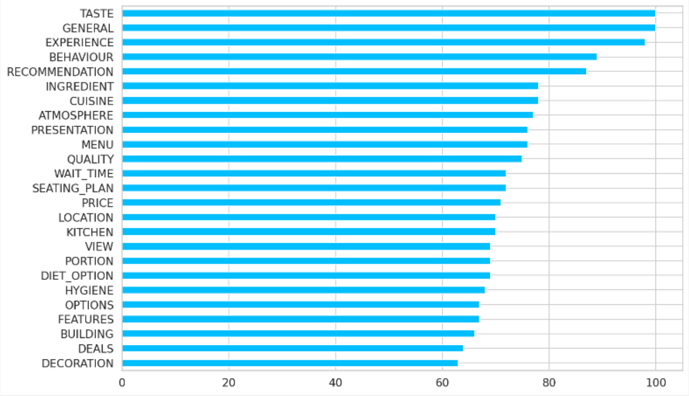
ASPECT	COUNT
DECORATION	63
DEALS	64
BUILDING	66
FEATURES	67
OPTIONS	67
HYGIENE	68
DIET_OPTION	69
PORTION	69
VIEW	69
KITCHEN	70
LOCATION	70
PRICE	71
SEATING_PLAN	72
WAIT_TIME	72
QUALITY	75
MENU	76
PRESENTATION	76
ATMOSPHERE	77
CUISINE	78
INGREDIENT	78

BEHAVIOUR	504
EXPERIENCE	553
GENERAL	564
TASTE	568

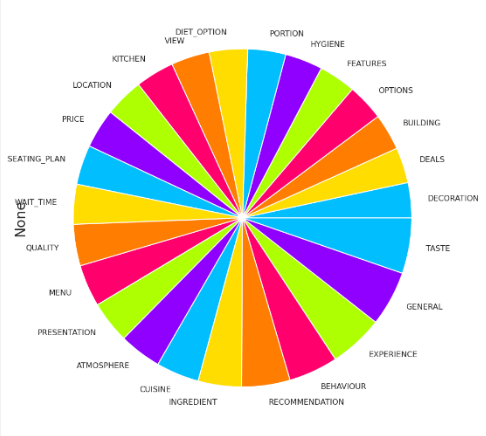
RECOMMENDATION	87
BEHAVIOUR	89
EXPERIENCE	98
GENERAL	100
TASTE	100

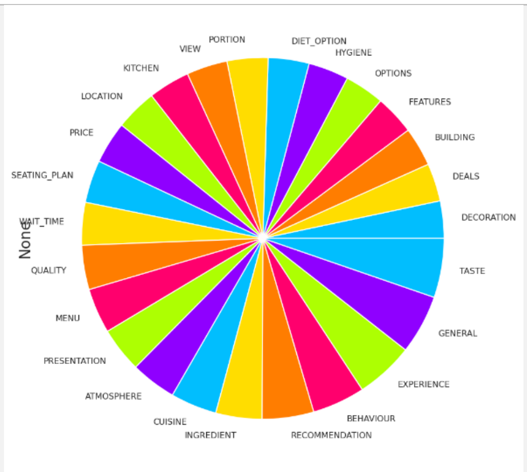
BAR POT





PIE PLOT





DATA LIMITATIONS AND BALANCING

As we explore Aspects classification under the Multi Class Multi Label Classification problem, we find that the data generated by CHATGPT is mostly limited to sentences with a single aspect, rather than multiple aspects that they might be associated with. This is due to the processing limitations of CHATGPT, which shifted the focus to generating non-contextual domain sentences and aspects. However, it has proven effective in generating sentences with a single aspect. As far as the data balancing is concerned Based on the above visualization, we can conclude that the data is almost balanced across each Aspect class, thanks to an extensive repetition cycle of queries to enhance the minority classes' data.

FINE TUNING BERT ON COMBINED DATA

CODE LINK

https://colab.research.google.com/drive/1WP-rh5ZDiplEO6-iOBIw9ZRTNcM-5AvV?usp=share_link

CONFIGURATION

N_EPOCHS = 15

BATCH_SIZE = 04

Steps per Epoch =length (Train Data Frame) // BATCH_SIZE

Training Steps = Steps per Epoch * N_EPOCHS

Warmup Steps = Training Steps // 5

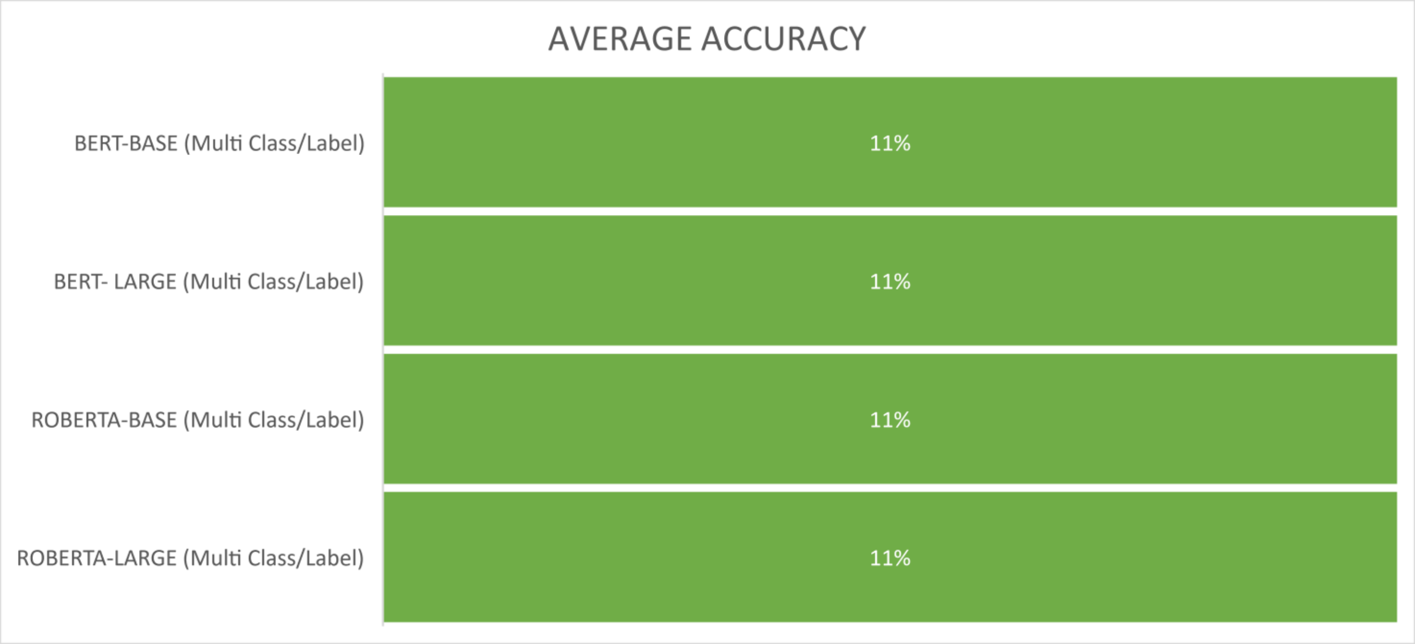
MODELS AND RESULTS

Model	Best Training Loss	Best Validation Loss	Cosine Similarity at Optimal Threshold	Weighted F1 Score at Optimal Threshold	Remarks
Bert-Base	0.310	0.324	0.229	0.11	As Bert-base is trained in less data as compare to Bert-large but both Bert

					variants under performed in terms of accuracy and performance these results may be attributed to the limitations of the Training Dataset.
Bert-Large	0.445	0.259	0.230	0.11	When it comes to F1 score, Bert-large is in the same league as Roberta-base and Roberta-large. However, when it comes to accuracy over the trained data, Bert-large falls short of expectations. The model's inability to perform as well as the others in terms of accuracy is a major drawback. This means that while Bert-large may be suitable for certain tasks, it may not be the best choice for our training dataset that is limited to single aspect in majority.
Roberta-Base	0.225	0.25	0.2307	0.11	Roberta-base, on the other hand, performed poorly in terms of accuracy. The model failed to learn the hidden patterns in the data, resulting in a significant reduction in accuracy. Given its lackluster performance, it's hard to make the model performance effective without revising the dataset.
Roberta-Large	0.199	0.229	0.2307	0.11	Similarly, Roberta-large also struggled to perform well. This can be attributed to the training dataset, which may not have been optimized for the model. While the model itself may

					not be at fault, the poor accuracy results make it challenging to use in many applications. Ultimately, revising the dataset for any given task requires careful consideration of the strengths and weaknesses of each option.

COMPARISON CHART



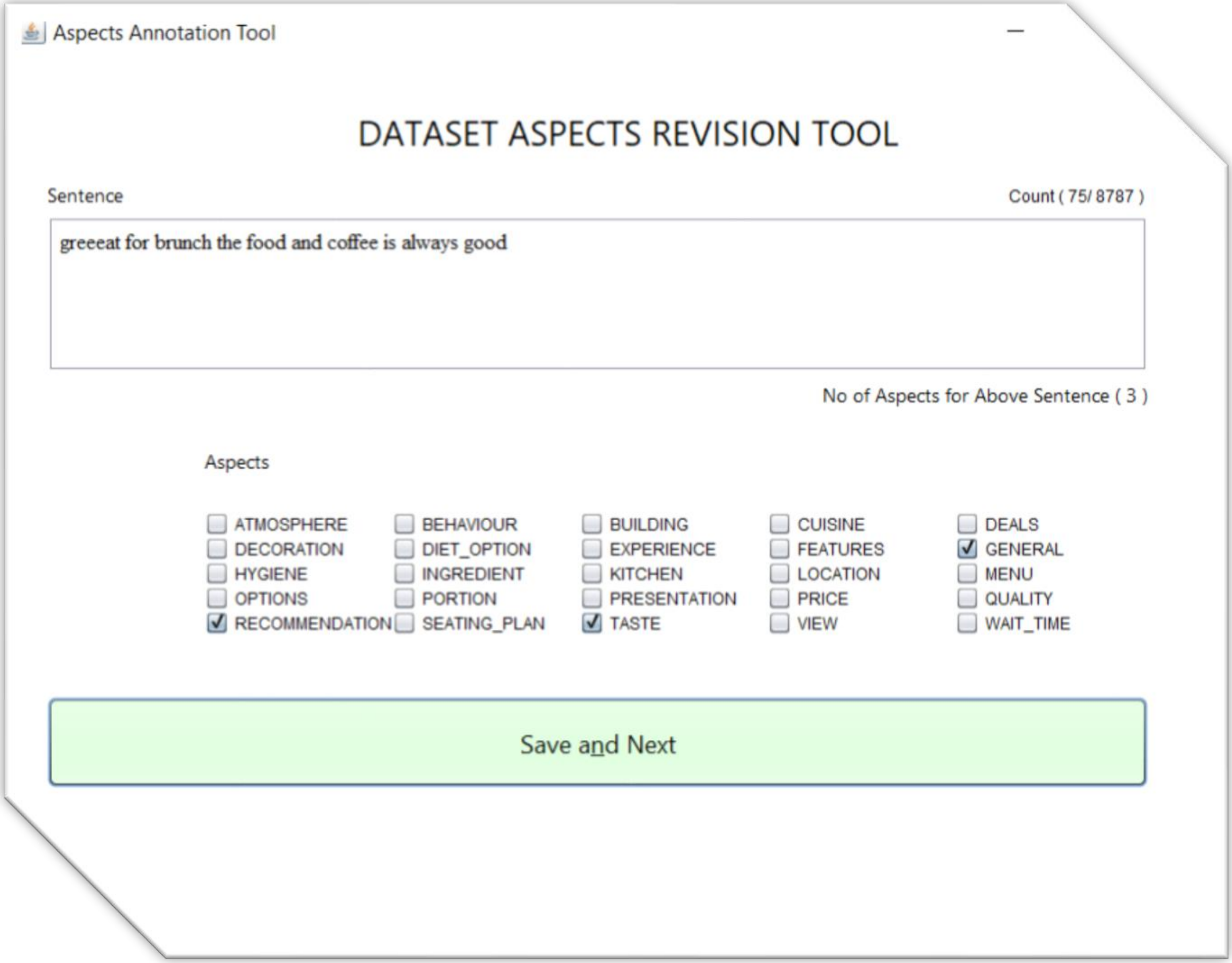
CONCLUSION

After conducting a thorough analysis and comparison, we have arrived at the conclusion that the performance of all Bert variants is not satisfactory when tested against the combined (CHATGPT generated and original) dataset. This could be attributed to the limitations of the data generated by CHATGPT, which although balanced and enhanced, has a majority of sentences associated with only a single aspect. As a result, the models are unable to effectively learn the hidden patterns in the training dataset and classify the aspects with high precision and accuracy. Even though some models have low training and validation losses but on optimal thresholding values, they generate poor predictions with low probability ranges and may be another reason behind the low accuracy is the inability to determine the optimal threshold.

To overcome this challenge, manual supervision of the data is necessary, either by directly modifying the data or by using a customized data revision tool to associate single sentences with multiple aspects. While this procedure may be lengthy, but worth it. For our analysis, we recommend using **Roberta-base**, as it outperformed all other BERT variants and has the potential to produce considerable results although its accuracy score is low but Roberta-base predict the aspects with high probability score so it would be good choice to go with Roberta-base for training on revised dataset.

CUSTOMIZED TOOL

We had already developed the Revision tool that will fetch each sentence and its aspect and allow the user to simply elect or deselect existing/aspects on his will in a effective and efficient way. upon saving the sentence with the revised aspects will get stored in new CSV file for later Training.



DATASET ASPECTS REVISION TOOL

Sentence

Count (75/ 8787)

greeeat for brunch the food and coffee is always good

No of Aspects for Above Sentence (3)

Aspects

☐ ATMOSPHERE ☐ BEHAVIOUR ☐ BUILDING ☐ CUISINE ☐ DEALS
☐ DECORATION ☐ DIET_OPTION ☐ EXPERIENCE ☐ FEATURES ☑ GENERAL
☐ HYGIENE ☐ INGREDIENT ☐ KITCHEN ☐ LOCATION ☐ MENU
☐ OPTIONS ☐ PORTION ☐ PRESENTATION ☐ PRICE ☐ QUALITY
☑ RECOMMENDATION ☐ SEATING_PLAN ☑ TASTE ☐ VIEW ☐ WAIT_TIME

MERGING ASPECTS AND CATEGOIRES MODEL

Merging of the Models is Possible in two ways.

- Independent Training and Combined Output
- Inter Dependent Training and Combined Output

MERGING BOTH MODELS TO TRAIN INDEPENDETTLY

Combining the Categories and Aspects to act as single model that would be able to predict the categories and aspects of the given sentence at once but without making the contributing models to be dependent on one another. Categories and aspects Models are trained independently from each other.

MODEL ARCHITECTURE

1. THROUGH PIPELINING

   The Previously trained both Models are Merged together in a pipeline to predict pass the Sentence to first model for Category Prediction and then to other Model for Aspects Extraction.

   CODE LINK

   https://drive.google.com/file/d/1_aySdtBw8YTM5HGb0599NZmGWH02S8VN/view?usp=share_link

   FINDINGS

   As both models are strictly independent in Training and evaluation, they have maintained the exact same accuracy as the individual Aspects and Category Model have.

2. THORUGH SEPARATE CLASSIFICATION HEAD AND COMBINED SUPERVISOR (MULTITASKING)

   The model has separate output layers for the categories and the aspects, and separate loss functions for each output. During training, the optimizer updates the model parameters for both the category and aspect output layers using the combined loss of both. However, the gradients are computed separately for each loss function and backpropagated independently for each output. The only characteristic is to combine the training of both models simultaneity and independently.

CODE LINK

https://drive.google.com/file/d/1ohAy9rMo9ZPBeFOLDTYtFlz49QwF3osW/view?usp=share_link

FINDINGS

Multitasking in BERT, which involves training the model on multiple tasks simultaneously, can have both positive and negative effects on accuracy.

Multitasking may promote better learning in some scenarios but on the other hand, multitasking can also lead to a reduction in accuracy if the tasks are too dissimilar as in our case and as well as if the model is not able to allocate sufficient resources to each task. This is because training on multiple tasks requires the model to divide its attention and resources among the tasks, which can lead to suboptimal performance on any individual task.

Additionally, multitasking can also increase the computational cost and training time of BERT, since the model must be trained on multiple tasks and with multiple loss functions simultaneously.

Overall, Merging the models through Multitasking leads to low accuracy and performance because of the issues discussed above so it may not worth using it.

MERGING BOTH MODELS TO TRAIN INTER DEPENDENTLY

To Train the Aspects and Categories classification to be a single model we could be using the dataset provided for the Categories Classification and Aspects Classification.

MODEL ARCHITECTURE

In inter dependence Training scenario there will be only output layer and there exist only one Classification head Classification of Aspects and Categories as well the model treats all labels as the same and concatenates the aspect and category labels to form a single vector of 31 labels. It then performs multi-label classification on this concatenated vector, without considering whether a particular label corresponds to an aspect or a category in such a way Bert would be enabled to learn the pattern inter dependently among aspects and categories.

DATASET CONSIDERATION AND PREPARATION

As we know that in both the Models are trained on Different datasets that is the Categories Dataset sentences are the subsets of the Aspect Dataset. To improve the accuracy, we enhanced the aspects data set that reached up to 8787 entries, but the Categories data set is about 2263 entries.

PROBLEM

The problem arises when we want to merge the individual datasets to make a single dataset with three columns as

- The first column should contain the sentence text.
- The second column should contain a list of one-hot encoded aspect labels, where each element in the list corresponds to a different aspect. The list should have the same length for every row, with a 1 in the position corresponding to the aspect(s) present in the sentence and 0s in all other positions.
- The third column should contain a one-hot encoded category label, with a 1 in the position corresponding to the category of the sentence and 0s in all other positions.

As every sentence have its associated categories and aspects but the dataset for the aspects is greater than the categories dataset concluding that about from categories dataset only 980 sentences have their Corresponding Aspects and Categories rest of the 1283 sentences have only categories classified to them because the data is generated from CHATGPT regardless of their inter relation (Aspects and Categories) and on the other hand in Aspects data only 980 Sentences have their classified Categories and Aspects rest of the 7784 sentences do not have their category classified but only have aspects because of the same reason as discussed for Categories dataset.

Note: The data in Original Dataset is 980 in Categories and 1003 in Aspects dataset because of the inner join filtration as well as removed duplications and merging.

Data Set Essentials	Categories Dataset	Aspects Dataset
Total Count	2263	8787
Sentences with Both Aspects and Categories	980	1003

Sentences with only Aspects and not with Categories	0	7784
Sentence with Only Categories not with Aspects	1283	0

PROPOSED SOLUTION TO DETERMINE THE MISSING ASPECSTS OR CLASSIFICATION

MISSING SENTENCE CATEGORIES IN ASPECTS DATASET

1. For aspects data we will proposing the solution to Determine Aspects Parent Categories to Make the Hierarchal Distribution of the 25 Aspects to the 6 Categories through manual and critical Analysis of the relationship among the Aspects and Categories of Sentences.

   Once The Hierarchal Distribution will get achieved then we will be splitting the Aspects dataset sentences that doesn't have the Category specified into the data frame that have the replicated sentences if the multiple aspects are there and then for every sentence, we will be mapping its corresponding aspects to its hierarchal parent Category and will specifying that category against the considered sentence. And finally rearranging the aspects dataset to contain the sentence its associated aspects in List and its Categories set without duplication.

   ### IMPORTANT CONSIDERATION

   - If the sentence has the multiple aspects from same category, then the category label will be the Parent Category of those Aspects.
   - If the Sentence have the Multiple aspects but from different Categories than output label for category will be the Parents of those Aspects.
   - Ensure no Duplication in the Categories Label for single sentences.

2. Another solution could be using the Pre trained Categories Model to Predict the Sentences Categories.

MISSING SENTENCE ASPECTS IN CATEGOIRES DATASET

For Missing sentences Aspects could be determined by the given below procedure.

1. We will be using the pre trained model to extract the Aspects from the 1283 sentences that have been missing associated aspects in Category dataset and Finally thresholding those aspects list to select top Aspects and specifying the aspect against the considered sentences but this may not be the worthy solution because the trained model for the aspect isn't that much precise and accurate so it may be a best option to go with annotation tool that we created for aspects dataset revision.

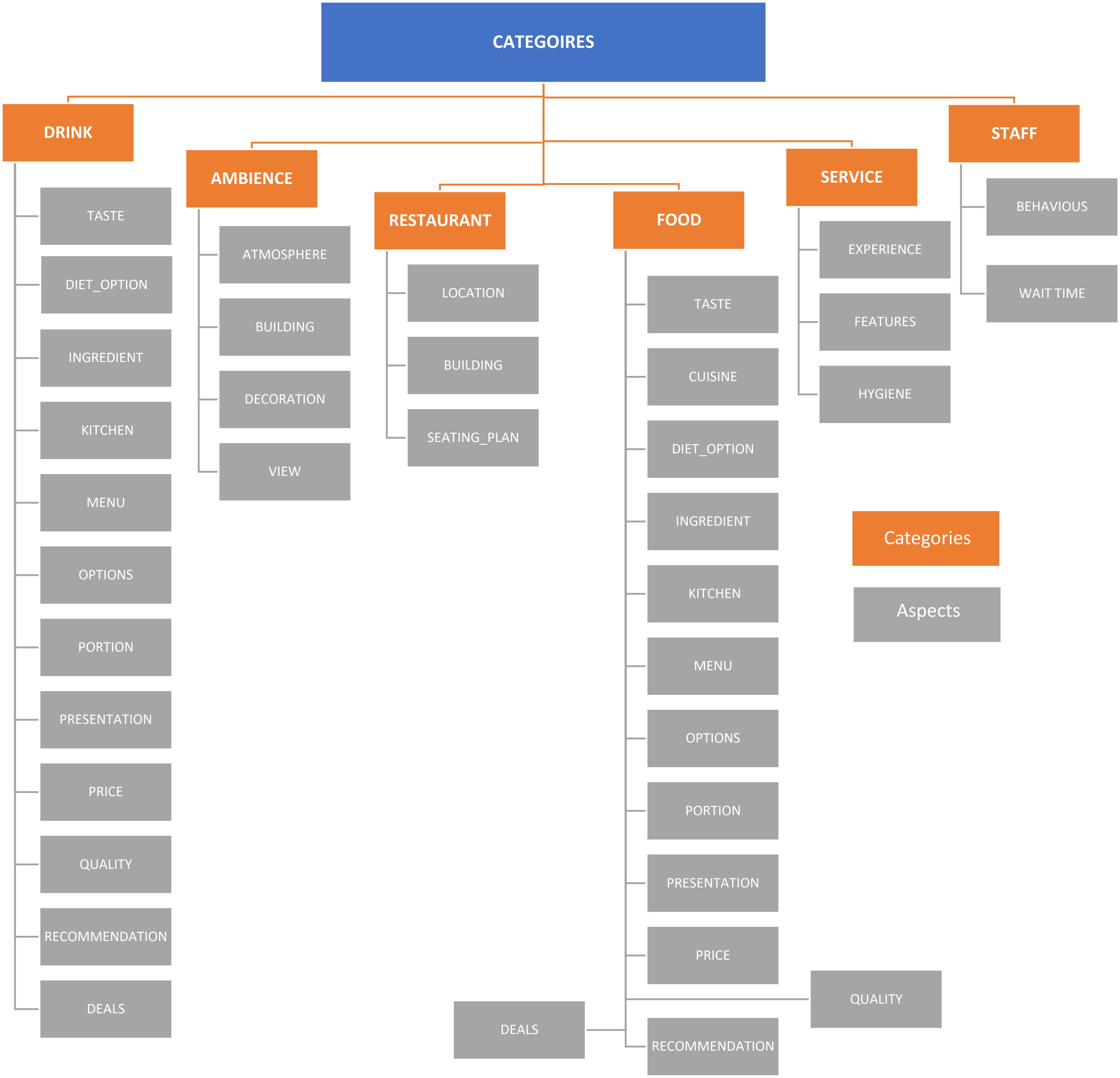
DATA AFTER MERGING CONTRIBUTING DATASETS

DATA SET	ENTRIES
ALREADY AVAILED DATASET WITH BOTH ASPECTS AND CATEGORIES	1011
ASPECT DATASET WITH BOTH ASPECTS AND CATEGORIES	7784
CATEGORIES DATASET WITH BOTH ASPECTS AND CATEGORIES	1283

Finalized Dataset with both aspects and Categories for Combined Training:

1011+7784+1283=10078

HIRARICHAL MAPPING OF ASPECTS TO CATEGOIRES



OBSERVATION AND SOLUTION CONVINIENCE

Upon Hierarchal mapping of the aspects one critical problem arises that is the aspects duplication under some categories as shown in figure such as Drink and Food Class may have same Aspects associated to it and other categories also encountered aspects replication. The problem could be solved by merging the categories that have same aspects, but it won't worth because we may have to truncate some aspects to be make the categories merged as well as we already trained the models on 6 categories if we merged them we may loss precision and the categories classification may get generalized for the individuals categories thus loss of information may also be the drawback. On the other hand, it will also make the more rigid to be tightly coupled with parent categories and increased complexity. Thus its not a great choice to use this solution we may pursue with the other solution that determines the categories of the sentences using pre trained model.

USING TRAINED MODEL TO PREDICT THE CATEGOIRES OF THE SENTENCES IN ASPECTS DATASET

ORIGINAL DATA WITH ASPECTS AND CATEGROIES

DATA LINK

https://drive.google.com/file/d/1VkEh9ZYq3PQgM-WBztlTQ42Ren7VDe8G/view?usp=share_link

ASPECTS DATASET WITH ASPECTS AND CATEGROES

DATA LINK

https://drive.google.com/file/d/15_PhPm1rNIVA4gQ2-F5PICEfN8cYNMe8/view?usp=share_link

PREDICTED CATERGOIRES OBSERVATION

It is with great pleasure that I present to you the outstanding performance of our Categories Model. This model has been meticulously developed and rigorously validated to ensure high precision and accuracy. In fact, the model has performed magnificently in predicting the missing categories from the aspect's dataset.

Through careful observation and thorough analysis, we can confidently conclude that the sentence categories have been classified with exceptional precision and accuracy. The model has truly left a benchmark in terms of its accuracy, setting a standard for others to strive towards.

CATEGROIRES DATASET WITH ASPECTS AND CATEGOREIS

DATA LINK

https://drive.google.com/file/d/1v1Id19L63gSjPBYGrEv6BuEPTcdjm1oY/view?usp=share_link

ANNOTATED ASPECTS OBSERVATION

Missing Aspects in the Categories data set are manually abbreviated with high precision and consistency we also trained a separate aspects classification on that data because the data is manually supervised and annotated and Trained Model outperforms in qualitative dimension by maintaining weighted F1 Score of 60%. We can also conclude that data balancing may not be solution to the accurate and outperforming Model but the Quality and consistency of the Aspects Annotation.

CODE LINK FOR ASPECTS CLASSIFICATION ON MANUALLY ANNOTATED DATA

https://colab.research.google.com/drive/1TxSZwRqDCRQrDzf3pNMT2VM4yLp7M1Mr?usp=share_link

COMBINED DATASET LINK

https://drive.google.com/file/d/11WnPm9u53fE99wjQ00H4JZkjLj6mTd7W/view?usp=share_link

PRE-PROCESSED COMBINED DATASET

https://drive.google.com/file/d/1PCduMDv9AxnrNudktI9Lg4UMBXRq7gRy/view?usp=share_link

PROCESSING CODE LINK

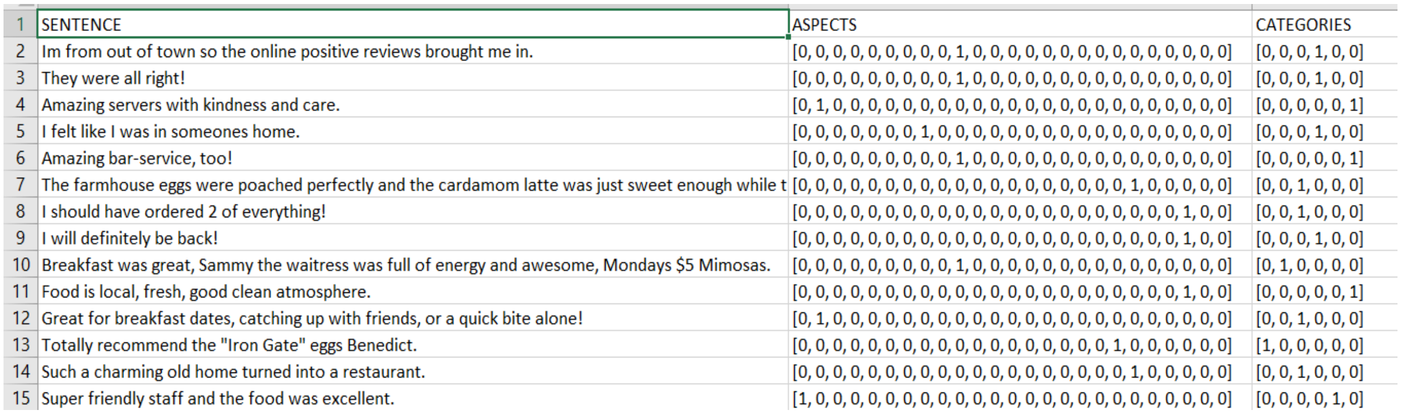
https://colab.research.google.com/drive/19wNejgrmnQvA25zzYk42RO76UFEsC-jO?usp=share_link

DATA CLEANING

- Data Cleaning Pipeline for Removing Stop words, Special Characters Html tags etc.
- Processing for Contraction Tackling and Spell Mistakes.
- Duplication Removing.
- Uncertainties and inconsistencies mitigation.

ONE HOT ENCODING

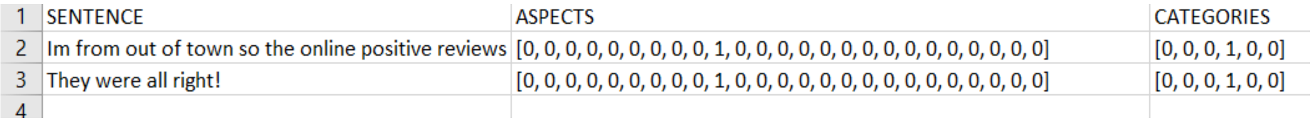
One hot encoding of the data in such a way that resultant data frame have three columns one of them is Sentences, Second one is the one hot encoded list of Aspects and last one is also the one hot encoded list of categories as in below image.



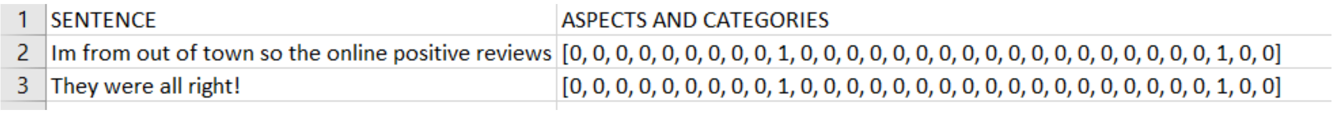
1	SENTENCE	ASPECTS	CATEGORIES
2	Im from out of town so the online positive reviews brought me in.	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 1, 0, 0]
3	They were all right!	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 1, 0, 0]
4	Amazing servers with kindness and care.	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 1]
5	I felt like I was in someones home.	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 1, 0, 0]
6	Amazing bar-service, too!	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 1]
7	The farmhouse eggs were poached perfectly and the cardamom latte was just sweet enough while t	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	[0, 0, 1, 0, 0, 0]
8	I should have ordered 2 of everything!	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	[0, 0, 1, 0, 0, 0]
9	I will definitely be back!	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	[0, 0, 0, 1, 0, 0]
10	Breakfast was great, Sammy the waitress was full of energy and awesome, Mondays $5 Mimosas.	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0]
11	Food is local, fresh, good clean atmosphere.	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	[0, 0, 0, 0, 0, 1]
12	Great for breakfast dates, catching up with friends, or a quick bite alone!	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 1, 0, 0, 0]
13	Totally recommend the "Iron Gate" eggs Benedict.	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0]
14	Such a charming old home turned into a restaurant.	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	[0, 0, 1, 0, 0, 0]
15	Super friendly staff and the food was excellent.	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 1, 0]

MERGING INTO SINGLE VECTOR

As we know that in interdependent learning and training there exist only one classification head for output layer for such purpose we have to merger the input features (Aspects and Categories) into single vector as below:



1	SENTENCE	ASPECTS	CATEGORIES
2	Im from out of town so the online positive reviews	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 1, 0, 0]
3	They were all right!	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 1, 0, 0]
4			



1	SENTENCE	ASPECTS AND CATEGORIES
2	Im from out of town so the online positive reviews	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
3	They were all right!	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

SINGLE VECTOR FEATURES

Considered Aspects Labels: 25

Considered Categories Labels: 06

Merged Vector Labels: 31

First 26 represent the Aspects and rest of the 6 represents the Categories and are filtered out to the correspondent Aspect or Category in predictions.

INTERDEPENDENT ASPECTS AND CATEGORY MODEL TRAINING

Merged Data is considered for the interdependent Training of both aspects and categories.

CODE LINK

https://drive.google.com/file/d/1h5XYTbzpe14r1ifATWCR-nggDUeDKVyb/view?usp=share_link

AVERAGE VALIDATION LOSS

0.204 about 20 %

AVERAGE TRAINING LOSS

0.217 about 21%

PROBLEM IN PREDICTION

Optimal Threshold for Aspects and Categories

SOLUTION

To make the First 25 labels of aspects to be threshold against their optimal thread hold and last 06 labels that corresponds to the categories be filtered on their sperate optimal threshold value.

EVALUATION AND CLASSIFICATION REPORT

Considered variant (Bert-base-cased and Roberta-base) both have approximately same performance and results.

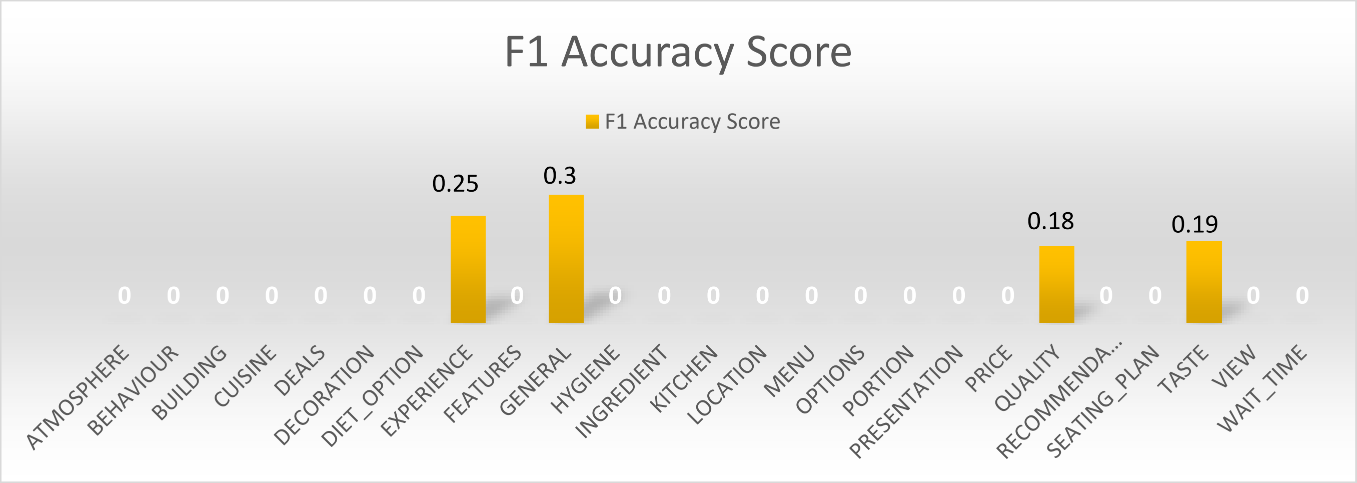
ASPECT	PRECISION	RECALL	F1-SCORE	SUPPORT
ATMOSPHERE	0.00	0.00	0.00	102
BEHAVIOUR	0.00	0.00	0.00	124
BUILDING	0.00	0.00	0.00	21
CUISINE	0.00	0.00	0.00	145
DEALS	0.00	0.00	0.00	19
DECORATION	0.00	0.00	0.00	75
DIET_OPTION	0.00	0.00	0.00	28
EXPERIENCE	0.14	1.00	0.25	218
FEATURES	0.00	0.00	0.00	16
GENERAL	0.21	0.53	0.30	51
HYGIENE	0.00	0.00	0.00	7
INGREDIENT	0.00	0.00	0.00	88
KITCHEN	0.00	0.00	0.00	6
LOCATION	0.00	0.00	0.00	7
MENU	0.00	0.00	0.00	68
OPTIONS	0.00	0.00	0.00	37
PORTION	0.00	0.00	0.00	16
PRESENTATION	0.00	0.00	0.00	13
PRICE	0.00	0.00	0.00	14
QUALITY	0.15	0.22	0.18	192
RECOMMENDATION	0.00	0.00	0.00	42
SEATING_PLAN	0.00	0.00	0.00	17
TASTE	0.10	1.00	0.19	158
VIEW	0.00	0.00	0.00	3
WAIT_TIME	0.00	0.00	0.00	45

CATEGORY	PRESICSION	RECALL	F1 SCORE	SUPPORT
AMBIENCE	0.21	1.00	0.34	309
DRINK	0.15	0.50	0.23	177
FOOD	0.19	1.00	0.32	290
RESTAURANT	0.26	1.00	0.41	389
SERVICE	0.15	0.43	0.22	183
STAFF	0.13	0.75	0.22	164

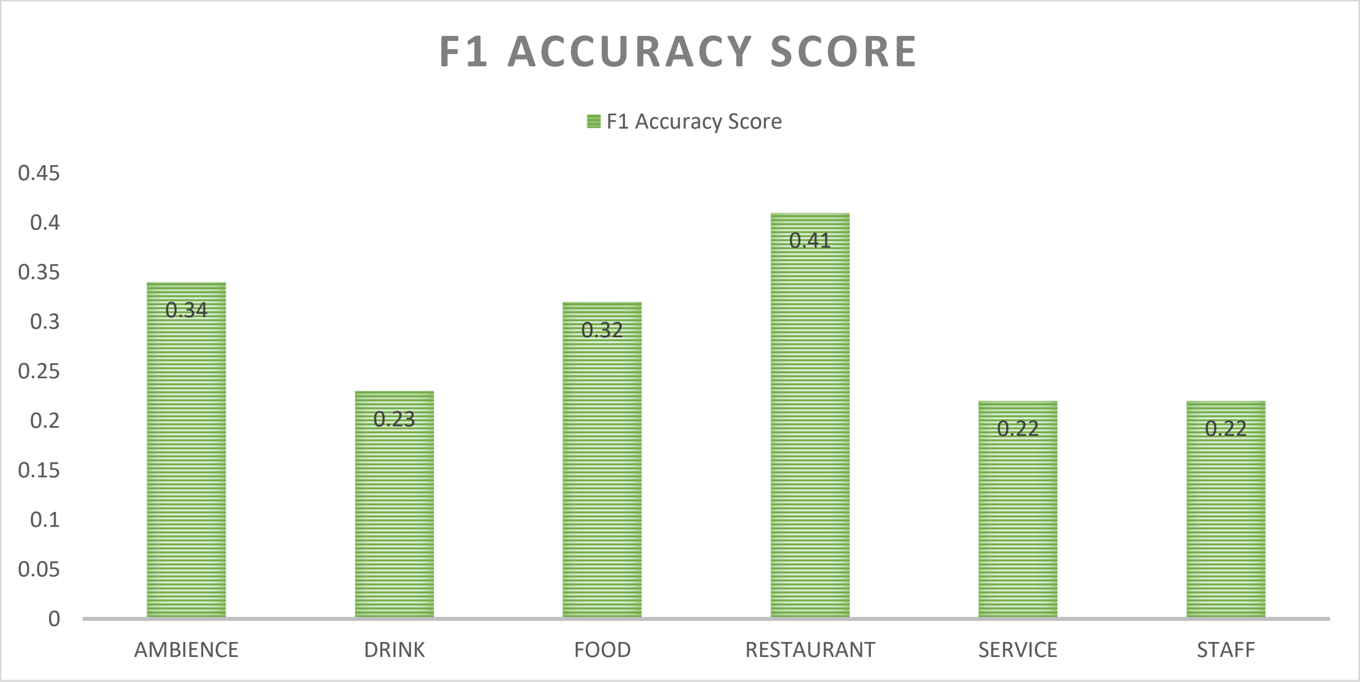
EVALUATION REPORT

MICRO AVG	**0.17**	**0.57**	**0.26**	**3024**
MACRO AVG	0.05	0.24	0.09	3024
WEIGHTED AVG	0.13	0.57	0.20	3024
SAMPLES AVG	0.18	0.57	0.27	3024

ACCURACY PER ASPECTS CLASS



ACCURACY PER CATGFORY CLASS



WHICH ONE TO CHOOSE? A CONCLUSION

After conducting a thorough analysis and comparison, it can be concluded that the Categories Model performs exceptionally well when trained independently. This is due to the model being fed consistent and cleaned data, which allows BERT to learn hidden patterns and make accurate predictions regarding the category of a given sentence, with a high probability distribution range.

On the other hand, the Aspects Model has shown good performance when trained on manually supervised data consisting of approximately 1263 rows. However, this model may not be suitable for the real world, as it could underfit and undergeneralize due to its limited data. Another Aspects Model trained on 8787 rows has shown decreased performance, likely due to imbalanced and inconsistent data as discussed earlier in the report. This model may require manual supervision for quality enhancement.

When considering interdependent training, the Training and Validation loss is low. However, it is essential to note that the independent Categories Model still outperforms the integrated Aspects and Categories Model in terms of accuracy. While the integrated model can predict both category and aspects, it does so with a lower probability.

Based on this analysis, it can be recommended that the Aspects dataset with 8787 rows be supervised manually by the proposed Data Revision tool to boost the performance of both the independently trained Aspects Model and the interdependent training of both Aspects and Categories Model. However, for the time being, it may be best to use the independently trained models for may not be the best but considerable results.

FUTURE WORK

1. AUGMENTATION PROCESS ALTERATION FOR QUALITY DATASET

To make the augmentation of the data by considering the augmented sentence semantics to validate its class by the Basic Model that we trained on our original dataset so to enhance the quality of the Dataset.

PROPOSED METHODOLOGY

- We might be using the following procedure.
- Get the Sentences belong to Single Category.
- Select a sentence to augment and their participation count.
- For every single Sentence get augmented sentences from the NLPAUG Model.
- From simply Selecting those Sentences for the training we might divert to alternate Method for Dataset Quality enhancement.
    - we will be Pass those augmented Sentences to the basic categories Classification model trained on our limited dataset to predict the sentences categories.
    - Validate the categories that if the predicted category of the augmented sentences and original sentence category matches with the probability of 75% or more then consider those sentences if not then drop those sentences and regenerate the augmented sentences.
- Repeat the procedure until desired size of dataset reaches.

2. DATASET REVISION FOR SENTENCE ASPECTS ENHANCEMENTS USING CUSTOMIZED TOOL

As the data set have lack of sentences with multiple aspects so to enable the model to learn effectively, we must revise the data thoroughly through manual supervision.

REFERENCES

https://nlpaug.readthedocs.io/en/latest/augmenter/word/word_embs.html

https://www.analyticsvidhya.com/blog/2021/08/nlpaug-a-python-library-to-augment-your-text-data/

https://neptune.ai/blog/data-augmentation-nlp

https://huggingface.co/

https://nlp.stanford.edu/projects/glove/

https://medium.com/gumgum-tech/creating-balanced-multi-label-datasets-for-model-training-and-evaluation-16b6a3a2d912

http://scikit.ml/stratification.html

https://github.com/trent-b/iterative-stratification

https://360digitmg.com/blog/bert-variants-and-their-differences

https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

https://github.com/scikit-learn-contrib/imbalanced-learn