ASPECT-BASED SENTIMENT ANALYSIS : EXPERIMENTS WITH NEURAL PRE-TRAINED MODELS

Rakesh Reddy Soma
MSc. in Computing (Artificial Intelligence)
20210927
rakesh.soma2@mail.dcu.ie

Krithika Sharon Komalapati
MSc. in Computing(Artificial Intelligence)
20211239
krithika.komalapati2@mail.dcu.ie

***Abstract*—Sentiment analysis has become increasingly important for businesses due to increasingly high volumes of online data. Using aspect-based sentiment analysis which is about the impact of aspects in a sentence to identify the polarity is the newest approach. This paper compares Naive Bayes, Logistic Regression and Support Vector Classifier models along with a lighter Bi-directional Encoder Representation of Transformers (BERT) called DistilBERT. One model is uncased distilBERT with text and the other is with text + aspect which is separated by the [SEP] token where the latter got a higher accuracy of 82.62%. Another model introduced in this paper is the Aspect classifier Model for each location and aspect individually giving an astounding accuracy of 94.71%. The various results obtained show that aspect classification for each aspect separately is more effective than with the whole dataset as one.**

***Index Terms*—Sentiment Analysis, Bi-Directional Encoder Representations from Transformers(BERT), Machine Learning Model, Aspect-Based, Accuracy.**

I. INTRODUCTION

Sentiment analysis (SA) is a natural language processing (NLP) task that determines the opinion expressed in a sentence and if it is positive, negative or neutral. Due to ample amount of opinionated data, sentiment analysis has become essential in the field of research and business like social media and e-commerce. Natural language sometimes contains ambiguity in its text which makes extraction of sentiment very challenging. Most of sentiment analysis' attention is on online reviews, where customers give opinions on products or services which can be utilized to measure customer satisfaction and make improvements.

Initially, sentiment analysis aimed at detecting the polarity of the sentence as a whole [13] which does not speak much about the product or service it provides. It is mostly about finding the overall expression of the sentence without knowing what it is about [5]. This causes confusion, especially when the text is referring to more than one topic or aspect or entity which have different sentiments. For example, restaurant reviews can be quite general (*eg, This restaurant is quite good*) but can also have many aspects like location, food, price (*eg. The food is delicious*). A review can contain more than one aspect (*eg. The food does not taste good and the prices are high*) and can sometimes have contradicting sentiments (*eg. The restaurant is very beautiful, but the food is costly*). Table 1 shows how a single sentence can used for multiple aspects and multiple sentiments.

A finer approach used is the Aspect-Based Sentiment Analysis (ABSA) where the polarity of a text is decided by aspect. Four stages of aspect-based sentimental analysis are Aspect Keyword Extraction (AKE), Aspect Term Extraction (ATE), Aspect Categorization (AC) and Sentiment Analysis (SA). AKE is having multiple words under an aspect (eg. Keywords like *cost, costly, expensive* are under the aspect *Price*) whereas ATE is taking a single word for aspect (eg. for the sentence *The wifi card is not good* the term is *wifi card*) [4]. AC is categorising a review based on ATE and AKE (eg. Categorising two aspects like food and safety into one and forming an association network) and SA is predicting the polarity of the review based on the text input (*eg. The restaurant has amazing* has a positive sentiment).

In this paper, different machine learning algorithms along with pre-trained BERT models are investigated. The pre-trained BERT is fine-tuned to get results on polarity and perform aspect classification with aspect classifier models. A comparative study is conducted to show that BERT performs better than other algorithms. Our contributions in this paper are as follows:
1. Aspect classifiers with pre-trained BERT produce better accuracy results individually than on the whole dataset.
2. Overall aspect classifiers perform better at sentiment classification than on Machine learning algorithms.
3. The BERT aspect classifier is superior to the SVM aspect classifier in terms of performance without considering the overall result.

The paper is organised as follows: Section II is literature review related to this paper; Section III introduces BERT and its features; Section IV describes the sentiHood data which is used for this paper; Section V is about the machine

Table I
EXAMPLE OF A SINGLE TEXT HAVING DIFFERENT ASPECTS AND POLARITIES [18]

Text	Label
location1 is very safe and location2 is too far	(location1,safety,Positive) (location1,transit-location,None) (location2,safety,None) (location2,transit-location,Negative)

learning algorithms used for aspect-based sentiment analysis; Section VI is about methodology for working of various models created; Section VII is the results and evaluation of this paper; Section VIII concludes on the paper's findings and suggested on some future enhancement. An appendix section is also discussed at the end of the paper which shows our analysis on a movie dataset to better understand the machine learning algorithms.

II. Literature Review

Initially, Sentiment Analysis aimed to predict the polarity of a text [18]. Some of the machine learning algorithms used to classify reviews are Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM). [10] found that Naive Bayes works well for problems with highly dependent features (eg. aspect, polarity) but it is surprising as the basic assumption of Naive Bayes is that the features are independent. So, [12] introduced a new model in which efficient approaches are used for feature selection, weight computation and classification. But, Support vector classifiers have more recall and accuracy than Naive Bayes in sentiment analysis hence proving that they are always the better option [11].

But a more fine approach such as Aspect-Based Sentiment Analysis (ABSA) was introduced in [9]. The aim of ABSA is to identify the aspects of the text which is reviewed and determine the polarity for each aspect. There are many ABSA models developed for customer reviews [7], restaurants [3] and on movie reviews [21]. For different datasets, there are different tasks adopted, wherein [3] there are 6 aspects with four polarities (positive, negative, conflict, neutral). Each sentence has one or more aspects with polarities for them. In [7] the dataset had aspect terms along with strength scores for them. There is the famous SemEval-2014 ABSA task which is based on laptop and restaurant reviews with 4 subtasks namely; Aspect term extraction, Aspect term polarity, Aspect category detection and Aspect category polarity [15]. These workshops continued in the following years for opinion mining for 2015 [16] and 2016 [17]. In [4], out-of-domain classification is done with state of art results using the SemEval dataset.

Targeted aspect-based sentiment analysis is carried out in [18], which uses the SentiHood dataset extracted from the question answering platform where urban neighbourhoods are described. Transformers play an important role for aspect classification in sentiment analysis and Transformers described in [22] is a neural architecture that uses multi-head self attentions to learn hidden representations of input texts. Bi-directional Encoder Representation of Transformers (BERT) is a pre-trained language model which captures the context from both left and right sides simultaneously [2]. In [2], BERT showed state of art results on 11 natural language processing tasks because BERT can be fine-tuned by adding an output layer that is precise for any particular task. Authors from [23] turn customer reviews to answer user questions, this is called review reading comprehension (RCC).

Another model that gave state of art results was when the BERT is added with the end to end task [8] and also perform a comparative study by utilizing a hold-out development dataset. In [20], 97.5% accuracy is achieved when BERT is paired with NLI on SemEval and SentiHood datasets. An auxiliary sentence is constructed from an aspect to sentence-pair classification task. Because BERT is pre-trained on reviews it has rich semantic knowledge about texts. DistilBERT has the same general architecture as BERT only but which is lighter and uses less computational power, it is best used in our paper because of less data and GPU. The token-type embeddings and the pooler are removed while the number of layers is reduced by a factor of 2 [19].

III. BERT

This section reviews BERT and its detailed implementation along with DistilBERT which is used on SentiHood data in this paper.

BERT is one of the recent innovations that sheds light on context-based representation learning [14] [6] [2]. BERT also has a unified architecture for any end task unlike ULMFit [6] or Elmo [14]. Bidirectional encoder representations from Transformers(BERT) is a pre-trained language model that is designed to consider the context of a word from both left and right sides simultaneously. BERT can extract more context features from a sequence compared to any of the machine learning algorithms. The model architecture of BERT is a multi-layer bidirectional transformer based on an original transformer implementation in [22].

There are two model sizes in BERT [23]:

- **BERT BASE** which has 12 layers, 768 hidden dimensions and 12 attention heads with 110M parameters.

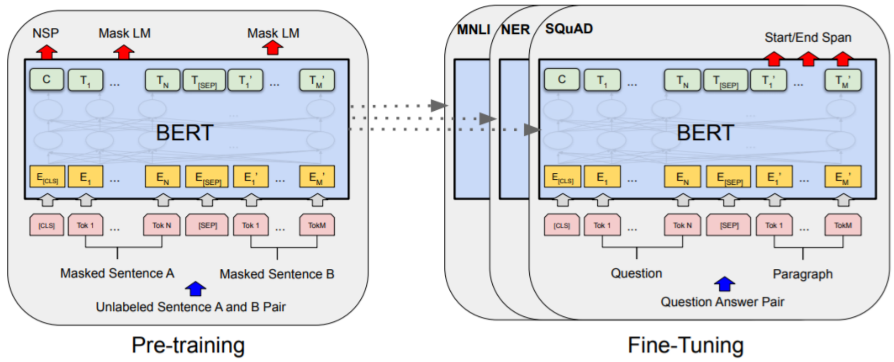

Figure 1. General Architecture of Pre-training and fine-tuning of BERT [1]

- **BERT LARGE** which as 24 layers, 1024 hidden dimensions and 16 attention heads with 340M parameters.

The Input/Output representation of BERT is decided based on the task at hand where the input can be a single sentence or two sentences packed together. Other ways of representation can be question and answer. In this paper, input is mostly a single sentence which is the text or two sentences packed in one i.e. text and aspect. The input representation in BERT is summed up with tokens, segments and position embeddings. For the output, it is mostly the polarity of a text which is 'positive', 'negative' or 'none'. But in the case of aspect classification tasks, the output is the aspect related to the sentence. The overall working of the BERT model is shown in figure 1.

A. Pre-Training

The Bi-directional Encoder Representation of Transformers (BERT) is trained on English Wikipedia and BookCorpus, a dataset including 10,000 books of various genres. It is crucial to utilize document-level corpus rather than sentence-level corpus to extract lengthy phrases. BERT uses two strategies while training:

Masked Language Model(MLM) replaces 15% of its words in its sequence with a [MASK] token. Based on the context of the unmasked words in the sequence, the model predicts the masked words. This is done by multiplying the output vectors by embedding matrix, making them into the vocabulary and calculating the probability of each word with softmax. Then the BERT loss function considered the prediction of masked words and not the non-masked words.

Next Sentence Prediction(NSP) is used when a pair of sentences are used in which BERT learns to predict the second sentence. For training the model in this way, the [CLS] token is used to indicate the beginning of the sentence and [SEP] to separate the two sentences from each other. A sentence embedding is also added to each token to indicate which sentence and a positional embedding is added to indicate each token's position. To predict the next sentence the output of the [CLS] token is transformed into the vector using a classification layer and then the IsNextSequence's probability is calculated with softmax.

B. Fine-Tuning

Fine-tuning the BERT is fairly simple because it applies the same understanding as Next Sentence Prediction. Fine-tuning the BERT model depends on the language task. For classification tasks such as in this paper, a classification layer is added on top of the transformer's output for the [CLS] token. Fine-tuning is performed based on the hyper-parameters selected. Hence, depending on the task, the BERT model is given with specific hyper-parameters (eg. *QA task, NLI task etc.*)

IV. Data

SentiHood dataset [1] is data taken from the question answering platform of yahoo! These are answers given by customers for questions related to neighbourhoods of London city. Each of the locations(entities) were queried to get answers and only the sentences having a location in them are kept while the others are discarded. The number of locations mentioned was more than 50 in one sentence. But, for forming the SentiHood dataset only two location sentences are annotated.

A. Data Set Description

The SentiHood dataset [18] has 5215 sentences in total, where 3862 contain a single location and 1353 contain two locations. These sentences are masked with two location entities LOCATION1 and LOCATION2 so that the task does not involve the identification of named entities. This dataset is divided into training data, validation data and testing data with 2977, 747 and 1490 respectively. Table II shows that there are a total of 12 aspects namely *live, safety, price, quiet, dining, nightlife, transit-location, touristy, shopping, green-culture and multicultural*.

[1]https://gitlab.computing.dcu.ie/komalak2/2021-mcm-master/tree/master/docs

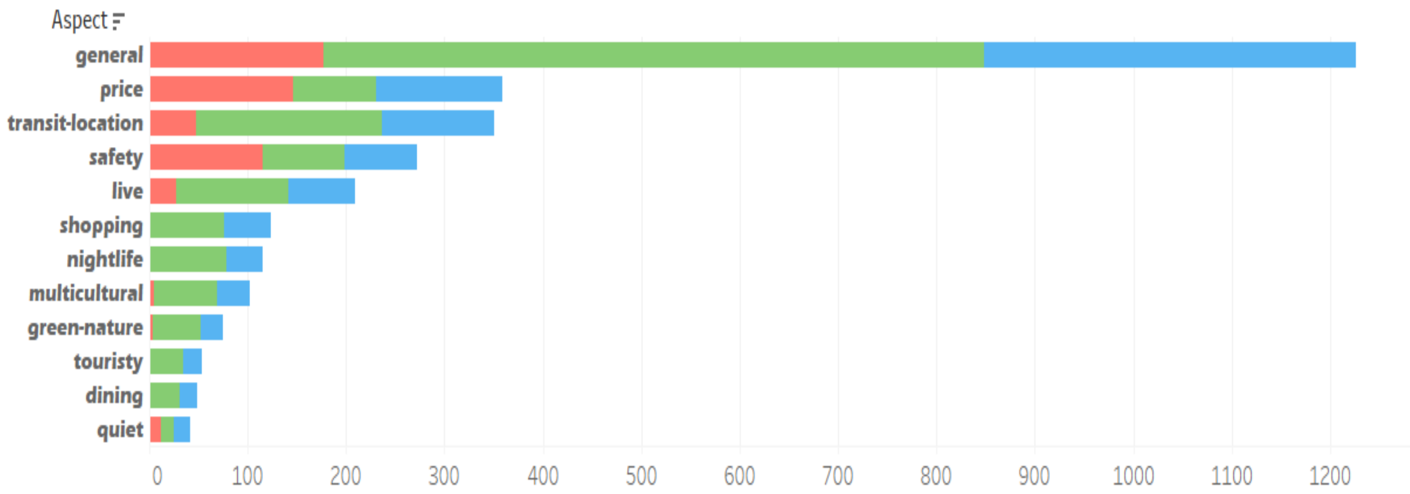

Figure 2. Bar Graph showing polarity with respect to aspect (positive-green; none-blue; negative:red)

Table II
ASPECT COUNT IN TRANING, VALIDATION AND TESTING DATA

	Training	Validation	Testing
General	1226	293	628
Transit-Location	351	98	177
Price	359	84	146
Safety	272	83	179
Live	209	58	100
Shopping	124	37	49
Nightlife	115	21	55
Multicultural	102	21	50
Dining	49	15	22
Green-nature	75	14	39
Quiet	41	13	17
Touristy	54	10	29

B. Data Exploration

The dataset contains three sentiments which are ”positive”, ”negative” and ”none”. Figure 2 shows that the ”positive” sentiment is more dominant than the other two sentiments showing that both locations have good feedback. The general aspect is the most frequent one with 2147 sentences and the least frequent one is touristy with 93 sentences. The transit-location and general aspects have the most ”positive” sentences showing that these aspects are good in these locations. The green-nature, touristy and dining aspects have no or very less negative sentences. In figure 2, the representation of sentiment is as follows: positive - green, none - blue, negative - red.

C. Data Cleaning and Preparation

The dataset is divided into training data, validation data and testing data is downloaded from github mentioned on the previous page. It is converted from .json format into a .csv format to facilitate data exploration and model building. All text is stored in dataframes along with the sentiment column. Another column is added with the 12 aspects corresponding to each text, each data line corresponds to only one aspect. The count of each aspect for training data, validation data and testing data is shown in Table III.

Table III
SENTIMENT COUNT IN TRAINING, VALIDATION AND TESTING DATA

	None	Positive	Negative
Training	956	1486	535
Validation	242	374	131
Testing	491	719	281

V. MACHINE LEARNING ALGORITHMS

This is divided into 2 sections: pre-processing and machine learning algorithms.

A. Pre-Processing

The training data, validation data and testing data are merged into a single dataframe. Then it is fitted with a label encoder in order to encode the categorical features as a one-hot numeric array and normalize labels. The sentiment category is encoded with none:0, negative:1 and positive:2. The aspect category is also encoded into numerical values for the model to perform better. Further, another column with the text and aspect combined is made for analysis with aspects present in the text. The training data, validation data and testing data is combined into a single dataframe.

First, the dataframe which is a combination of training, validation and test data is split using the train_test_split into training data of 3911 and test data of 1304. Tokenization is performed on both test and text by using the TF-IDF vectorizer. Our text sentences are converted to TF-IDF vectors, and the entire corpus is transformed into a TF-IDF matrix.The TF-IDF score formula is as follows:

$$\begin{aligned} \mathrm{tfidf}(t,d,D) &= \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D) \\ \mathrm{tf}(t,d) &= \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \\ \mathrm{idf}(t,D) &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

From the above formula, TF is the term frequency and IDF is the inverse document frequency which is a product that gives the TF-IDF score. Term frequency is the number of terms in each document by the number of times a particular term occurs in a document. The inverse document frequency is the log of a total number of documents in the corpus by the number of documents where that particular term occurs. The TF-IDF score is calculated when the frequency of text is calculated and then inverted into weights to encode new text and give the unique words more weight.

In another instance, one-hot encoding is also performed to the aspects so that they can be converted into categorical values with a specific number assigned to them. One-hot encoding is applied to both the train and test labels whereas TF-IDF is applied to the text. One hot-encoding is the integer representation of bits with unique values.

B. Algorithms

In order to get a better understand of how the text works with each algorithm, we applied the machine algorithms with slight variations. All these algorithms are implemented using scikit learn. They are as follows:
1. Predicting the sentiment of a text by using only the text
2. Predicting aspect of a text by using the text
3. Predicting the sentiment of a text using the sentence and aspect together.

The Machine Learning Algorithms used are listed below:

1. Naive bayes: Naive bayes classifier is a supervised machine learning algorithm that applies Bayes' theorem with the assumption of conditional independence between every pair of features. It is based on a simple assumption that the property is independent of each other when given the target value. Multinomial naive bayes is applied when multiple probabilities for a certain event are generated given the features. This can be effectively used for text classification where the events represent the occurrence of a word in the corpus. The Naive Bayes formula is as follows:

$$P(class/features) = \frac{P(class) * P(features/class)}{P(features)}$$

P(class/features) : Posterior Probability
P(class) : Class Prior Probability
P(features/class) : Likelihood
P(features) : Predictor Prior Probability

2. Logistic Regression: It is also a supervised machine learning algorithm and the model learns and approximates a mapping function f(Xi) = Y from input to output. The difference between the Naive Bayes and logistic regression is that logistic regression is a discriminative classifier. Discriminative classifiers directly model P(y/x) without modelling the joint probability P(x,y). The general formula for is as follows:

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where, β_i are the parameters of the model
p is the probability of the event that Y=1 (Y is the response variable)

3. Support Vector Machines(SVM): It is also a supervised machine learning classification technique that plots the data points in a two-dimensional space. It predicts the class based on side of the spaces where the points lie, in this case, many hyperplanes are drawn. Among these hyperplanes one will be chosen which has a maximum margin that classifies data points effectively. Support vector classifiers have shown to be extremely capable of handling aspect-based sentiment analysis because of maximum margin. Text data is ideally suited for SVM classification because of the sparse nature of the text, in which few features are irrelevant, but they tend to be correlated with one another and generally organized as linearly separable categories. The formula is as follows:

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)\right)\right] + \lambda\|\mathbf{w}\|^2$$

where, λ determines the increase of the margin to make $\mathbf{x}_i$ is on the right side
$\mathbf{w}$ is the normal vector to the hyperplane
x and y are the points in the dataset

VI. Methodology

Pre-trained language models are providing a context to words, that have previously been learning the occurrence and representations of words from unannotated training data. This section explains the type of BERT models used in this paper.

A. Setting up the data

Apart from the pre-processing done from section V(A), the only difference is that the training data, validation data and testing data are kept the same without combining. Label encoder is used for the aspects to change them into integer categories. The sentiment is changed into none: 0, negative: 1 and positive: 2.

From the DistilBert Model, the DistilBertTokenizerFast is used to tokenize the text and to convert it into a model readable format. The fast version of its tokenizers is used because of its performance benefits. The tokenizer converts sentences to the sequence of tokens using [CLS] at the beginning and [SEP] at the end. Tokenizing and encoding are carried out for the training data, validation data and testing data. The TFDistilBertForSequenceClassification is used for the sequence classification of the model, and all the weights are not changed to achieve what is required.

B. Pre-trained Model

Transformers have thousands of pre-trained Bert models from the hugging face library [2] in which DistilBERT is used in this paper. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% fewer parameters than Bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark. DistilBert is also pre-trained just like the BERT. The TFTrainingArguments has information about the training epochs, batch sizes and weight decay which can be changed while initializing the model. Using the TFTrainer the model is trained with train data and validation data for evaluation.

C. Hyper-parameters

To get more performance out of the DistilBERT Optimizer with multiple learning rates i.e. 2e-5, 3e-5, 5e-5 are applied. The optimizer used is Adam which is a replacement optimizer algorithm for stochastic gradient descent which helps in training models. We use the accuracy metric for compiling the model which batch sizes 16 and 32. The training data and validation data are considered for fine-tuning. After getting accuracy for them, results are predicted on testing data.

D. Aspect Classifier Model

After fine-tuning the BERT model, based on the results that were obtained, it is decided to make an analysis for each aspect separately. For each location in the dataset, LOCATION1 and LOCATION2 are used along with 12 aspects which are dining, general, green-nature, live, multicultural, nightlife, price, quiet, safety, shopping, touristy, transit-location. Aspect classifiers are defined for each aspect to find differences in each of them. For each location and for each aspect there is an aspect classifier defined to predict polarity for the sentiHood dataset. The process for using this classifier is the same as DistilBERT model where training data, validation data and testing data are used.

All data is converted into tensor objects and sent through an optimizer. Predictions are made using the testing data with softmax activation. These predicted values are stored in a dataframe along with the true values of the same data for each aspect. Then the overall accuracy of all the aspects when comparing the predicted values with the true values is evaluated.

Another model with the same data is created, known as the SVM classifier. This is applied for every aspect for both the locations same as done with the aspect classifier model. The predicted and actual values are stored together and accuracy is calculated based on that.

E. Text [SEP] Aspect Analysis

For this analysis, the aspects and sentiments are converted into labels. The training data, validation data and testing data are combined with their respective aspects along with a [SEP] between them. The format is text + [SEP] + aspect. For example, *LOCATION1 is transforming and the prices will go up and up [SEP] price*.

After applying the separator, the same process is described in section VI(A, B, C). Another way where a separator is not used is also executed where the text and aspect are combined together without [SEP]. For example, *LOCATION1 is transforming and the prices will go up and up price*.

VII. Results and Evaluation

A. Performance Metrics

All predicted values must be validated to determine the performance of the model. Most conclusions and discussions are often based on the predicted values obtained. The metric used in this paper is accuracy.

1) Accuracy: It is a measure further used for a particular task eg. Aspect-Based Sentiment Analysis (ASBA) and it is the by the fraction of a number of correct predictions over a total number of predictions. Precision may be thought of as a metric for how accurate a classifier is. Recall of a classifier is a measure of its accuracy..

[2] https://huggingface.co/transformers/pretrained_models.html

Table IV
SENTIMENT AND POLARITY RESULTS FOR MACHINE LEARNING ALGORITHMS

Model	**Text + Aspect** Sentiment	Aspect	**One-Hot Encoding** Sentiment	Aspect
Naive Bayes	61.5%	41.1%	63.5%	49.1%
Logistic Reg	75%	58.6%	75%	66.3%
SVM	75.5%	62.1%	**76.6%**	66.4%

Table V
ASPECT CLASSIFICATION AND SVM CLASSIFICATION RESULTS FOR LOCATION1 AND LOCATION2

Aspect/Location	**Aspect Classifier** LOCATION1	LOCATION2	**SVM Aspect Classifier** LOCATION1	LOCATION2
General	85.64%	76.28%	78.4%	78.09%
Transit-Location	91.07%	90.72%	91.81%	90.72%
Price	95.10%	90.2%	94.70%	88.40%
Safety	95.43%	93.81%	95.97%	93.29%
Live	94.03%	96.90%	95.43%	95.36%
Shopping	98.65%	97.93%	98.05%	96.39%
Nightlife	98.32%	97.93%	98.05%	98.45%
Multicultural	98.65%	98.48%	98.18%	98.19%
Dining	98.85%	97.68%	**99.32%**	**98.71%**
Green-nature	98.65%	97.93%	98.05%	98.19%
Quiet	**98.99%**	98.19%	98.85%	98.19%
Touristy	98.79%	**98.71%**	98.59%	**98.71%**
Overall		94.71%		95.24%

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$
$$\text{Precision} = \frac{tp}{tp + fp}$$
$$\text{Recall} = \frac{tp}{tp + fn}$$

where, tp = true positives(equal to true hit)
tn = true negatives(equal to correct rejection)
fp = false positives(False alarm/ type 1 error)
fn = false negatives(miss/ type 2 error)

B. Results and Discussion

Machine learning algorithms are applied to the SentiHood dataset and sentiment is predicted using the test data from the dataset. Predicting the aspects also was carried out. The results are shown in Table IV.

From table IV we can say that the support vector machine (SVM) perform the best i.e. 76.6% while predicting the polarity of the SentiHood data by applying one-hot encoding to the aspects and polarity. In the case of aspect prediction support vector machine (SVM) and logistic regression (LR) produce an accuracy of 66.4% with one-hot encoding.

The next table, Table V is about the results on aspect classifier and SVM which use aspect classification. The overall results for both are slightly different, where the aspect classifier has an accuracy of 94.71% and the SVM aspect classifier has an accuracy of 95.24%. The aspects dining, quiet and touristy perform higher than 98.5% for both locations whereas the general aspect performs the least with as low as 78.09% for location1- SVM classifier.

Table VI
RESULTS ON DISTILBERT

	DistilBERT-base-uncased
Text	80.68%
Text + Aspect	79.54%
Text [SEP] Aspect	**82.62%**

Table VI shows the results for DistilBERT-base-uncased, where the highest results(82.64%) are achieved when Text [SEP] Aspect are given as input instead of just plain text. This paper does not aim to achieve state of art results on any models but wants to show a comparative study.

The results above show that the SVM classifier performs better than the aspect classifier when calculated overall. But when observed for the general aspect LOCATION1, we can see that the aspect classifier performs better. The general aspect has all the polarities i.e. *positive, negative, none* are equally distributed, whereas in the other minor (less in number) aspects i.e. *quiet, touristy, green-nature* the none polarity is significantly more. Due to this, when the models are trained repeatedly, the minor aspects are over-trained to get such results. Because the minor aspects have greater none polarity sentences in them, the model prediction is biased. All this is due to SentiHood data's class imbalance.

VIII. Conclusion and future work

In this paper, A series of experiments have been conducted on the SentiHood dataset. DistilBERT model for the overall text with and without aspects is created. Aspect classifier models with location and aspect separately are also created which gives the highest accuracy of 94.71%. This paper addresses that better results are achieved with aspect classifiers than with one single model.

For future work, better enhanced in fine-tuning can be applied to the models used in this paper. More pre-processing with the text or aspects can be done to find any potential loss rate. working more on the SVM aspect classifier more, in order to find out why the results are varied. Addressing the class imbalance of the SentiHood dataset would help to better understand the polarities and aspects overall.

References

[1] Jan Christian Blaise Cruz and Charibeth Cheng. "Evaluating Language Model Finetuning Techniques for Low-resource Languages". In: (July 2019). DOI: 10.13140/RG.2.2.23028.40322.

[2] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: (June 2019), pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[3] Gayatree Ganu, Noemie Elhadad, and Amélie Marian. "Beyond the Stars: Improving Rating Predictions using Review Text Content." In: (Jan. 2009).

[4] Athanasios Giannakopoulos et al. "Unsupervised Aspect Term Extraction with B-LSTM & CRF using Automatically Labelled Datasets". In: (Sept. 2017), pp. 180–188. DOI: 10.18653/v1/W17-5224. URL: https://aclanthology.org/W17-5224.

[5] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. "Aspect-Based Sentiment Analysis using BERT". In: (Sept. 2019), pp. 187–196. URL: https://aclanthology.org/W19-6120.

[6] Jeremy Howard and Sebastian Ruder. "Universal Language Model Fine-tuning for Text Classification". In: (Jan. 2018), pp. 328–339. DOI: 10.18653/v1/P18-1031.

[7] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews". In: KDD '04 (2004), pp. 168–177. DOI: 10.1145/1014052.1014073. URL: https://doi.org/10.1145/1014052.1014073.

[8] Xin Li et al. "Exploiting BERT for End-to-End Aspect-based Sentiment Analysis". In: (2019). arXiv: 1910.00883 [cs.CL].

[9] Bing Liu. "Sentiment analysis and opinion mining". In: *Synthesis lectures on human language technologies* 5.1 (2012), pp. 1–167.

[10] Luis Martin-Domingo, Juan Carlos Martín, and Glen Mandsberg. "Social media as a resource for sentiment analysis of Airport Service Quality (ASQ)". In: *Journal of Air Transport Management* 78 (July 2019). DOI: 10.1016/j.jairtraman.2019.01.004.

[11] M. Neethu and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques". In: *2013 4th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2013* (July 2013), pp. 1–5. DOI: 10.1109/ICCCNT.2013.6726818.

[12] Zhen Niu, Zelong Yin, and Xiangyu Kong. "Sentiment Classification for Microblog by Machine Learning". In: (Aug. 2012), pp. 286–289. DOI: 10.1109/ICCIS.2012.276.

[13] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification Using Machine Learning Techniques". In: *EMNLP* 10 (June 2002). DOI: 10.3115/1118693.1118704.

[14] Matthew E. Peters et al. "Deep contextualized word representations". In: (2018). arXiv: 1802.05365 [cs.CL].

[15] Maria Pontiki et al. "SemEval-2014 Task 4: Aspect Based Sentiment Analysis". In: (Aug. 2014), pp. 27–35. DOI: 10.3115/v1/S14-2004. URL: https://aclanthology.org/S14-2004.

[16] Maria Pontiki et al. "SemEval-2015 Task 12: Aspect Based Sentiment Analysis". In: (June 2015), pp. 486–495. DOI: 10.18653/v1/S15-2082. URL: https://aclanthology.org/S15-2082.

[17] Maria Pontiki et al. "SemEval-2016 Task 5: Aspect Based Sentiment Analysis". In: (June 2016), pp. 19–30. DOI: 10.18653/v1/S16-1002. URL: https://aclanthology.org/S16-1002.

[18] Marzieh Saeidi et al. *SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods*. 2016. arXiv: 1610.03771 [cs.CL].

[19] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: (2020). arXiv: 1910.01108 [cs.CL].

[20] Chi Sun, Luyao Huang, and Xipeng Qiu. "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence". In: (2019). arXiv: 1903.09588 [cs.CL].

[21] Tun Thura Thet, Jin-Cheon Na, and Christopher S.G. Khoo. "Aspect-based sentiment analysis of movie reviews on discussion boards". In: *Journal of Information Science* 36.6 (2010), pp. 823–848. DOI: 10.1177/0165551510388123.

[22] Ashish Vaswani et al. "Attention Is All You Need". In: (2017). arXiv: 1706.03762 [cs.CL].

[23] Hu Xu et al. "BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis". In: (June 2019), pp. 2324–2335. DOI: 10.18653/v1/N19-1242. URL: https://aclanthology.org/N19-1242.

Appendix

For a better understanding of the working of machine learning algorithms, a different dataset was used. A movie-review dataset is used for sentiment analysis This helps to work with different pre-processing techniques and m machine learning algorithms for exploration.

A. Data description

The dataset contains movie-review ratings which are given with positive or negative polarity. This dataset contains 1000 positive and 1000 negative documents with processed reviews [13]. Documents contain many texts that are reviews of the movie. This dataset comes under supervised learning because the sentiment is already specified.

B. Pre-Processing and Algorithms

Move-review dataset present in 2000 documents are made into lists and converted into datasets to apply machine learning algorithms to them. Add a column for the sentiment of text where positive:1 and negative:0. Pre-processing of this dataset is in the following way:
1. Punctuation is not removed and count vectorization is applied to convert text into tokens with frequencies.
2. Punctuation is kept as it is and TF-IDF is applied to convert into tokens with probabilities.
3. Punctuation and Stopwords are removed with TF-IDF applied to the dataset.
4. Punctuation and Stopwords are removed with first Count vectorizing and then applying TF-IDF vectorizer to the dataset.

Machine Learning Algorithms applied are
- Naive Bayes Algorithm
- Logistic Regression
- Support Vector Classification

C. Results and Discussion

The results for the above experiments are displayed in the below table:

Table VII
MODEL RESULTS

Model	Logistic	SVM
With Punct(CV)	69%	68%
With Punct(TFIDF)	69%	70%
without Punct (TFIDF)	68%	69%
without Punct (CV and TFIDF)	68%	69%

From Table VII, it is observed that there is not much difference between the machine algorithms even with the different pre-processing steps. After these results, SentiHood data can be used without major pre-processing in future.