Aspect Based Sentiment Analysis of Student Housing Reviews

Aniket Mukherjee
Manipal University Jaipur
mukherjeeaniket007@gmail.com

Shiv Jethi
Manipal University Jaipur
shivjethi04@gmail.com

Akshat Jain
Manipal University Jaipur
jain.akshat546@gmail.com

Ankit Mundra
Manipal University Jaipur
ankitmundra8891@gmail.com

2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) | 978-1-7281-7132-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/PDGC50313.2020.9315324

***Abstract*—According to a 2016 report by the Indian Ministry of Human Resource Development, there were 39,658 student hostels across India. In recent years, owing to the growing number of students residing in such hostels, there has been an interest in helping students know more about these hostels by providing them with information and reviews from residing students. We aim to categorize these based on various aspects and give greater insights about them using applications of aspect based sentiment analysis. We have used a neural network based approach to pre-process the texts and propose two models, one for aspect extraction and classification and the other for sentiment polarity analysis. Further, we have presented an extensive evaluation of our models and have achieved an accuracy of more than 75% on both the models.**

***Keywords*—Sentiment, Aspect, Neural Network**

I. Introduction

A. Motivation

The demand for student housing facilities has seen a tremendous upward spike in recent years. Often, students want to get an in-depth analysis of the housing facilities surrounding a university or college before they finally make the decision to enroll in the institution. As a result, housing facilities and other amenities associated with it play a vital role in the decision process of students and parents alike. Moreover, the recent developments in machine learning techniques for analyzing sentiments have shown promising results.

Aspect based sentiment analysis (ABSA) can be defined as a technique for text analysis that dissects a text into aspects and calculates a sentiment level for each aspect. An aspect refers to a certain attribute or feature of the entity under consideration. A sentence may talk about multiple attributes associated with different entities.

The quintuple definition of aspect based sentiment analysis is given below.

$$S = (e_i, a_{ij}, s_{ijkl}, h_k, t_l) \tag{1}$$

where:
e_i is the entity under consideration
a_{ij} is a particular aspect associated with the entity
h_k is a sentiment holder
t_l is the time at which the sentiment is held
s_{ijkl} is the sentiment about aspect a_{ij} of entity e_i held by h_k at time t_l

B. Objective

This paper presents an approach centered around aspect based sentiment analysis that uses massive review data to effectively classify student housing facilities based on various aspects, and presents the sentiment tendency of said classifications.

Under this approach, we have proposed two models for aspect classification and subsequent sentiment analysis. To implement the aforementioned, we scrape reviews from distinct sources on the internet and annotate them. This ensures that our dataset accounts for the words more prevalent amongst Indian users.

The following example facilitates a high level understanding of our approach (Fig. 1).

Review: ["The hostel mess provides with good quality food"]
Aspect Classification Model Output: [Food and Mess]
Sentiment Polarity Model Output: [Positive]

II. Background and Related Work

Sentiment analysis helps to detect the sentiment/emotion in texts, making it one of the most widely used applications in the domain of machine learning. Sentiment Analysis is widely used while analyzing textual data on online platforms.

ABSA goes one step further than native sentiment analysis and takes into consideration the words related to the aspects and distinguishes the sentiment with respect to each aspect. ABSA model requires aspect categories and its corresponding aspect terms to extract sentiment for each aspect from the content corpus. One can create a domain-centered model for a specific implementation, however, general language models can also be used.

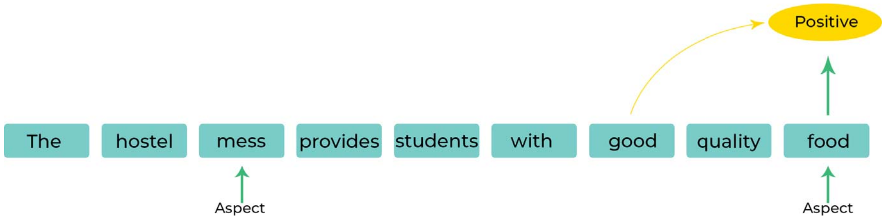

Fig. 1. Example of a review

ABSA broadly consists of two sub-tasks, aspect-description extraction and classification, followed by sentiment analysis on the aspect level. ABSA is more fine-grained than text-level and sentence-level sentiment analysis methods and can obtain more accurate and detailed information from textual data.

The goal of aspect-description extraction is to extract and identify the objects to be evaluated from reviews through a series of operations. For a product, the attribute description is the performance characteristics of the product in various aspects such as appearance, function, price, etc. In hostel review data, residents often evaluate various characteristics of the facilities from multiple perspectives such as security, location, dining, accommodation, staff, etc. Each attribute evaluation has a positive, negative, or neutral emotional tendency.

At present, sentiment word extraction and classification methods mostly make use of a sentiment dictionary. Such methods depend on the emotional dictionary, and its quality plays a decisive role in the judgment of emotional polarity. In comparison, methods based on neural networks are more efficient and more intelligent because of the strong learning ability of neural networks. As a result, neural networks based approaches have gained a prominent place in the mainstream focus for such tasks.

Applications of ABSA have been used in a vast variety of fields. Features based on lexicons and word frequency were combined with traditional machine learning methods in [1] to analyze movie reviews. On the other hand, [2] employed a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) model to analyze movie reviews. In the field of electronic commerce, [3] proposed a trust representation model to conduct sentiment similarity analysis.

When it comes to hostel reviews, the problem is a bit more complicated. Most of the sentiment holders are young students, who make use of locally popular conventions (or slang words) to refer to a universal feature that can be associated with any community. Therefore, such reviews often contain many words that essentially mean the same but are difficult to define grammatically. Traditional sentiment analysis methods are not optimal for dealing with such reviews.

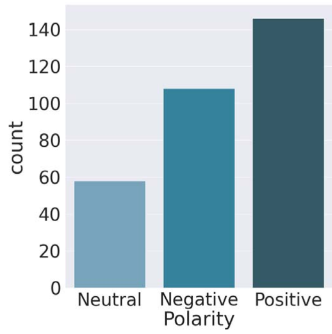

Fig. 2. Distribution of Polarity in the dataset

III. Dataset

The dataset initially consisted of three columns "Reviews", "Aspects", and "Polarity", where "Reviews" consisted of all the human-authored annotations by students, scraped from various websites. We used *beautifulsoup4* and *requests* to scrape the content and generated a Comma Separated Values (CSV) file. We manually identified each student review to a particular aspect, which was maintained in the column "Aspects" and we assigned each review a polarity, which was subsequently added to the column "Polarity". The dataset consisted of 320 reviews that were subsequently annotated. The distribution of the Polarity and Aspects are given in (Fig. 2) and (Fig. 3) respectively.

A. *Data Cleaning*

We used regular expressions to clean the reviews by removing auxiliary emojis and hashtags encountered within each review.

B. *Data Preprocessing*

We generated a vocabulary of words from all the reviews. The vocabulary was of size 843. Using *word2vec* semantics, each review was converted into a vector of size [1,843]. Further, by using label encoding methods from *scikit-learn*, we converted aspect labels into a numeric form. To perform aspect term extraction, we identified the most similar nouns pertaining to the given aspect categories. We achieved this using *noun_chunks* method from *Natural Language Toolkit*

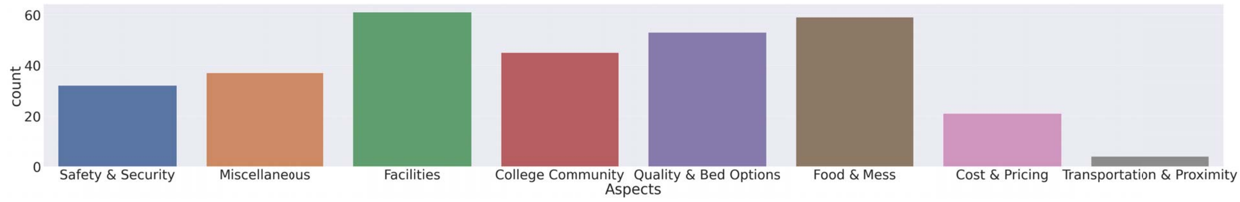

Fig. 3. Distribution of Aspects in the dataset

(NLTK) in Python, and after doing so, added the same to a separate column "Aspect Terms".

IV. Proposed Approach

The earliest design of the basic structure of a neural network (Fig. 4) can be traced back to [4], which was inspired by the structure of the human brain.

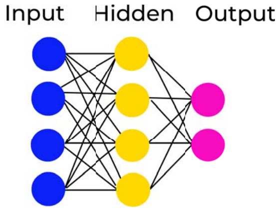

Fig. 4. Architecture of a Neural Network

The three components of a neural network are an input layer, one or more hidden layers, and an output layer. Depending on the structure and method, the nodes (or neurons) can either be partially or fully connected.

The interrelationship between input and output values of a hidden unit can be defined as:

$$Y_i = g(\sum_j W_{ij} * a_j) \tag{2}$$

The relationship can be exemplified in (Fig. 5). The output value Y_i of a node i is calculated as shown in (2), where a_j refers to the input variables, W_{ij} is the weight of input node j on node i and g is the activation function, which is normally a nonlinear function (e.g. Sigmoid or Gaussian function) to transform the linear combination of an input signal from input nodes to an output value.

The training of an Artificial Neural Network (ANN) is done by iterative modification of the weight values in the network to optimize the error margin between predicted and true values, typically through back-propagation methods [5].

Neural networks offer a varied set of features such as adaptive learning, parallelism, fault tolerance, and generalization. Due to their deep architectures with hidden layers, they generally tend to represent more intelligent behavior than shallow architectures like Support Vector Machines (SVMs).

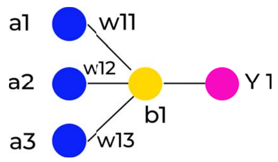

Fig. 5. Architecture of a Neuron

A. Architecture

We created two neural networks, an aspect classification model and a sentiment polarity model. The neural networks consisted of 4 dense fully connected layers. In a dense layer, all the neurons of a layer are connected to all the neurons of the next layer. It provides learning features from all the combinations of the previous layers [6].

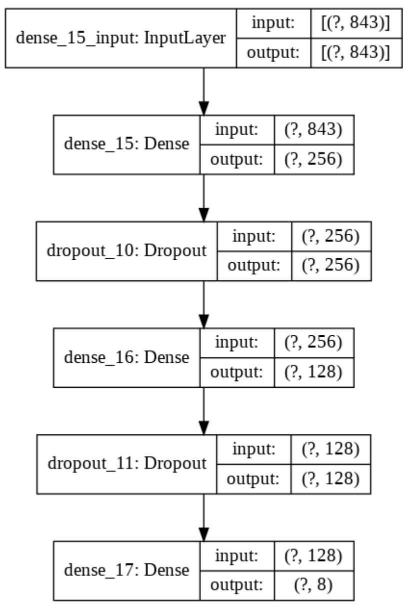

Fig. 6. Architecture of the Aspect Classification Model

The input to layer 1 is a vocabulary of words of size 843. The layer is a dense layer of size 843 with a Rectified Linear Unit (ReLU) activation. This is beneficial to our network as ReLU returns a value of zero for all negative inputs and thus, is sparsely activated. ReLU offers cheap computations and

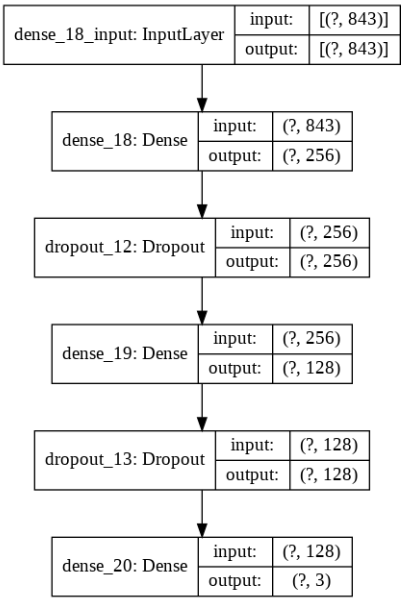

Fig. 7. Architecture of the Sentiment Polarity Model

performs better than hyperbolic tangent networks [7].

The next two-layer groups are subsequent dense and dropout layers. We use dropout layers to prevent overfitting and drop roughly 20% of the neurons and their connections during the training [8].

For the aspect classification model, the last layer is a dense sequential layer of size 8 with a softmax activation function that converts the logits into probabilities of sum 1. The output vector of size 8 contains the probabilities of a review belonging to each aspect. Through an argmax function and an inverse label encoder, we obtain the aspect to which the review has the highest probability of belonging.

In the same manner, we created a similar neural network to classify the sentiment polarity of the reviews. The output of the sentiment polarity model is a vector of size 3 which contains the probabilities of a review being positive, neutral, and negative respectively. We use an argmax function and a reverse label encoder to get the sentiment of the review.

Lastly, the aspect classification model and the sentiment polarity model were combined to present the findings.

B. Training

We used the categorical cross entropy loss function (3) to penalize the outputs that were being wrongly predicted with a high probability. Categorical Cross Entropy works well with a multi-classification problem and serves as the loss function for our networks [9].

In general, cross entropy can be defined as:

$$H_{y'}(y) = -\sum_i y'_i \log(y_i) \tag{3}$$

where y'_i is the predicted probability for class i and y_i is the true probability for that class.

V. Experimental Evaluation

A. Experimental Setup

We implemented our approach in Python. We used the LabelEncoder, train_test_split and preprocessing implementation provided by *scikit-learn*. The two Neural Network models were implemented in Python using implementations of Dropout, Dense, Activation layers by TensorFlow. All experiments were run on a Google Colab instance with 12 GB of random-access memory.

We evaluated the performance of our models utilizing the accuracy metric by TensorFlow. It was used to calculate the frequency of correct predictions. The higher the accuracy, the better the performance of a model.

B. Results

To evaluate our approach, we used an accuracy measure on 20% of the train dataset as a validation feature. We tuned the hyperparameters and decreased the learning rate as the model loss converged.

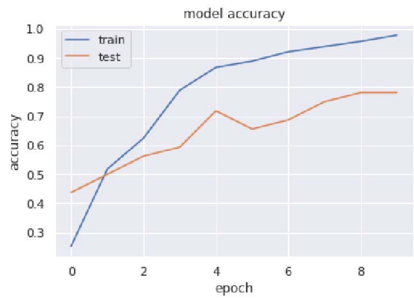

Fig. 8. Aspect Classification Model Accuracy

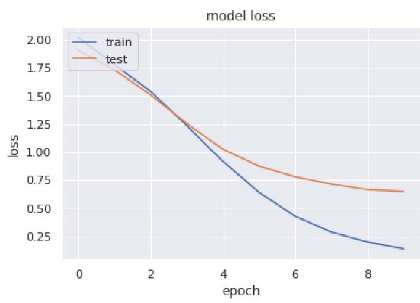

Fig. 9. Aspect Classification Model Loss

TABLE I
ASPECT CATEGORY MODEL EVALUATION

	Dataset size	Accuracy
Train	290	98%
Validation	30	92%
Test	45	80%

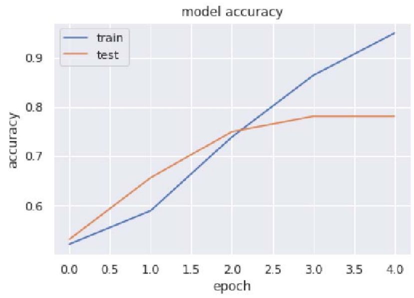

Fig. 10. Sentiment Polarity Model Accuracy

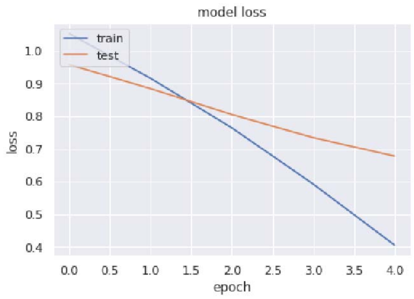

Fig. 11. Sentiment Polarity Model Loss

TABLE II
SENTIMENT POLARITY MODEL EVALUATION

	Dataset size	Accuracy
Train	290	95%
Validation	30	84%
Test	45	77%

Since, such a task had not been done before for an Indian audience, there was no benchmark for comparison. This made it difficult for us to judge the viability of our approach in real-world scenarios.

We used a hold-out set to evaluate the test accuracy. The distribution of aspects and sentiment polarities in the hold-out set was proportional to the dataset. This made sure that it was prepared to be as representative of real world scenarios as possible. The experimental results are consistent with expectations, which means that the model accuracy meets the requirements of practical application.

We used and optimized the Adam optimizer with suitable hyperparameters to get the best possible results. Adam [10] is a combination of Root Mean Square Propagation (RMSProp) and Stochastic Gradient Descent, which has been shown to work well with deep neural networks.

We managed to achieve a train accuracy of almost 98% and a test accuracy of almost 80% on the aspect classifier neural network. Similarly, the sentiment polarity neural network yielded almost 95% train accuracy and 77% test accuracy respectively.

VI. CONCLUSION

In this paper, we have presented an approach centered around ABSA for evaluating student housing facilities. It ensures effective use of data and better judges the emotional tendency of a user towards various aspects.

We have built two neural networks to extract and classify the aspect on which a review is based and then classify the sentiment polarity. We conducted extensive experiments to verify the effectiveness of the approach. The experiments showed that our models performed well.

Through the experiments, we found that the proposed method has many advantages. It manages the extraction and classification of text with high accuracy. Our approach has been fine-tuned to words and phrases often used by Indian users, which makes it suitable for practical applications.

The future work is to expand the dataset to include similar reviews in local languages to further widen the scope. We plan to experiment with new deep learning techniques to handle the text classification task. We also intend to implement our approach as a browser extension. This would help to classify reviews on various websites quickly.

REFERENCES

[1] A. Kumar, S. Kohail, A. Kumar, A. Ekbal, and C. Biemann, "Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, CA, USA, p. 1129–1135.

[2] A. Yenter and A. Verma, "Deep cnn-lstm with combined kernels from multiple branches for imdb review sentiment analysis," in *IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, New York, NY, USA, p. 540–546.

[3] Z. Shaozhong and H. Zhong, "Mining users trust from e-commerce reviews based on sentiment similarity analysis," *IEEE Access*, vol. 7, p. 13523–13535.

[4] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, p. 115–133.

[5] S. Dreyfus, "The computational solution of optimal control problems with time lag," *IEEE Transactions on Automatic Control*, vol. 18, p. 383–385.

[6] M. Rampurawala, "Classification with tensorflow and dense neural networks." [Online]. Available: https://heartbeat.fritz.ai/classification-with-tensorflow-and-dense-neural-networks-8299327a818a

[7] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, p. 315–323.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, p. 1929–1958.
[9] R. Gómez, "Understanding categorical cross-entropy loss, binary cross-entropy loss, softmax loss, logistic loss, focal loss and all those confusing names." [Online]. Available: https://gombru.github.io/2018/05/23/cross_entropy_loss
[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, San Diego, CA, USA.