ASPECT BASED SENTIMENT ANALYSIS

ABSTRACT

The Aspect Based Sentiment Analysis is an analysis to identify and determine the sentiment for specific aspects in the reviews. The data given to us was analyzed in two approaches. In the first approach, a sentence-based analysis was done wherein we have considered the reviews in the data provided and preprocessed the text to train the classifiers and determine the accuracy of the classifier. In the second approach, we have taken into consideration of aspect term along with the text in the reviews and applied various techniques like dependency parsing, TF-IDF vectorization to train the classifiers and predict if the sentiment of the reviews as positive, negative or neutral.

INTRODUCTION

In the recent years, sentiment analysis and opinion mining have gained a lot of attention. The sole reason being the user-generated data on the internet, that consists of opinions and reviews about certain products and services on various shopping websites, blogs and review forms. These reviews are given by users who have certain feedbacks after using a product or service. They are beneficial for other customers while making decisions before purchasing a certain product or service. These reviews are also valuable for companies in this business to know what their customers need and what changes need to be done in their product or services.

For example, in case of a restaurant review page, there would be various reviews, written by the people who have visited the restaurant, considering different opinions about the dishes served, various cuisines available, the service provided, the overall ambience of the restaurant, even the waiting time to get the seat and how happy they are about visiting the place. We could get similar reviews when we consider a database for a laptop. The users might consider various aspects of the laptop like the storage capacity, speed, processor and if the laptop suffices their needs or not. However, not all the reviews are important and searching through this vast collection of reviews is a very tedious and time-consuming task. Thus, sentiment analysis has given a way to analyze these reviews to get the most relevant information to the users.

Aspect based sentiment analysis emphasis on the aspects present in the reviews and depending on a particular aspect in a sentence, the polarity of the review is determined. The polarity can be positive, negative or neutral with respect to every aspect in the sentence. Furthermore, these analyses are usually converted into graphs (this is beyond the scope of this project) and displayed on the websites that help the customers and companies to know the reviews about a particular aspect of their product or services.

TECHNIQUES

In this project, we have been given two datasets consisting of restaurant and laptop reviews respectively on which aspect-based sentiment analysis has been applied. The data preprocessing, features used in this project and the classification models that we have tried are discussed below:

I. Data Preprocessing:

Firstly, we have converted all the data to lowercase so that it becomes easier for the computer to understand. There are some techniques that count the individual words very precisely and, in this case, they might count a lower-case and upper-case of the same word separately. The '[comma]' is converted into the actual symbol ',' for getting a proper statement. The data is then resampled. Resampling is used to repeat certain sentences to balance the classes. In the data provided the neutral class sentences were very few compared to positive and negative class and thus resampling was done. Later, these reviews are tokenized. Tokenization breaks the review text into words and thus a bag of words was created.

II. Features:

The features are then extracted from this bag of words. After the features are extracted the stop words are removed. Then the TF-IDF vectorization is applied using these features as vocabulary. The Stanford core NLP libraries consists of various relations between words. In this project we have used amod(adjective modifier), advmod(adverb modifier), rcmod(relative clause) and cc(coordination). These modifiers find the adjective, adverb, clauses and conjunction words from the sentence that help to determine the sentiment of the aspect term. Then these words are given higher weights in the vector formed from the vectorization.

The dependency parsing is done by creating a tree. This tree is then converted into a graph for easy retrieval. In this graph, the words act as the nodes and the edges between the nodes are the relation between the words. This makes it easier to access the words that give opinion about every aspect in the sentence.

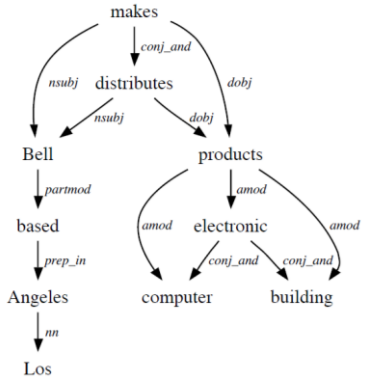

Figure 1: Graphical representation of the Stanford Dependencies for the sentence: Bell, based in Los Angeles, makes and distributes electronic, computer and building products. [1]

III. Classification Methods:

The 10-fold cross validation is applied on the given datasets and supervised learning is processed to get the accuracy of the models. The models used for the system are:

Random Forest Classifier: This is an ensemble classifier that consists of decision trees for individual sentences and output classes for each. It then outputs the mode of these classes. This was the best classifier in our case.

Naïve Bayes Classifier: This technique is based on the Bayes Theorem which assumes independence among the features. The naïve bayes classifier gave a pretty low score for the neutral class in our class. Thus, it wasn't used further.

Support Vector Machine(SVM): This technique is based on the separating hyperplane. The points are plotted and SVM finds the optimal boundary between the possible outputs. However, in our case, the f1-score and accuracy were biased towards the positive class more than the negative and neutral classes.

EVALUATION

The above methods were used in both the approaches, namely, the sentence-based analysis and the aspect-based sentiment analysis and were applied on both the datasets (i.e. Restaurant reviews and Laptop reviews). The results obtained after training the above classifiers are as below:

The results for Data-1:

Approach-1:

DATA_1	**RANDOM FOREST**		**NAÏVE BAYES**		**SVM**	
	Positive	Negative	Positive	Negative	Positive	Negative
Precision	0.7706	0.6926	0.7431	0.7136	0.7608	0.6998
Recall	0.7674	0.7776	0.8085	0.7139	0.7908	0.7744
F1 score	0.7676	0.7321	0.7737	0.7136	0.7751	0.7349
Accuracy	0.7154		0.7103		0.7117	

Approach-2:

DATA_1	**RANDOM FOREST**			**NAÏVE BAYES**			**SVM**		
	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral
Precision	0.9802	0.9782	0.9671	0.9608	0.9862	0.9340	0.9726	0.9762	0.9345
Recall	0.7635	0.8423	0.8609	0.7333	0.6854	0.6780	0.6312	0.6615	0.5349
F1 score	0.8582	0.9041	0.9091	0.8301	0.8063	0.7718	0.7632	0.7836	0.6517
Accuracy	0.8186			0.6822			0.5795		

The results for Data-2:

Approach-1:

DATA_2	**RANDOM FOREST**		**NAÏVE BAYES**		**SVM**	
	Positive	Negative	Positive	Negative	Positive	Negative
Precision	0.7015	0.5085	0.7578	0.5283	0.7250	0.5089
Recall	0.8669	0.3371	0.8049	0.4483	0.8578	0.4164
F1 score	0.7745	0.4033	0.7790	0.4806	0.7846	0.4487
Accuracy	0.6379		0.6413		0.6511	

Approach-2:

DATA_2	**RANDOM FOREST**			**NAÏVE BAYES**			**SVM**		
	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral
Precision	0.9791	0.9798	0.9667	0.9571	0.9710	0.9282	0.9628	0.9764	0.9243
Recall	0.8316	0.8706	0.8471	0.7553	0.7171	0.6800	0.6552	0.6737	0.5138
F1 score	0.8984	0.9217	0.9016	0.8403	0.8238	0.7737	0.7751	0.8128	0.6500
Accuracy	0.8448			0.7022			0.5974		

CONCLUSION

In this project, we concluded that for the given datasets and the techniques that we have used the Random Forest classifier worked the best for us. Later, for the demonstration, the 10-fold cross validation was removed, and the new datasets given to us were used as the testing data whereas the previous datasets were entirely used for training purposes. The random forest classifier was trained and dumped and was later used to determine the polarities of the reviews given to us in the testing datasets. The future scope of this project could be, to extract more information from the dependency tree by using the other relations (approximately 50 more relations available) present in the Stanford core NLP libraries.

REFERENCES

1. https://nlp.stanford.edu/software/dependencies_manual.pdf
2. https://www.sciencedirect.com/science/article/pii/S2090447914000550
3. https://link.springer.com/content/pdf/10.1007%2Fs10791-008-9070-z.pdf
4. http://www.nltk.org/book/ch06.html
5. https://www.thinkful.com/projects/building-a-text-classifier-using-naive-bayes-499/
6. https://bbengfort.github.io/tutorials/2016/05/19/text-classification-nltk-sckit-learn.html
7. https://ieeexplore.ieee.org/document/7960005/?reload=true
8. http://nlp.cs.aueb.gr/theses/ipavlopoulos-thesis.pdf
9. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7960005
10. https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6
11. https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a
12. https://www.cs.uic.edu/~liub/publications/acl17_LifelongCRF.pdf
13. http://www.aclweb.org/anthology/S14-2145
14. https://www.oreilly.com/ideas/sentiment-analysis-with-apache-mxnet