ASPECT BASED SENTIMENT ANALYSIS

LAKSHMI DIVYA JILLELLAMUDI KAMALA SIVARAMAN LAKSHMIPATHY

ABSTRACT:

The aspect-based sentiment analysis technique has a greater potential of analyzing sentiments for given statements more accurately than the general sentiment analysis as this approach intends to use the aspect of the given sentence to predict the opinions or the sentiments. There is a lot of active research going on in this area. Here, we built multiple classifiers, out of which, SVM classifier achieved the higher accuracy for the given datasets. The data pre-processing, models utilized, and results are also included as a part of this report.

INTRODUCTION:

Two sets of training data were provided, one each in the tech and food review domain. The training data consists of the sentiments based on the aspects of a particular domain. The opinions were of three types [+1, 0, -1] corresponding to positive, neutral and negative opinions. The task was to predict the opinion for a statement given the aspect terms. The data was preprocessed to be used by the classifier, followed by feature engineering in order to evaluate the aspect's opinion and the results were generated using stratified cross-validation with 10 folds.

DATA PRE--PROCESSING:

1. Removal of stop words:
The stop words in the English language were removed using the NLTK corpus.

2. Basic string pre-processing:
In order to utilize the given data, few basic string processing techniques were utilized. We replaced [comma] with a "," and converted all the characters to the lower-case for easier processing. We also considered a maximum characters threshold as 500 in a sentence and dropped the sentences that did not fit this condition.

3. Stemming:
The words in the statements were converted to their stem forms using the technique of stemming in NLP.

FEATURE ENGINEERING:

1. TF-IDF:

We used this scheme to represent the statements in the dataset based on their importance in the statement corresponding to the given dataset. The TF-IDF values give the importance of words in the dataset per statement. Every statement is converted to a vector format consisting of TF-IDF values of each word. We experimented with different n-gram models and are utilizing uni, bi and tri-gram models.

The tf-idf score formulation is as follows:

$$\mathbf{tfidf_{t,d} = tf_{t,d} \times idf_t}$$

2. Opinion Lexicon incorporation with proximity value calculation:

Professor Bing Liu's opinion lexicon[1] is utilized to generate a proximity value for each opinion word in a sentence. Based on the proximity of the opinion word to the aspect term, weighted values will be assigned based on the polarity of the opinion word. Finally, the weighted values are added to obtain a weighted opinion value for the sentence with respect to the aspect term. This value was appended to the TF-IDF vector generated from the dataset for each sentence.

3. Sampling:

Since the provided training data has class imbalance (class 0 had significantly lesser number of examples.).Three different sampling techniques were experimented and the best results were achieved using oversampling.

MODELS UTILISED:

1. Logistic regression classifier:

We are trying to predict the probability that an aspect term has a certain opinion given a sentence context. This classifier is one of the basic classifiers we experimented by extending the one versus all classification method.

2. Support Vector Machines:

SVMs are known to perform well in the sentiment analysis problems. On extending the same idea to include the aspect terms decent accuracies were obtained initially. On introducing the bi-gram and tri-grams, SVMs performed the best among all the experimented classifiers. 'Crammer_singer' model was used for multiclass classification, which performed better than the SVM Kernels (poly and rbf were utilized).

3. Naive-bayes classifier:
Naive Bayes works on the conditional independence of a pair of features. This enabled higher weight representation of bi-grams and tri-grams thus producing better results. Naive Bayes was also chosen due to high performance in text-classification in cases involved in multinomial distributions.

4. Adaboost classifier:
It is one of the ensemble methods. It was implemented with the logistic regression classifier as the base classifier. However, this method produced lower accuracy which is attributed to the non-linear nature of the underlying classifier.

5. Decision tree classifier:
It was one of the earliest classification techniques applied to solve the problem. It was chosen due to its ability to select the features with most importance for the classification process, thus picking out most opinionated words automatically. However, this approach was not effective across for all the three targeted classes.

Experiment results:

Dataset 1: Tech dataset

Classifier	Positive class			Negative class			Neutral class			Accuracy
	P	R	F1	P	R	F1	P	R	F1	
Logistic regression	0.7744	0.8115	07925	0.7260	07934	0.7582	0.6464	0.4655	0.5413	0.6362
SVM	0.8414	0.8040	0.7697	0.7947	08336	0.8137	0.8292	0.8592	0.8439	0.8208
Naive Bayes	0.6041	0.8688	0.7127	0.8274	0.5213	0.6396	0.7963	0.7420	0.7682	0.6785
Adaboost	0.7873	0.7756	0.7814	0.7491	0.7020	0.7248	0.6984	0.4541	0.5504	0.7233
Decision Tree	0.7476	0.7587	0.7531	0.7280	0.6734	0.6996	0.6663	0.3689	0.4748	0.7136

Dataset 2: Food dataset

Classifier	Positive class			Negative class			Neutral class			Accuracy
	P	R	F1	P	R	F1	P	R	F1	
Logistic regression	0.7263	0.8830	0.7970	0.5440	0.3838	0.4501	0.4267	0.2717	0.3320	0.6640
SVM	0.8978	0.7841	0.8371	0.8171	0.8590	0.8375	0.7993	0.8595	0.8283	0.8342
Naive Bayes	0.7165	0.7791	0.7465	0.8060	0.7721	0.7887	0.7671	0.7324	0.7494	0.7615
Adaboost	0.8713	0.7844	0.8251	0.6365	0.6897	0.6550	0.4821	0.5967	0.5384	0.7265
Decision Tree	0.6727	0.8694	0.7585	0.3981	0.2258	0.2881	0.3466	0.1987	0.2526	0.4721

Future work and conclusion:

Out of all the models, SVM performed best with an accuracy of 82% and 83% on the given datasets. Further work could involve better-weighted approach given multiple aspects in a sentence, taking variable window size similar to Google's word2vec implementation. Additionally, deep learning methods could be utilized to gain higher accuracy.

References:

1. [1] Opinion Lexicon (Hu and Liu, KDD-2004)
http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar
2. A lexicon model for deep sentiment analysis and opinion mining applications (Isa Maks, Piek Vossen)
3. Web data mining: exploring hyperlinks, contents, and usage data (Liu, Bing)
4. Opinion Mining and Sentiment Analysis (Bo Pang, Lillian Lee)