A Project Report

On

Aspect Based Sentiment Analysis

Submitted for the course

Lab-based Project (CSN-300)

of Bachelor of Technology in Computer Science and Engineering

by

Deepanshu Singhal
(Enroll No. 14114022)
Tarun Kumar
(Enroll No. 14114068)
Tejal Singh
(Enroll No. 14114069)

Under guidance of
Dr. Sugata Gangopadhyay
Associate Professor, IIT Roorkee

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE
ROORKEE- 247667 (INDIA)

SPRING, 2017

Contents

Abstract

This project discusses how to provide the grading of an item (restuarant, laptops etc.) according the users' reviews. This project presents the model to quantify the text(reviews) from the user review forum into the fixed set of Aspects along with their ratings. These aspects are then divided into certain categories (food,price etc.) and users' overall opinion towards these categories is predicted. In this project we discuss various approaches to meet the required goal.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning, Support Vector Machines, Stanford CoreNLP

1 Introduction

E-commerce, as a new shopping and marketing channel, has led to an upsurge of review sites for a variety of services and products. To help the consumers and to maintain the reputation, firms require to monitor their consumers' feedback and for consumers, it is required to decide with whom they should have their shopping deal. In this context, Aspect Based Sentiment Analysis (ABSA) i.e.- mining and summarizing opinions from text about specific entities and their aspects.
Recent years has seen rapid growth of research on sentiment analysis. Sentiment analysis has both business importance and academic interest. So far, most sentiment analysis research has focused on classifying the overall sentiment of a document into positive or negative. We would, however, often like to understand what are the specific sentiments towards different aspects of an entity, e.g. a restaurant review "Food is decent but service is so bad." contains positive sentiment towards aspect food but strong negative sentiment towards aspect service. Classifying the overall sentiment as negative would neglect the fact that food was actually good.
newpage

1.1 Modules

Entire project work is divided into Four Modules:

1. Aspect Term Extraction
2. Aspect Term Polarity Extraction
3. Aspect Category Classification
4. Category Polarity Extraction and Grading

1.2 Dataset

Restaurant Review Dataset: We have used the dataset from Aspect Based Sentiment Analysis (ABSA), Task 4 of SemEval-2014. This dataset consists of

over 3K English sentences from the restaurant reviews of Ganu et al. (2009).
Example:

```
<sentence id="3121">
<text>But the staff was so horrible to us.</text>
<aspectTerms>
    <aspectTerm term="staff" polarity="negative" from="8"
    to="13"/>
</aspectTerms>
<aspectCategories>
    <aspectCategory category="service" polarity="negative"/>
</aspectCategories>
</sentence>
```

2 Aspect Term Extraction

2.1 Preprocessing

We used POS Tagger which is a part of Stanford CoreNLP. We extracted input in two formats. One containing only POS tag of words and the other one containing POS Tags as well as dependencies between words.

2.2 Approaches

1. **Hu and Liu, 2004 method**
   In this approach, we used first input format containing only POS tags of words. We first stored all the nouns present in reviews. There are multi-word aspects also in the dataset. For that we took consecutive nouns in a sentence and assumed that to be multi-word aspect. For simplification purpose, we only took the last word of our extracted multi-word aspects and modified frequency. We now took frequency of aspects by not considering multi-word aspects. If a noun has frequency greater than a certain threshold value (10, in our case), then it is assumed to be an aspect. Now, there may be some aspects that may be less frequent and are left out because of above approach. To, include them, we took use of opinion words. Opinion words are mostly adjectives. We extracted opinion words present with the selected aspects. Then, we checked for those opinion words in the reviews and if there is an undetected noun present near it, then that noun is also included as aspect term.

2. **Hu and Liu method with Lemmatization**
   This method uses second format of input. In this approach, we used

lemmatization along with above mentioned approach. The frequency for lemmatized nouns were considered instead. For adjectives, instead of searching them in sentences, we used the dependencies - 'nsubj' and 'amod' for relation between noun and adjectives. Adjectives were also used after lemmatizing.

3 Aspect Term Polarity Extraction

3.1 Approaches

We extracted aspect terms and their corresponding polarity labels from given XML corpus. This extracted data will be then used in both of the following approaches. We divided this data in ratio 4:1 for training and testing respectively.

1. **SVM**
   In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.
   We created a model pipeline consisting of CountVectorizer, TfidfTransformer and OneVsRest LinearSVC. In this model we provide the input data and their corresponding polarity labels and train the classifier.
   CountVectorizer converts a collection of reviews to a matrix of token counts.
   Transform a count matrix to a normalized tf or tf-idf representation. We gained accuracy of close to 60% and F-score also close to 60%.

2. **LSTM**
   Long short-term memory (LSTM) is a recurrent neural network (RNN) architecture (an artificial neural network) proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber. Like most RNNs, a LSTM network is universal in the sense that given enough network units it can compute anything a conventional computer can compute, provided it has the proper weight matrix, which may be viewed as its program. We used pre trained embedding weights (glove) which will be used in the first layer of our neural network. MAX_SEQUENCE_LENGTH is the maximum number of words in any sentence feeded to LSTM. We created embedding matrix using these weights of sentence in which each element represent an embedding vector corresponding to the word at that index. We concatenated the aspect term at the end of this embedding. Next Layer of our neural network is LSTM layer having input size of 100 embedding sequences. Third layer has input dimension of 128 which uses "relu" activation funciton. We used dropout of 20%. Last layer has activation function "softmax" and has three outputs corresponding to three labels (positive, negative, neutral). We gained accuracy of 62% on testing data.

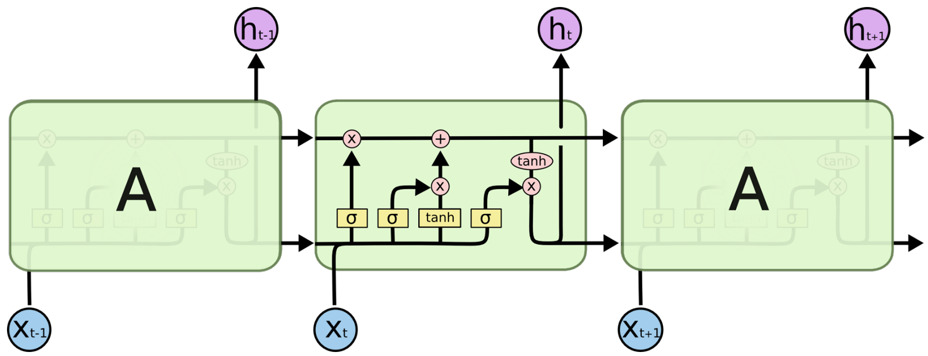

Figure 1: Structure of LSTM

4 Aspect Category Classification

4.1 Introduction

We classified aspects into four different categories - food, service, ambience, and price.

4.2 Approaches

1. **Similarity Matching - aspects with categories**
   In this approach, we used wordnet from nltk.corpus. We matched similarity of each aspect with each category. We chose category with highest matching percent with the aspect and selected it only if it is greater than a threshold value (40%, in our case).

2. **Similarity Matching - aspects with aspects**
   This approach is an extension to above method. After implementing above method, similarity of an aspect is matched with every other aspect. If matching percent is greater than the similarity of the aspect with its category, then it is transferred from its previous category, to the category of highest matched aspect.

3. **Using Support Vector Machines**
   We used SVM Model similar to that we applied on aspect polarity detection. The input data is aspect terms and label is corresponding category. So this becomes category classification problem. We used data from 'SemEval-2016 : Semantic Evaluation' to train our model and testing is done on data from 'SemEval-2014 : Semantic Evaluation'.
   The variation of various metrics with confidence level of prediction (decision function) is shown in the results section. If Decision Function value is less than 0.25 for given prediction then we assume that it is giving false category, so we removed that aspect from values which will be used subsequently.

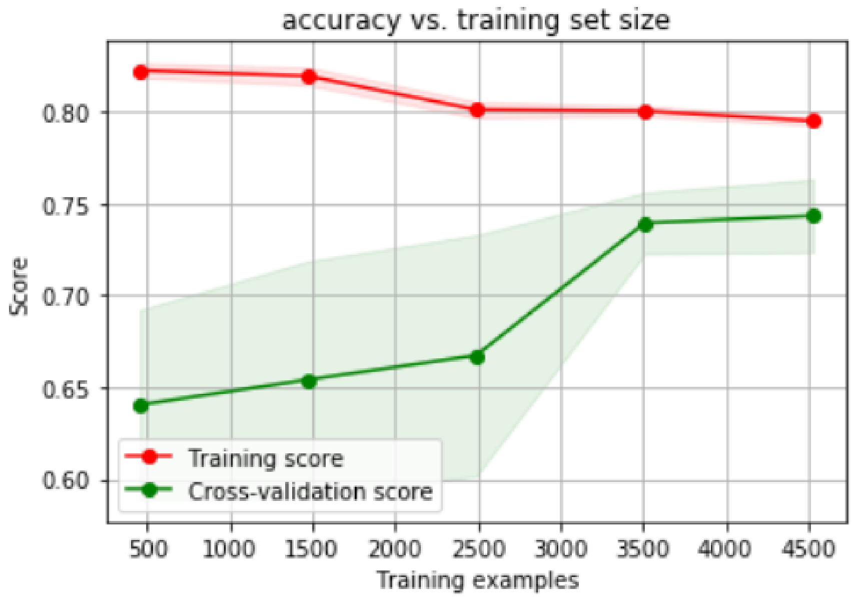

Figure 2: Score vs Dataset Size- Using SVM

5 Category Polarity Extraction and Grading

For finding the rating of a given category, first we are finding total number of aspects under each category that we have used in our project {food, ambience, price, service}. Then we aggregated the polarity number {-1(negative) ,0(neutral) ,1(positive)} aspects under given category. The resulting sum is then used to calculate the overall rating of each category.

6 Results

6.1 Aspect Term Extraction

	Hu&Liu	Hu&Liu with lemmatization
Precision	0.3497	0.3454
Recall	0.6277	0.5030
F-score	0.4491	0.4099

In our project we have used the first approach - Hu & Liu. As mentioned above in our case, all multiword aspects are coming in false negatives. Same is the case with false positives. So if we discard the effect of multiword aspects then count of false positives as well as false negatives is reduced significantly. Below graph (Figure 3) depicts the relation between scores and the frequency of nouns used for the aspect term extraction task.

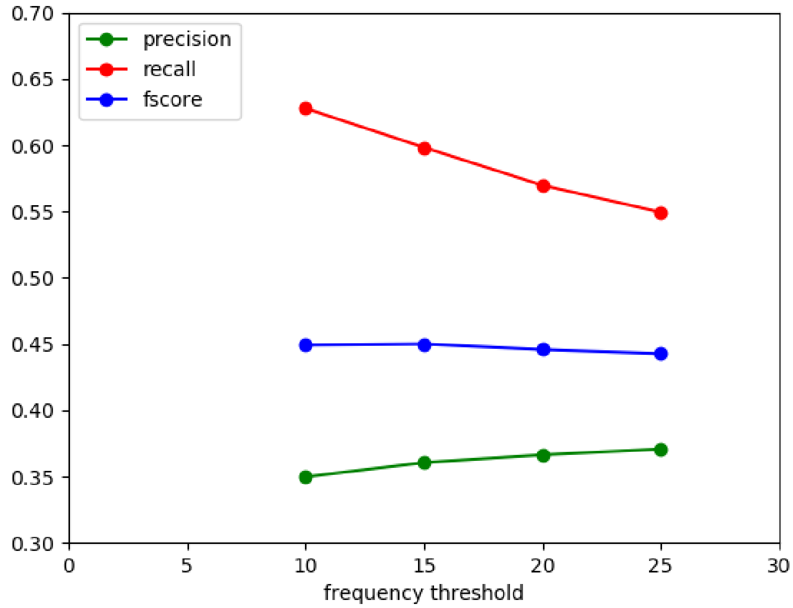

Figure 3: Hu & Liu Approach

6.2 Aspect Term Polarity Extraction

Figure 4 shows the F-score using the SVM model applied on Aspect Term Polarity Extraction.
Figure 5 shows the relation of validation accuracy with number of epochs when using LSTM model to extract the polarity of aspect terms.

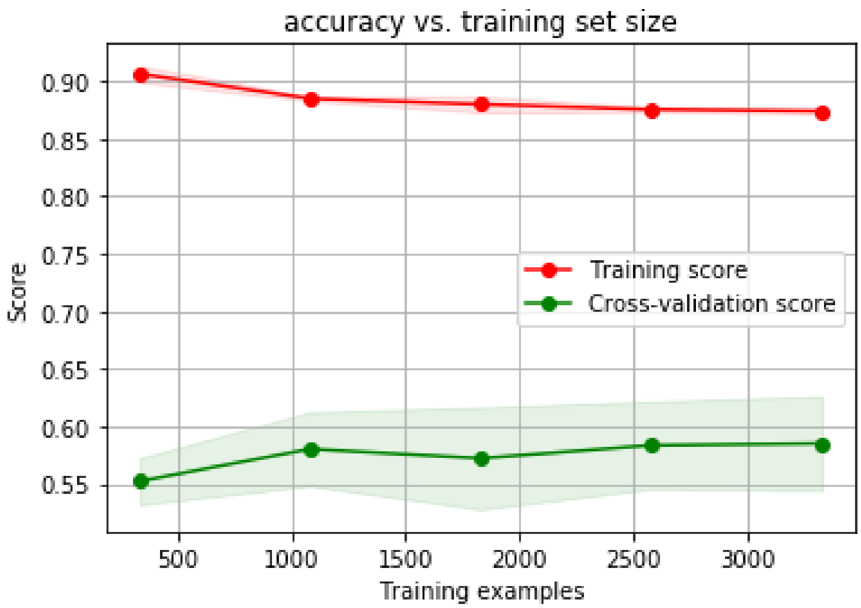

Figure 4: Using SVM

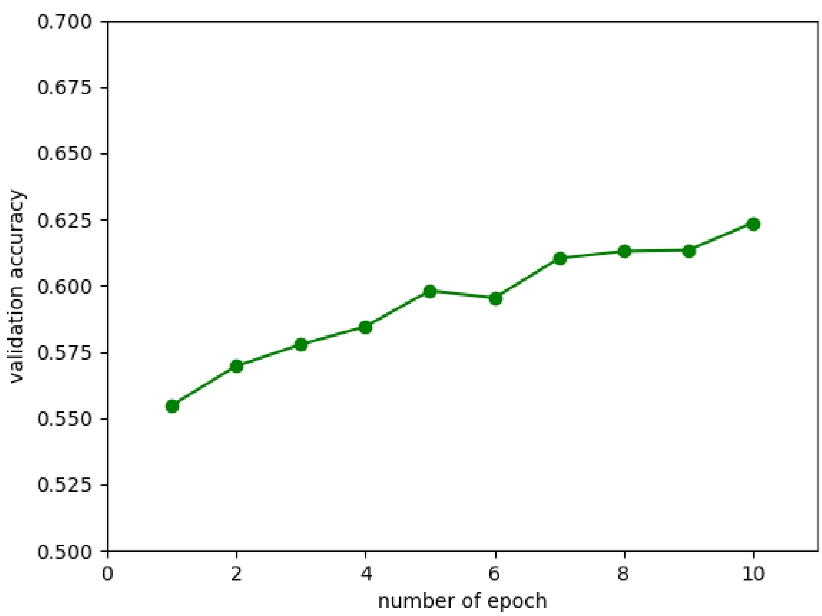

Figure 5: Using LSTM

6.3 Aspect Category Classification

	Similarity Matching		
	aspects with categories	aspects with aspects	Using SVM
Precision	0.6420	0.5384	0.7748
Recall	0.5532	0.5455	0.6656
F-score	0.5944	0.5658	0.7161

The above table shows the results we got from using different approaches. It can be seen that the results have significantly improved from what we got after aspect term extraction. This is because, for multi-word aspect we took only the last word and that word is correctly classified into category. For example, if we consider aspect "Grilled Chicken", then we extract the aspect "chicken" from it. This increases our false positive and false negative. But now this aspect is classified into category "food" as should be the case with "Grilled Chicken". We also tested the system by combining the approaches 1 and 3 mentioned in "Aspect Category Detection" section. But the result we got was similar to SVM approach, so we did not mention it.

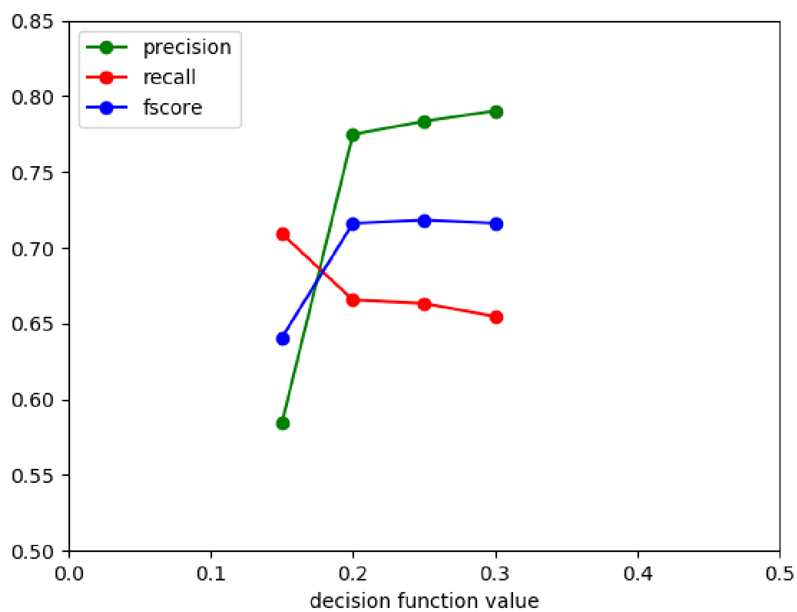

Figure 6: Final Approach - using SVM

6.4 Category Polarity Extraction and Grading

The following table shows the ratings of different categories along with the percentage of positive, negative and neutral reviews. The ratings and percentage of reviews were calculated on the basis of methodology explained in section 5 - Category Polarity Extraction and Grading.

	Rating	Positive Reviews	Negative Reviews	Neutral Reviews
Food	5	0.64	0.18	0.17
Ambience	6	0.72	0.16	0.10
Price	5	0.71	0.20	0.08
Service	3	0.60	0.27	0.13

References

[1] Hu and Liu(2004),
`https://www.cs.uic.edu/ liub/publications/kdd04-revSummary.pdf`

[2] GloVe: Global Vectors for Word Representation,
`https://nlp.stanford.edu/projects/glove/`

[3] Stanford CoreNLP,
`http://stanfordnlp.github.io/CoreNLP/`

[4] Long Short Term Memory (LSTM) Model,
`http://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/`