Aspect Sentiment Analysis Using Supervised Learning

Jurat Shayiding Yashika Goyal

Department of Computer Science University of Illinois at Chicago

Abstract

With the excessive growth of social media and online services, the customer reviews and opinions serve as a platform for decision making. It is important to analyze the reviews/opinions to help other users to validate their opinions or make purchases from online sites. A human mind forms a biased view with his preferences about an opinion. So, a system is trained to analyze the reviews. In this project, we propose some of the techniques we used for aspect sentiment analysis on restaurant and laptop data. Data preprocessing is done first on the provided training data. We use various type pf feature extraction i.e. TF-IDF vectorization, keywords extraction, POS tag features, dependency parsing. Then different classifiers like Multinomial Naive Bayes, Logistic Regression, SVM, SGD, Random Forest are used to create the model to classify new set of data and their performances are discussed in the report. The performance of the model is determined by F1 score, accuracy, precision and recall. The best accuracy for the test data was achieved by SGD model.

1 Introduction

Sentiment Analysis gives the sentiment of a document in general, but it is more useful to get the sentiments of individual components. So, the term aspect is introduced which refers to the components of entity. Aspect Sentiment Analysis is a research problem where a model is trained to find aspect terms in a document and sentiments of that particular aspect entity in the document. It analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language. ASPA has expanded to the field of health, online reviews, marketing, recommendations, social media, blogs, forums and more.

Sentiments and opinions are important to us for decision making. We validate our decisions on the basis of other people's' opinions, thoughts, sentiments. The choices we make, the way we see form perceptions about the reality is dependent on how other people view it. The recent growth in social media, web and technology has provided with a possibility of gathering the data and analyzing it to apply in the real-world applications. It is an active research problem in Natural language processing and data mining, text mining.

2 Emprical methods

Empirically, we have experimented with extracting various type of linguistic features including bag of words, POS tagging, key phrase, top key words with TF-IDF scheme,vectorization, N-gram models to build candidate features for training various type of machine learning models. In this project, we set up a procedure to extract linguistic features, where we use entropy and information gain to select the most relevant ones for training various machine learning models. In the following sections, we will outline data pre-processing, feature extraction and machine learning models that we used.

2.1 Data Pre-processing

we performed data preprocessing which is essential step to do. However, the controlling the quality of data preprocessing is critical to determine overall accuracy of training machine learning model. Our task includes the removal of all stopwords, punctuation, twitter hash tags, non-letter special characters and other symbols. However, the dataset had ‘commas’ instead of ‘ , ’ to avoid errors with csv file format, the commas in the dataset are replaced by space “ ”. The two datasets have stop words as in the English grammar, but they are not required in the features, yet stop words are obtained from NTLK corpus and removed. Note that by removal of stop words, the accuracy decreases sometimes. The tags like ["</?.*?>"," <> "] are removed using regular expression operations. We also removed twitter hash tags,emoji icons from given text sequences. The special characters of the form ["(\\d|\\W)+"] are removed using regular expression operations. New line character of the form [’(\r)+ ‘] and non-ascii character of the form [’[ˆ\x00-\x7F]+’] are also removed using regular expression operations. Last but not least, punctuation is removed from the datasets.Note: by removal of punctuation, the accuracy decreases sometimes.

2.2 Feature Extraction

In this project, we conduct various type of feature extraction that can be used for training our models. In general, we have tried to mine following features.

TF-IDF vectorization

In machine learning domain, using TF-IDF vectorization is commonly used as one of primal features to train model. To do so, we use scikitLearn library to convert text sequences into bag of words.

Keywords extraction

we are also interested to find too key words from sentence, to do so, we also use TF-IDF scheme to top n key words from the text sequences.

POS tag features

In NLP domain, exploring POS tag pattern in given sentence or text always bring us potential semantic and syntactic structure of given text sequence. To extract POS tag features for each sentence, we use stanfordcoreNLP library to mine POS tag features. In this project, we extract POS tag features by unigram, bigram, and trigram wise, to see using POS tag features can have significant impact on sentiment analysis prediction.

Nearrest adjective to given aspect term

Since dataset comes with aspect term, we are interested in to extract nearest adjective to the aspect term because adjective could give more information about the aspect of entity. To do so, we use powerfull NLP library SPacy to find nearest adjective.

bigram phrase detecton with TF-IDF scheme

To achieve better accuracy in sentiment prediction, we want to find out most relevant features to the aspect term and the context of sentence. We use gensim model to detect bigram phrase with TF-IDF score. We speculate that more fetures we can get, we can able to select the most relevant features that has significant impact on sentiment prediction.

Noun-adjective, verb phrase extraction

the last but not least, we use dependency parsing to extract verb phrase, and noun- adjective phrase from sentence respectively. To do so, we use SPacy library phrase extraction based on dependency parsing sequence in the sentence.

2.3 Machine learning Models

In this project, we use various machine learning models to assess overall performance of each in sentiment prediction. The models were trained by selecting different features iteratively. To make training model more efficient, we create one hot vector to place all extracted features in one place, which can be easier to train each models with different features recursively. We notice that using bigram phrase with TF-IDF score on stochastic gradient descent classifier render best performance as 75%. However, we are going to briefly introduce machine learning models that we have used in this project.

Multinomial Naive Bayes classifier

In web data mining domain, Naive Bayes classifier always shows amazing good per- formance on text classification task. Naive Bayes predicts the class considering each feature to be independent of the other. It is perhaps the most widely used machine learning techniques for sentiment analysis tasks. In this project we choose Naive Bayes classifier because it has strong classification performance on text classification.

Logistic regression

logistic regression is one of the important binary classifier it is commonly used to build predictive model which can be applicable to sentiment classification tasks. In our work, logistic regression shows fairly well performance on sentiment class prediction.

SVM

feature based support vector machine have been strongly capable of handling aspect based sentiment analysis tasks. In our task, SVM shows faily modest performance on training phrase, the reason we speculate that providing optimal parameters to SVM may get better accuracy.

SGD

SGD Classifier implements regularized linear models with Stochastic Gradient Descent algorithm. In general, SGD classifier requires some hyper parameter tuning to be done, to do so, we tried gridsearchCV to find best parameters, yet SGD classifier without tuning hyper parameters render one of the best accuracy along with logistic regression and Naive Bayes classifier.

Random forest classifier

random forest classifier is an ensemble model that fits many shallow decision trees on sub-samples of input data set. It has shown reasonably well performance than various machine learning models and it can be considered as wise choice than vanilla decision trees. Thus, we choose random forest classifier for sentiment classification task.

to sum up, we trained above machine learning models with different features itera- tively. We notice that controlling the quality of data preprocessing is critical for overall sentiment prediction, thus we leave a room to consider the the quality of text sequences after the data preprocessing which used for feature extraction.

3 Results

We use laptop review and food review dataset for feature construction and training different machine learning models. In our pipeline, training SGD and logistic regression with bigram phrase with TF-IDF score shows the best performance on sentiment prediction where corresponding overall accuracy up to 73% 75% respectively. In this project, nearest adjective to aspect term didn't show any significance on training phase, we believe that correctly capture semantic relation between certain parsing patterns and respective aspect terms might have a gap to fill. Top n key words with TF-IDF scheme also considered as second most relevant features which shows fairly good results on SGD, logistic regression and multi-nomial naive bayes classifier. However, in this project, we have done efficient feature engineering to mine most relevant features that facilitate sentiment prediction tasks. In our work, we select top three machine learning modesl which render best performance on laptop review and food review dataset. The statistical table of performance metric for best three machine learning models on laptop review and food review dataset can be shown in Figure **1** and **2**

Figure 1. models performance on laptop review dataset

classifier	class	Precision	recall	F1-score
Multinomial naive bayes	1	0.69	0.75	0.72
	0	0.88	0.17	0.29
	-1	0.68	0.89	0.77
	Avg	**0.73**	**0.70**	**0.66**
Logistic regression	1	0.74	0.84	0.79
	0	0.72	0.33	0.46
	-1	0.73	0.83	0.78
	Avg	**0.73**	**0.73**	**0.72**
SGD	1	0.78	0.76	0.77
	0	0.57	0.56	0.57
	-1	0.75	0.77	0.76
	Avg	**0.72**	**0.72**	**0.72**

Figure 2. models performance on food review dataset

classifier	class	Precision	recall	F1-score
Multinomial naive bayes	1	0.77	0.25	0.38
	0	0.76	0.10	0.18
	-1	0.66	0.99	0.79
	Avg	**0.70**	**0.67**	**0.66**
Logistic regression	1	0.72	0.40	0.51
	0	0.75	0.30	0.43
	-1	0.72	0.97	0.83
	Avg	**0.73**	**0.73**	**0.69**
SGD	1	0.68	0.61	0.65
	0	0.58	0.51	0.54
	-1	0.82	0.88	0.85
	Avg	**0.75**	**0.76**	**0.75**

Conclusion

To conclude, we performed aspect sentiment analysis on two datasets. We preprocessed the data by removing tags, stop words, punctuation, special characters. Then we trained our features using TF-IDF vectorization, POS tagging, dependency parsing. We trained multiple models i.e. Logistic regression, SVM, SGD, Random Forest, Naïve Bayes. For this project, the best performance was given by SGD (food review dataset - F1 score: 0.75 and laptop review dataset - F1 score: 0.72). In this project, we did not include the user profile, the time of post, holder, multiple aspects and other factors in our features. In future, we would like to include these factors. Also, for dependency parsing, we want to include more rules for feature extractions. We will also train the features using other models like stochastic gradient descent, semi-supervised learning or deep-learning models.

References

- Bing Liu. 2006. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, 2006 and 2011: Springer.
- Bing Liu. 2012. Sentiment Analysis and Opinion Mining https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf
- Scikit-learn: http://scikit-learn.org/
- NLTK: http://www.nltk.org/genindex.html
- TF-IDF vectorization : https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html