Topic :	IMDB Review classification using Sentiment Analysis
Team Members Name / NET-ID	Mary Pratima Yedluri [**mxy150630**] Kanchan Waikar [**kpw150030**]

Project Report: Opinion Polarity Detection Approaches

(IMDB Review classification using Sentiment Analysis)

1. Introduction

Users typically provide their reviews on IMDB website for movies as they are released. Movie reviews by expert users play a very important role in prediction of the movie's success. More positive reviews predict successful movie whereas bad reviews simply reflect badly on the entire movie crew. It is important to make machine understand and do sentiment detection based on natural language processing features that exist in the review comments that users provide. Our target is to automate polarity extraction of data in order to help machine identify the polarity of the review comment.

2. Problem Definition and Algorithm

2.1 Task Definition

In order to help the machine understand the polarity of the review without it being explicitly told is what the project intends to automate. The Project uses mathematically proven Machine learning algorithms in order to train the machine and predict the sentiment of the review based on several linguistic features. We compare different Machine learning approaches namely, bag of words model vs. statistical natural language feature heuristics.

2.2 Algorithm Definition

We need to train the model in order to predict polarity of the data. We start of with Bag of words model in which we use different word frequencies and their conditional probabilities in order to arrive to the model. But we must remember that this model is tied to the domain of training. When we use test dataset from a different domain, the model may not work very well.

In order to solve this problem, we want to see if statistical linguistic features of the review data could be used in order to train the model and to check whether the model will perform better than a coin flip.

The most important Linguistic features that we intend to use from dataset are as follows.

1. Vocabulary (Extracted at runtime using Multiterm Noun/verb entity extractor)
2. Review length
3. Number of Positive - number of negative words
4. Number of Stop words
5. Number of nouns, Verbs and adjectives

3. Experimental Evaluation

3.1 Methodology

Our hypothesis assumes that these features help predict the overall sentiment of the raw data up to at least 60% accuracy - better than the coin flip. We use SVN in order to form a linear regression line and execute the trained model on the unseen test data. Doing a 5 fold cross validation will help tune the model further.

We also use naive bayes classifier on the individual words in order to predict the sentiment class of the review - positive or negative.

Once done we intend to compare all three and share results of our experiment.

Data set used:

We are using the data set was compiled by Stanford Linguistics Professor Chris Potts and Stanford CS PhD student Andrew Maas. There are 25000 training examples and 25000 test examples, each of which is a textual review of a movie on IMDB. Some preprocessing (namely, bag of words encoding) has already been performed on the raw text.

- 80 MB compressedData set Link : Movie Dataset 1
  http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
  - 25000 training examples, 25000 test examples (used for Naive bayes)
    - 12,500 Positives
    - 12,500 negatives
  - 1500 samples (used for Rest of the algorithms)
    - 750 Positives
    - 750 Negatives

Another dataset that we intend to use purely for testing purposes is from a different source altogether. This data set was Introduced in Pang/Lee ACL 2004. Released June 2004. IT contains 1000 positive and 1000 negative processed reviews.

- Polarity Detection Dataset V2.0 : Movie Dataset 2

- 1000 positives and 1000 negatives

This data is available at
http://www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz

Input Feature Set Preparation:

We decided to extend our feature set by using the additional statistical information that could be extracted from the data. We used Stanford Part of Speech tagger in order tag the sentence and sentiWordNet in order to extract the polarity of the adjective as well as verb.

Initially when we started, we used LESK algorithm in order to understand the sense of the word but we realized that wordnet did not include information about word sense polarity, hence we used SentiWordNet for each of the Verb/adjective found in the text

- stopWordsProportion
  - Percentage of stop words present in entire review text. Number of stop words can have impact on word polarity detection. More stop words may mean less shallow opinion which is why we considered this as one of the input features. This is calculated using the formula numStopWords/numWords.
- numWords
  - Number of words in a review may give an insight into how detailed a review is. Shorter reviews imply less importance unless they are from very highly sought after reviewer. Since we wanted to keep the implementation very generic, we did not include features that could not be applied on any other data apart from imdb or even review data.
- positiveWordProportion
  - This is a Percentage number between 0-100 corresponding to number of words that are positive - this is calculated using the formula Positives/total number of words
- negativeWordProportion
  - This is a Percentage number between 0-100 corresponding to number of words that are negative- this is calculated using the formula negatives/total number of words
- numNouns
  - Although at shallow level it does not seem to be a valid feature input, but it does give an insight into how relevant a review could be. More Proper nouns can hint more appropriate review as opposed to the one in which there are very few.
- numAdjectives
  - Number of adjectives have impact on polarity depth of the content
- numVerbs

    - Number of verbs can have impact on polarity depth of the content

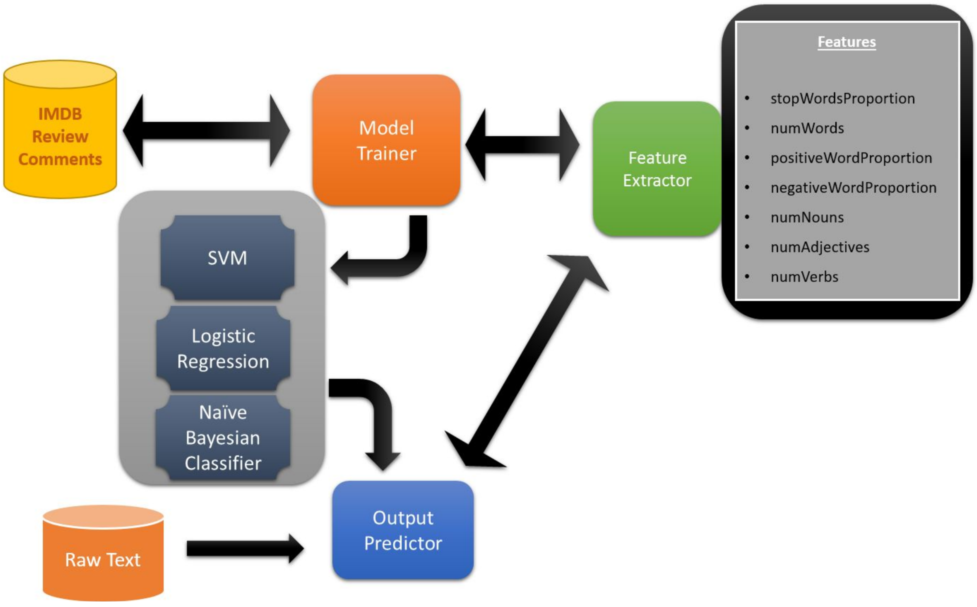
Technical framework:

- Java 1.8
- NLTK Libraries
- Stop Words List/Positive negative words list available on the internet.
- Senti-WordNet API for adjective/Verb polarity extraction
- Java - ML
- Stanford POS Tagger
- Weka Library

System Architecture:

The system would have following important components.

- Feature extractor
    - This Extractor is named as DataPreparationHelper and it creates a CSV file that needs to be converted into ARFF format
- model trainer
    - This class is named as GenericClassifier and has different Java-ML Implementations.



Architecture Diagram

3.2 Results:

Input Data Set used: Data Set 1

Sr. No	Approach	Feature Set	Training Accuracy	Test Accuracy
1.	Logistic Regression Lambda - 0.01 Learning Rate - 0.01 NumIterations - 150	Bag of Words Without Stop Words	94.54	74.7
2.	Logistic Regression Lambda - 0.01 Learning Rate - 0.01 NumIterations - 150	Bag of Words with StopWords file	100%	76.29
3.	Naive Bayes (On complete 25k dataset)	Bag of words (Without Stop Words)	94.996	82.316
4.	Naive Bayes (On complete 25k dataset)	Bag of words (With StopWords file)	96.284	83.0

Feature Set Used: (Custom Features)

When we use the feature Set of stopWordsProportion , numWords, positiveWordProportion, negativeWordProportion, numNouns ,numAdjectives, numVerbs, as input we get following as our output.

- Training : resources/training.arff
- Test: resources/test.arff

Statistics

Sr. No	Approach	Training Accuracy	Test Accuracy	5 fold testing results

1.	Naive Bayes	58.78%	57.45%	TP=564.0,
				FP=456.0,
				TN=295.0,
				FN=187.0

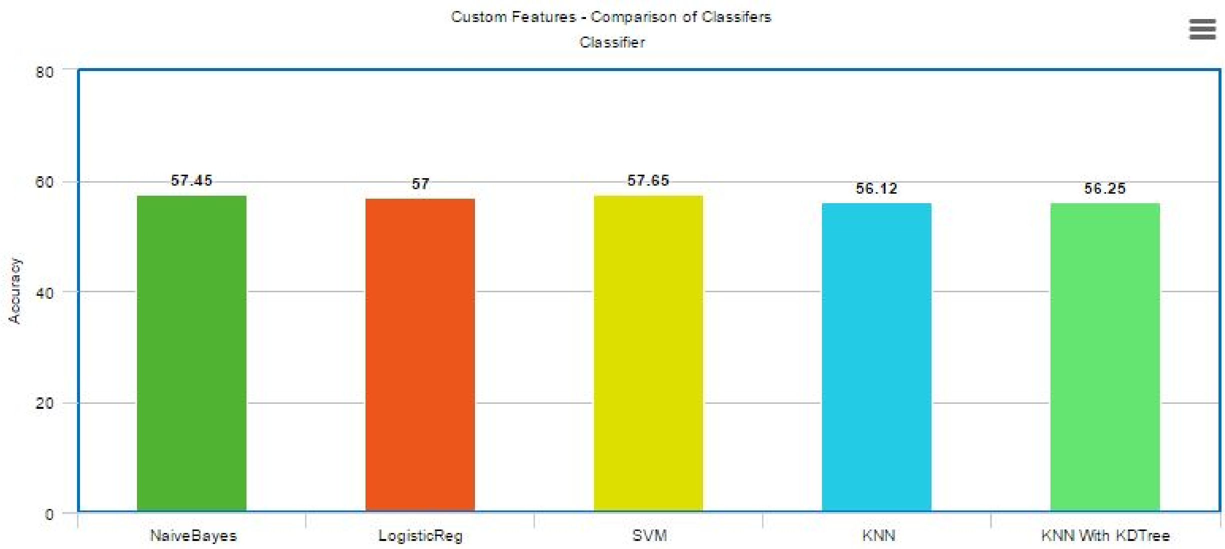
2.	Logistic Regression	61.38%	57.05%	TP=466.0,

				FP=303.0,
				TN=448.0,
				FN=285.0

3.	Lib SVM	60.51%	57.65%	TP=497.0,
				FP=413.0,
				TN=338.0,
				FN=254.0

4.	K Nearest Neighbours
	with k=75
		55.85%	56.12%	TP=484.0,
				FP=411.0,
				TN=340.0,
				FN=267.0

5.	KNN with KD tree
	With k=75
		58.58%	56.25%	TP=448.0,
				FP=380.0,
				TN=371.0,
				FN=303.0



We can see here that SVM performs better than other classifiers for same datasets.

Take 2: Training Data from Dataset 1, Test data from Data set 2

Sr. No	**Approach**	**Feature Set**	**Training Accuracy**	**Test Accuracy**

1.	Naive Bayes	Bag of Words Without Stop Words	99.26	63.8
2.	Naive Bayes	Bag of Words With Stop Words	99.46	58.0

Feature Set Used: (Custom Features)

When we use the feature Set of stopWordsProportion , numWords, positiveWordProportion, negativeWordProportion, numNouns ,numAdjectives, numVerbs, as input we get following as our output.

- Training : resources/training.arff
- Test: resources/test.arff

- Training : resources/training.arff
- Test: Data set 2 -resources/test.arff

Statistics

Sr. No	Approach	Training Accuracy	Test Accuracy	5 fold testing results

1.	Naive Bayes	58.78%	50.6%	TP=564.0,
				FP=456.0,
				TN=295.0,
				FN=187.0

2.	Logistic Regression	61.38%	49.95%	TP=466.0,
				FP=303.0,
				TN=448.0,
				FN=285.0

3.	Lib SVM	60.51%	49.9%	TP=495.0,
				FP=338.0,
				TN=413.0,
				FN=256.0

4.	K Nearest Neighbours
	with k=75
		55.85%	52.15%	TP=448.0,
				FP=381.0,
				TN=370.0,

				FN=303.0

5.	KNN with KD tree
	With k=75
		58.58%	52.15%	TP=448.0,
				FP=380.0,
				TN=371.0,
				FN=303.0

3.3 Discussion

We can clearly see that when test data does not belong to the domain of training data, the model suffers when bag of words approach is used. This is because the training model and test model deviate from each other a lot and there is not much that supports the inference of the predicted value.

Statistical approach does show almost constant results as they do the prediction that is independent of the domain. This proves that linguistic feature information can be further tuned and can be used for domain independent prediction of the polarity.

4. Related Work

There have been studies that have tried to solve the problem of polarity identification on single domain data and have proven that SVM performs better than other classifiers [

Possible enhancements:

- Tuning Input feature extraction further
  - Using alternate POS tagging mechanism
  - using stemming for detecting polarity.
- Trying other classifiers on the input data
- Implementation of SVM on Raw dataset

5. Bibliography

1. http://stanford.edu/~cpiech/cs221/homework/finalProject.html
2. Learning Word Vectors for Sentiment Analysis
3. http://sentiwordnet.isti.cnr.it/
4. https://wordnet.princeton.edu/
5. CSV to ARFF conversion tool - http://slavnik.fe.uni-lj.si/markot/csv2arff/csv2arff.php
6. http://www.cs.cornell.edu/people/pabo/movie-review-data/
7. http://acl2014.org/acl2014/W14-26/pdf/W14-2621.pdf