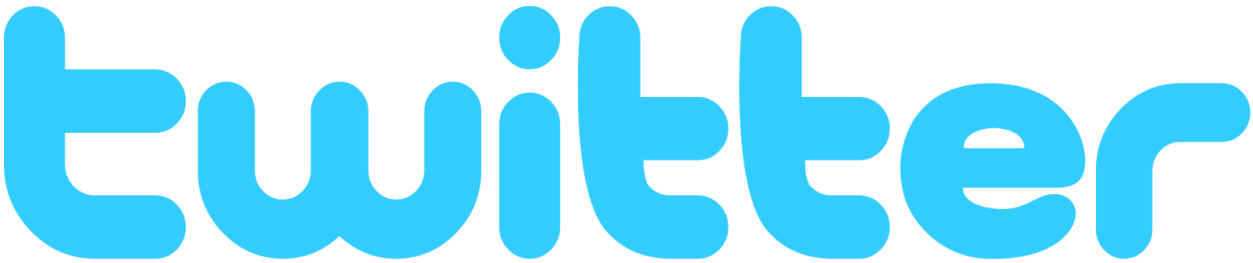
INFORMATION RETRIEVAL AND EXTRACTION

Sentiment Analysis on Twitter Data

Major Project



Abstract/Problem Statement

The aim of this project is to classify tweets based on their polarity mainly into three categories positive or negative or neutral.

Introduction

Microblogging websites have evolved to become a source of varied kind of information. This is due to nature of microblogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product. Many times these companies study user reactions and reply to users on microblogs. One challenge is to build technology to detect and summarize an overall sentiment.

In this project, we look at one such popular microblog called Twitter and build models for classifying “tweets” into positive, negative and neutral sentiment. We

build models for two classification tasks: a binary task of classifying sentiment into positive and negative classes and a 3-way task of classifying sentiment into positive, negative and neutral classes. We experiment with four types of models: unigram model, bi-gram model, tri-gram model and a feature based model.

Sentiment Analysis

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials.

Consumers can use sentiment analysis to research products and services before a purchase. Production companies can use the public opinion to determine acceptance of their products and the public demand. Movie-goers can decide whether to watch a movie or not after going through other people's reviews.

Microblog data like Twitter, on which users post real time reactions to and opinions about "everything", poses newer and different challenges. We have used tweets ending in positive emoticons like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative. We have built models using Naive Bayes and Support Vector Machines (SVM), we have come to a conclusion that SVM outperforms other classifiers.

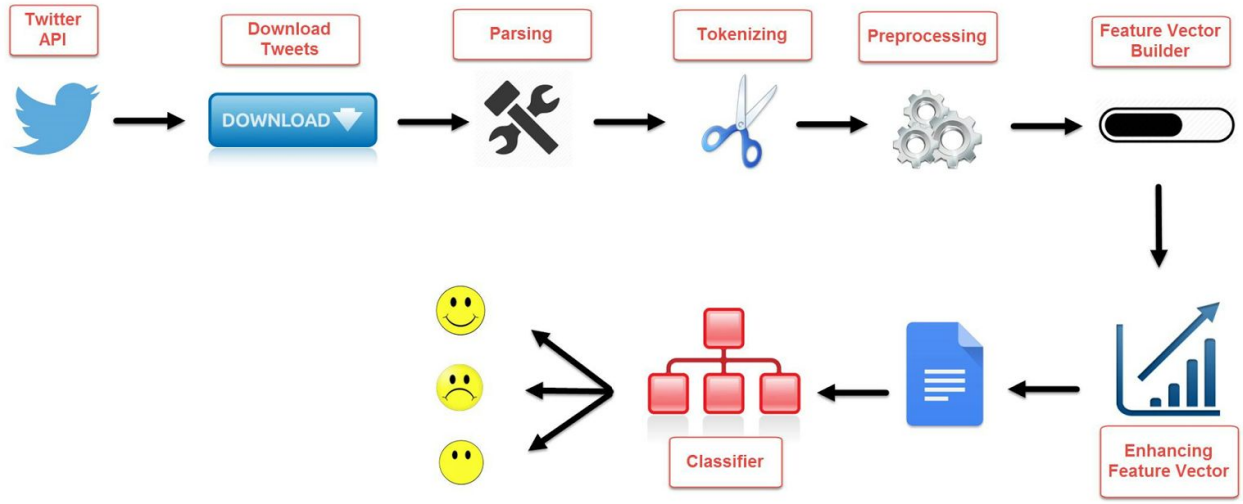
Goal

- Input: It will be a tweet in textual format downloaded from the twitter API.
- Output: It should be the label that specifies the polarity (positive, negative, neutral) of the given tweet.

Challenges

- Usernames are mentioned more often than not. Usually they consist of some alphabets and numbers, and do not contribute much towards sentiment classification, except for increasing the size of the feature vector.
- URLs too are not required in our task.
- Repeated letters People often repeat letters in some words, in order to stress upon a particular emotion. For example:- sad, saaaad, saaaddd. All of them mean the same, yet it is not possible to distinguish between them if guided only by their spellings.
- Hashtags Words in hashtags may be read different from the same word without the hash tag.
- Punctuation and additional spaces.

Approach/Flow



Download Tweets

The tweets are downloaded using the Twitter API.

Parsing

- We have totally 9684 tweets for training the algorithm and 8987 as testing tweets.
- Out of these tweets, few tweets are incomplete which are not useful.
- We removed those tweets and updated the data set.

Tokenization

- We didn't use ARK Tokenizer for tokenizing the tweets.
- Instead, we coded the tokenizer following the steps given in the paper "Sentiment Analysis of Twitter Data" by Apoorv Agarwal et al.
- So after parsing the original tweets will be processed by above steps and will be written to separate file which will be used in later stage.

Pre-Processing

Steps involved in pre-processing:

- Replacing emoticons with weights without disturbing it's polarity.
- Replacing usernames, URLs with symbols like ||T||, ||U||.
- Replacing words with more than 3 continuous repeating characters with only 3 occurrence.
- Removing all the stopwords and punctuation.
- Replacing all hashtags with the name in it.

- Replacing all the abbreviations with their full-forms.

Feature Vector

- With the processed dataset, we created feature vectors.
- The basic feature that was considered was of unigrams.
- A list of all unique unigrams(tokenized words) across the training set was constructed and it formed the feature vector for all the tweets.

Enhancing Feature Vectors

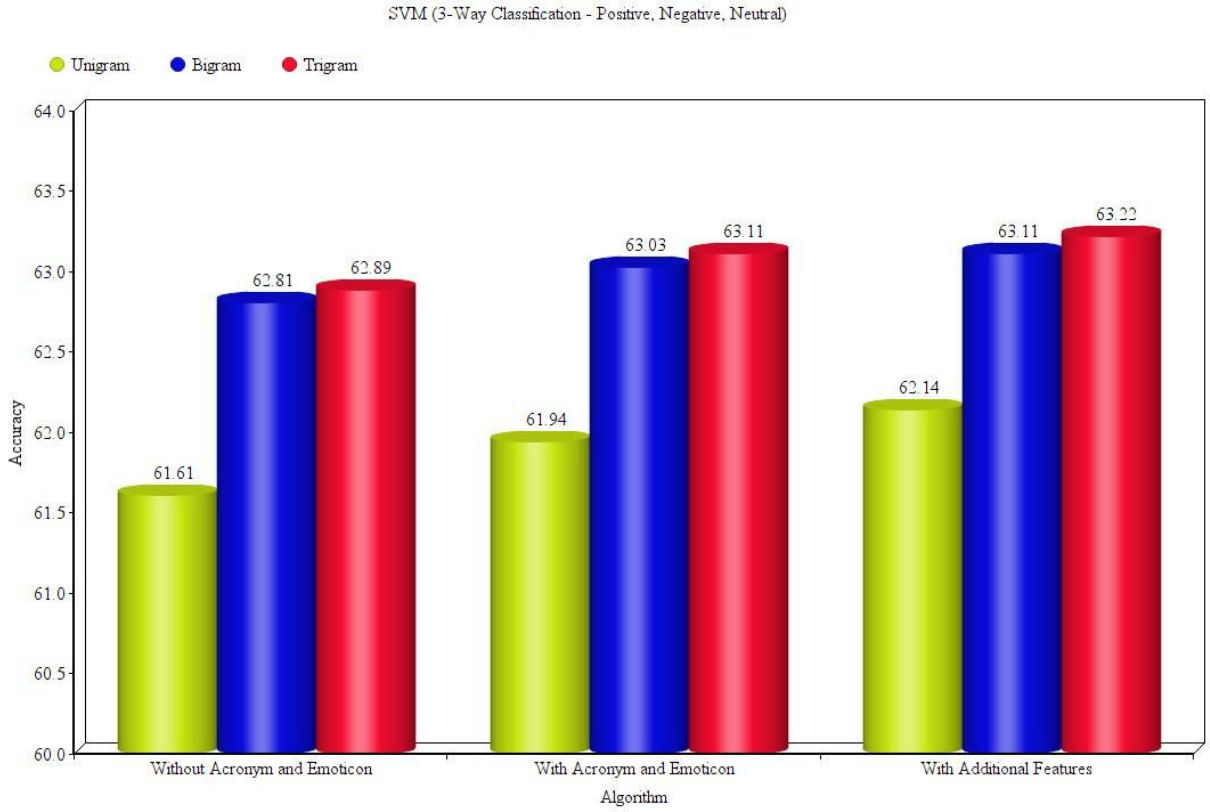
- To improve the efficiency of the algorithm, we have added few more features to the existing one.
- In addition to the unique unigrams, few features like the following were also added
    - Count of hashtags.
    - Bigrams.
    - Trigrams.
    - Count of emoticons.
    - Count of Negative words.

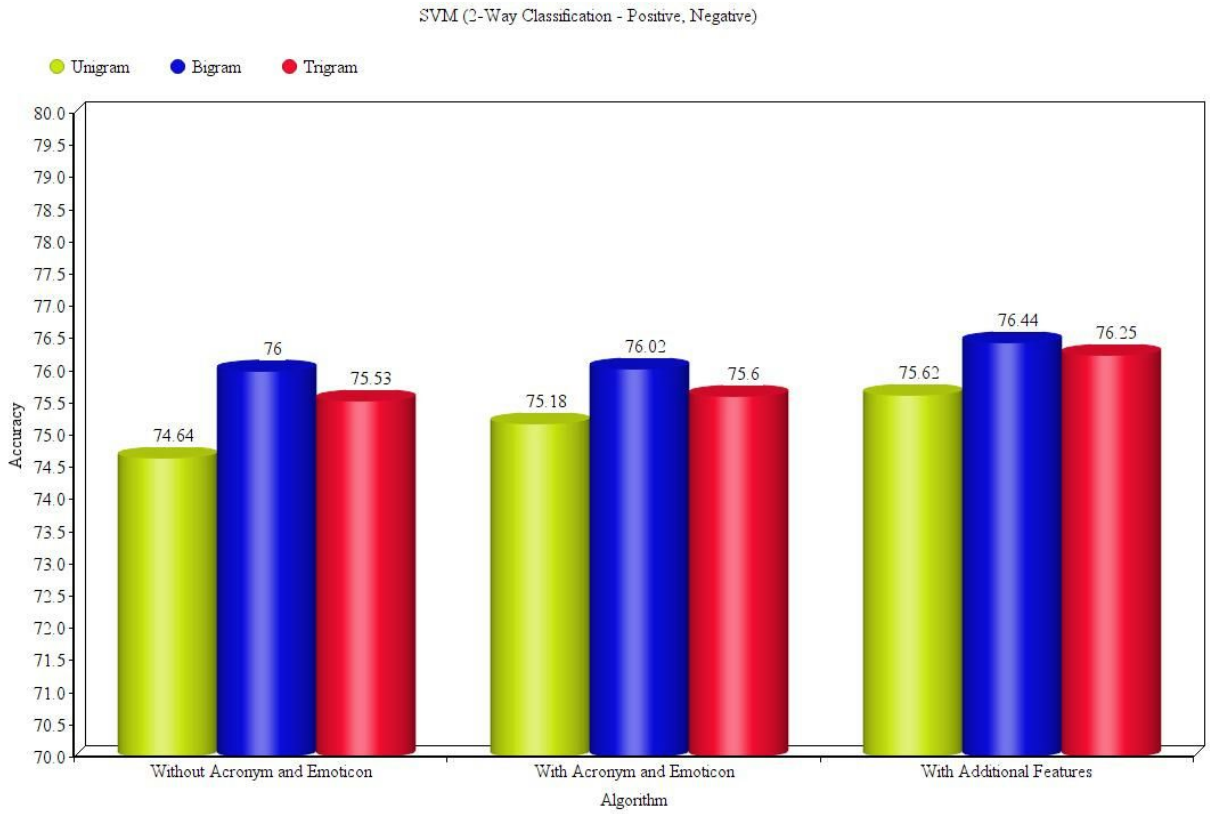
SVM

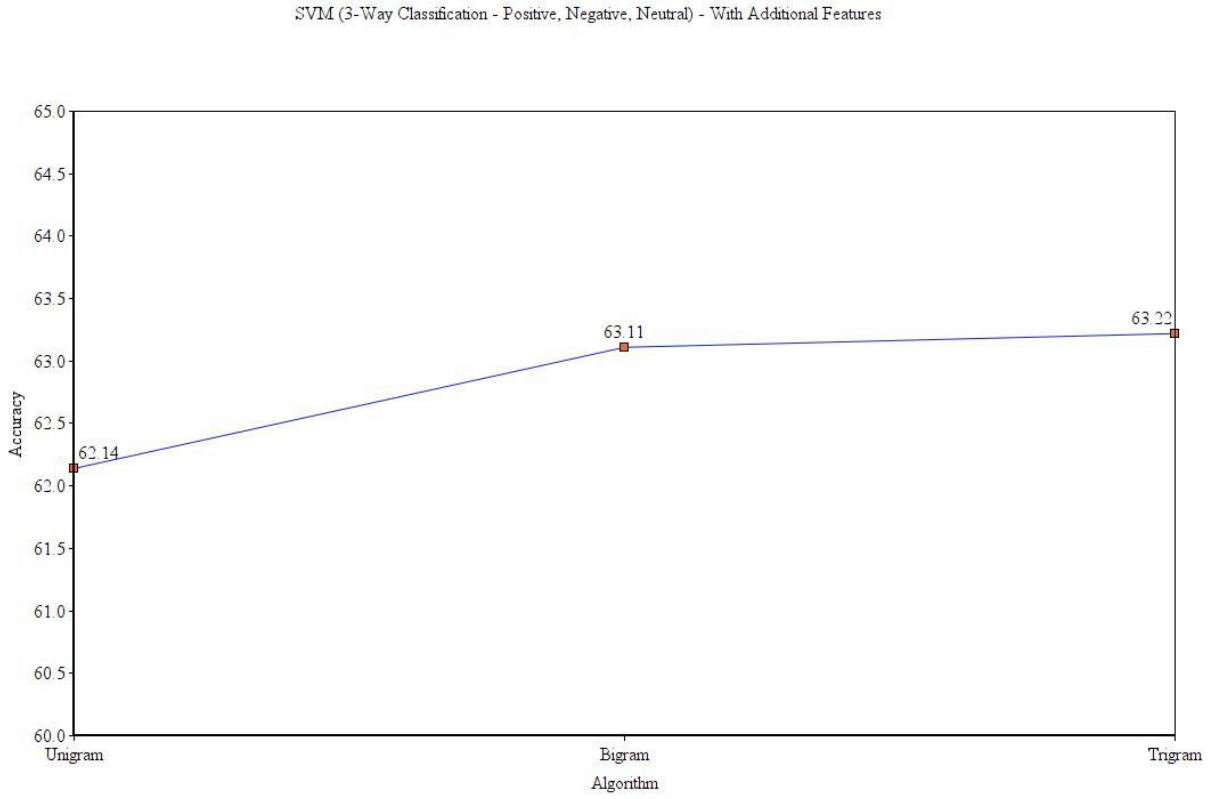
- The Feature Vector was written into a file in the format expected by libsvm classifier.
- A SVM Classifier was used and trained with the training file as input and the classifier is built.
- That classifier is used to predict the polarity of tweets in testing file.

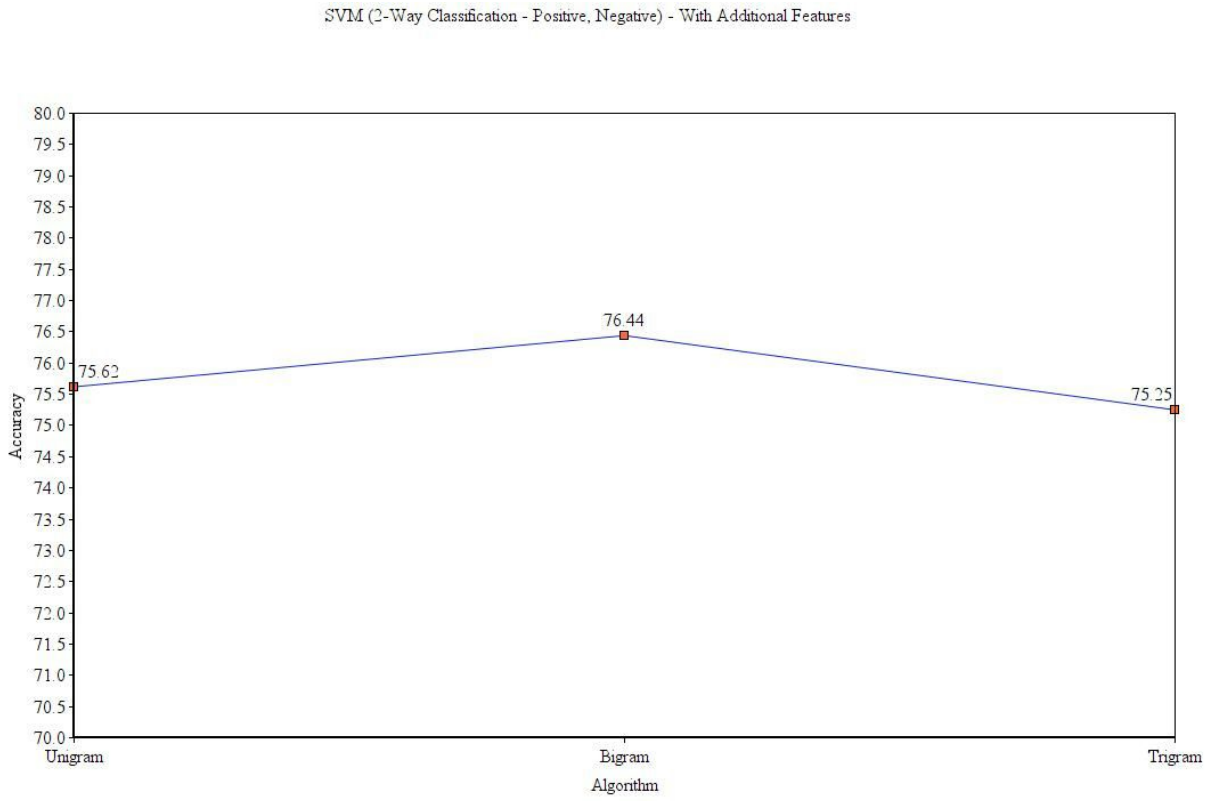
Results

		Without Acronym Emoticon handling	With Acronym Emoticon handling	With Additional Features
3-Way Classification	Unigram	61.61%	61.94%	62.15%
	Bigram	62.82%	63.03%	63.11%
	Trigram	62.89%	63.11%	63.23%
2-Way Classification	Unigram	74.64%	75.18%	75.63%
	Bigram	76.04%	76.02%	76.44%
	Trigram	75.53%	75.60%	76.26%




SVM (2-Way Classification - Positive, Negative)
Unigram
Bigram
Trigram
Accuracy
80.0
79.5
79.0
78.5
78.0
77.5
77.0
76.5
76.0
75.5
75.0
74.5
74.0
73.5
73.0
72.5
72.0
71.5
71.0
70.5
70.0
74.64
76
75.53
75.18
76.02
75.6
75.62
76.44
76.25
Without Acronym and Emoticon
With Acronym and Emoticon
With Additional Features
Algorithm


SVM (3-Way Classification - Positive, Negative, Neutral) - With Additional Features
65.0
64.5
64.0
63.5
63.0
62.5
62.0
61.5
61.0
60.5
60.0
Accuracy
62.14
63.11
63.22
Unigram
Bigram
Trigram
Algorithm



Important Terms

Information Retrieval and Extraction Course

IIIT-H

Major Project

Sentiment Analysis

Social Network

Natural Language Processing

Twitter

Support Vector Machine

Naive Bayes

Text Analysis