INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

KULLIYYAH OF INFORMATION AND COMMUNICATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

CSC 3304 – Machine Learning

Semester 1 2017/18

Section: [1]

Group Project

[SVM and Maximum Entropy Model for Sentiment Analysis of Tweets]

Group- 5

Group Members

[UMAR IBN ALI	1328435]
[MD. SHAIKOT HOSSEN	1233903]
[MD ABDUL QUADIR	1311903]
[S M RAJU	1416463]

Supervised by

Assoc. Prof. Dr. Amelia Ritahani Ismail

Table of Contents

1.0 ABSTRACT

Nowadays, Internet has become an online platform to exchange ideas and share opinions. Twitter is rapidly gaining popularity as it allows people to share and express their views about topics, or post messages all over the world. There are a lot of tweets regarding Apple products world-wide, as it is one of the leading technology companies in the world today. It is difficult for us to know how people feel about Apple on a major scale. It is not possible to manually extract the sentiment by considering each tweet. We proposed exclusive sentiment polarity detection approaches. SVM and Maximum Entropy can both be used to analyze sentiments. Our experiment will evaluate the algorithm using accuracy, precision, recall and F-measure as metrics. Thus, we can reach a conclusion is more effective for mining sentiments of tweets.

2.0 INTRODUCTION

From millions of online websites and social medias, millions of texts are generated basis on several issues and factors such as stories, discussions, decisions, blogs, feedback, twitter, postings etc. on daily basis. These data of texts have been reshaping corporations for analyzing their services, impacts of public sentiments, public emotions etc. These data are great opportunity to impact our social and political views and methods. But to analyses these vast datasets are not easy. Several researches and algorithms have been developed over these as sentiment analysis and opinion mining. For our project, our subject texts are collected from twitter. These tweets will be extracted as texts and filtered as plain texts, which will feed the process of sentiment analysis. During the process, the content as well as the texts is classified into several polarity measures as positive, negative and neutral. Here in this experiment, SVM and Maximum entropy algorithms will be used regarding the sentiment analysis to test their accuracy.

2.1 PROJECT OBJECTIVES

1. To classify the twitter texts into positive, negative and neutral sentiments.
2. To evaluate which algorithm does this classification with more accuracy.

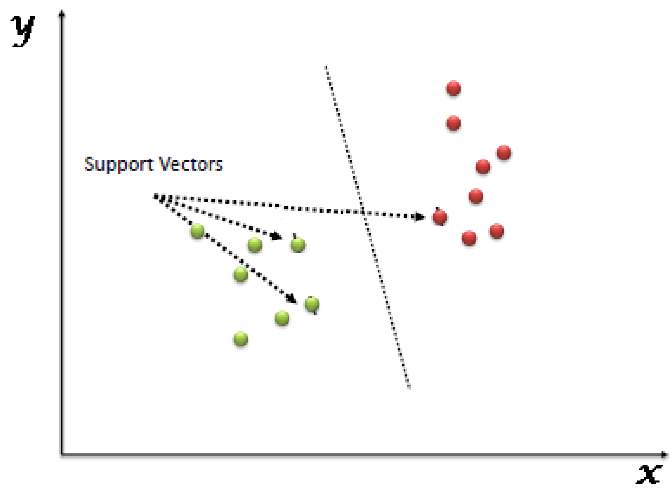
2.2 PROJECT SCOPE

Using twitter, we will mine the sentiments of people about Apple company around the world and analyse the sentiments to compare the two algorithms.

2.3 CHOSEN ALGORITHM

Our chosen algorithms are Maximum Entropy Model and SVM, we would like to find out which algorithm is better at classification of sentiment.

- SVM: Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



[Figure 1]

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

- Maximum Entropy Model: The Maximum Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit the training data, selects the one which has the largest entropy. The Max Entropy classifier can be used to solve a large variety of text classification problems such as language detection, topic classification, sentiment analysis and more.

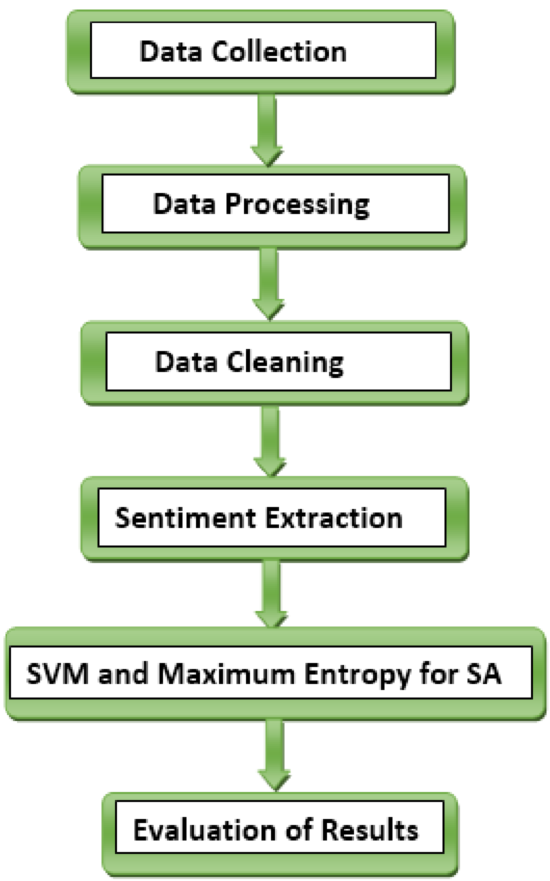
2.4 DATASET

Data is extracted from twitter. A twitter developer account is created, and tweets are extracted with keys generated by twitter. Tweets with the following word, '@Apple' are extracted from twitter.

2.5 SENTIMENT ANALYSIS

Sentiment analysis (SA) is the system of extracting the polarity of individuals' subjective opinions from plain normal language texts. Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". For example, a review on a website might be broadly positive about a digital camera, but be specifically negative about how heavy it is. Being able to identify this kind of information in a systematic way gives the vendor a much clearer picture of public opinion than surveys or focus groups do, because the data is created by the customer.

2.6 WORK FLOW DIAGRAM



3.0 LITERATURE REVIEW

Pak and Paroubek (2010) [1] proposed a model to classify the tweets as objective, positive and negative. They created a twitter corpus by collecting tweets using Twitter API and automatically annotating those tweets using emoticons. Using that corpus, they developed a sentiment classifier based on the multinomial Naive Bayes method that uses features like N-gram and POS-tags. The training set they used was less efficient since it contains only tweets having emoticons.

Parikh and Movassate (2009) [2] implemented two models, a Naive Bayes bigram model and a Maximum Entropy model to classify tweets. They found that the Naive Bayes classifiers worked much better than the Maximum Entropy model.

Go and L.Huang (2009) [3] proposed a solution for sentiment analysis for twitter data by using distant supervision, in which their training data consisted of tweets with emoticons which served as noisy labels. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM). Their feature space consisted of unigrams, bigrams and POS. They concluded that SVM outperformed other models and that unigram were more effective as features.

Xia et al. [4] used an ensemble framework for Sentiment Classification which is obtained by combining various feature sets and classification techniques. In their work, they used two types of feature sets (Part-of-speech information and Word-relations) and three base classifiers (Naive Bayes, Maximum Entropy and Support Vector Machines) . They applied ensemble approaches like fixed combination, weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

Luo et. al. [5] highlighted the challenges and an efficient technique to mine opinions from Twitter tweets. Spam and wildly varying language makes opinion retrieval within Twitter challenging task.

Analysis of Twitter information has been the focus of many contemporary researches within the domain of sentiment analysis. Many researchers are seeking to combine the sentiment evaluation and text mining as subsequent generation discipline. Level classification is most promising subject in sentiment evaluation document in Sentiment classification. Reference [6] showed that there is a correlation between sentiment measures computed utilizing phrase frequencies in tweets and both client self-assurance polls and political polls. Accordingly, they illustrated that inclination of public towards special entities might be examined through analysis of tweets. Reference [7] measured presidential efficiency over an exact time interval by way of extracting public sentiment from Twitter. For this motive they used the SentiStrength lexicon [8]. As already acknowledged, [9] adopts a suite of sentiment features as well as some non-sentiment facets to procedure and analyze a manually annotated data set of tweets.

Many of the already existing PD systems nevertheless, participate in polarity detection without even defining a target phrase that their SA is directed at. In many-actual world issues, especially within the domain of purchaser products, assessment between a target item and its competitors will have to be handled. Hence, performing a target-oriented SA is important. Nevertheless, a unigram-situated model would become aware of the equal sentiment polarity for each goal of

SA, as well as, its rivals that arise within the identical document. On this paper we reward a novel set of sentiment features to participate in goal-oriented PD on Twitter knowledge. We show that even by utilizing a very small set of elements, the unigram model (which is overall considered as a baseline for sentiment polarity analysis) is outperformed.

Many researchers have developed distinctive methods for sentimental analysis. The researcher Seyed-Ali Bahrainian et al. [10] presented a novel approach to SA of quick informal texts with a primary focus point on Twitter posts referred to as “tweets”. He also compares state-of-the art SA procedure towards a novel hybrid process. The hybrid process utilizes a sentiment lexicon to generate a new set of features to instruct a linear support vector machine classifier. [11] awarded an adaptive sentiment analysis method called S-PLSA+, which no longer most effective can capture the hidden sentiment factors within the stories, but has the talents to be incrementally up-to-date as extra information grow to be on hand. And in addition exhibit how the proposed S-PLSA model can be utilized to sales efficiency prediction utilizing the ARSA model.

Additionally, the process proposed by means of Noriaki Kawamaet al. In [11] the place “the hierarchical technique to sentiment analysis, identifies each an item and its score by means of dividing topics, which is mainly handled as one entity. [12] developed novel sentiment ontology to conduct context-sensitive sentiment evaluation of on-line opinion posts in stock markets. ZHU Nanli et.Al. [13] introduced a survey on the cutting-edge progress in sentiment evaluation, and makes an in-depth introduction of its research and application in industry and Blogsphere. Reference [14] confirmed that there's a correlation between sentiment measures computed utilizing word frequencies in tweets and both patron self-assurance polls and political polls. Thus, they illustrated that inclination of public in the direction of one-of-a-kind entities might be examined by analysis of tweets.

Pak and Paroubek(2010) [15] proposed a model to classify the tweets as objective, positive and negative. They created a twitter corpus by collecting tweets using Twitter API and automatically annotating those tweets using emoticons. Using that corpus, theydeveloped a sentiment classifier based on the multinomial Naive Bayes method that uses features like Ngram and POS-tags. The training set they used was less efficient since it contains only tweets having emoticons.

Parikh and Movassate(2009) [16] implemented two models, a Naive Bayes bigram model and a Maximum Entropy model to classify tweets. They found that the Naive Bayes classifiers worked much better than the Maximum Entropy model.

Go and L.Huang (2009) [17] proposed a solution for sentiment analysis for twitter data by using distant supervision, in which their training data consisted of tweets with emoticons which served as noisy labels. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM). Their feature space consisted of unigrams, bigrams and POS. They concluded that SVM outperformed other models and that unigram were more effective as features.

Barbosa et al.(2010) [18] designed a two phase automatic sentiment analysis method for classifying tweets. They classified tweets as objective or subjective and then in second phase, the subjective tweets were classified as positive or negative. The feature space used included retweets, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS.

Bifet and Frank(2010) [19] used Twitter streaming data provided by Firehouse API , which gave all messages from every user which are publicly available in real-time. They experimented multinomial naive Bayes, stochastic gradient descent, and the Hoeffding tree. They arrived at a conclusion that SGD-based model, when used with an appropriate learning rate was the better than the rest used.

Agarwal et al. (2011) [20] developed a 3-way model for classifying sentiment into positive, negative and neutral classes. They experimented with models such as: unigram model, a feature based model and a tree kernel based model. For tree kernel based model they represented tweets as a tree.The feature based model uses 100 features and the unigram model uses over 10,000 features. They arrived on a conclusion that features which combine prior polarity of words with their parts-of-speech(pos) tags are most important and plays a major rolein the classification task. The tree kernel based model outperformed the other two models.

Davidov et al.,(2010) [21] proposed a approach to utilize Twitter user-defined hastags in tweets as a classification of sentiment type using punctuation, single words, n-grams and patterns as different feature types, which are then combined into a single feature vector for sentiment classification. They made use of K-Nearest Neighbor strategy to assign sentiment labels by constructing a feature vector for each example in the training and test set.

Po-Wei Liang et.al.(2014) [22] used Twitter API to collect twitter data. Their training data falls in three different categories (camera, movie , mobile). The data is labeled as positive, negative and non-opinions. Tweets containing opinions were filtered. Unigram Naive Bayes model was implemented and the Naive Bayes simplifying independence assumption was employed. They also eliminated useless features by using the Mutual Information and Chi square feature extraction method. Finally , the orientation of an tweet is predicted. i.e. positive or negative.

Pablo et. al. [23] presented variations of Naive Bayes classifiers for detecting polarity of English tweets. Two different variants of Naive Bayes classifiers were built namely Baseline (trained to classify tweets as positive, negative and neutral), and Binary (makes use of a polarity lexicon and classifies as positive and negative. Neutral tweets neglected). The features considered by classifiers were Lemmas (nouns, verbs, adjectives and adverbs), Polarity Lexicons, and Multiword from different sources and Valence Shifters.

Turney et al [24] used bag-of-words method for sentiment analysis in which the relationships between words was not at all considered and a document is represented as just a collection of words. To determine the sentiment for the whole document, sentiments of every word was determined, and those values are united with some aggregation functions.

Kamps et al. [25] used the lexical database WordNet to determine the emotional content of a word along different dimensions. They developed a distance metric on WordNet and determined semantic polarity of adjectives.

Xia et al. [26] used an ensemble framework for Sentiment Classification which is obtained by combining various feature sets and classification techniques. In their work, they used two types of feature sets (Part-of-speech information and Word relations) and three base classifiers (Naive Bayes, Maximum Entropy and Support Vector Machines) . They applied ensemble approaches like fixed combination, weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

Luoet. al. [27] highlighted the challenges and an efficient technique to mine opinions from Twitter tweets. Spam and wildly varying language makes opinion retrieval within Twitter challenging task.

4.0 DATA COLLECTION AND PROCESSING

4.1 CODE TO EXTRACT TWEETS

```
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
from subprocess import STDOUT
import time
import json

#Variables that contains the user credentials to access Twitter API (twitter keys hidden)
access_token = ""
access_token_secret = ""
consumer_key = ""
consumer_secret = ""

class Listener(StreamListener):

    def on_data(self,data):
        try:
            print (data)
            saveFile=open('tweetscollect.csv','a')
            saveFile.write(data)
            saveFile.write('\n')
            saveFile.close()
            return True
        except BaseException as e:
            print ('failed ondata,'),str(e)
            time.sleep(5)

    def on_error(self,status):
```

```
        print (status)

if __name__ == '__main__':

    l = Listener ()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    twitterStream = Stream(auth,l)

    #This line filter Twitter Streams to capture data by the keywords:
    twitterStream.filter(track=['Apple’])
```

4.2 RAW DATA

The raw data contains many columns. The list of columns are: created_id, id_str, text, source, in_reply_to_status_id, in_reply_to_status_id_str, in_reply_to_user_id, name, screen name, location, language, retweet status etc. This is illustrated in Figure 2.

["created_	id:929932(	id_str:"92!	text:"RT @	source:"\u	truncated	in_reply_1	in_reply_1	in_reply_1
["created_	id:929932(	id_str:"92!	text:"UK-t	maker of	has raisec	source:"\u	truncated	in_reply_1
["created_	id:929932:	id_str:"92!	text:"RT @	source:"\u	truncated	in_reply_1	in_reply_1	in_reply_1
["created_	id:929932:	id_str:"92!	text:"RT @	source:"\u	truncated	in_reply_1	in_reply_1	in_reply_1
["created_	id:929932:	id_str:"92!	text:"Mac	display_te	113]	source:"\u	truncated	in_reply_1

[Figure 2]

These steps must be taken to our raw data now:

- The tweets must be parsed to ensure retweet status is zero. Thus, the dataset will not have a repetition of text.
- Only the text is necessary, that is why we will only extract the text from all these information about the tweet.

4.3 CODE TO PARSE TWEETS

```
import json

with open('tweets_ML.csv', 'r') as data_file:
    count=0
    #def on_data(self, data):
    for line in data_file:
        try:
            jsonData = json.loads(line)
            createdAt = jsonData['created_at']
            text = jsonData['text']
            lang=jsonData['lang']
            loc=jsonData['user']['location']
            if ('RT @' not in text) and ('Retweeted' not in text):
                saveFile=open('parsed_ML_tweets.csv','a')
                saveFile.write("text: "+ text+'\n')
                saveFile.write('\n')
                saveFile.close()
                #return True
                count=count+1
        except Exception:
            pass
    print (count)
```

4.4 PARSING RESULTS

As shown in Figure 3, only the text of the tweet has been saved to a new comma separated value (csv) file.

I have to say, Apple has by far the best customer care service I have ever received! Apple AppStore
iOS 7 is so fricking smooth and beautiful!! ThanxApple Apple
LOVE U APPLE
Thank you apple, loving my new iPhone 5S!!!!! apple iphone5S pic.twitter.com/XmHJCU4pcb
apple has the best customer service. In and out with a new phone in under 10min!
apple ear pods are AMAZING! Best sound from in-ear headphones I've ever had!
Omg the iPhone 5S is so cool it can read your finger print to unlock your iPhone 5S and to make purchases without a passcode Apple Apple
the iPhone 5c is so beautiful Apple
AttributeOwnership is exactly why apple will always be one! apple marketing marketer business innovation fb
Just checked out the specs on the new iOS 7...wow is all I have to say! I can't wait to get the new update ?? Bravo Apple
I love the new iOS so much!!!!! Thnx apple phillydvibing
Can't wait to get my Iphone5S!!! apple
V2vista Fingerprint scanner: The killer feature of iPhone 5S. This is so bloody brilliant. apple timesnow http://toi.in/W0o-3Z

[Figure 3]

4.5 PROCESSING AND CLEANING DATA

The language and grammar of tweets is informal, and oftentimes the same tweet text can be posted dozens of times through retweets. To process our tweet text, we converted our text to lowercase. We also removed specific words like “RT”. It was necessary to do so because these words did not relate to the sentiment expressed in the tweet. Moreover, removing these parts of tweets helped us better identify duplicate tweets and avoid overfitting.

Tweets that are collected are manually assigned to be positive, neutral and negative. Labeling each tweet based on sentiment was very time-consuming since we had to hand label and cross-check each tweet in both our training and testing datasets to check classifier accuracy. This distinction is important because a tweet can express joy (“I am so happy that Apple is not that popular anymore!!”), but be classified as negative because of its opinion against Apple. This is the main reason behind classifying the tweets manually.

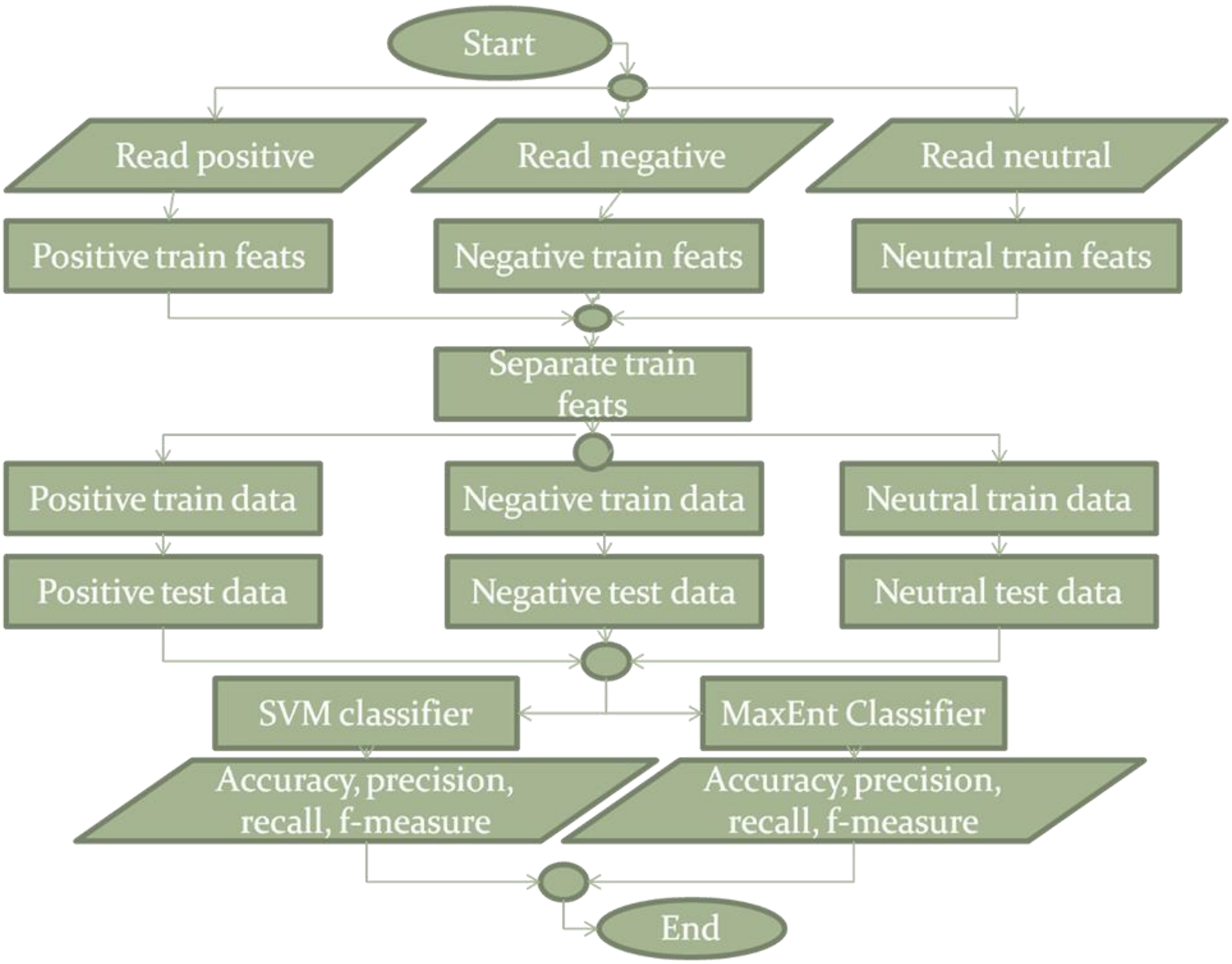
The positive tweets, negative tweets and neutral tweets are then placed in separate csv files for our experimentation.

5.0 EXPERIMENT AND EXPERIMENTAL RESULTS

5.1 METHODOLOGY

In python, there are special classifiers in nltk corpus for sentiment analysis. In the nltk classify corpus there are different classifier defined for different models such as NaiveBayesClassifier, MaxentClassifier, SklearnClassifier, DecisionTreeClassifier, etc. In , SklearnClassifier, there are multiple sub-classifiers such as svm with kernel=rbf, svm with kernel=linear, LinearSVC etc.

In this sentiment analysis project, we used Vectorizer approach to classify our data from plain texts to matrices of datasets each containing keywords and their Boolean values “True” or “False”. First, we split all the sentences using splitter tools from the Positive, Negative and Neutral datasets. Then we create training feats and test feats for all 3 datasets. After that, we assigned 75% of each dataset to be train data and 25% of each datasets to be test data. Then we train the classifiers of the spitted texts using nltk MaxentClassifier for classifying Maximum entropy and, SklearnClassifier, Linear SVC for SVM. Once trained, we pass the reference sets (train data) and test sets (test data) to the classifiers to evaluate. Then we extract the scores of the classifiers using nltk metrics corpus to get accuracy, precision, recall, f-measure of each algorithm for all Positive, Negative and Neutral datasets. In the end, we show our scores in bar chart using matplotlib for both models SVM and Maximum entropy classifiers.



[Figure-4: Flowchart for SA extraction]

5.2 SVM AND MAX_ENTROPY CODE

```
import collections
import nltk.classify.util, nltk.metrics
from nltk.classify import MaxentClassifier, SklearnClassifier
import csv
from sklearn.svm import LinearSVC, SVC
import random
from nltk.corpus import stopwords
import itertools
```

```
from nltk.metrics import precision, recall, f_measure
import numpy as np
import matplotlib.pyplot as plt

positive = []
negative = []
neautral = []

acrcy=[]
prcsn=[]
rcall=[]
fmsr=[]

with open('Positive.csv', 'r') as myfile:
    reader = csv.reader(myfile, delimiter=',')
    for val in reader:
        positive.append(val[0])

with open('Neutral.csv', 'r') as myfile:
    reader = csv.reader(myfile, delimiter=',')
    for val in reader:
        neautral.append(val[0])

with open('Negative.csv', 'r') as myfile:
    reader = csv.reader(myfile, delimiter=',')
    for val in reader:
        negative.append(val[0])

def splitter(d):
    d_new = []
    for w in d:
        w_filter = [i.lower() for i in w.split()]
        d_new.append(w_filter)
    return d_new

def feats(w):
    return dict([(w, True) for w in w])

# Calculating Precision, Recall & F-measure
def evaluate_classifier(featx):

    negfeats = [(featx(f), 'negative') for f in splitter(negative)]
    posfeats = [(featx(f), 'positive') for f in splitter(positive)]
    neautralfeats = [(featx(f), 'neautral') for f in splitter(neautral)]
    negcutoff = int(len(negfeats)*3/4)
    poscutoff = int(len(posfeats)*3/4)
    neautcutoff = int(len(neautralfeats)*3/4)

    trainfeats = negfeats[:negcutoff] + posfeats[:poscutoff] + neautralfeats[:neautcutoff]
    testfeats = negfeats[negcutoff:] + posfeats[poscutoff:] + neautralfeats[neautcutoff:]
    # Max Entropy and SVM classifiers
```

```
    classifier_list = ['maxent', 'svm']

    for cl in classifier_list:
        if cl == 'maxent':
            classifierName = 'Maximum Entropy'
            classifier = MaxentClassifier.train(trainfeats, 'GIS', trace=0, encoding=None,
labels=None, gaussian_prior_sigma=0, max_iter = 1)
        elif cl == 'svm':
            classifierName = 'SVM'
            classifier = SklearnClassifier(LinearSVC(), sparse=False)
            classifier.train(trainfeats)

        refsets = collections.defaultdict(set)
        testsets = collections.defaultdict(set)

        for i, (feats, label) in enumerate(testfeats):
                refsets[label].add(i)
                observed = classifier.classify(feats)
                testsets[observed].add(i)

        accuracy = nltk.classify.util.accuracy(classifier, testfeats)

        pos_precision = precision(refsets['positive'], testsets['positive'])
        if pos_precision is None:
            pos_precision = 0.0
        pos_recall = recall(refsets['positive'], testsets['positive'])
        if pos_recall is None:
            pos_recall = 0.0
        pos_fmeasure = f_measure(refsets['positive'], testsets['positive'])
        if pos_fmeasure is None:
            pos_fmeasure = 0.0

        neut_precision = precision(refsets['neautral'], testsets['neautral'])
        if neut_precision is None:
            neut_precision = 0.0
        neut_recall = recall(refsets['neautral'], testsets['neautral'])
        if neut_recall is None:
            neut_recall = 0.0
        neut_fmeasure = f_measure(refsets['neautral'], testsets['neautral'])
        if neut_fmeasure is None:
            neut_fmeasure = 0.0

        neg_precision = precision(refsets['negative'], testsets['negative'])
        if neg_precision is None:
            neg_precision = 0.0
        neg_recall = recall(refsets['negative'], testsets['negative'])
        if neg_recall is None:
            neg_recall = 0.0
        neg_fmeasure = f_measure(refsets['negative'], testsets['negative'])
        if neg_fmeasure is None:
            neg_fmeasure = 0.0
```

```
    print ('\n')
    print (classifierName)
    print ('accuracy:', accuracy)
    acrcy.append(accuracy)
    print ('precision', (pos_precision + neg_precision + neut_precision) / 3)
    prcsn.append((pos_precision + neg_precision + neut_precision) / 3)
    print ('recall', (pos_recall + neg_recall + neut_recall ) / 3)
    rcall.append((pos_recall + neg_recall + neut_recall ) / 3)
    print ('f-measure', (pos_fmeasure + neg_fmeasure + neut_fmeasure ) / 3)
    fmsr.append((pos_fmeasure + neg_fmeasure + neut_fmeasure ) / 3)

evaluate_classifier(feats)

#Plotting:

msvm=(acrcy[1],prcsn[1],rcall[1],fmsr[1])
mmaxent=(acrcy[0],prcsn[0],rcall[0],fmsr[0])

fig, ax = plt.subplots()
index = np.arange(4)
width = 0.35
err_config = {'ecolor':'0.3'}

r1=plt.bar(index,mmaxent,width,alpha=0.4,color='b',error_kw=err_config, label='MaxEntropy')
r2=plt.bar(index+width,msvm,width,alpha=0.4,color='g',error_kw=err_config, label='SVM')
plt.xlabel('Values')
plt.ylabel('Algorithms')
plt.title('MaxEntropy VS SVM')
plt.xticks(index+width/2,('Accuracy','Precision','Recall','F-measure'))
plt.legend()
plt.tight_layout()
plt.show()
```

5.3 EVALUATION METRIC

Accuracy- This is the measurement of the correct readings divided by the total number of readings.

Precision- The proportion of positive examples that are truly positive; in other words, when a model predicts the positive sentiment, how often is it correct?

Recall- The measure of how complete the results are. This is defined as the number of true positives over the total number of positives.

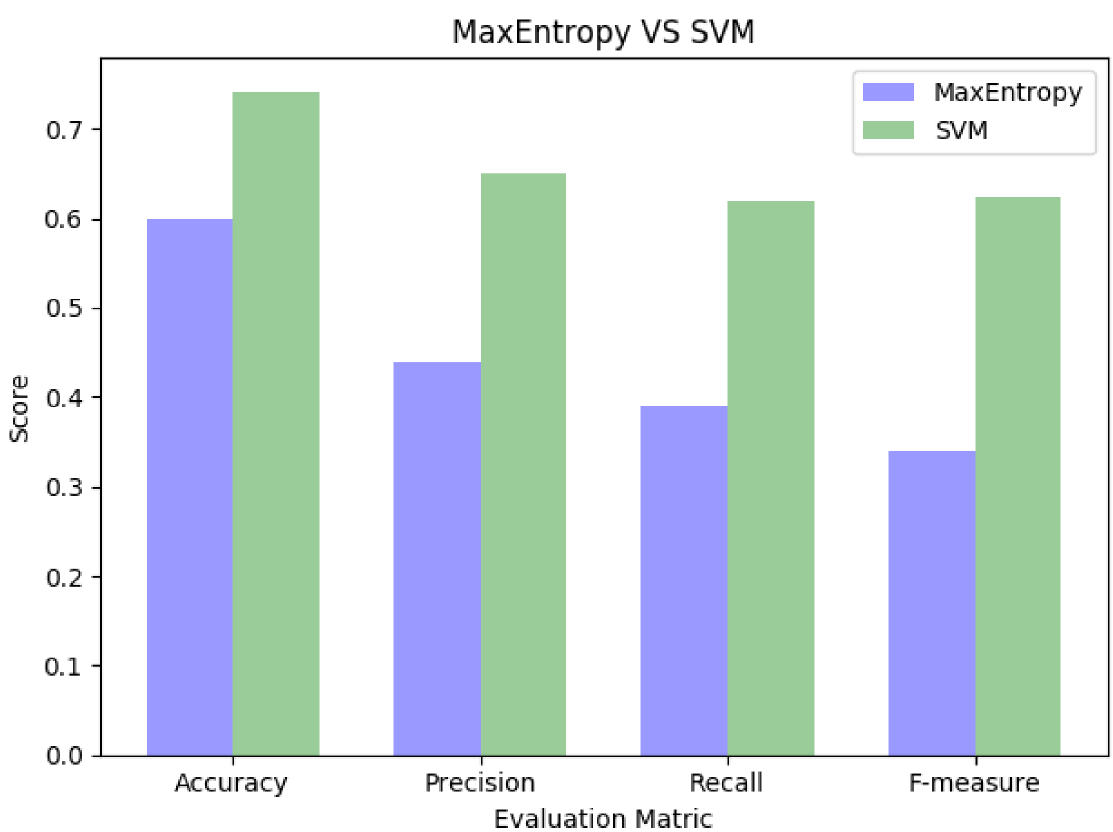
F-Measure- This combines precision and recall using the harmonic mean, a type of average that is used for rates of change. The harmonic mean is used rather than the common arithmetic means since both precision and recall are expressed as proportions between zero and one, which can be interpreted as rates.

5.4 OUTPUT OF EXPERIMENT

```
Maximum Entropy
accuracy: 0.5989010989010989
precision 0.4393053340421762
recall 0.3900709219858156
f-measure 0.33950304259634895


SVM
accuracy: 0.7417582417582418
precision 0.64997964997965
recall 0.6189357984609475
f-measure 0.6238981069455741
```

5.5 VISUALIZATION AND ANALYSIS OF RESULTS



[Figure:5 Visualization of MaxEnt vs SVM]

From both runtime output and Bar chart of the outputs, we can see that SVM has much higher rates of accuracy, precision, recall, f-measurements than Maximum Entropy.

6.0 CONCLUSION AND FUTURE WORK

In our paper, we tested machine learning classifier SVM and Maximum Entropy for Sentiment Analysis of tweets. We conclude that sentiment analysis of tweets gives the best result using SVM classifier. Furthermore, more the cleaner data, more accurate results can be obtained.

Our results improvement is proportional to dataset size and cleanliness. For future work, tweets in different languages can be used as dataset to improve our scope. In SVM, there are 3 subclassifiers with different kernel values, those are kernel=rbf, kernel= linear and LinearSVC. Each of these subclasses returns different results of the SVM. For future work, this 3 kernel subclasses can be merged for better SVM results.

REFERENCES

[1] A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326

[2] R. Parikh and M. Movassate, "Sentiment Analysis of User- Generated Twitter Updates using Various Classi_cation Techniques", CS224N Final Report, 2009

[3] Go, R. Bhayani, L.Huang. "Twitter Sentiment Classification Using Distant Supervision". Stanford University, Technical Paper,2009

[4] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011.

[5] ZhunchenLuo, Miles Osborne, TingWang, An effective approachto tweets opinion retrieval", Springer Journal onWorldWideWeb,Dec 2013, DOI: 10.1007/s11280-013-0268-7.

[6] O'CONNOR, B.; BALASUBRAMANYAN, R.; ROUTLEDGE, B.; SMITH, N.. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. International AAAI Conference on Weblogs and Social Media, North America, may. 2010.

[7] Lai, P., Extracting Strong Sentiment Trends from Twitter, 2011.

[8] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas. A., Sentiment strength detection in short informal text. Journal of the

[9] American Society for Information Science and Technology, pages 2544- 2558, 2010.

[10] S.-A. Bahrainian and A. Dengel, in: 2013 IEEE/WIC/ACM International Joint Conferences on Sentiment Analysis Using Sentiment Features, Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Vol. 3, 17–20 Nov. 2013, pp. 26–29.

[11] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R., Sentiment analysis of Twitter data. In Proceedings of the Workshop on Languages in Social Media (LSM '11). Association for Computational Linguistics, Stroudsburg, PA, USA, 30-38,2011.

[12] Xiaohui Yu,Yang Liu, Aijun An" An Adaptive Model for Probabilistic Sentiment Analysis", IEEE Computer Society ,Volume, Issue No. : 4191-4/10, pp-661-667, November 2010.

[13] Ley, Z., Riddhiman, G., Mohamed, D., Meichun, H., & Bing, L. (2011). "Combining lexicon-based and learningbased methods for twitter sentiment analysis". HP Laboratories, Technical Report HPL-2011, 89.

[14] Liu, B. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

[15] A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326

[16] R. Parikh and M. Movassate, "Sentiment Analysis of User- GeneratedTwitter Updates using Various Classi_cation Techniques",CS224N Final Report, 2009

[17] Go, R. Bhayani, L.Huang. "Twitter Sentiment ClassificationUsing Distant Supervision". Stanford University, Technical Paper,2009

[18] L. Barbosa, J. Feng. "Robust Sentiment Detection on Twitterfrom Biased and Noisy Data". COLING 2010: Poster Volume,pp. 36-44.

[19] Bifet and E. Frank, "Sentiment Knowledge Discovery inTwitter Streaming Data", In Proceedings of the 13th InternationalConference on Discovery Science, Berlin, Germany: Springer,2010, pp. 1-15.

[20] Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, "Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011Workshop on Languages in Social Media,2011 , pp. 30-38

[21] Dmitry Davidov, Ari Rappoport." Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". Coling 2010: Poster Volumepages 241{249, Beijing, August 2010

[22] Po-Wei Liang, Bi-Ru Dai, "Opinion Mining on Social MediaData", IEEE 14th International Conference on Mobile Data Management,Milan, Italy, June 3 - 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5, http://doi.ieeecomputersociety.org/10.1109/MDM.2013.

[23] Pablo Gamallo, Marcos Garcia, "Citius: A Naive-Bayes Strategyfor Sentiment Analysis on English Tweets", 8th InternationalWorkshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland,Aug 23-24 2014, pp 171-175.

[24] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics, pp. 417–424, Association for Computational Linguistics, 2002.

[25] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using wordnet to measure semantic orientations of adjectives," 2004.

[26] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011.

[27] ZhunchenLuo, Miles Osborne, TingWang, An effective approachto tweets opinion retrieval", Springer Journal onWorldWideWeb,Dec 2013, DOI: 10.1007/s11280-013- 0268-7.