# COMPARATIVE ANALYSIS OF ACTIVE LEARNING STRATEGIES IN TWITTER DOMAIN

by

Kousar Aslam

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of
Master of Science

Sabancı University
December 2015

COMPARATIVE ANALYSIS OF ACTIVE LEARNING STRATEGIES IN TWITTER DOMAIN

APPROVED BY:

Prof. Dr. Yücel SAYGIN ..............................................

(Dissertation Supervisor)

Assoc. Prof. Dr. Şule Gündüz Öğüdücü ..............................................

Assoc. Prof. Dr. Hakan Erdoğan ..............................................

DATE OF APPROVAL: 30-12-2015

# Acknowledgements

My sincere gratitude is to my supervisor Yücel Saygın for his guidance, patience, and immense knowledge.

I would like to thank Stefan Rabiger who was always available for his guidance and insightful discussion throughout my thesis work.

I am thankful to my thesis committee members for their comments.

I am very much thankful to my family for their immense support and to my friends Zoya Khalid, Maria Khalid, Asma Almurtadha and Rashid Zaman for their matchless friendship which kept me highly motivated and relaxed all the time during my studies abroad.

COMPARATIVE ANALYSIS OF ACTIVE LEARNING STRATEGIES IN TWITTER DOMAIN

Kousar Aslam

Computer Science and Engineering, Master's Thesis, 2015

Thesis Supervisor: Yücel SAYGIN

**Keywords:** Active learning, Sentiment Analysis, Twitter

## Abstract

Since its launch in the year 2006, Twitter has been one of the most popular social media platforms where users are free to share opinions, ideas and feelings. Latest statistics tell us that nearly 350,000 tweets are being posted every minute on Twitter. Also twitter is the first place to track the response to any important incident or events in the world. For this reason, Twitter has attracted the researchers from many fields, including Sentiment Analysis which deals with opinion mining from text. Twitter data is rich in containing the sentiments but is inherent with the problem of being very informal and unstructured, which makes it very difficult to convert this data into information. Labeling this large amount of data to build classifiers for supervised learning is next to impossible. So we make use of Active Learning which is a sub-field of Machine Learning and concerns with the selection of most informative instances to train the classifiers, thus saving labeling efforts. This thesis deals with the comparative analysis of selected Active learning sampling strategies with twitter domain. The results show Uncertainty Sampling beats Random Sampling and Query by Committee consistently. An analysis of agreement levels among annotators for twitter data has also been presented.

Twitter alanında aktif öğrenme stratejileriin karşılaştırmalı analizi

Kousar Aslam

Computer Science and Engineering, Master's Thesis, 2015

Thesis Supervisor: Yücel SAYGIN

**Keywords:** Active learning, Sentiment Analysis, Twitter

# Özet

Twitter, 2006 yılında kullanıma açıldığından bu yana insanların fikirlerini ve hislerini özgürce paylaşabilecekleri bir ortam olarak en populer alanlardan biri oldu. Son istatistiklere göre dakikada 350.000 Tweet atılmaktadır. Bunun yanında Twitter herhangi bir olaya karşı tepkiyi takip etmek için bakılan ilk yerlerden biridir. Bu bakımdan, Twitter duygu analizi gibi birçok alandan araştırmacıların dikkatini çekmiştir. Gerçekten de Twitter verileri toplumun hissiyatını içermesi açısından önemli ancak düzensiz ve informal yapısından dolayı da çalışılması zor bir ortamdır. Büyük çapta verilerin olduğu bu alanda denetimli öğrenme amacıyla etiketleme yapmak neredeyse imkansızdır. Amacımız, makine öğrenmesinin bir alt dalı olan aktif öğrenme teknikleri kullanarak en fazla bilgi içeren örneklerin etiketlenmesi ve bu yolla etiketleme için gereken eforun azaltılmasıdır. Bu tezle, belirlediğimiz aktif öğrenme stratejilerinin Twitter alanında karşılaştırmalı analizini yapmayı hedefledik. Sonuçlar bize belirsiz örnekleme yöntemlerinin rastgele örnekleme ve komite ile sorgulama yöntemlerinden daha başarılı olduğunu göstermektedir. Bir başka analizde ise etiketleyen kişilerin davranışlarının aktif öğrenmeye etkisini gözlemledik.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Importance & Popularity of Social Media in Data Analysis

The last decade proved to be revolutionary when it comes to the discussion about social media. Although the origin of social media concept i.e. enabling users to share their content and be invoked for social networking traces back to 1997, when the first social media site SixDegrees.com was launched[1]. But this idea did not take that boom as could Facebook and Twitter which appeared in year 2004 and 2006 respectively. With parallel development in the evolution of cell phones getting smart and Internet becoming ubiquitous, the volume of data being put on social media is also increasing rapidly, thus attracting the attention of researchers to convert this data into useful information.

As per Twitter conference[2] held in 2010, Twitter had 106 million users and there is an increase of 300.000 users per day. At the time of this writing, latest statistics tell that Twitter has 320 million monthly active users and every minute, round 350,000 tweets are being posted. This makes Twitter an attractive data source for many research areas like Sentiment Analysis, Data Mining, Machine Learning and Network Analysis etc.

In Machine Learning, we need to learn models to understand and comprehend the data. The learning of these models can be done in supervised or unsupervised manner. Both supervised and unsupervised learning methods intend to perform good on unseen data, which is also called test data. In supervised learning we want to predict the labels of unseen examples or data (which will be referred to as

[1] http://smallbiztrends.com/2013/05/the-complete-history-of-social-media-infographic.html

[2] businessinsider.com/twitter-stats

instances in rest of the document) based on the training of model done with the set of labeled instances. The model thus tries to learn a function from the training data which it can apply on the test data and thus predict the labels by generalizing the information learnt. The training data is in the form of pair of input feature vector and a label. Feature selection along with some other factors like the representation and quality of data instances contribute towards the better performance of model but if the model needs to learn some truly complex function then a lot of labeled data is needed to train the model.

## 1.2 Motivation

The motivation behind this work is to increase the capabilities for understanding and making useful inferences from plentiful twitter data. Twitter data drives the attention because it is huge real time data. In this work, by Twitter data we refer to the tweets which are being posted by the users round the clock on Twitter. The tweets are short texts which can be of maximum 140 characters. The data is very informal because it is just a small text in which a user expresses whatever is going on in his mind. This makes it a difficult task to extract required information but the process is rewarding. The data serves as a potential research material for the fields of Sentiment Analysis, Topic Categorization, Concept drift etc. The public opinion about the physical world is being analyzed through the social media. Also there are few examples which show the strong impact of opinions shared and networking done in social media on the real world. One such instance of ensuring social revolution is the Arab Spring which is believed to be greatly aided by twitter [1].

To build the notion of this work, let us suppose that there is a public relations company interested in knowing how people think about a politician X. Let us say there are 100,000 tweets available for that particular person on Twitter. Now, the public relations analyst can look at each tweet manually - that is quite tedious and infeasible. Making the use of Machine Learning (ML), supervised learning can be applied which also requires to label again each tweet to learn a model that can predict the sentiment for future tweets. So again we face the same problem of cumbersome labeling. What the analyst can do is to sample the 100,000 tweets and label only 1000. But how would he know that which tweets are representative. By chance he can pick the tweets that are actually redundant for the classifier, in other words not helping it to discover learning patterns in the data. To handle this we can leave it up to the classifier to pick the instances to be labeled because the algorithm knows best which instances it needs to be labeled (leaving behind the redundant ones). This can be achieved with Active Learning (AL), a sub-field of ML as it is based on the idea that some labels do not provide further information to classifiers, but rather confuse them. Also the labeling effort is reduced this way.

There are several AL strategies developed for choosing next instances. And it is difficult to decide which one performs better in which particular scenario. The goal of this thesis is to shed light on the question of investigating popular AL strategies in the Twitter domain to predict sentiment of single tweet. We have done a comparison between Random Sampling, Query by Committee (QbC) and Uncertainty Sampling (confidence, entropy and margin have been implemented). Also we have implemented a framework that allows humans to annotate tweets after these are chosen from the dataset with these AL strategies. This is of particular importance because usually AL strategies are only tested under artificial conditions i.e. simulation mode as opposed to real-world scenarios in which we need to acquire labels from humans.

## 1.3 Research Questions

To express precisely the problems we have addressed in this thesis, we will now explicitly state the research questions that we tried to answer in our work.

1. Which AL sampling strategy outperforms when doing Sentiment Analysis with Twitter data.

2. Is the annotation of tweets based on the presence of emoticons reliable enough for performing Sentiment Analysis.

3. What features of tweet text contribute to the agreement on class labels of tweets among human annotators.

## 1.4 Main Outcomes

The thesis contributes the following main findings by providing answers to above stated questions.

1. The results show that Random Sampling performs the worst and Uncertainty Sampling supersedes all in our scenario. QbC also lagged in performing with twitter data. Among the three implemented methods of Uncertainty, Confidence tends to perform good even with less training instances. And Entropy is best when we have good amount of training data available. These are the results specific to our dataset.

2. The comparison of the results obtained after experimenting the emoticoned dataset for Sentiment Analysis reveals that this labeling pattern is not perfect and works moderately for this purpose. Additional information is needed to train a good classifier. Also the dataset does not have a class abel for Neutral or Objective class.

3. The detailed analysis reveals that features like length of tweet, positive and negative words and presence of some special characters like exclamation mark and question mark make tweets easy to be labeled and thus reduce dis-agreement among annotators.

The thesis is organized as follows: Chapter 2 gives information about the background and related work on our research area. In Chapter 3, we have presented literature review related to our topic. In Chapter 4, our implementation approach and experimental design has been discussed. In Chapter 5, we have presented the setup of experiments and analysis of results. In Chapter 6, a detailed qualitative and quantitative analysis of the annotator's behaviour with twitter dataset has been done to give an insight of agreement level among annotators. In Chapter 7, we conclude the thesis and mention about future works.

# Chapter 2

## Preliminaries & Background

In this chapter, we will discuss in detail the background knowledge required for building the foundation for the understanding of experimental work presented in this thesis. Starting with the discussion on different learning methodologies in Computer Science we will explain AL and where it fits into picture. Later selected AL strategies will also be described.

## 2.1 Machine Learning

Machine learning has been discussed considerably in literature. Tom Mitchell defines it in a much formal way in [2] as ”A computer program is said to *learn* from *experience* E with respect to some class of *tasks* T and *performance measure* P, if its performance at tasks in T, as measured by $P$ , improves with experience E”. Building the concept with this definition, several questions arise in mind. Firstly, what can be the task, what experience (information) is needed to solve that particular task and how that experience is used to solve tasks.

In Machine learning there are different tasks and their nature is dependant upon the feedback available to the learning system and the input signals that it takes. The two main learning types are Supervised Learning and Unsupervised Learning.

### 2.1.1 Supervised Learning

In supervised learning the model is trained with the labeled data. The labeled data consists of tuples of input feature vector (values of attributes for a particular instance) and its label. The model infers a function from the analysis of training data. Supervised learning is considered to be a purpose driven learning system. The aim here mainly is to predict the hidden label from known attributes. An example of this learning is suppose we have a bunch of molecules with information given about which molecule is a drug. Now we need to train a model which can predict whether a given new molecule is a drug or not.

### 2.1.2 Unsupervised Learning

The unsupervised learning is also named as "Learning without a teacher" which means the model is given a dataset comprising of instances which do not contain labels [3]. This learning is data driven. Associating the same drug molecules example to Unsupervised learning scenario, consider a case in which we have a set of molecules, part of them are drug ones and rest are not but we do not know whether a molecule belongs to cluster of drug or non-drug. So for Unsupervised learning we will need to develop an algorithm which can differentiate between the two.

## 2.2 Active Learning

Artificial intelligence and Machine learning extend branches to Active Learning as a sub-field which is characterized by a process of continuous and interactive learning. As discussed earlier, there are lots of examples in the real world where the unlabeled data is abundantly available and processing this data can yield useful information. But acquiring the labels for this data is quite expensive in terms of time, labour and expertise. Labeling itself is a cumbersome job but for some datasets a lot of knowledge is needed to assign the labels [4]. So the goal is to achieve higher performance with fewer labeled instances. A few examples include the classification of documents like articles or reviews on websites. If information is to be extracted from the data for specific knowledge domains like labeling genes the annotators require PhD level knowledge. In such a scenario Active Learning paces the process of learning and improves the performance of classifier because in AL the core assumption is that if the algorithm picks only the most informative instances, it will be able to learn faster. In supervised Machine Learning humans decide which labels to use for learning a model, while in AL the algorithm itself decides which labels it needs [4]. Additionally, many labels are inherently noisy, i.e. they could confuse a learner (like SVM, Naive Bayes etc.). When we avoid such noisy labels in training, it helps in reducing the number of instances needed to learn a model of satisfactory performance. [Now you should give 1-2 examples of successful applications of AL from somewhere to show that these assumptions do hold in some scenarios, e.g. in X the autors apply AL to problem Y and need 70

Here it is worth mentioning that AL particularly suits the scenario where many unlabeled instances are available or it is easy to collect these, whereas the labeling cost is high enough to discourage the annotation. There maybe other situations in which only a small number of instances are required to train the model. In such cases the cost of implementing AL framework might be greater than just acquiring the labels.

### 2.2.1 Pool-based vs Stream-based Active learning

There are two different scenarios explained by [4] for AL mechanism, stream-based AL and pool-based AL. In pool-based AL it is assumed that initially there is a small pool of labeled data and a large pool of unlabeled data available. During the process of learning, unlabeled instances are chosen from unlabeled pool which is supposed to be static. Queries are chosen according to the utility measure with which we iterate over all the instances in unlabeled pool. In this way the entire collection is ranked before choosing the best instance. Pool-based scenario is much more common for applied research in AL.

Opposite to Pool-based scenario, in Stream-based AL [5] constantly new unlabeled instances arrive from some streaming data source. The model evaluates the instance and decides for each query individually. This type of Active Learning is particularly useful when we have memory constraints and are unable to store large amount of unlabeled data. Also it is useful when we have less computational power to iterate over whole unlabeled pool to calculate the usefulness of each instance.

### 2.2.2 Exploitation vs Exploration

Exploitation is relying solely on the model for AL and hence is model driven. Exploration on the other hand explores the instance space and is thus data driven. Neither pure exploitation nor complete exploration is useful for AL.

In case of exploitation, what if initially there are insufficient labeled instances and the model is poor. Also as we will see in Uncertainty sampling that as instances having high uncertainty are chosen for labeling, possibility of selection of redundant instances i.e. identical instances and instances belonging to same class is unavoidable. A trade-off is needed to get an optimized solution. Apart from the aforementioned problem, the exploitation approach sticks to the mistaken hypothesis of the location of true decision boundary.

Exploration set asides the model and hunts for diversity in instance selection. In this process it can ignore critical uncertain instances required for specifying the true decision boundary. As model is not taken into consideration, exploration may select areas of the feature space where the model is poor.

## 2.3 Selected Active Learning Strategies

For the experimental setting in this work we have considered the Pool-Based AL scenario. The process is pictorially represented by [4] shown in Figure 2.1

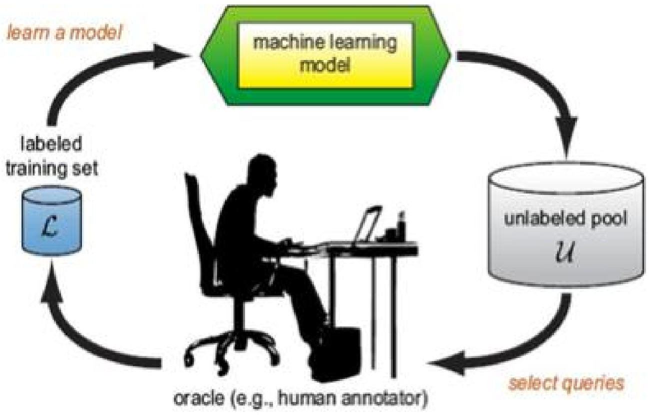


Figure 2.1: Pool-based Active Learning

The process proceeds like this, initially we have a model, an AL strategy picks an unlabeled instance from the pool of unlabeled instances (the most informative instance from the pool), expects an oracle which normally is a human annotator to provide the label for that instance. The instance is now put into the pool of labeled instances. The current model is then updated and the process continues until some defined stopping criterion is reached. How this most informative instance is picked, is dependent on the AL sampling strategy. There are several sampling techniques for choosing useful instances with AL. Some of these are Random Sampling, Uncertainty

Sampling, Query-by-Committee, Expected model change, Expected error reduction and Variance reduction [6]. For this work few of them have been chosen to be tested against each other with the proposed implementation. The main goal is to investigate the performance of these strategies with twitter data.

### 2.3.1 Random Sampling

As the name indicates this sampling strategy just randomly chooses instances from the unlabeled pool. In each run, it picks one or more instances, trains the classifier with it and adds the instances to the labeled pool. This strategy is a pure exploratory approach as the model does not interfere in the selection or rejection of instances. Random Sampling is problematic when dealing with highly unbalanced data among classes. In such cases, the instances from minority class have rare chances to be selected which will surely result in bad models if one is interested in minority class.

Despite its simplicity and problem with skewed class distribution, Random Sampling has been a consistent performer so far. This is said because all other strategies work better in certain scenarios but none has been proven to outperform Random Sampling in every domain. In a challenge [7] on Active Learning, the second most popular sampling strategy happened to be Random Sampling.

### 2.3.2 Uncertainty Sampling

Settles [4] quotes Claude Shannon saying "Information is the resolution of uncertainty". Uncertainty sampling is a pure exploitative strategy which is totally dependant on current model. The basic concept for this learning is that a model can learn fast if it asks only for the labels of instances about which it is least confident. It is a very popular strategy and often used as baseline in comparative studies of sampling strategies [3]. The notion of Uncertainty can be explained differently with different learners. For some of these, like Logistic Regression or support vector machines it can be defined as the distance of an instance from the decision boundary. If the calculated value for confidence comes to be near 0.5 it depicts that the classifier is uncertain about the instance and needs more information or should acquire its label from the oracle. Also it is easy and efficient to be computed so it works well in situations where the unlabeled data is abundant but we need a quick strategy to choose informative instances to be added to labeled pool. Like any other strategy, Uncertainty Sampling cannot also be claimed to work well in all situations. [8] exemplifies that for probabilistic classifiers the strategy can be problematic in regions with high Bayesian error. For learners with decision boundary, if the initial boundary drawn is detrimental then there are chances of sub-concepts to be missed.

Now we will discuss three different measures based on posterior probabilities to calculate uncertainty of given instances.

### Confidence

This is a simple strategy being employed by [9]. The model will query for instances about whose labels it feels most uncertain. Using the formula presented by [4], the instance for which the model is least confident will be calculated as follows

$$x^*_{LC} = \underset{x}{argmax} \quad 1 - P_\theta\left(\hat{y}|x\right) \tag{2.1}$$

This instance has been chosen by the model $\theta$ from unlabeled pool because the model found itself least confident in determining the label for this instance. The model calculated the highest posterior estimate for the class $y$ so $\hat{y} = \underset{y}{argmax} P_\theta\left(y|x\right)$. This measure only considers the highest posterior of instances, so it ignores the rest of useful information about the estimates of other class. To handle this shortcoming there is another measure which uses the notion of margin and will be discussed now.

### Margin

This alternative strategy has been proposed by [10]. In contrast to confidence, where only the lowest posterior estimate is considered, margin considers the difference between the prediction for most likely class and the second most likely class. More is this difference, easier it will be for the classifier to assign the class to the instance. When the distance is very small, the model becomes confused about which class to be exactly assigned to the instance. Expressing this in the formula as done by [4] the chosen instance from this strategy is calculated as :

$$x^*_M = \underset{x}{argmin} \quad P_\theta\left(\hat{y_1}|x\right) - P_\theta\left(\hat{y_2}|x\right) \tag{2.2}$$

As we are talking in terms of probabilities so it fits into the picture for probabilistic classifiers. For others, like logistic regression and support vector machines, margin is defined to be the distance from decision boundary. Though margin takes into account more information than confidence, it still ignores information about rest of posterior distributions. The measure that considers all this information is entropy, whose discussion follows now.

**Entropy**

In this sampling strategy all class probabilities are taken into account [11]. The calculated entropy is considered to be the uncertainty. This is the most general and most common uncertainty measure. Using the formula of [12] the most uncertain instance from the unlabeled pool chosen by this strategy is calculated with the formula:

$$x_H^* = \underset{x}{argmax} \quad -\sum_{i=1}^{Y} P_\theta\left(\hat{y}_i|x\right) logP_\theta\left(\hat{y}_i|x\right) \tag{2.3}$$

In machine learning entropy is meant to be the measure of impurity. There is one another understanding of the concept of entropy, which is the predicted number of bits required to represent the posterior class probability of the model.

### 2.3.3 Query by Committee (QbC)

In this sampling strategy we train a variety of models on currently labeled data. Then the unlabeled pool is iterated and the models vote on the class labels for all these instances. The dis-agreement between the models is then analysed and the instances for which there is high level of dis-agreement between the models is sent to the oracle for labeling. QbC is an effective learning approach and has been applied successfully to different classification problem [13]. It requires less computation than some other sampling strategies like Error-reduction. In our case we have selected Multinomial Naivebayes, Stochastic Gradient Descent (SGD) and Hoeffding tree as the three models to form the committee. Multinomial Naivebayes will be explained later. SGD is a simple but efficient learner. It has been applied to large-scale and sparse machine learning problems often encountered in text classification and Natural Language Processing. The classifier uses gradients of loss functions which are estimated from subsets of training data and updates the parameters in an online fashion. In this way it requires much less training time in practise [14]. Hoeffding tree is a decision tree for streaming data. It is an increamental decesion tree capable to learn from massive data streams. It works with streams on the assumption that distribution generating examples do not change over time. One striking theoretical feature of Hoeffding tree is that it has sound guarentee of performance. It makes use of the fact that a small number of instances can work to make choice for optimally splitting attribute. This is mathematically supported by hoeffding bound, which can be used to show that it can achieve the same performance with few examples as can a non-increamental classifier after the use of a large number of instances.

## 2.4 Bag of Words

An important decision in text classification is the choice for representation of the data. Different representations of documents have been used in the past. The simplest and most widely used is the Bag of words [15]. Bag of words is an orderless document representation used in the fields of Natural Language Processing and Information Retrieval. Each sentence in text is represented by a vector containing the word count for a particular sentence. This vector does not consider grammar or word order but only keeps track of multiplicity [16] The review of literature shows that various attempts have been made to replace or aid bag of words representation with richer features. For instance, a survey [17] represents a work experimenting with the so far available approaches for representation of data with text classification and concludes that adding complex features do not bring more gain when combined with popular classifiers like Support Vector Machines. One possible reason for this result can be that these approaches consider only the relevance of features and not their redundancy.

## 2.5 Multinomial NaiveBayes

Multinomial NaiveBayes is a variant of NaiveBayes model. NaiveBayesian classifier is an easy to build model based on Bayes theorem. NaiveBayes works on a strong assumption that every feature it uses, is independent of every other feature, for a given class. This is called conditional Independence. Although this assumption does not always hold true in real world, NaiveBayes surprisingly shows good classification results. Some theoretical reasons have been proposed for this efficient and effective performance of NaiveBayes by [18].

There are several variations of NaiveBayes classifier like Multinomial NaiveBayes, the Binarized Multinomial Naive Bayes, Bernoulli Naive Bayes etc. Which variation to use depends on the nature of the problem to be solved. Since we used the Bag of words approach, the suitable variant with this representation is Multinomial. This variant calculates the conditional probability of a particular word or token given a class as relative frequency of term $t$ in documents belonging to any one particular class $c$. This is expressed in the Equation 2.4

$$P\left(t|c\right) = \frac{T_{ct}}{\sum_{t'eV} T_{ct'}} \tag{2.4}$$

Here $t$ is the count of a word, $T_{ct}$ is number of occurrences of t in training document from class c. $V$ is the extracted vocabulary from the documents

# Chapter 3

## Related Work

In this chapter we will present the literature review of the work that has been done in the past in Active Learning on Sentiment Analysis with Twitter data. After that we will present our research hypothesis and how this work is different from the existing work. We have structured this review to separate the discussions for Active Learning with Twitter data, Active Learning with Sentiment Analysis and Studies of Comparative analysis of AL strategies.

### 3.1 Active Learning with Twitter Data

In [19], authors have worked on the classification of tweets to detect latest hype on twitter. They have involved AL for a stream based setting which decides weather the tweet should be labeled or not as it arrives the system. They have used the Entropy method of Uncertainty Sampling to choose next instances. In [20], authors included network information for classification of tweets and AL has been introduced to deal with the lot of unlabeled data available. The inference made based on the results obtained for Random Sampling and Uncertainty Sampling is in accordance with ours.

### 3.2 Active Learning for Sentiment Analysis

AL has been applied in the past to address the problem of sentiment analysis. [21] presents the work on sentiment analysis with twitter data but they have experimented with stream-based based setting and the tweets are specific to financial domain. Their results advocate that using Active Learning the classification power of a sentiment analysis classifier is enhanced. Also in [22], the authors have involved crowd sourcing to deal with the sentiment analysis in twitter domain but again it concerns with tracking economic sentiment in online media.

## 3.3 Comparative Analysis of AL strategies

[8] has presented a comparative analysis of selected AL strategies but the experiments have been performed on nominal UCI datasets and text classification has not been considered. [7] presents a challenge for a comparison between AL strategies on different datasets including handwriting and speech recognition, document classification, vision tasks, drug design using recombinant molecules and protein engineering. Here also no one strategy can be considered to outperform all the time, but specific to each dataset conclusion can be drawn. The most popular strategy turned out to be Uncertainty Sampling and next to it was Random Sampling. Both represent the extremes of AL, Uncertainty Sampling being purely exploitative and Random Sampling being purely explorative.

## 3.4 Our Work

From this literature review, it can be seen that a considerable work has been done in AL with Twitter data, AL with Sentiment Analysis and also some comparative studies. Our work distinguishes from the previous work because we are presenting a single study to comprehend all these tasks i.e. we present how well different AL strategies perform when the task is Sentiment Analysis with Twitter data.

# Chapter 4

## Experimental Design and Implementation

In this chapter, we will describe the functionality and implementation design of our system. In the past several studies have been done which involve building programs and frameworks for comparative analysis of different Active Learning Strategies. [8] recently designed a framework for comparing some active learning strategies. Utilizing this implementation as foundation of our work, we extended the framework to fit to the Model-View-Controller (MVC) architecture. But before proceeding to the discussion on MVC, we will describe in detail the already available implementation. The implementation presented by [8] is a standalone application which has been built with the aim of reproducible AL experiments. The experiments have been designed for pool-based setting but to have the validation of results cross-validation has been applied. In the beginning the dataset is randomized, divided into parts for cross-validation and each of these parts is once considered as a test set. All the other parts comprise the training test then. Initially we need a ground truth, which is also called seed set (a set of labeled instances) from the training set to train the classifier. Absence of ground truth is known as cold-start problem. There are some example cases where Active Learning was started without training with initial ground truth. After taking out this subset of labeled instances, rest of the instances are referred to as pool of unlabeled instances and the Active Learning sampling strategy is supposed to pick out most informative instances from this pool. It can also be the case to use two different classifiers in the process i.e. one for labeling and one for training the classifier and predicting on labeled data. In this way experiments can be conducted with regard to label re-usability.

This framework uses some already developed components which have been used extensively in Data mining application. Two of the important ones are WEKA [23], which is a suite of machine learning algorithms and secondly Massive Online Analysis (MOA) [24], which is specific to the algorithms for stream mining. Both WEKA and MOA have been developed using JAVA. The reason for building the foundation of our implementation on this work also lies in the choice of these frameworks because firstly, for AL where the classifier is retrained in every iteration (in our case on one instance) so the choice of MOA is perfect because it includes incremental classifiers which can be updated with single instance. Almost all other data mining tools deal in static mode and retrain the classifier from scratch. Secondly MOA and WEKA work with similar data structures so it is convenient to prepare and pre-process data with WEKA and feed into MOA. We extended our implementation in play framework which is used to build applications with JAVA and Scala. So it was suitable to integrate the existing work with ours using play framework, as both use JAVA.

The process proceeds like this: In each iteration, the sampling strategy which is coupled with the learner, picks up an instance from the unlabeled pool, acquires the label, trains the model with this labeled instance and records the performance of the learner by testing it on test set in terms of some performance measure like AUC, Accuracy or mis-classification loss. This process continues until the budget exhausts. Budget is defined here in terms of number of instances that a strategy can buy to train the classifier. With MOA any classifier can be used as the implementation is based on an interface *ActiveSamplingClassifier* which makes sure the implementation of all MOA functions. It also forces the implementation of *getGain(Instance x)* and another function *update(Instance x)*. The *getGain(Instance x)* function gives the score which determines how much an AL strategy favours any instance. The *update(Instance x)* function retrains the MOA classifier. The simplest in implementation is Random Sampling in which the *getGain(Instance x)* function of the class *ActiveRandomClassifier* just returned a random number and then the *update(Instance x)* function simply trains internal MOA classifier with chosen instance. For other strategies the required logic has been implemented in their respective classes to select the next instance.

## 4.1 Our Implementation Design

One motive of this implementation is to provide the user with a tool where they can input their data file in a particular format and test it with different AL strategies. To achieve this we have provided a graphical user interface to ease the user with these operations. There are two modes of operation available in this tool: Simulation mode and Annotation mode. In Simulation mode the Active learning Strategy chooses the instance and discovers its label from the dataset i.e. we are considering a scenario where dataset already has the labels and it will be unhide for the chosen instance which will then be used to retrain the classifier. The other mode is called Annotation mode which is more near to the real world scenario for the sake that we often do not have labels for the large amount of data that is available. Here we initially train the classifier with the ground truth, and the strategy chooses the next instance to be labeled but this instance is shown to the annotator on the interface to let an oracle provide the label. Unfortunately, we could not experiment much with the Annotation mode due to the lack of availability of people for labeling but it will serve as a source of many potential future works.

We have used the Model-View-Controller (MVC) paradigm for extending the framework. MVC is a software paradigm for developing web applications. The source code of the web application is separated into three different components namely Model, View and Controller. The conceptual pictorial representation of MVC architecture is shown in the Figure 4.1 below taken from [25]

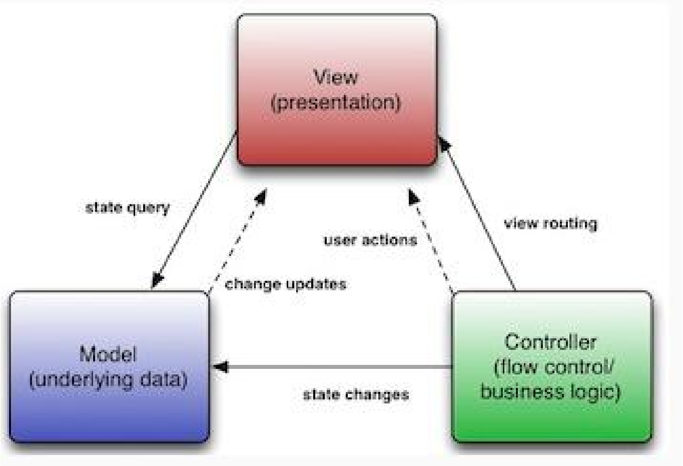


Figure 4.1: Conceptual MVC Approach

Now we will explain the three components of the framework with specification to the parts of our application fitting into each of these.

### 4.1.1 Model

The state of the application is basically handled by the Model as these are the classes that help us encapsulate the relevant data into the modules. As mentioned earlier, the application has two modes of operation, Annotation and Simulation. This component aids the users to perform AL with their datasets in these modes. As the code for different modules is separated, therefore changes in any module will not have any effect on implementation of another and so it is easy to add, remove or modify some functionality of these modules. There is a request router on the top of Web Application layer which receives the HTTP requests from the user and direct them to relevant modules.

In the Model component, we have a class *PoolBasedActiveLearningEvaluation* which simulates the AL strategy. Several functions of this class, when called by the controller in an order, train the classifier with initial seed set, choose the instance from the unlabeled pool and after acquiring its label adds it to labeled pool to retrain the classifier until the budget, which has been stated in terms of maximum number of instances to be bought, exhausts. So the whole process of Active Learning is local to the Model. After this, evaluation is made on the already separated test set with WEKA Evaluation class methods which are invoked by the Model class *CrossValidation*. Here we record the model measurement for different performance metrics to be analysed later.

For the Annotation mode, we needed to apply a different logic to involve both controller and view in the process. This is because we are getting the label from an annotator for an instance which was chosen by the Active Learning strategy. And this instance will retrain the classifier which, coupled with the AL strategy, will choose an instance again until the budget finishes. To accomplish this functionality, communication between Controller, Model and View has been implemented in which View passes the label to Controller which in turn calls the Model to select next instance. This instance is passed to View by Controller and so all the required labels are obtained.

### 4.1.2 View

Views stand on the other side in respect to the Model and are used for the presentation of the data to the user. The View is usually identified as the Graphical User Interface of the application i.e. here the user interacts with the system. In Figure 4.2 we have shown the interface for simulation mode of our tool. Using this the user can upload his dataset to test the implemented AL strategies on it.

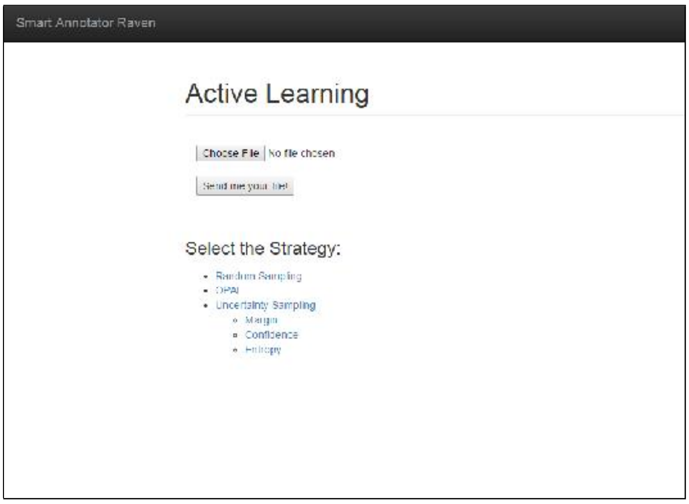
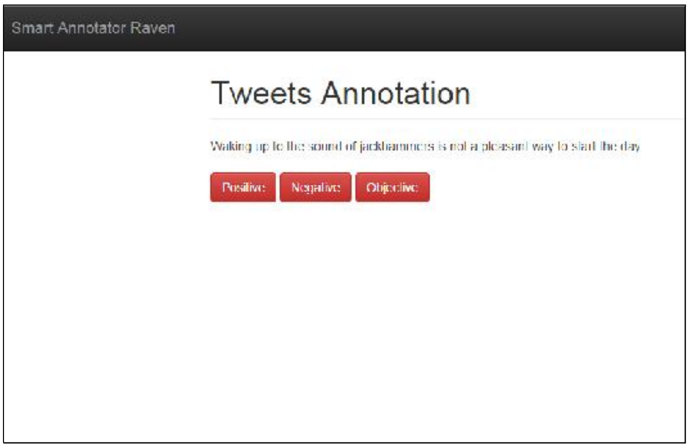


Figure 4.2: Simulation Mode

Figure 4.3 shows the annotation mode of our tool. The AL strategy chooses the most informative instance to be presented to annotator for labeling. Annotator assigns the label to the tweet and thus it is added to the labeled pool to retrain the model. Annotator keeps on receiving the tweets until the budget is exhausted.



Waking up to the sound of jackhammers is not a pleasant way to start the day

Figure 4.3: Annotation Mode

### 4.1.3 Controller

Controller processes user's requests, builds appropriate models and passes it to view to be shown to user. Code in the controller runs on server (back end of the application). This component controls the flow of logic for the application. It is responsible for all the manipulations to be performed on the data objects. The logic for experimenting with the AL strategies for both Simulation and Annotation mode has been placed in this component. In this application we have developed two separate controllers for the two modes.

# Chapter 5

## Experimental Results

In this chapter we will present the analysis of the results obtained after experimenting with different datasets. Here it is necessary to talk about the few assumptions we have made while working for this thesis. These assumptions are common to scientific setting and have also been discussed in [4].

- All labels cost the same.
- The labels that are bought are always correct.
- The classifier type does not change during the Active Learning process.

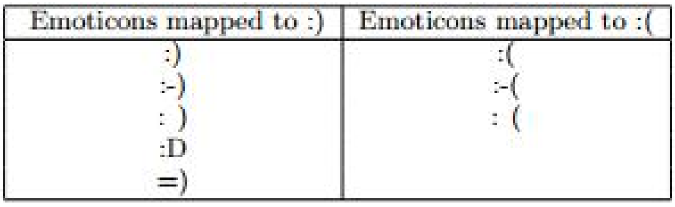
We make and follow these assumptions because it makes the interpretation of results easier. These assumption may not necessarily hold in real world scenario but in a lab setting these make sense. For instance, in the first assumption we state that we are not dealing with the cost that maybe associated with the wrong label of an instance. But it maybe problematic in the scenario where some critical lab test is to be performed for a disease. With less intensity, this also applies to the case when the spam emails are being separated from the real ones. Assigning the spam label to an important email can surely be annoying to the user. Similarly for the second assumption, the labels may not always be correct if the annotators do not possess the required background knowledge of an area necessary for understanding the content to be labeled. Especially in crowd-sourcing, the performance and interest of oracle tends to degrade with the passage of time due to huge amount of data so the correctness of labels cannot be guaranteed. Also we stick to one classifier during the process, but in real world it is common to change classifiers to match with the state of art approaches. But it makes analysis of the results harder in lab practice.

Throughout these experiments we have used Multinomial NaiveBayes classifier and Bag of words representation of dataset.

## 5.1 Dataset

In order to train a classifier, we need a dataset which has been genuinely labeled. There is a large range of topics being discussed on twitter all the time, so it is not easy to manually collect and label the data to train a classifier for sentiment analysis. [26] introduced the approach of using distant supervision, which means the classifier is being trained with tweets which have been labeled on the basis of presence of emoticons. Thus the emoticons are used as noisy labels. The presence of :) emoticon will mark the tweet as Positive and the presence of :( will label it as Negative. In this way a lot of human effort and time is being saved which would be spent in labeling otherwise. In this work we have used a dataset presented in [27] which has been prepared in the same way.

The authors have used the twitter Application Programming Interface (API) [28] for collecting the tweets. The parameters have been set to collect tweets only in English language. The tweets collected are from the time period between April 6, 2009 to June 25, 2009. Several emoticons can refer to Positive and Negative sentiments separately. For example :) and :-) both refer to Positive polarity. A complete list of emoticons queried to twitter for collection of tweets is shown in the Figure 5.1.



| Emoticons mapped to :) | Emoticons mapped to :( |
|---|---|
| :) | :( |
| :-) | :-( |
| : ) | : ( |
| :D | |
| =) | |

Figure 5.1: List of Emoticons

After collecting the data, it has been processed by the authors to remove junk or unwanted content. So first of all emoticons were stripped off. Also the tweets containing both positive and negative emoticons were removed. There is a trend on twitter of re-tweeting the tweets someone agrees to, i.e. copying someone's tweet and posting it to another account. These tweets usually start with RT. All these were removed so that each tweet has the same weight in the training data. Also the tweets containing ambiguous emoticons like :P are removed because it may not always mean a negative sentiment.

The dataset available publicly after this whole processing has first 80,000 tweets with Positive emoticons and next 80,000 tweets with Negative emoticon thus making 160,000 tweets in total. We took out randomly 2000 positively labeled and 2000 negatively labeled tweets from this whole dataset to make our dataset.

We pre-processed the data with python scripts to remove usernames, dates, digits, punctuation marks and hyperlinks. Using WEKA functions stemming was performed and stop words were also removed.

The experimental results have been presented for four varying datasets. To test a particular strategy, we have first separated 20% of the total dataset as Test set and the remaining is considered as Training set. After that we split the Training set into ground truth, which is also called seed set, and Unlabeled pool. Increasing the value of Seed set from 1% to 5% with increment of 1% each time, we conducted 5 separate experiments on each dataset with all strategies. For each experiment, to exclude the effect of randomness in separating the Test set, Training set and Seed set, we randomized the whole dataset with 10 different random seed values. Also after the whole allocated budget is consumed and all the labeled instances have been added to Training set we perform Cross-validation with WEKA Evaluation class. Cross-validation has been done to validate the performance because we wanted to make sure that we get the closely correlated results for cross-validation and test set.

Now we will evaluate the performance of selected Active Learning strategies on varying datasets on the basis of these assumptions and will draw conclusions about which strategy outperforms and in which particular scenario.

## 5.2 Emoticon-ed labeled Dataset

From now on we will use therm Emoticon-ed to refer the dataset which has been labeled based on the presence of emoticon. This dataset, as already discussed, is a class balanced dataset with 2000 Positive and 2000 Negative instances making a total of 4000 instances. After taking out 20% of these which makes 800 instances, we are left with 3200 instances. These 3200 instances comprise the Training set. There are no Objective or Neutral tweets in this dataset because it has been collected and labeled on the basis of presence of emoticon, but the absence of 'Objective' class label is surely a limitation because after the emoticon is stripped off, for many tweets the text alone is insufficient to be assigned some label due to its vagueness. For five different experiments shown in the figure, the distribution of instances among seed set and training set each time seed set is increased can be seen in Table 5.1

| Seed set% | Seed set | Unlabeled Pool |
|---|---|---|
| 1% | 32 | 3168 |
| 2% | 64 | 3136 |
| 3% | 96 | 3104 |
| 4% | 128 | 3072 |
| 5% | 160 | 3040 |

Table 5.1: Instance distribution for Emoticon-ed dataset

Fig 4.2- Fig 4.6 show the learning curves for Accuracy, and Fig 4.7 - Fig 4-11 show the F-measure respectively. The F-measure is the harmonic mean of precision and recall. We take the harmonic mean because both precision and recall are expressed as ratios. F-measure is thus a measure of correctness of results because it considers the ratios of false positives and false negatives.

In the graphs shown, it is a common observation that with the increase in seed set size and queried instances, the performance of the classifier improves because we are increasing the size of training set. Comparing the strategies, Random Sampling under-performs in general. But it can also be seen that during the coarse of learning, it ties and at few points slightly beats other strategies like Uncertainty Sampling (Entropy and Margin specifically). As we are dealing with the two class problem here, all the three methods of Uncertainty sampling are monotonic function of each other. For binary problem they all are symmetric and as a result these all reduce to choosing the instance that is closest to decision boundary. For the same reason we do not see difference among the results of metrics of Uncertainty sampling. The slight difference is due to the randomization of data which we actually did to exclude the effect of lucky distribution of instances among different sets.

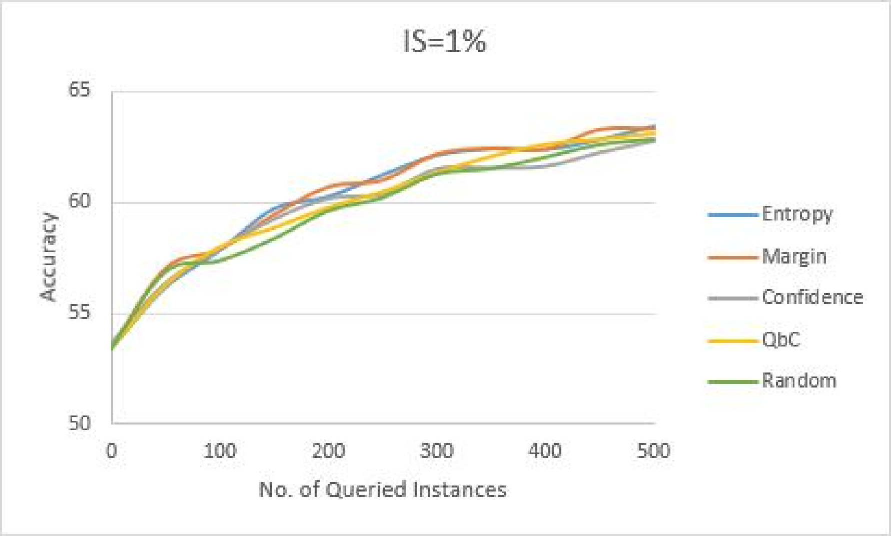


Figure 5.2: Accuracy for 1% seed set using Emoticon-ed data

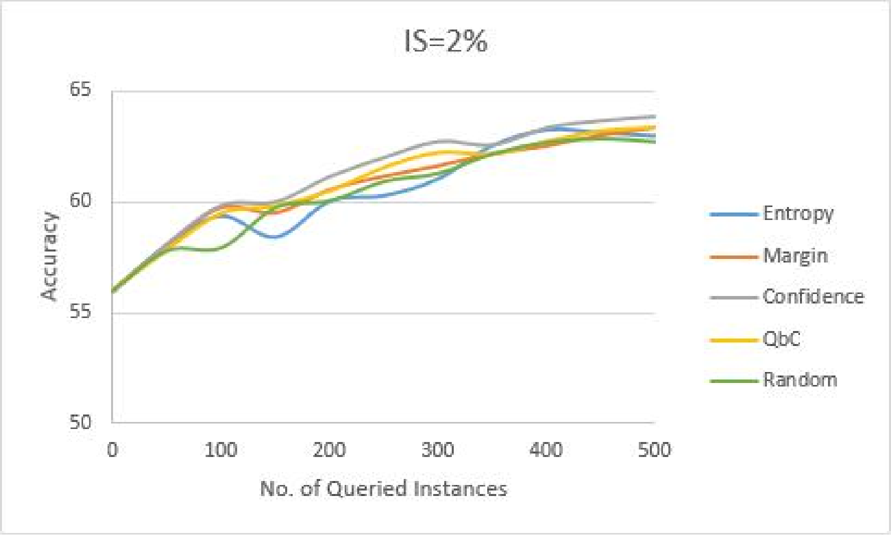


Figure 5.3: Accuracy for 2% seed set using Emoticon-ed data

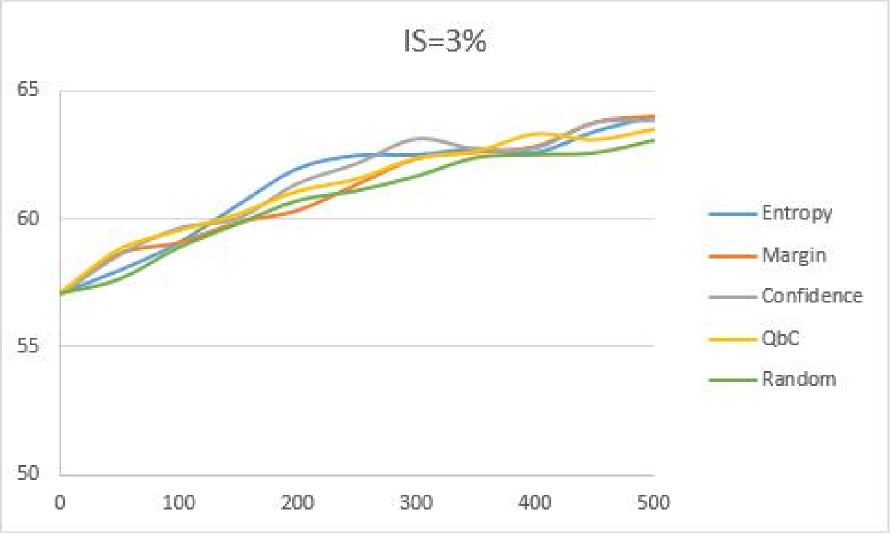


Figure 5.4: Accuracy for 3% seed set using Emoticon-ed data

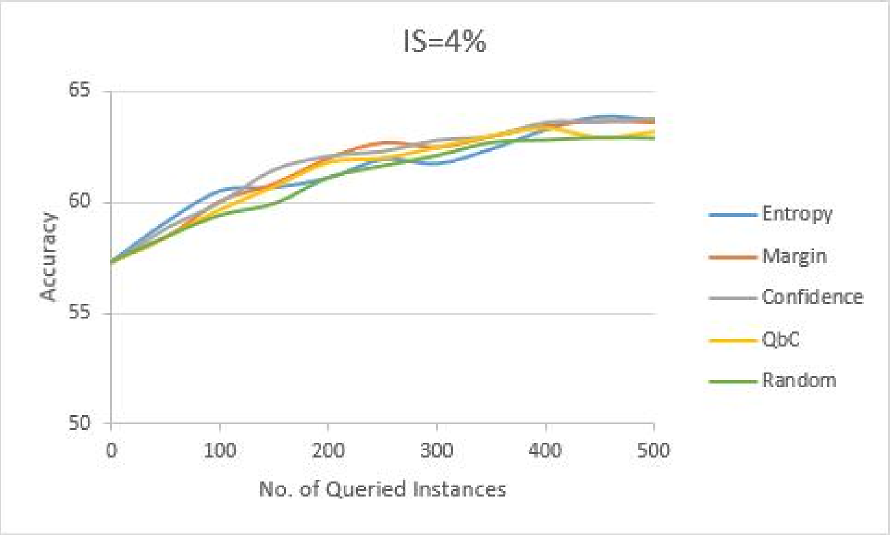


Figure 5.5: Accuracy for 4% seed set using Emoticon-ed data

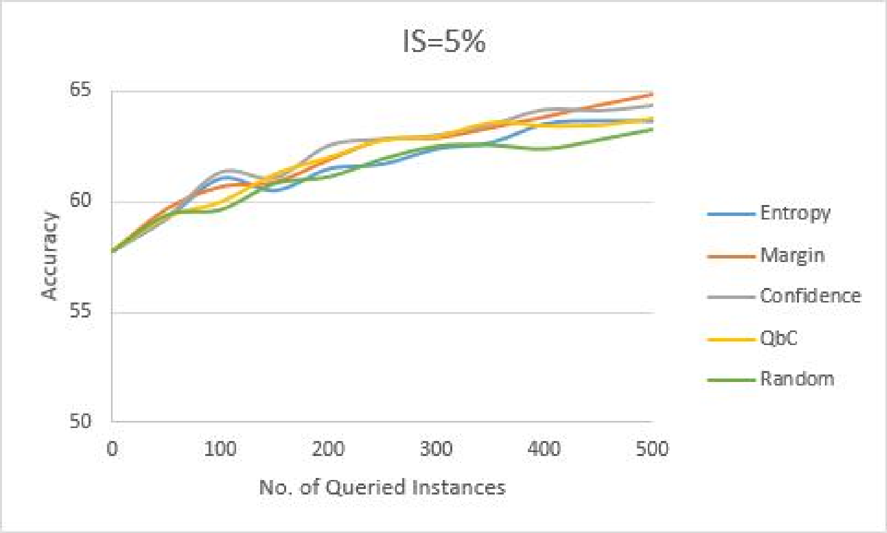


Figure 5.6: Accuracy for 5% seed set using Emoticon-ed data

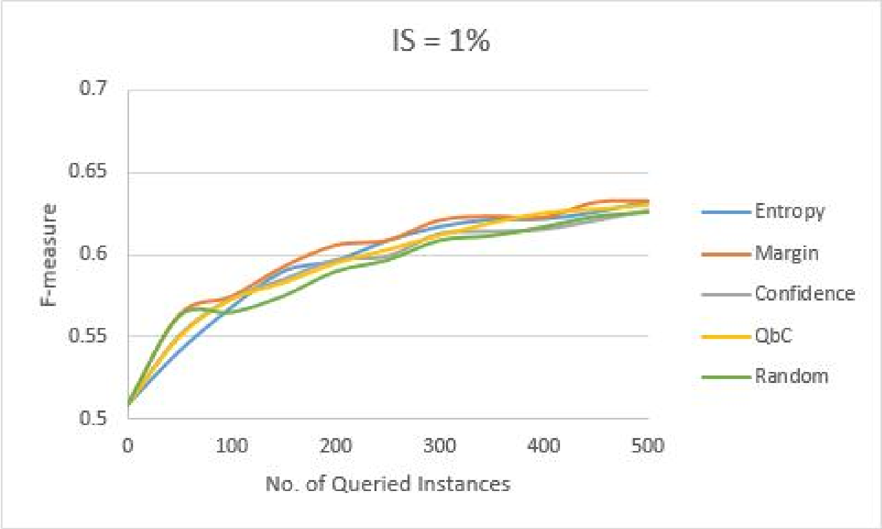


Figure 5.7: F-measure for 1% seed set using Emoticon-ed data

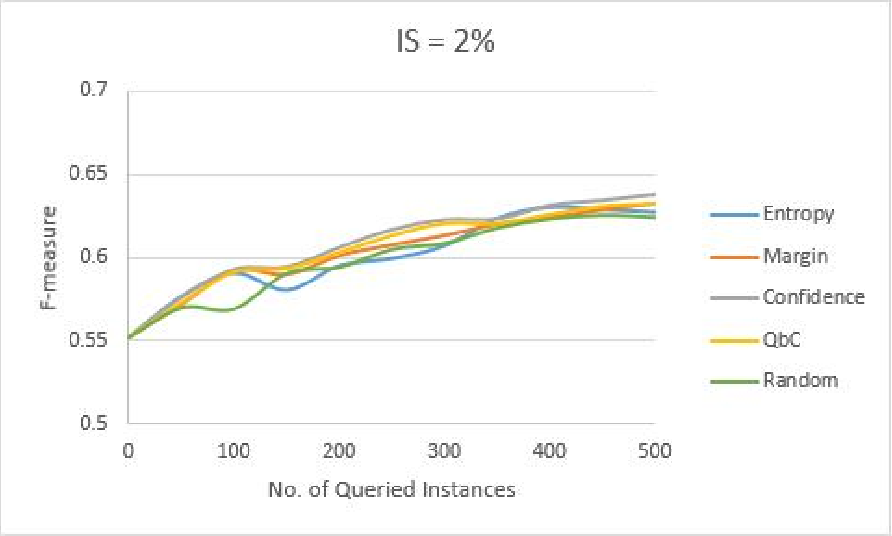


Figure 5.8: F-measure for 2% seed set using Emoticon-ed data

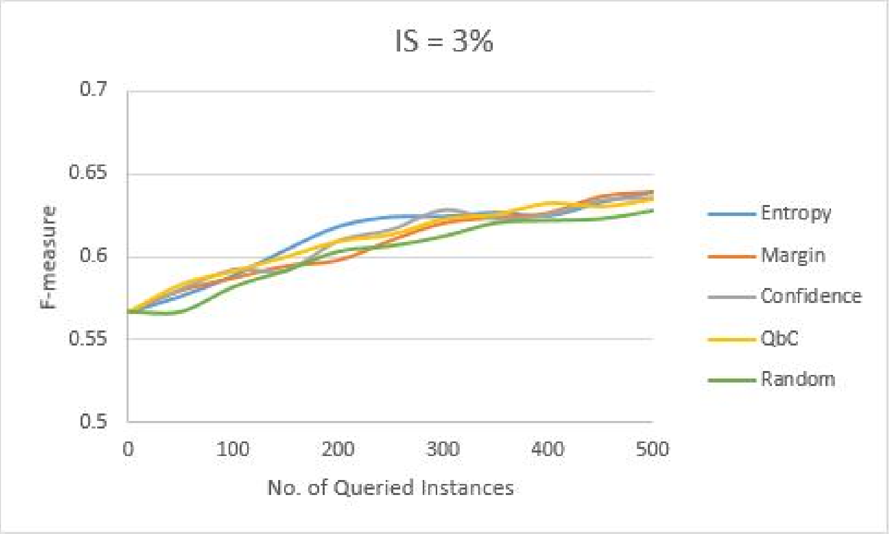


Figure 5.9: F-measure for 3% seed set using Emoticon-ed data

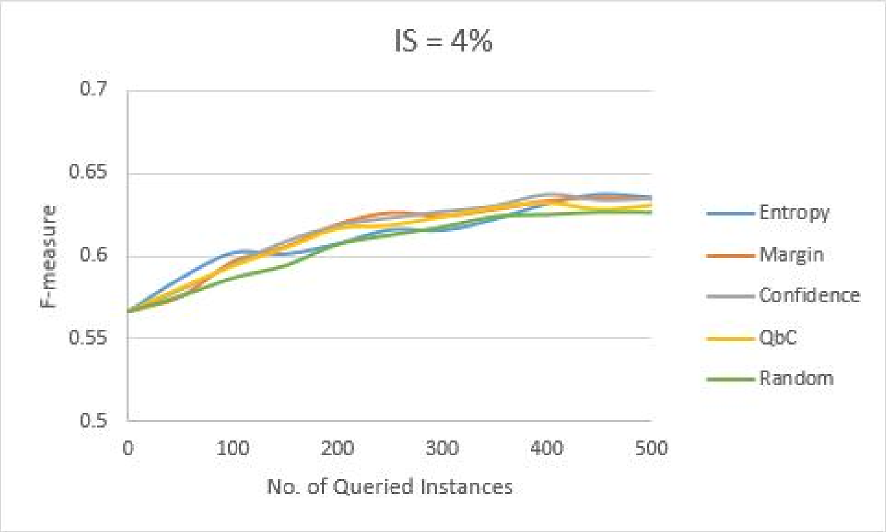


Figure 5.10: F-measure for 4% seed set using Emoticon-ed data

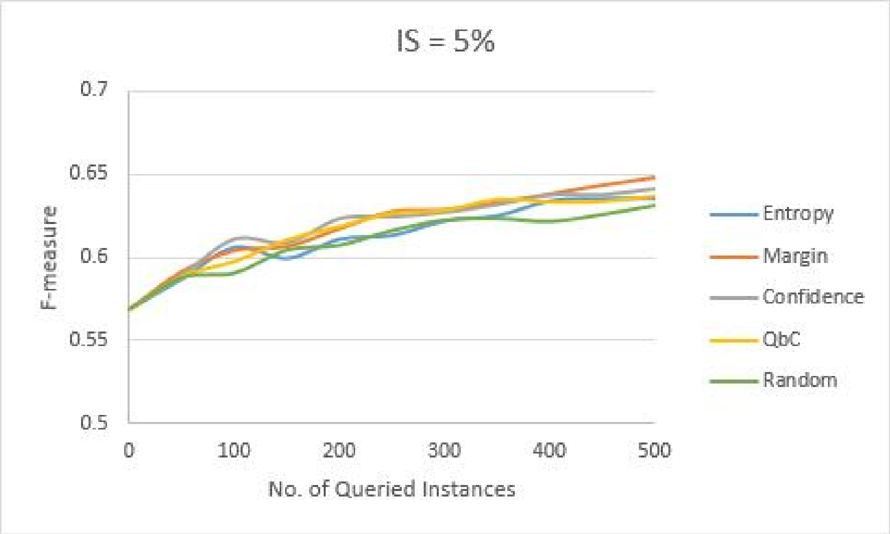


Figure 5.11: F-measure for 5% seed set using Emoticon-ed data

## 5.3 Comparison of Strategies on Manually Labeled data

The labeling done on the basis of presence of emoticon is noisy meaning that these are not genuine labels and are not always consistent with the actual polarity of the tweet text. Even many times the tweet text does not contain any sentiment but due to the presence of emoticon it is assigned a label. Also, this tweet collection is based on the query of presence of emoticon so we do not have any tweet labeled as Neutral in the dataset. This is a problematic assumption in the real world scenario. The manual inspection of tweets also revealed that there are many tweets which do not contain any sentiment. Now the question is, how the AL strategies will perform if we get this data labeled manually which will turn out to be a multi-class problem. For handling this case, we got these tweets manually labeled so that we can also include the 'Objective' class. We had ten annotators available for this task. All of the annotators were Machine Learning students which implies that all are well aware of the twitter data and its use in research of data analysis. The dataset was distributed among the annotators such that any three annotators receive the same tweet so that each tweet gets labeled thrice. Finally the labels from all annotators were collected and compiled on the basis of majority voting i.e. we assigned the most occurring label as the final label. There were tweets which received different label from all the three annotators. For this particular experiment, we presented those tweets to a fourth annotator to receive another label. This way we were able to remove the tie among labels.Out of 4000 total instances, 1791 belong to Positive class, 1354 belonging to Objective class and 855 belong to Negative class.

For five different experiments done with different seed sizes, the distribution of instances between seed sets and training sets is same as for Emoticon-ed dataset because the total number of instances in both datasets is the same. This time we have three classes and unequal instance distribution among classes, so we will be using weighted F-measure which is weighted average of F-measure of classes, weighted by proportion of number of elements in each class.

Having a look at the results for Accuracy and F-measure, the first observation that we make is that we are getting low values for accuracy and F-measure as compared to the previous experiment but we are having 3-class problem here so the results are more or less consistent.Among the strategies, Uncertainty Sampling supersedes the rest from the beginning till the end. Random Sampling under-performs all strategies in general but in the beginning QbC is performing even poor than Random Sampling for smaller seed sets and low budgets. All the strategies show a learning trend with increase in number of queried instances and seed set both of which comprise the training set. Among Uncertainty Sampling, Confidence performs well on less training data as well, but after a considerable learning Entropy and Margin supersedes Confidence.

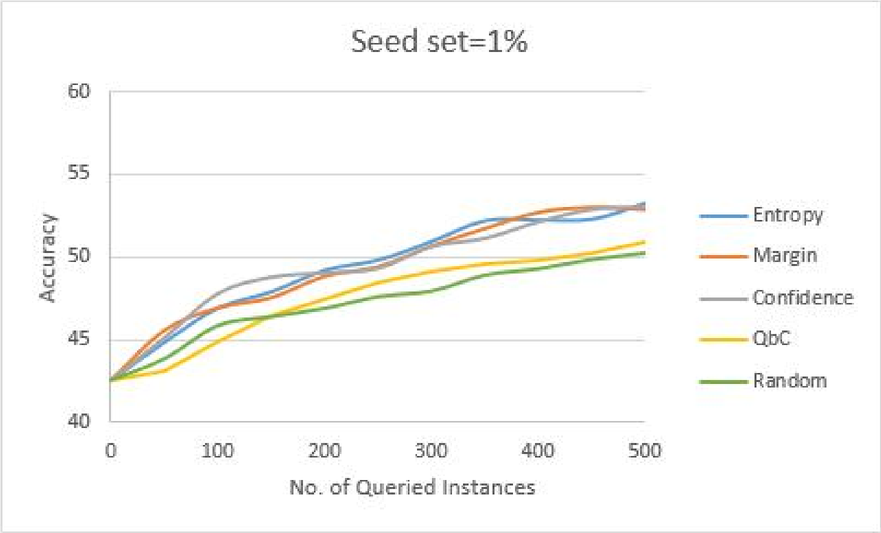


Figure 5.12: Accuracy for 1% seed set using Manually annotated data

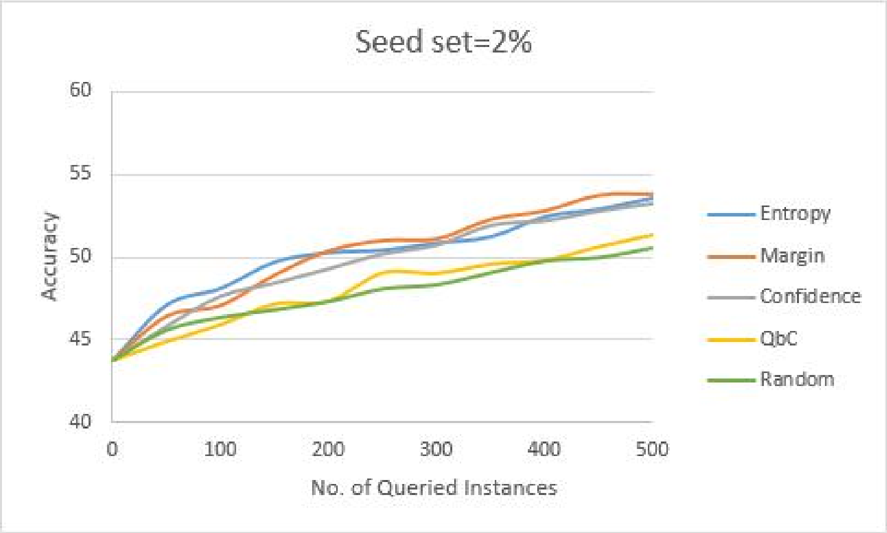


Figure 5.13: Accuracy for 2% seed set using Manually annotated data

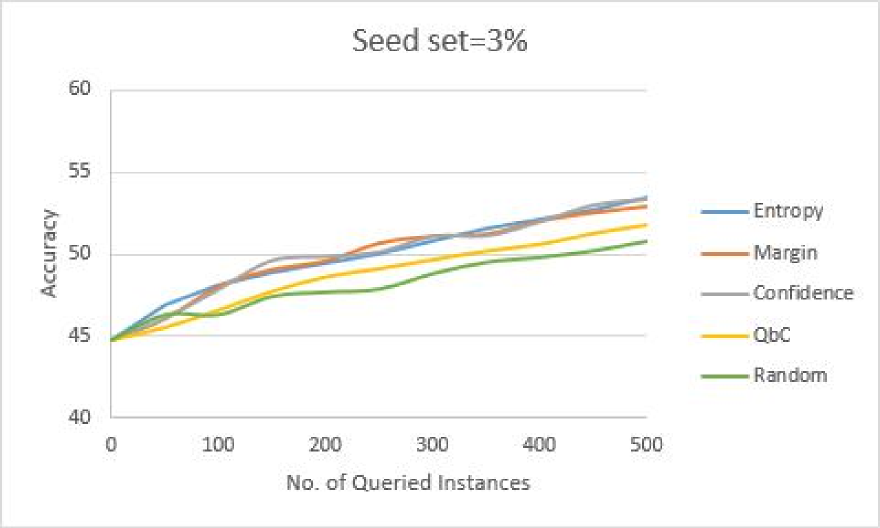


Figure 5.14: Accuracy for 3% seed set using Manually annotated data

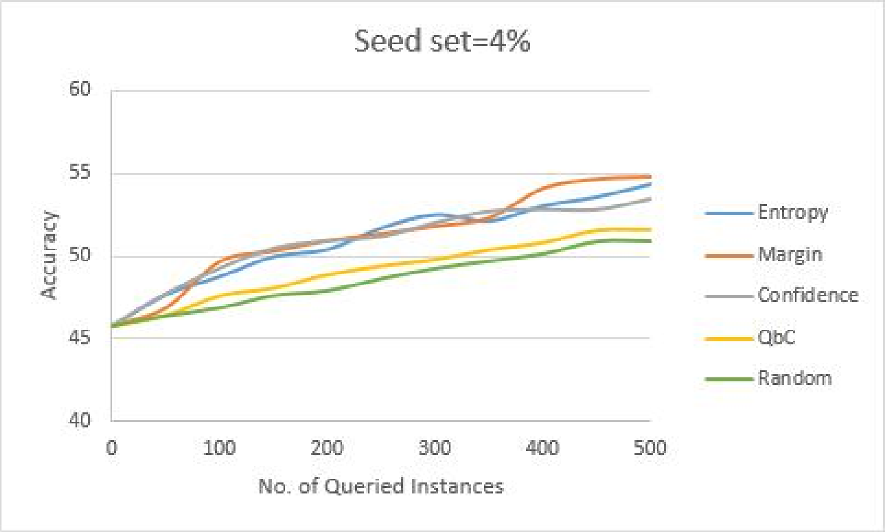


Figure 5.15: Accuracy for 4% seed set using Manually annotated data

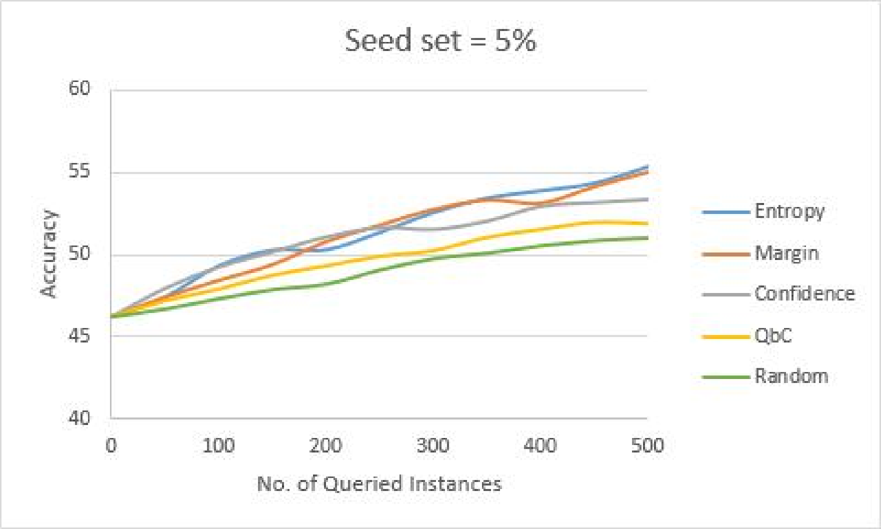


Figure 5.16: Accuracy for 5% seed set using Manually annotated data

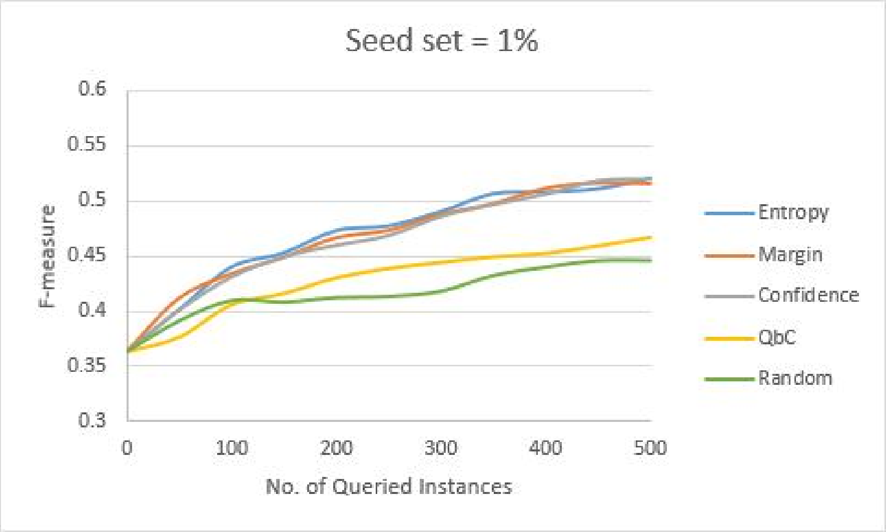


Figure 5.17: F-measure for 1% seed set using Manually annotated data

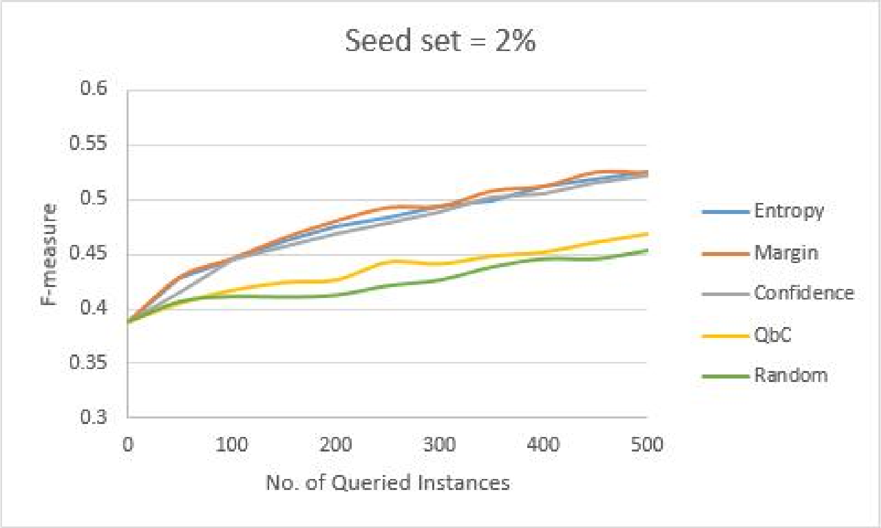


Figure 5.18: F-measure for 2% seed set using Manually annotated data

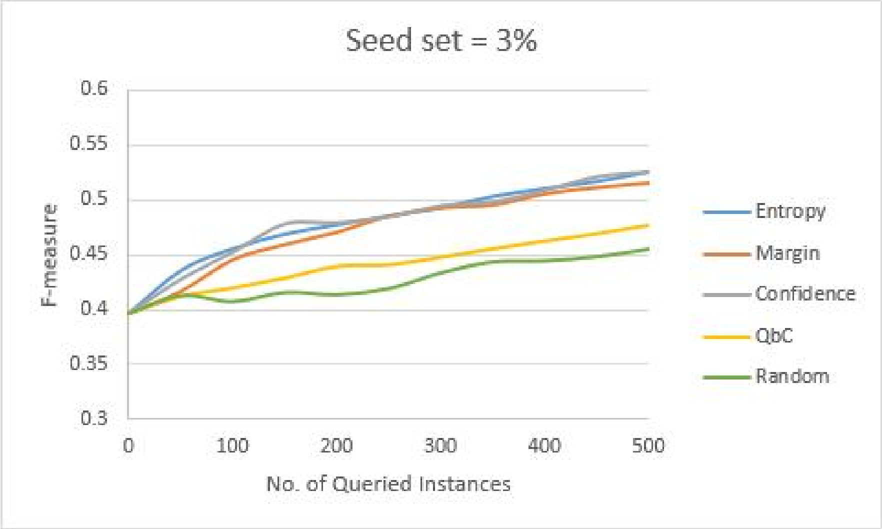


Figure 5.19: F-measure for 3% seed set using Manually annotated data

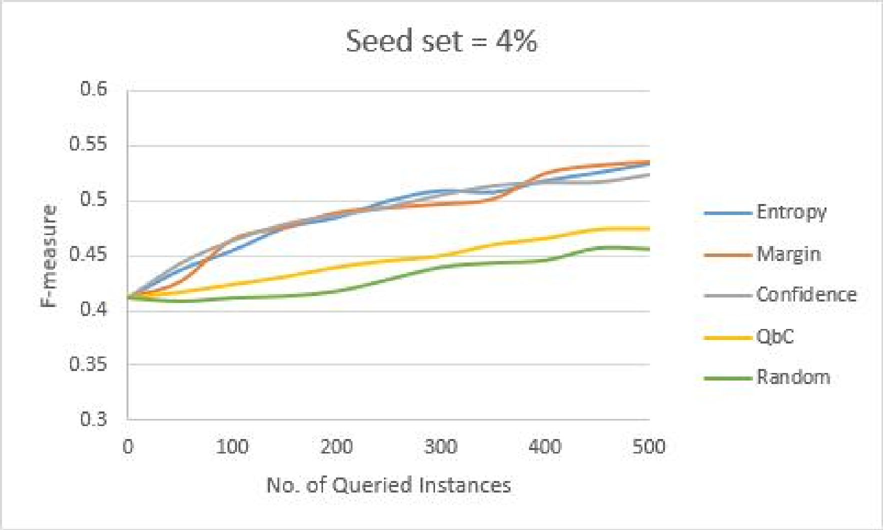


Figure 5.20: F-measure for 4% seed set using Manually annotated data

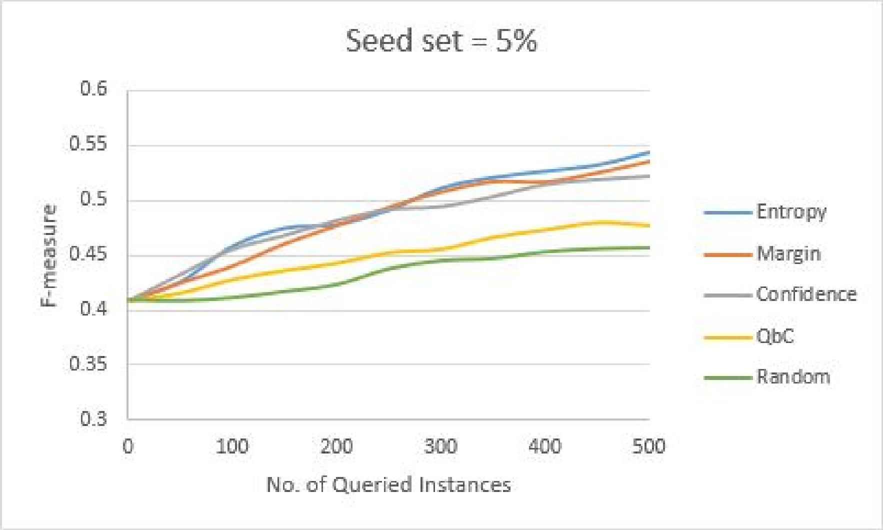


Figure 5.21: F-measure for 5% seed set using Manually annotated data

## 5.4 What if Strong Dis-agreement among Labeling is Removed?

The analysis of the results obtained on the dataset labeled by three annotators reveals that the tweets are too vague to train a classifier for sentiment analysis problem. This claim is made because we could achieve a maximum of 55% accuracy and an F-measure of 0.53 which is not very high. As already mentioned we have tweets in the dataset which received distinct labels from all three annotators. Though we previously managed with those by involving a fourth annotator but it maybe the case that eliminating such ambiguous tweets decrease the confusion occurring in the training of classifier. So in this experiment we want to see whether removing such tweets from the dataset is beneficial in terms of improvement in performance. There are 208 such instances out of total 4000 instances. So we are left with 3792 instances. Out of these there are 1744 Positive instances, 804 Negative instances and 1244 Neutral instances. For this experimental setting, the distribution of instances among unlabeled pool and seed set with the increment in seed set is shown in table.

| Seed set% | Seed set | Unlabeled Pool |
|---|---|---|
| 1% | 30 | 3004 |
| 2% | 61 | 2973 |
| 3% | 91 | 2943 |
| 4% | 121 | 2913 |
| 5% | 152 | 2882 |

Table 5.2: Instance distribution in Dataset without Strong Dis-agreement

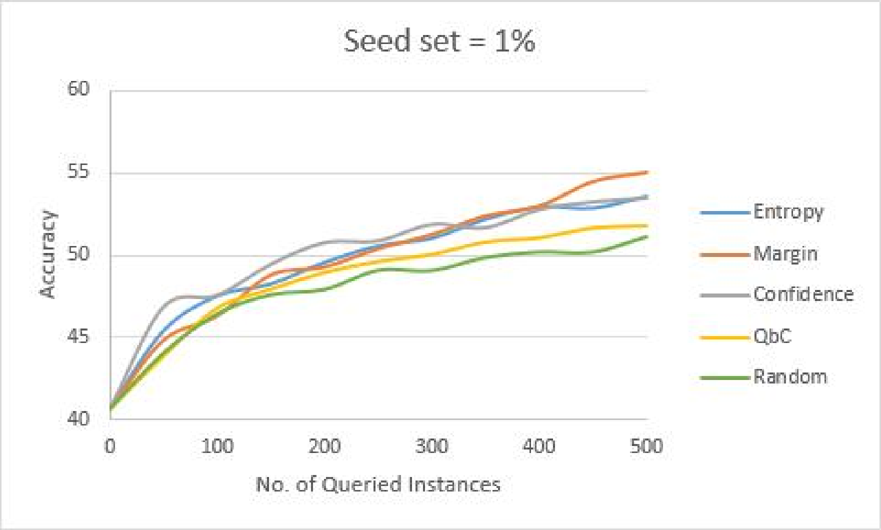


Figure 5.22: Accuracy for 1% seed set after removing strong dis-agreement

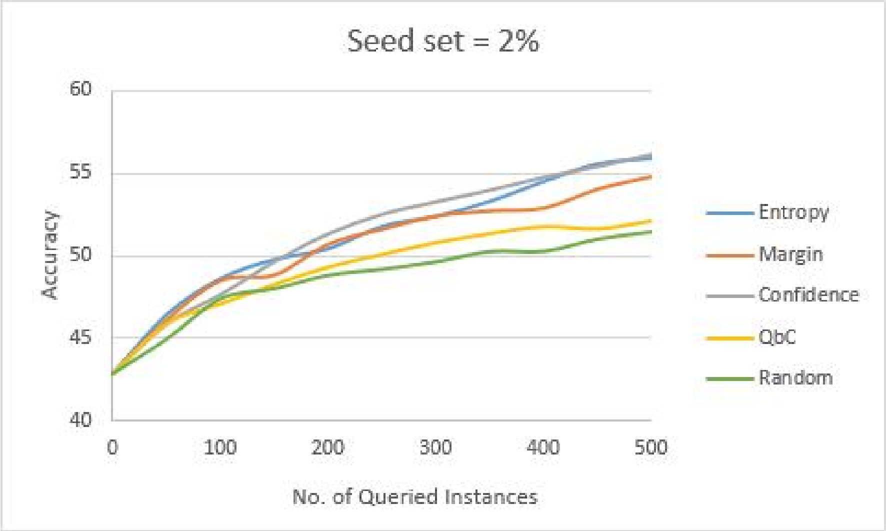


Figure 5.23: Accuracy for 2% seed set after removing strong dis-agreement

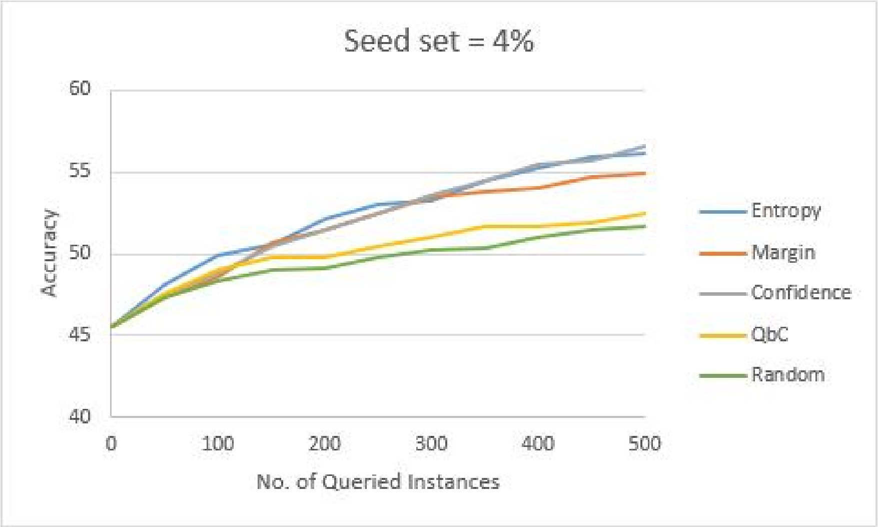


Figure 5.24: Accuracy for 3% seed set after removing strong dis-agreement

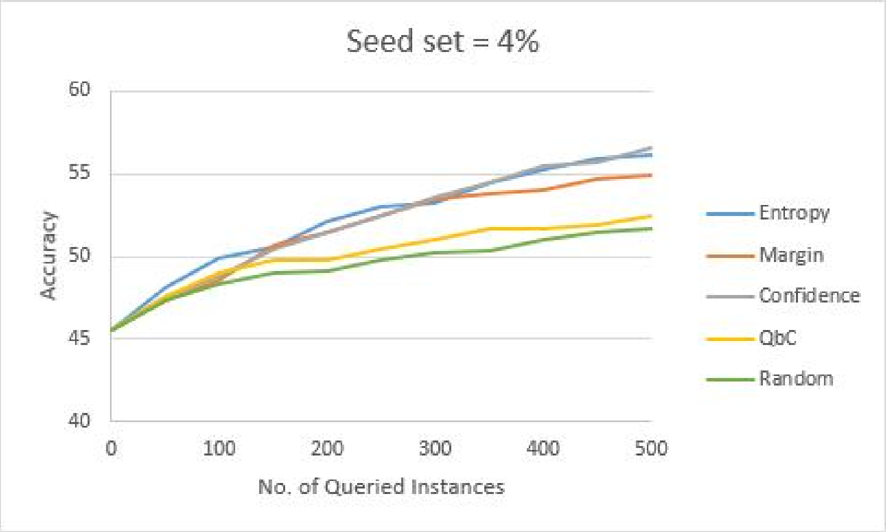


Figure 5.25: Accuracy for 4% seed set after removing strong dis-agreement

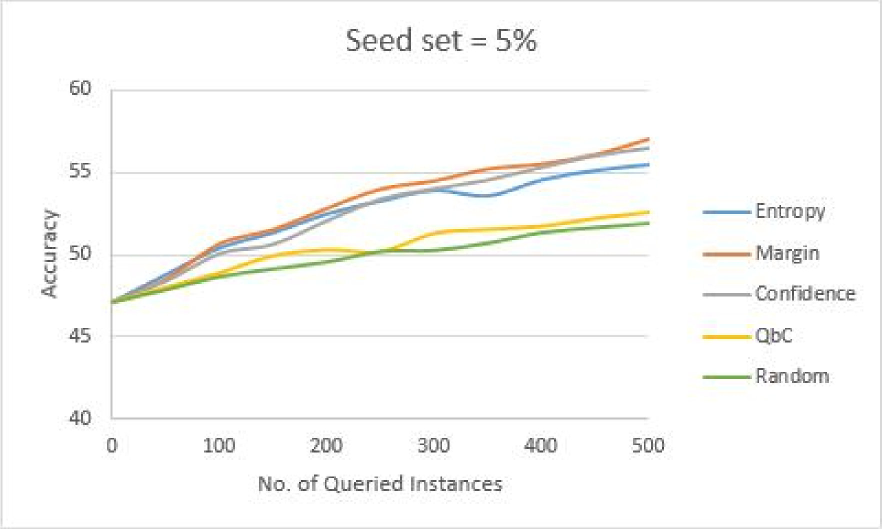


Figure 5.26: Accuracy for 5% seed set after removing strong dis-agreement

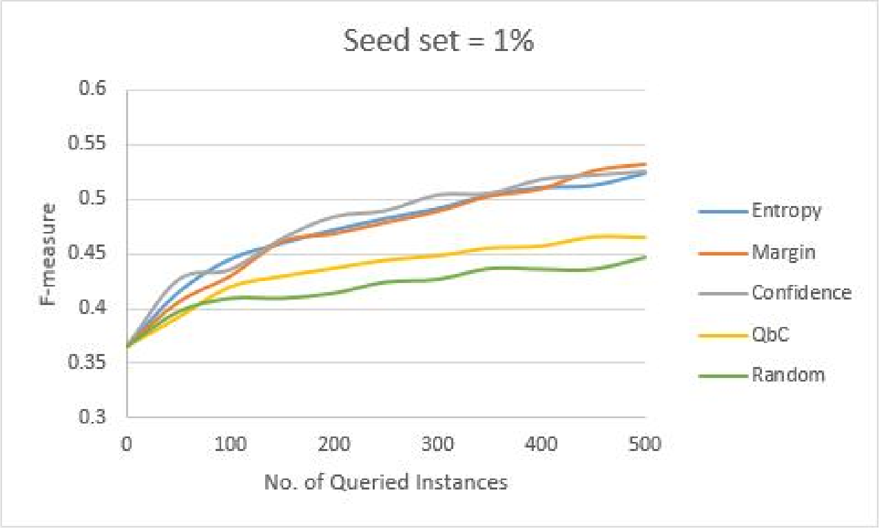


Figure 5.27: F-measure for 1% seed set after removing strong dis-agreement

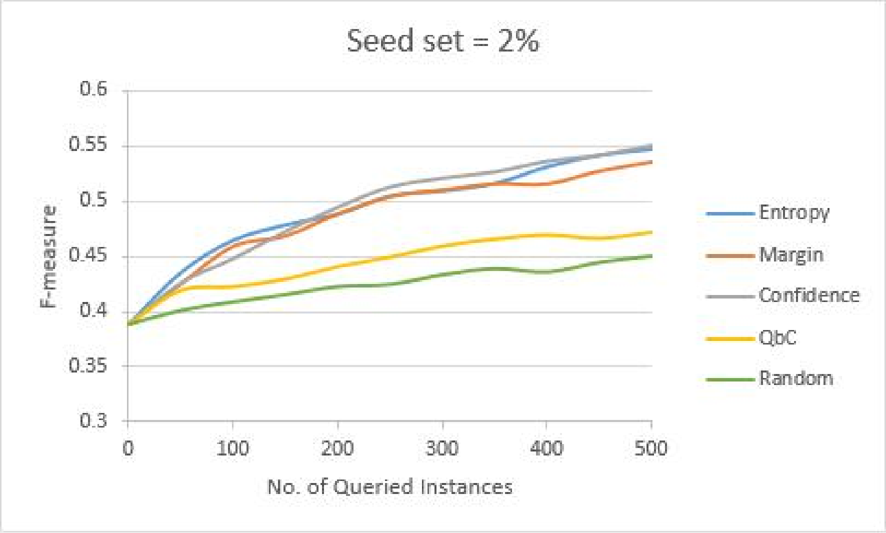


Figure 5.28: F-measure for 2% seed set after removing strong dis-agreement

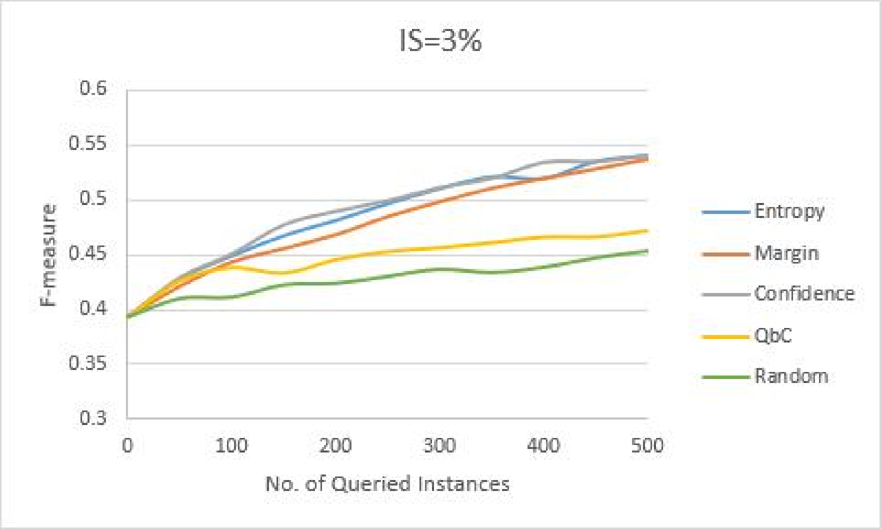


Figure 5.29: F-measure for 3% seed set after removing strong dis-agreement

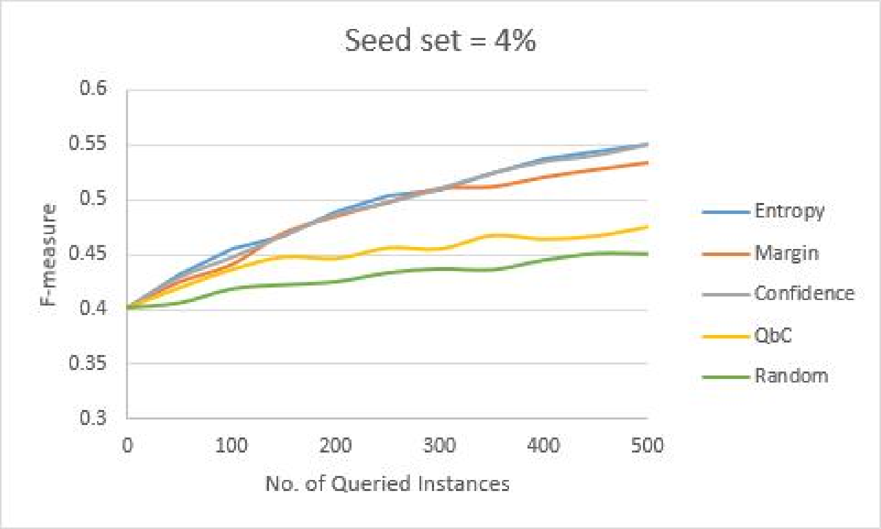


Figure 5.30: F-measure for 4% seed set after removing strong dis-agreement

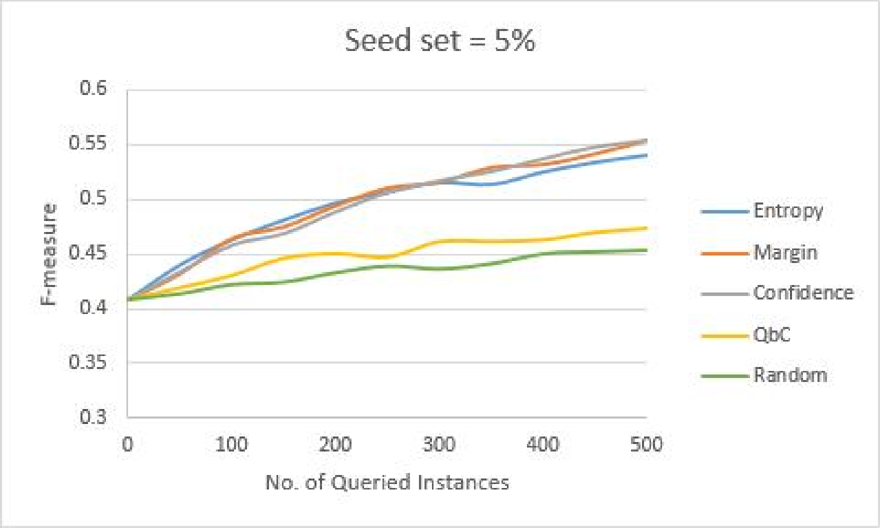


Figure 5.31: F-measure for 5% seed set after removing strong dis-agreement

The result analysis shows that in the beginning when the when the seed set is only 1%, we achieved 55% accuracy and an F-measure of 0.53. On the utilization of maximum available training set i.e. 5% seed set and number of queried instances being 500, we could achieve 56% accuracy and 0.55 F-measure. Though there is not much difference from the previous experiment but still there is a raise of 2-3% in accuracy and increment in F-measure by 0.02 points. This somehow supports our hypothesis that the data contains noise, removing which can help in better training. 208 instances make 5% of the total dataset, so it may not be too much to bring huge difference but it has given us a motivation to again analyse the dataset and look for more ways to train a better classifier. About the strategies, again Random Sampling under performs consistently. Only in the beginning with very less training size, QbC lags Random Sampling a bit. Uncertainty Sampling leads in general and among different methods of Uncertainty, Confidence appears to be suitable when less training data is available and it also behaves in same consistent leading manner when we have more labeled instances available, however margin and entropy show a variable behaviour which tends to improve with learning.

## 5.5 What if Only Agreement among Labeling is considered?

The previous experiment shows that if we can decrease the ambiguity in the data presented to the classifier, we can achieve a better performance. Because when a classifier is trained with unclear data, which was even troubling for the annotators, it cannot result in training an intelligent classifier. Moving ahead with this observation we decided to experiment with those tweets from this dataset which got same labels from all the three annotators i.e. they have been labeled as Positive, Negative or Objective unanimously by all annotators. We found that out of 4000 we have 1800 such tweets. And among those 1800 we have the class distribution as 945 Positive, 393 Negative and 462 Neutral ones. Here the distribution of instances among unlabeled pool and seed set with the increment in seed set is shown in Table 5.3.

| Seed set% | Seed set | Unlabeled Pool |
|---|---|---|
| 1% | 14 | 1426 |
| 2% | 29 | 1411 |
| 3% | 43 | 1397 |
| 4% | 58 | 1382 |
| 5% | 72 | 1368 |

Table 5.3: Instance distribution in Dataset with Complete Agreement dataset

Now we present the learning curves we obtained when the above mentioned dataset is experimented upon.

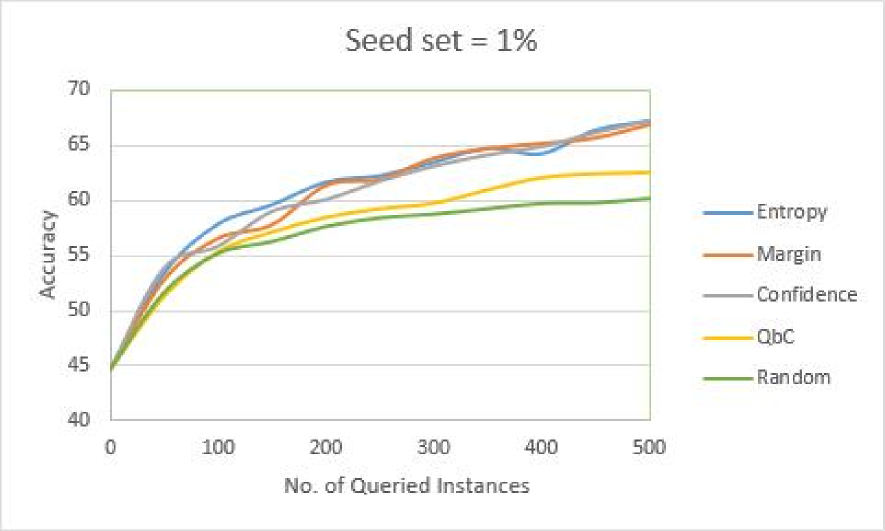


Figure 5.32: Accuracy for 1% seed set with only strong agreement data

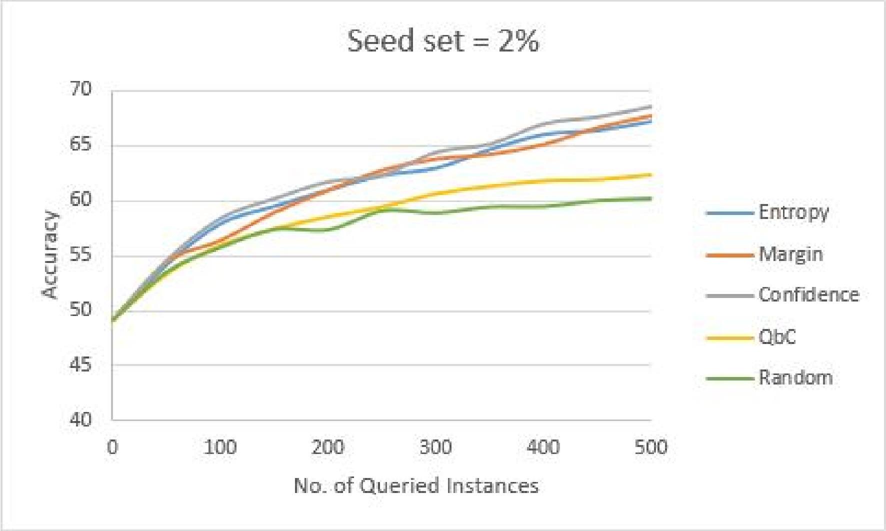


Figure 5.33: Accuracy for 2% seed set with only strong agreement data

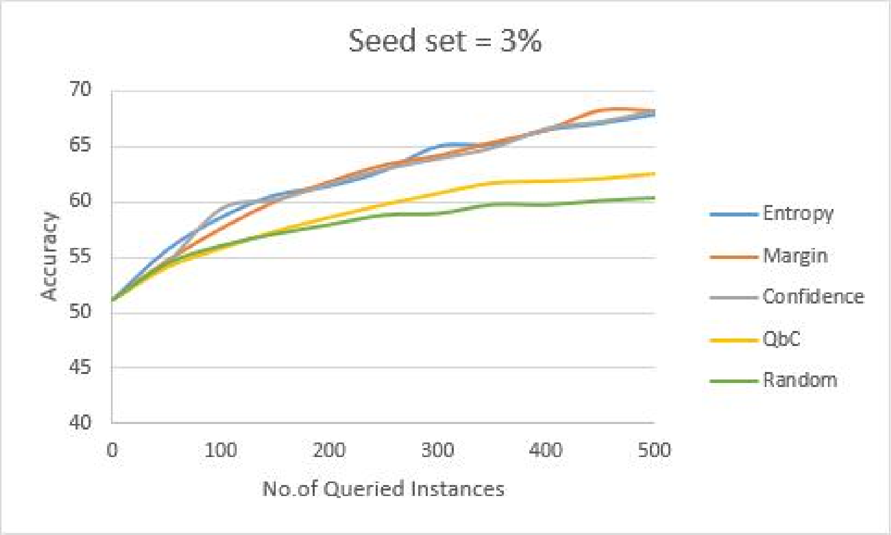


Figure 5.34: Accuracy for 3% seed set with only strong agreement data

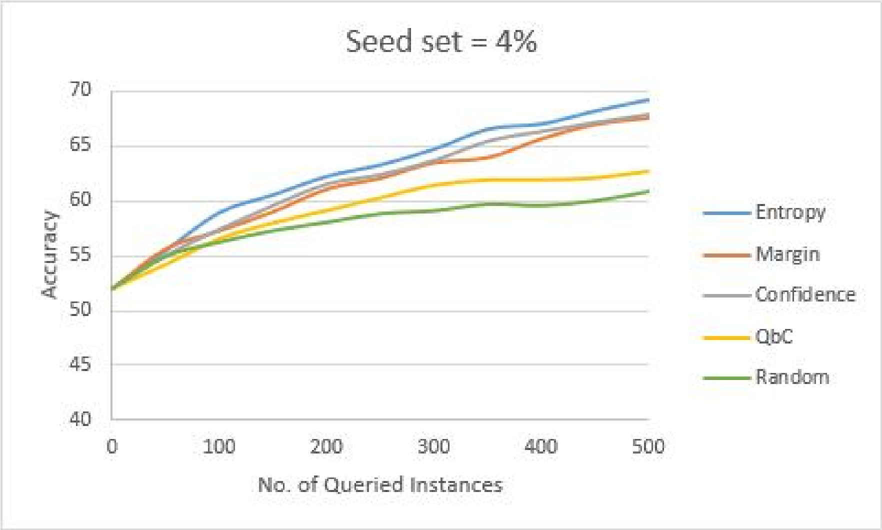


Figure 5.35: Accuracy for 4% seed set with only strong agreement data

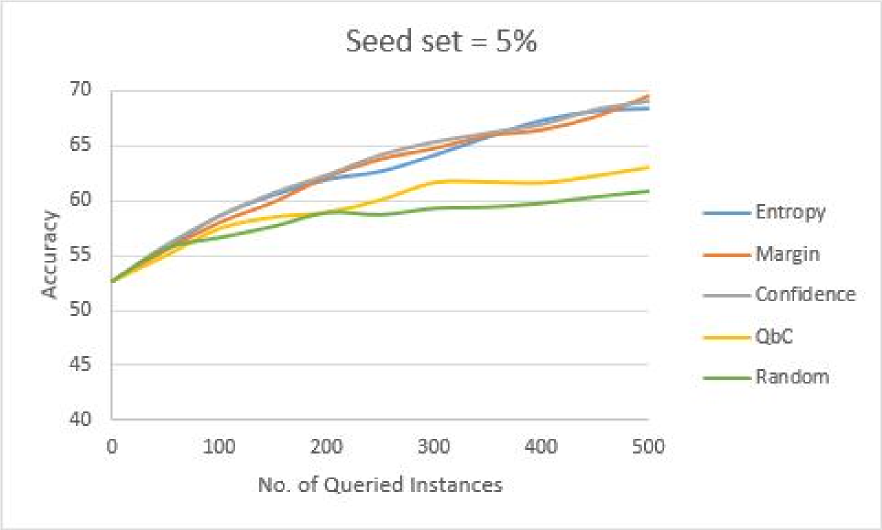


Figure 5.36: Accuracy for 5% seed set with only strong agreement data

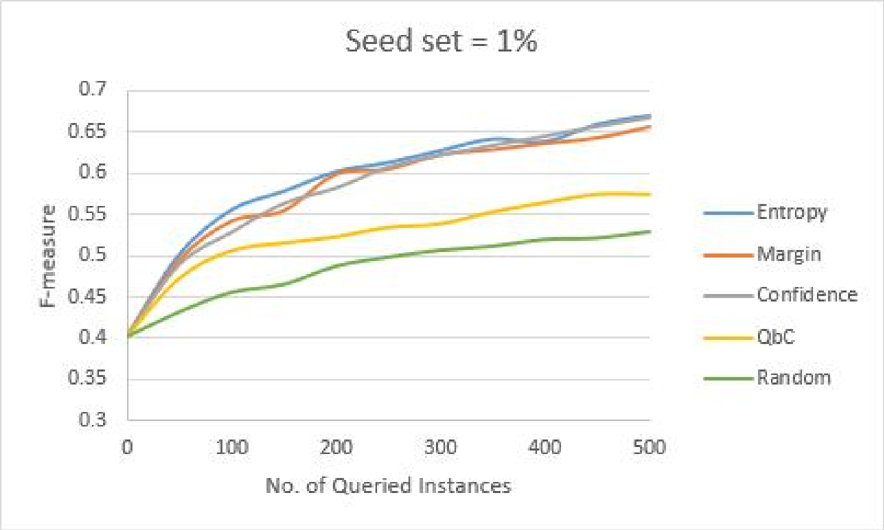


Figure 5.37: F-measure for 1% seed set with only strong agreement data

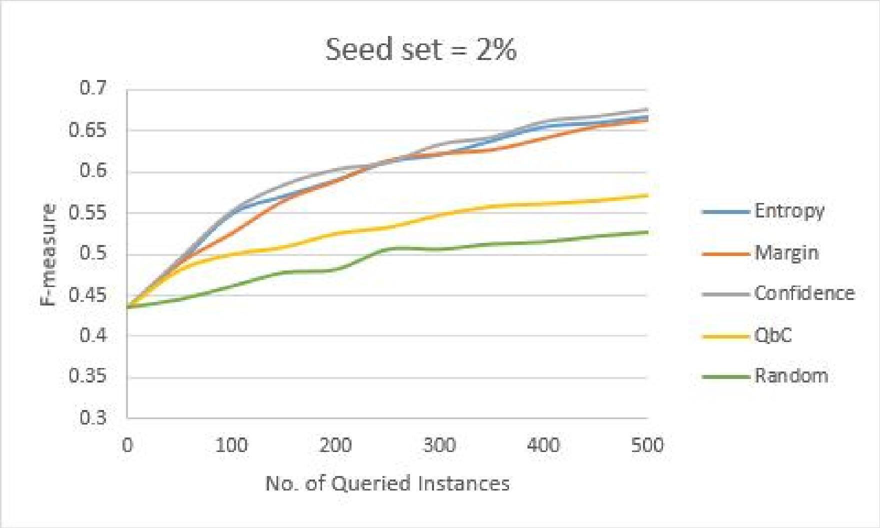


Figure 5.38: Accuracy for 2% seed set with only strong agreement data

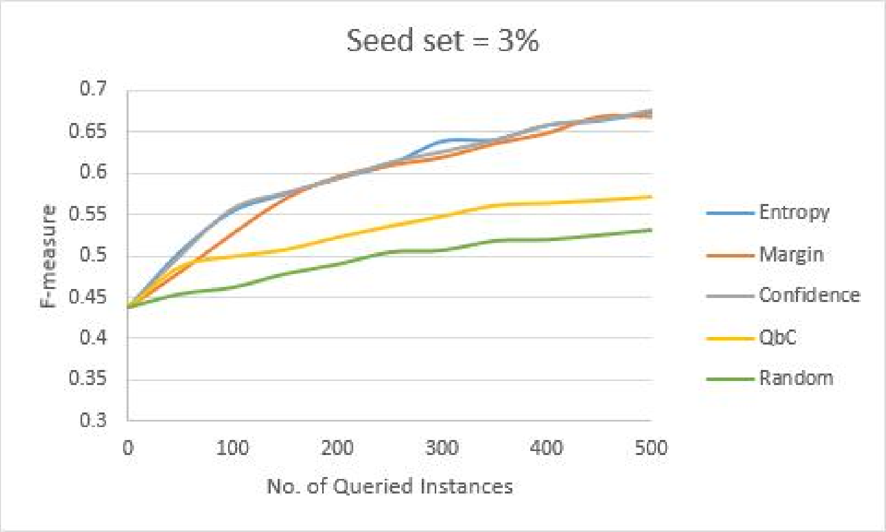


Figure 5.39: Accuracy for 3% seed set with only strong agreement data

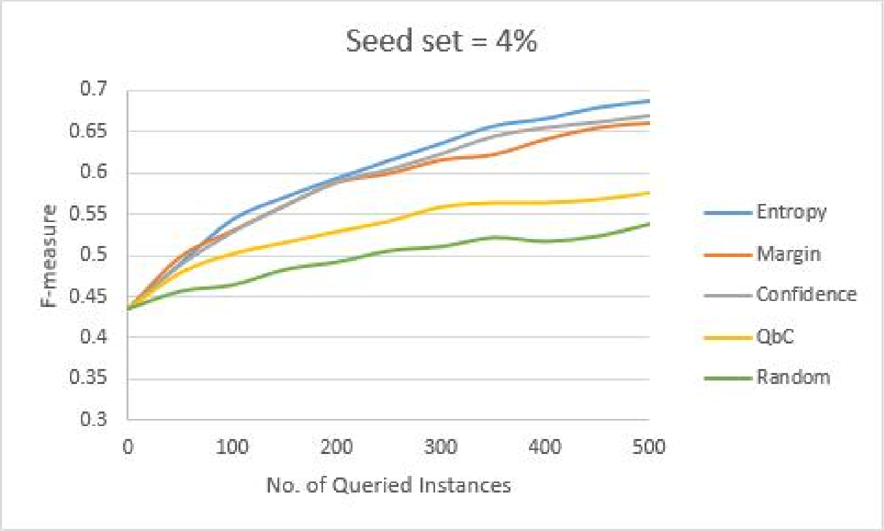


Figure 5.40: Accuracy for 4% seed set with only strong agreement data

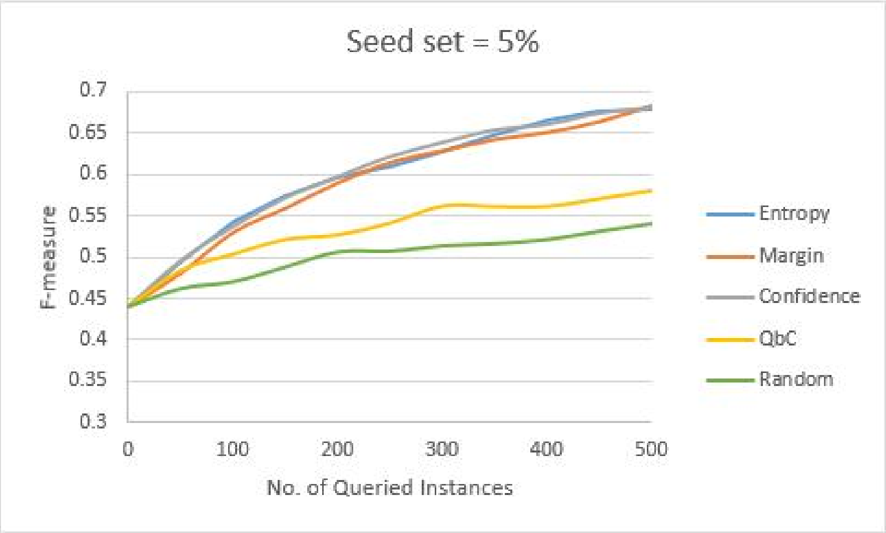


Figure 5.41: Accuracy for 5% seed set with only strong agreement data

Summarizing the comparison among strategies, Random Sampling performs worst and immediate better than it appears to be QbC. Uncertainty Sampling is best and among its methods Confidence and Margin perform closely, Entropy supersedes after learning from good number of examples probably because it includes all more information than Margin and Confidence both about the probabilistic distribution among all classes. Observing the graphs, we see a straight 12-15% increase in the accuracy and F-measure. These results are in accordance with our hypothesis that the data, both Emoticoned and manually annotated, contained same tweets which were ambiguous and quite vague to train a good classifier for sentiment analysis. When we presented only those tweets to the classifier which had same labels from three annotators, means we are training the classifier with clear examples which have a good association and mapping between the attributes and class labels. These problems are inherent to the twitter data. So in the next chapter we present a detailed analysis of labeling behaviours of annotators to understand what causes agreement and dis-agreement among annotators for twitter data.

# Chapter 6

## Analysis of Annotator Behaviour with Twitter Data

After having discussed the experimental results in detail in the previous chapter, we will now present a qualitative and quantitative analysis of the dataset used in this work. As explained earlier the dataset contained 4000 tweets which were firstly labeled on the basis of emoticons contained. Later the emoticons were removed and each tweet was presented to three different annotators. We did the majority voting among the labels and most occurring labels were selected as the final label.

### 6.1 Agreement Levels between Annotators

After this, lets analyse the data based on the agreement and disagreement among the annotators for different tweets. We defined three levels of agreement/disagreement for labels. We have three labels, Positive, Negative and Objective represented by P, N and O respectively. The uppermost level of agreement is that all three annotators assign same class to the tweet e.g it is marked 'P' by all of them. Next to it is a dis-agreement level, i.e. two annotators agree on same class but the third annotator has a different opinion. For instance a tweet is marked 'P' by two annotators and the third one labels it 'N' or 'O'. The upper most level of dis-agreement is that all the three annotators have different opinion about the class to be assigned. So we have some 208 tweets which have been assigned 'P', 'N' and 'O' labels from the three annotators separately. To carry out the analysis we have labeled the tweets now as 'A' for Agreement, 'D' for dis-agreement and 'SD' for strong dis-agreement. This agreement/dis-agreement level along with class distribution is shown in Table 6.1.

| Agreement Level | Interpretation | No. of Instances |
|---|---|---|
| Agreement | All annotators agree | 1800 |
| Dis-agreement | Two annotators agree | 1992 |
| Strong-Disagreement | All annotators disagree | 208 |

Table 6.1: Agreement Level among Annotators

## 6.2 Examples

In this section we will present some tweets from the dataset to make an understanding of the agreement levels between annotators.

### 6.2.1 Examples from class A

First of lets consider tweets from class A. *"love the french I tell people here in the south i am french and they snarl at me french are beautiful people"*. This tweet has been unanimously labeled Positive by annotators because the tweet is structured, clear in meaning and having words like love and beautiful which show the positive polarity. Considering another example from same class A, *"Just woke up to find this coldfluillness type thing, isnt going without a fight and apparently beats you up in your sleep!!!"*. This long tweet is much clear in revealing the annoyance tweeter is facing depicted by the words fight, beats you up etc so it has been labeled Negative by all. Having a look at another tweet *"@point_moot I was just listening to the scotch_mist version of Stepout when you replied"* . This tweet is not related to any sentiment and is clear in being neutral so it got the label Objective.

### 6.2.2 Examples from class D

Here we wil discuss about tweets which got same labels from two annotators. The first one is *"Shoot For The Moon! Even if you miss, you will land among the stars."* This tweet has final label Positive but one annotator assigned it Objective, maybe because the annotator finds this a general thought and not specific to any particular personal feeling. Another tweet that got Negative label from two annotators and Positive from the third one is *@stephenkruiser I am so sorry to hear that! Take* *care!*. The exclamation marks and the word 'sorry' bring the understanding closer towards Negative class but the phrase 'Take care!' induces a positivity which results in confusion. Last example from the class Objective is *@JonathanRKnight I am beginning to think that you are now finally Twiverted.* This is a non-sentiment bearing tweet so majority voting makes it fall into Objective class but one annotator assigned it Positive class which means labeling of such tweets is totally subjective to the annotator. The number of tweets falling in this class is 1992 out of 4000. The examples showed that even the two annotators agreed there is some level of ambiguity in the tweet text which makes it difficult to be understood or interpreted. This definitely is going to contribute in the low performance of classifier as it couldn't be trained with distinguishing examples.

### 6.2.3 Examples from class SD

Here we will quote a couple of instances where tweet text is so vague that all annotators had different opinion. One such tweet is *@IamMarkus jep I could use some sleep* which is very unclear. Looking for a long stronly dis-agreed tweet we have *"@rmolden LOL yes - but I truly can't believe what I forgot! OMG Oh ! perhaps I am doing too much?! lol love! hehe help..."* . The tweet does not mean anything related to sentiment analysis. Another such tweet is *"@NaiveLondonGirl Why howdy maam. I'm fine n dandy. And if ur naive I'm a dutch man's uncle."* This tweet is also ambiguous and its labeling is subjective to the inference annotator makes from it so it was labeled differently by all.

## 6.3 Manual Annotation feedback

We asked the annotators to share their experience of labeling these tweets. Compiling their experience here we infer that as the data did not belong to any one particular topic, it was very difficult for them to judge a tweet as Positive or Negative. It turned out to be mainly related to the interpretation of the annotator. So even for manual annotation, the labeling task was highly subjective. The main observations and difficulties highlighted by the annotators in labeling these tweets are listed here.

1. Some tweets contain sentimental information intrinsically, those tweets are

easier to label. However, other tweets do not contain sentimental information but rather reflect an opinion of the tweet owner. It is difficult to label those tweets and the best label that could be given to them is most probably ”Objective”.

2. It is difficult to assign labels to tweets if they are obviously part of longer conversations. For example, a tweet, such as ”I love my life” is self-contained and can easily be identified as ”Positive” But it is not the case if tweet is part of a longer dialog, e.g. ”I agree with you wholeheartedly”.

3. The long tweets at times contain mixed polarity. e.g. ”It was a nice day, but I'm so exhausted now”. It is difficult to infer the actual sentiment of the tweet.

4. Some tweets contain sentiment, but in a way that positive and negative feelings are revealed at the same time. Then the annotator should somehow compare the levels of positive and negative sentiments in the tweet and decide the label of the given tweet accordingly. That is a very difficult and time consuming part of the labelling. For instance, ”I hate 2 be allergic, I want a puppy so bad that I've already thought some cool names”.

5. Tweet length contributes to the easiness of labeling tasks because such tweets are better structured and complete in meaning too.

6. Among the shorter tweets, the easier tweets were those where the tweeter explicitly mentioned the sentiment e.g. I loved this , I am feeling super excited about , or some expressions like yupee!!, Urgh! etc.

7. The hardest part of labeling seemed to be the need to figure out if the sentiment is determined by the underlying idea or directly what was written as the tweet.

8. There is an excessive use of the slang language in some of the tweets. Although the number is not that high, some of the tweets are so much unstructured that it is difficult to understand the sense of that tweet.

9. Some of the tweets are very easy to be labeled. If it was saying ”I hate Monday morning” it is pretty self explanatory, and easy to be labeled.

As the labels for this dataset initially came from the emoticons, this qualitative analysis, low positive correlation between Emoticon-ed labels and manual labeling of two datasets and annotators feedback show that emoticons may not be very helpful in understanding the sentiment.

## 6.4 Quantitative Analysis

Now we will discuss some quantitative analysis that we did on the data to understand dis-agreement levels.

## 6.5 Correlation between Emoticon-ed and Human Labeled dataset

Now we will present an analysis of this labeled data. First of all we want to know how much these labels are in accordance with the labels initially acquired from emoticons. To know this, we calculated Pearson correlation between the two. Pearson correlation is a measure of linear correlation between two variables X and Y. The range of this correlation is between -1 and 1, both inclusive. The interpretation is like 1 is total positive correlation, 0 means no correlation and -1 is total negative correlation. Equation 6.1 states the formula for calculating Pearson correlation between $x$ and $y$ denoted by r$_{xy}$

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sqrt{(\sum x_i^2 - n\bar{x}^2)}\sqrt{(\sum y_i^2 - n\bar{y}^2)}} \tag{6.1}$$

Here $n$ is the number of values for $x$ and $y$, and are the mean of all values of $x$ and $y$. To calculate the correlation, we removed the tweets with class label 'Objective' from the manually annotated data and also the same corresponding tweets from Emoticon-ed data because Emoticon-ed data does not contain the Objective class. The correlation between the two datasets came out to be 0.6189. This implies that the data is considerably positively correlated. This also gives an indication that the labels we achieved with emoticons are not very reliable.

## 6.6 Feature Extraction

In the classification of tweets for sentiment analysis as the main task, we have considered bag of words representation of data. This means all the features are actually the words occurring in tweet text. To understand the reasoning of different agreement levels, this time we have decided to extract some other features from tweet texts to analyse which are the most contributing features towards agreement or dis-agreement among annotators. The features, which have been extracted with a python script for each tweet are listed in the Table 6.2

| Feature | Explanation |
|---|---|
| tweet_length | No. of characters in the tweet |
| positive_seed_words | No. of positive seed words |
| negative_seed_words | No. of negative seed words |
| positive_emoticons | No. of positive emoticons |
| negative_emoticons | No. of negative emoticons |
| exclamation_marks | No. of exclamation marks |
| question_marks | No. of question marks |
| negative_dictionary_words | No. of negative dictionary words |
| positive_dictionary_words | No. of positive dictionary words |
| elongated_words | No. of elongated words |
| caps_words | No. of words in CAPS |

Table 6.2: Extracted Features

We used senticwordnet [29] to determine positive and negative dictionary words. A word is counted as positive if the positive polarity outweighs the negative one. Similarly for negative words the negative polarity is larger than positive polarity. But the language used in tweets is not formal and the spellings are also mostly not correct (people mostly use short spellings and slang words due to limitation of characters in tweet text), we decided to extract positive and negative words specific to these tweets. For this purpose we extracted the tweets which were labeled as Positive or Negative by all three annotators. These tweets were then tokenized and with the manual inspection we compiled two lists of Positive and Negative words which were used to extract positive_seed_words and negative_seed_words. We were able to come up with 313 Positive and 169 Negative sentiment bearing words from the tweets, few of which are listed in Table 6.3

| Negative seed words | Positive seed words |
|---|---|
| wow | sick |
| yaay | sigh |
| soooper | urrrggg |
| HILARIOUS | Smugness |
| Congrats | bitter |

Table 6.3: Sentimental bearing words

After collecting these features, we performed attribute selection with WEKA. The goal is to identify most contributing attributes towards the classification between agreement and dis-agreement. For this particular purpose, we used our dataset with two different underlying concepts. First, we did feature selection for dataset with three classes i.e. A, D and SD representing Agree, Dis-agree and Strongly dis-agree respectively. We performed 10 fold cross-validation with ClassifierSubsetEval and WrapperSubsetEval using Decision tree (J48). The results of WEKA show the contribution of the attributes in the classification process in terms of percentage. As per results, the most significant features for this particular problem are

- tweet_length
- positive_seed_words
- negative_seed_words
- positive_dictionary_words
- exclamation_marks
- question_marks

Next we merged strong dis-agreement and dis-agreement into one class represented by 'D', so now we have a two class problem. Repeating the same attribute selection process with WEKA we obtained the below mentioned attributes as highly ranked ones.

- tweet_length
- positive_seed_words
- negative_seed_words
- exclamation_marks
- positive_dictionary_words
- negative_dictionary_words
- question_marks
- caps_words

# Chapter 7

## Conclusion & Future work

### 7.1 Results Summary

The main goal of this work was to present a comparative analysis between popular AL strategies for building a classifier for sentiment analysis with Twitter data. The strategies included Random Sampling, Uncertainty Sampling and QbC. These strategies are well understood and computationally efficient to be used in real world scenario. For this analysis we performed experiments with different seed set sizes which were randomized several times to remove the chances of lucky distribution of data. There are several useful findings that we concluded from the work done in this thesis.

Among the Active learning strategies, Random Sampling was under-performing most of the times. The only strategy which lagged Random-Sampling few times turned out to be QbC. Uncertainty Sampling performed well consistently and among its three implemented methods, Confidence turned out to perform well even with less earned labels. As the amount of training instances increased, in terms of seed set and budget allocated for AL, Entropy was performing well.

We performed these experiments with bag of words representation of Twitter data which was first labeled on the basis of emoticons and later was manually labeled by three annotators. The results of the experiments have been shown as learning curves for Accuracy and F-measure. We considered Accuracy because it is most intuitive performance metric. But Accuracy is a good measure only with symmetric class distribution. With skewed datasets, which is mostly the case in real world, it is not a reliable measure. So we considered also the F-measure. F-measure is not as intuitive as Accuracy but is useful for uneven class distribution datasets.

We couldn't achieve very high accuracy or F-measure which directed us to do a detailed analysis for the experimented dataset and annotators behaviour. We found out that the inherent vagueness in Twitter data caused difficulties in labeling. So there was a considerable level of dis-agreement between the labels acquired from the annotators which resulted in confused training of the classifier. Also we worked on finding the most significant attributes of tweets contributing to the ease of labeling and agreement between the annotations.

## 7.2 Future Work

In this work we have used the bag of words representation of dataset. In future we would like to experiment with different representations to build more authentication for the results. Also we have done a comparative analysis between popular and computationally efficient sampling strategies. It would be interesting to analyse the performance of some sophisticated strategy with Twitter data and optimize it for efficient computations.

During this work, we developed a tool to provide an annotation mode for testing some AL strategies with Twitter data. It is near to real world because it provides an interface to involve human annotators for labeling and normally we need to get labels from annotators to label the data. The tool selects useful instances based on the chosen AL sampling strategy. Due to short of time and unavailability of annotators we could not perform much experiments with this tool but in future we can find interesting findings about the behaviour of annotators and the factors contributing to good quality labeling. We can also make useful interpretations about the time consumed in selection of next instances by the strategy compared to the time taken by human to comprehend and assign a label to tweets.

So, this thesis provides a foundation for to understand the interesting phenomenon of Active Learning and its importance in today's world of overwhelmed data which needs to be converted into useful information applying as little manual effort and time as possible.

# Bibliography

[1] Y. Takeichi, K. Sasahara, R. Suzuki, and T. Arita, "Twitter as social sensor: Dynamics and structure in major sporting events," in *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, vol. 14, 2014, pp. 778–784.

[2] T. M. Mitchell, *The discipline of machine learning.* Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006, vol. 17.

[3] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval.* Springer-Verlag New York, Inc., 1994, pp. 3–12.

[4] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[5] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 27–39, 2014.

[6] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory.* ACM, 1992, pp. 287–294.

[7] I. Guyon, G. C. Cawley, G. Dror, and V. Lemaire, "Results of the active learning challenge." *Active Learning and Experimental Design@ AISTATS*, vol. 16, pp. 19–45, 2011.

[8] C. Beyer, G. Krempl, and V. Lemaire, "How to select information that matters."

[9] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," in *AAAI*, 2005, pp. 746–751.

[10] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Advances in Intelligent Data Analysis.* Springer, 2001, pp. 309–318.

[11] L. P. Evans, N. M. Adams, and C. Anagnostopoulos, "When does active learning work?" in *Advances in Intelligent Data Analysis XII.* Springer, 2013, pp. 174–185.

[12] C. E. Shannon, "A note on the concept of entropy," *Bell System Tech. J*, vol. 27, pp. 379–423, 1948.

[13] P. Melville and R. J. Mooney, "Diverse ensembles for active learning," in *Proceedings of the twenty-first international conference on Machine learning.* ACM, 2004, p. 74.

[14] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1.* Association for Computational Linguistics, 2009, pp. 477–485.

[15] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval.* New York, NY, USA: McGraw-Hill, Inc., 1986.

[16] C. Boulis and M. Ostendorf, "Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams," in *Proc. of the International Workshop in Feature Selection in Data Mining*, 2005, pp. 9–16.

[17] A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study," in *Advances in Information Retrieval.* Springer, 2004, pp. 181–196.

[18] H. Zhang, "The optimality of naive bayes," in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, V. Barr and Z. Markov, Eds. AAAI Press, 2004.

[19] U. Paquet, J. Van Gael, D. Stern, G. Kasneci, R. Herbrich, and T. Graepel, "Vuvuzelas & active learning for online classification," in *NIPS Workshop on Comp. Social Science and the Wisdom of Crowds*, 2010.

[20] X. Hu, J. Tang, H. Gao, and H. Liu, "Actnet: Active learning for networked texts in microblogging." in *SDM.* Citeseer, 2013, pp. 306–314.

[21] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Stream-based active learning for sentiment analysis in the financial domain," *Information Sciences*, vol. 285, pp. 181–203, 2014.

[22] A. Brew, D. Greene, and P. Cunningham, "Using crowdsourcing and active learning to track sentiment in," 2010.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[24] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "Moa: Massive online analysis," *The Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.

[25] J.Yarrow. (2010) Mvc tutorial.

[26] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL student research workshop.* Association for Computational Linguistics, 2005, pp. 43–48.

[27] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[28] J.Yarrow. (2010) Twitter conference.

[29] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” in *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, 2006, pp. 417–422.