Islamic University of Gaza
Faculty of Information Technology

"Sentiment Analysis of Microblogs in Education Domain"

By

Alaa AlHaddad

Master Thesis

A Master Thesis presented to the Faculty of Information Technology of Islamic University of Gaza in partial fulfillment of the requirements for the degree of Master of Science in Information Technology.

Supervisor: *Dr. Rawia Awadallah*

Gaza, April, 2015

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

"Sentiment Analysis of Microblogs in Education Domain"

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's name: Alaa I. AlHaddad اسم الطالب: آلاء ابراهيم زكريا الحداد

Signature: التوقيع:

Date: 26/07/2015 التاريخ: 2015/07/26

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي: 1150

الرقم Ref ج س غ/35/
التاريخ Date 2015/04/29م

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحثة/ **الاء ابراهيم زكريا الحداد** لنيل درجة الماجستير في **كلية *تكنولوجيا المعلومات*** **برنامج تكنولوجيا المعلومات** وموضوعها:

تحليل الآراء للمدونات في مجال التعليم

Sentiment Analysis of Mircroblogs in Education Domain

وبعد المناقشة التي تمت اليوم الأربعاء 10 رجب 1436هـ، الموافق 2015/04/29م الساعة الواحدة ظهرًا، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. راوية فوزي عوض الله	مشرفاً و رئيساً	.....
أ.د. علاء مصطفى الهليس	مناقشاً داخلياً	.....
د. إيهاب صلاح زقّوت	مناقشاً خارجياً	.....

وبعد المداولة أوصت اللجنة بمنح الباحثة درجة الماجستير في **كلية *تكنولوجيا المعلومات*/ برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحها هذه الدرجة فإنها توصيها بتقوى الله ولزوم طاعته وأن تسخر علمها في خدمة دينها ووطنها.

والله ولي التوفيق،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 هاتف ص.ب. 108 الرمال، غزة، فلسطين
public@iugaza.edu.ps www.iugaza.edu.ps

Dedicated to my parents, and my adorable daughter and son

Acknowledgment

First and foremost, all praise and thanks are due to God for giving me the power to believe in my passion and pursue my dreams. I could never have finished this achievement without the faith I have in Him.

I am heartily thankful to my supervisor, Dr. Rawia Awadallah for her splendid guidance and encouragement during the course of my studies. Her support and insightful advice were crucial to my academic success as a researcher.

I am also grateful to my amazing friends who never hesitate to support me, especially Abeer Barakat. Thank you Abeer for being a steadfast source of encouragement and inspiration to me.

Finally, I would like to express my greatest and deepest gratitude to my family. Mother, Father, Sisters and Brothers specially my Brother Bahaaeddin.

Last but not least, thanks to my kids, Islam and Omar, for being so cute and filling my life with joy and hope. And being patient all the busy time to finish my thesis.

Abstract

During the recent years, microblogging and social media have become very popular where millions of people post short text about different things. Topics range from personal life and work, to current events, news, and interesting observations and political thoughts.

Education institutes become aware of the benefits engaging in such technology, and many instructors use social media in teaching courses they offer. Courses adopting social media in the learning process allow students to discuss with each other and with their teacher different topics and express their opinions on various aspects of these topics. The huge amount and variety of opinions generated out of these discussions create new opportunities for assessing teaching courses. Manual methods for analyzing opinions in these huge amount of data are infeasible.

Sentiment analysis is a research field that focuses on automatically identifying the subjectivity and the polarity (e.g. positive or negative) of a given text on an entity or a topic. It is a classification problem, where learning algorithms are used. Most of previous works focus on using supervised algorithms, however such algorithms are very expensive since we need to manually annotate a large amount of data for training the classifiers, in addition it is domain dependent (e.g. products, movies, politics, etc.). Besides, certain characteristics of social media content introduce challenges in their analysis. Informal English blended with abbreviations, slangs and context specific terms; lacking in sufficient context and regularities and delivered with an indifferent approaches to grammar and spelling, all at the top of these characteristics.

Most of previous works on sentiment analysis tackle domains such as economic, products, movie reviews, and political domain. There is a paucity of literature in the education domain. Our research is a contribution to this field. In particular, we propose a sentiment analysis prototype for microblogs posted in learning activities. The prototype automatically classifies microblogs of learning activities into positive and negative with less costs in terms of learning requirements. Our approach aims to achieve this objective using a novel combination of features extraction and engineering methods, and using a semi-supervised sentiment classification model based on label propagation algorithm.

The results returned of the experiments we conducted to evaluate the model were competitive to existing works. The F-measures of our approach using different datasets has an average value of $\approxeq 80\%$.

Keywords: Semi-supervise, Opinion mining, Sentiment Analysis, Microblogs.

Contents

List of Figures

List of Tables

1

Introduction

Seeking opinions is a very important stage for decision making in various fields. Traditional tools can be used to collect these opinions and get feedback such as interviews, questionnaires, focus groups, brain storming, and many other tools. These tools can be biased by many factors such as mental situation for people. So we need an unbiased tools that cannot be affected by humanity factors, where people can deliver their opinion freely any time anywhere as their behavior affect the decision will be taken. During the recent years, microblogging and social media have become very popular where millions of people post short text about their personal life and work, to current events, news, and interesting observations and political thoughts. These posts are published on the personal page and sent to their followers or friends. By following a group of people, users manage awareness of what is happening to their family, friends, and communities even the world. In addition, they share their life online and express their personal feelings, opinions, and comments about anything they are concerned. The huge amount and variety of generated content and the relationships between users generating this content create new opportunities for understanding web-based practices and building socially intelligent applications. Investigations around social data can be broadly categorized as follows [62]: 1) Understanding aspects of the user-generated content, 2) Modeling and observing the user network that the content is generated in, and 3) Characterizing individuals and groups that produce and consume the content.

Certain characteristics of social media content introduce challenges in their analysis. Informal English blended with abbreviations, slangs and context specific terms; lacking in sufficient context and regularities and delivered with an indifferent approaches to grammar and spelling, all at the top of these characteristics. Another spices added to microblogs that gave it its taste, are emotions, URL, and hashtags. Single emotion can tell all what the user feel and want to express. Hashtags are used not only to add context and

metadata to the post, but also for promotion and publicity. Sharing URLs, will support the user's post content and feelings. Many earlier researches developed content analysis technique for traditional and formal genre like news, Wikipedia or scientific articles not translated well to microblogs content [30].

Education institutes become aware of the benefits of engaging in such a technology. Many instructors use social media in teaching courses they offer. Many recent projects work on how to incorporate social media to courses and classrooms, and lot of efforts done to show the best practice and types of activity to improve students experience, and to show how this can improve student's skills [14]. Some researchers conducted practical experiments to measure the effectiveness of using Twitter in education over using traditional teaching methods [51, 16]. These experiments give a guide for the best practice of how to bring Twitter to classrooms [17, 37, 14]. The studies revealed that using such tools have many advantages such as teachers and students can contact each others outside class rooms keeping discussions and breaking shyness barriers, customization of learning depending on the student, using multimedia and connect to other learning communities and new educational content can easily fetched [14]. As a result, there is an increase in the amount of data within these social media that reflects what students are learning and how well they are learning. Courses adopting social media in the learning process allow students to discuss with each other and with their teacher different topics and express their opinions on various aspects of these topics. The data generated out of these discussions constitutes a valuable resource on which teachers can rely in order to conduct course assessment. This can make both teachers and students aware of holes in knowledge or understanding. This leads teachers to address specific content and provide additional learning strategies to fill in these holes, and also leads students to set goals and track their progress toward achieving them [13]. Manual methods for analyzing opinions in these discussions are infeasible. Opinion mining and sentiment analysis research field focus on proposing methods for automating the process of opinions analysis.

Our research is a contribution to the field of opinion mining and sentiment analysis. In particular, we seek to automatically identify sentiments (e.g. positive or negative) of microblogs using a semi-supervised approach. Since the education is one of the important domains that incorporate such a media to courses and classrooms, and since there is a lack of research on semi-supervised sentiment analysis for such a domain, we will be focusing on it as a main case study.

1.1 Opinion Mining and Sentiment Analysis

The first appearance of considering opinions and its subjectivity was by J. Wiebe in his paper [60] describing an approach using distributional similarity to distinguish between subjective sentence that hold opinion and objective one that present factual information.

With the rapid growth in the use of technology especially web 2.0 technology, there is a corresponding increase in the amount of opinions available on social media (e.g., reviews, forum discussions, blogs, microblogs, Twitter, and social networks). This crowded opinions is worthy to be considered and pay effort to extract what people feel. Analyzing these opinions and feelings is very essential for decision making. Opinions with all its related notions such as sentiment, sensation, emotion, attitudes, appraisals, and the assessment are the core of the sentiment analysis and opinion mining [33].

Manual methods for analyzing opinions is time consuming and very expensive. Automating this process is a very challenging task. Opinion mining and sentiment analysis researches seek to propose efficient methods for automatically analyzing opinions [33]. These researches use natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Mining opinions expressed in the user generated content is a challenging yet practically very useful problem. It can be applied for many different domains such as customer products, financial markets evaluation, studying trading strategies, characterize social relations, health, disaster management, and many other applications [20, 39, 33]. There is a huge demand of sentiment analysis. Online advice and recommendations are usually used by many people before they buy any product. Companies want to know ”How successful was their last campaign or product launch” based upon the sentiments of the customers on social media.

Some of the challenges in Sentiment Analysis are: people express opinions in complex ways and tend to express a lot of remarks in the form of sarcasm, irony, implication, etc. which is very difficult to interpret. For Example, ”How can someone buy this camera” is extremely negative sentiment yet contains no negative lexographic word. Even if a opinion word is present in the text, their can be cases where a opinion word that is considered to be positive in one situation may be considered negative in another situation. In informal medium like Twitter or blogs (Social media), more likely people combine different opinions in the same sentence which is easy for a human to understand, but more difficult for a computer to parse. In addition, as there is constrain on the number of characters used for each message, users follow informal grammar. They use misspellings,

creative spellings, slang, URLs, emoticons, new words, and genre-specific terminology and special abbreviations [23]. Working with such informal text opens a new trend in natural language processing [24].

There are few main fields of research predominate in sentiment analysis: subjectivity analysis, sentiment classification, and opinion summarization [33].

To determine whether a text holds any subjective information or opinion, subjectivity analysis is performed. The text pieces may or may not contain useful opinions or comments. The subjective sentences are the relevant texts, and the objective sentences are the irrelevant texts. The subjective sentences are those sentences having useful information for the sentiment analysis.

Sentiment classification deals with classifying text or review according to the opinions towards certain objects or towards features of certain objects. For example, classifying sentiments on the laptop in general or classifying the sentiments only on the screen quality. Classification methods require learning. Learning methods can be supervised, semi-supervised, or unsupervised. Supervised learning algorithms such as Naïve Bayes, Support Vector Machine (SVM), Dependency Tree-based classifier and Genetic Algorithm are usually used for sentiment classification tasks. Such methods require a large amount of manually annotated datasets for training the classifiers. Unsupervised learning algorithms such as rule-based ones rely on using syntactic pattern, distributional similarity, and dictionaries which are not always easy to obtain [33]. On the other hand, semi-supervised learning algorithms such as Label Propagation require small labeled datasets [33].

There are different levels in analysis depending on granularities; 1) Document level classify the whole document whether it holds an overall positive or negative sentiment, 2) Sentence level goes to sentences in a document to determine its polarity, 3) Entity and aspect level is a finer-grained analysis that not concern just on the whole sentiment but tie the sentiment with a target to better understand sentiment problem [33].

The task of Opinion Summarization is different from traditional text summarization because only the features of the product are mined on which the customers have expressed their opinions. Opinion summarization does not summarize the reviews by selecting a subset or rewrite some of the original sentences from the reviews to capture the main points as in the classic text summarization.

1.2 Opinion Mining and Sentiment Analysis in Education

One of the major academic goals for any university is to improve teaching quality. This can be achieved by following up student's attitudes. Student comments about courses can be considered as a significant informative resource that can be manipulated and analyzed to examine student's satisfaction [18].

1.2.0.1 Learning Analytics and Formative Assessment

Developing learning and teaching initiatives to improve retention and progression in education process is an important academic concern, which mainly depends on monitoring student performance and exploiting student feedback. Teachers and instructors need to uncover what and how well the student understands throughout the course of instruction. A teacher engaging in **formative assessment** uses information from a particular assessment analytically and diagnostically to measure the process of learning to inform himself/herself or the students of progress. This will guide for further learning, and adjust instructional strategies in a way intended to further progress toward learning goals. Formative assessment makes both teachers and students aware of holes in knowledge or understanding, leading teachers to address specific content and provide additional learning strategies to fill in these holes, leading students to set goals and track their progress toward achieving them [13]. **Learning Analytics** is a research field that goes in line with the goals of formative assessment. It seeks to enhance the learning process through systematic measurements of learning related data, and informing learners and teachers of the results of these measurements, to support the control of the learning process. The prime data source for most learning analytic applications is data generated by learner activities, such as learner participation in continuous, formative assessments. That information is frequently supplemented by background data retrieved from **Learning Management Systems** [61]. However, the recent ubiquitous access to social networks like microblogs (e.g. Twitter) become a critical part of learner's online identity, and an expected part of learning platforms and analytic research as well. Learning Analytics should as a result face the challenges of finding ways to capture and analyze learning data generated in social media networks. With the rapidly growing interest in and technical ability to leverage these data in educational settings, there is a sense that many recent educational technology and big data initiatives will provide education committee with

different point of view for what they know about learning and teaching. It will help educators better assess their pedagogical practices, and devise innovative educational methods, all with the goal of improving education.

1.2.0.2 E-learning and Sentiment Analysis

In E-learning, and distance learning, teacher and students are out of space and time. This separation results in lack of face-to-face communication which lead up the absence of emotion, and hence might affect learning process. As students may have negative feelings like frustrated or confusion if this problem cannot be resolved over a long period of time, resistance feelings will directed toward learning [8]. Opinion mining offer new opportunities as it can reflect reasonable needs, because emotion by written words are more mature, specific, and reasonable. An improvement can be achieved when applying opinion mining. As teacher be aware of his student's opinions, he can evaluate effectiveness and efficiency of all e-learning aspects. In the same time face huge challenge to mine valuable information from learning-related text, as information are exchanged in free style over broad topics [8]. Alvaro et al in [45, 36], represent the SentBuk, an application to support sentiment analysis in Facebook, this application can be used by adaptive e-learning systems to support personalized learning to tackle users emotion over time. This can serve as feedback for teachers, on the other hand offer a guide each student through the learning process according to his/her particular needs and preferences over time. Few of research papers mention sentiment analysis for e-learning educational sectors. A glance for statistics about the growing and expanding of e-learning systems shows that in 2013, about 7.1 million of higher education students had one or more online course in the United States, with The 6.1 % growth rate [4]. E-learning is expected to grow even more and at a quicker pace with the growth of the Internet and information technology infrastructure [28].

1.3 Motivation and Importance of Research

- In our research, we propose a semi-supervised approach for sentiment analysis of microblogs. Such an approach overcomes the limitations of supervised learning approaches in terms of learning requirements.

- The proposed approach can be applied on different domains and employed for different applications where microblogs sentiment analysis is needed.

- One of the domains that started to incorporate microblogs is Education. The lack of research work on semi-supervised sentiment analysis for education domain, motivate us to consider it as a case study in our research.

As we have mentioned in Section 1.2, **course assessment** provides essential feedback on student's learning process. Teachers can monitor their students and collect information whether they are mastering the goals and objectives or there are gaps in students' learning. These collected information helps teacher to develop or modify the teaching plan based on a student's achievement of curricular goals.

With the recent growth of using social media in education, in particular microblogging environments such as Twitter, there is a corresponding increase in the amount of data within these social media that reflect what students are learning and how well they are learning. Students discuss with each other and with their teacher different topics and express their opinions on various aspects of these topics. The data generated out of these discussions are valuable resources for course assessments. One way to make use of this data is to organize it based on "subtopics" relation. We envision a network that consist of nodes, each node represents a topic or subtopic of a course. The nodes are linked to each other based on "subtopic" relation. Each node is associated with microblogs discussing and expressing opinions on this topic. The size of the node reflects the number of microblogs posted on the node's topic. On the other hand, the color of the node illustrates the degree of sentiment (e.g. positive or negative) of the posts on a specific topic. These kind of networks evolve over time in terms of number of topics, number of students' contributions, and polarity of opinions expressed on these topics. This envisioned network would provide a quick feedback about whether the majority of students class has mastered a specific topic and its related subtopics based on the volume and the polarity of the discussions on this topic. Consequently the teacher would be able to make decisions and to take steps in order to overcome limitations at early points of time.

Building such a network is the outcome of an ongoing project, and our work on sentiment analysis is part of this project.

Figure 1.1 shows an example of the kind of network we envision. The network in the figure represents a course about "Networking". Each node represents a topic or a subtopic in the "Networking" course content; the larger the node the more tweets discussing this topic and the color reflects the total sentiment about this topic. For example "NW Fundamental" has a dark green color which implies a strong positive sentiment about this topic.

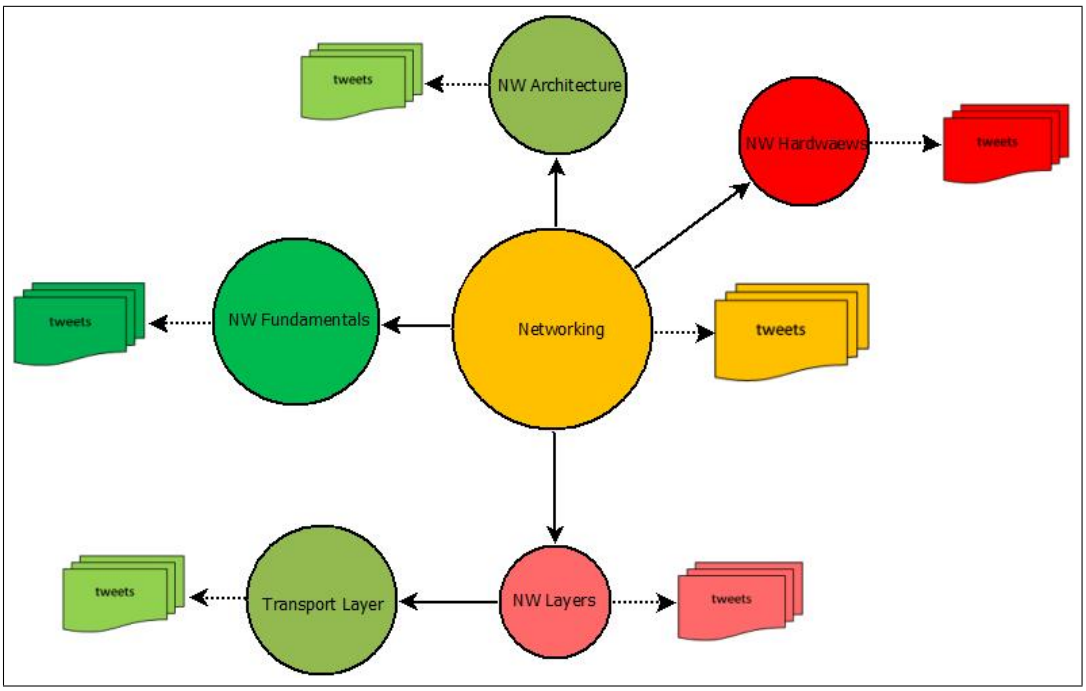


Figure 1.1: Assessment Network of Topics

In general opinion mining and sentiment analysis face the challenge of language diversity and ambiguity. Moreover, opinion mining and sentiment analysis usually rely on supervised machine learning approaches that require a large amount of manually annotated examples for training and learning the classification model, the thing makes the whole learning process very expensive and time consuming. Another challenge added by microblogs comes from the fact that the input is noisy text, short and informal, which makes the analysis of these microblogs very complicated.

1.4 Problem Statement

Given a set of microblogs, collected during learning activities for and educational course, the question we address in this research is how to automatically identify the sentiment of these posts with low model learning costs.

1.5 Objectives

Our main objective is to build a semi-supervised sentiment analysis prototype for microblogs that is able to automatically classify microblogs into positive and negative with

less costs in terms of learning requirements. **The main objective of our research includes the following sub-objectives:**

- Develop a method to collect microblogs.
- Build a preprocessing mechanism that takes into consideration the special characteristics of microblogs.
- Implement a method for subjectivity classification of microblogs.
- Develop a method for polarity (positive/negative) classification of microblogs.
- Conduct experiments in order to evaluate the precision, and the recall of the proposed prototype.
- Visualize the sentiment distribution of the automatically classified microblogs over the different topics.

1.6 Contributions

The main contributions of this research are:

- A solution is proposed for semi-supervised sentiment classification of microblogs with less costs in terms of learning requirements.
- Providing people interested in assessing the learning activities with feedbacks on the learner opinions on the different topics being discussed.
- The proposed prototype can deal with any type of online discussions data and not necessarily with microblogs or tweets.
- The proposed prototype can be extended to other domains (e.g. economic discussions) with no extra efforts.

These main contributions achieved by the following:

- Building a novel model to describe the dataset.
- Using novel features for best presentation of the dataset.
- Developing spelling correction algorithm to handle misspelled words.

1.7 Scope and Limitation

We aim to build a sentiment analysis prototype for microblogs that adheres to the following limitations and assumptions:

- The prototype handles English microblogs only.
- The extraction and disambiguations of the entity/topic of each microblog is out of the scope of this research.
- The topic of each microblog is given as an input to the prototype, and is not in the scope of our research.
- An assumption is made that each microblog includes one entity/topic.
- Sentiment analysis holds on document level.
- The data used for evaluating the prototype is a collection of microblogs posted in learning activities.
- The assessment of course based on the feedback the instructor receives from our prototype is not included in our study. User satisfaction of the prototype requires a considerable amount of time in order to be evaluated.

1.8 Thesis Structure

The thesis is structured as follows: Chapter 2 reviews some related work for researchers. Chapter 3 gives detailed explanation about our approached, and suggested model. Chapter 4 will handle the technical methods used to build and realized each part in the proposed model. Chapter 5 is about the experiments conducted in order to evaluate our approach. Finally, a conclusion and future works are provided in Chapter 6.

2

State of the Art

This is preamble chapter, that gives a glance about each topic and sub topic we handle in our research.

2.1 Opinion Mining and Sentiment Analysis in Microblogs

Two long and detailed surveys were presented by Pang and Lee [47] and Liu [33], they focused on the applications and challenges in sentiment analysis. They mentioned the techniques used to solve each problem in Sentiment Analysis. Cambria and Schuller et al. [9], Feldman [20] and Montoyo and Martý´nez-Barco [40] gave a short surveys illustrating the new trends in sentiment analysis. Tsytsarau and Palpanas [58] presented a survey which discussed the main topics of sentiment in details. For each topic they illustrated its definition, problems, development and categorized the articles with the aid of tables and graphs.

In general sentiment analysis approaches follow two main steps: 1) Data preprocessing and 2) Sentiment classification. In what follows, we present methods dealing with these two steps, proposed by research works in the context of microblogs sentiment analysis, as our research work is highly related to these works.

2.1.1 Data Preprocessing

Preprocessing simply means preparing the input data for manipulation. Preprocessing can be subdivided into two subtasks: 1) Cleaning and normalization, and 2) Features extraction [28].

2.1.1.1 Cleaning and Normalization

Tweets are informal text different from formal text such as newswire. This kind of text needs special treatment as it is full of unstructured text such as emotions, slangs and URLs, etc.. Different methods were proposed to clean and normalize such text [39, 23]. Most of preprocessing subtasks depending on tokenization and Part Of Speech (POS) tagging to identify each token in the microblog. For example, replacing all URLs with a special token for example ”twitterurl”, and replacing all Twitter usernames with ”@twitterusername”. Some other methods proposed expanding slangs and special abbreviations using special noslangs dictionaries. Other works propose converting emotions and smiley faces to its corresponding meaning text. Also emphasized words, elongated words, which contains sequences of many repeated characters are replaced with three characters, for example gooood is replaced with goood. Dealing with negative expressions has special treatment, appended ”neg” to all words from one position before a negation word to the next punctuation mark. Elimination of stop words may enhance the output results [30].

2.1.1.2 Features Extraction

Features extraction aims at splitting text into meaningful tokens. Microblogging text has its own special grammar and abbreviation. A tokenizer that able to recognize for example the hash tag, emoticons, URLs and the genre-specific terminology [23] is required. Then different combination of features can be selected to model the tweets then tested and examine the output to achieve the desired accuracy.

Part Of Speech (POS) tagging, a linguistic category of words, is also proposed for microblogs in order to recognize lexical categories such as verbs, nouns, adjectives, etc.. StanfordNLP, LingPipeNLP and OpenNLP are on-the-shelf tools that were trained on news datasets are usually used for formal English. Using such tools for microblogs proved to degrade the performance [21]. Therefore a special POS taggers were proposed to deal with this problem. For example, Olutobi et al in [46], achieved the state-of-the-art tagging results on both Twitter and IRC POS tagging tasks. Their tagging tool is called ArkTweetNLP. Applying it for tweets improved the tagging from 90% to 93% accuracy as a result for their experiments. ArkTweetNLP developed specifically for informal, online conversational text tagging that contains many non-standard lexical items and syntactic patterns based on sequential learning method Hidden Markov Models (HMM). ArkTweetNLP use unsupervised word clustering and lexical feature to improve accuracy [46].

Han et al in [23], look for statistical features, such as number of occurrence of the

Unigrams and big grams in the tweet, number of times each POS big gram appears in the tweet, stylistic features as number of times an emoticons appears in the tweet, number of words which are written in all capital letters, number of words containing characters repeated consecutively more than three times.

Saif et al in [39], were interested in other features such as word and character ngram, number of words with all-caps, number of occurrence of each POS, and number of hash tags. Using multiple sentiment lexicon, lexicon features were created for all tokens in the tweet, for each part- of-speech tag, for hashtags, and for all-caps tokens. Finally they calculate a score for each token with respect to each polarity.

2.1.2 Sentiment Classification Approaches

2.1.2.1 Supervised Learning Approaches

Zhu et al in [66], developed Twitter sentiment analysis system for message level with co-occurrence rate model feed with nine types of independent features for each tweet, using supervised method which is similar to the Naive Bayes classifier. Also discuss the value added to the result by adding or removing some features, for example the effect of adding or removing POS and the adding or removing stop words. They did an experimental study to test the contribution of each features on the F-measure value of the system. The results shows that Unigrams and stop words are the most important features.

Han et al in [23], proposed a machine learning based technique with a combination of SVM classifier and emoticons-smoothed language model to classify tweets. They got an overall accuracy ranges 41% to 64% depending on various combination of features they used. Their methodology gone as follow, first stage prepare training dataset by converting each tweet to a vector of features, then choosing the top features by using Mutual Information and 10-fold cross validation. Second stage produce initial predictions for each tweet in the development dataset. Third use the vote from the language model to give the final prediction. They investigate by their experimental study that cross validation average accuracy most affected by lexical and POS features.

Saif et al in [39], built the State-of-the-Art SVM learning classifier. SVM is proved to be effective on text categorization tasks and robust on large feature spaces. In their experiments they follow the known pipeline for sentiment analysis. First preparing the data to train the classifier, this task consist of normalization and feature extraction, the feature's vector consist of word grams, character N-grams, all-caps, POS, hashtags, negation, number of punctuation, emoticons, elongated words, and lexicon features . By

experiments it is interesting to note that classification benefited mostly from the sentiment lexicons features and attention to negations improved performance. The f-measure they achieved rang from 72% to 79% depending on various combination of features they used.

Alvaro et al in [45, 36], proposed a combination of techniques consisting of machine-learning and lexicon-based to produce a hybrid classifier. The lexicon-based approach was the only feasible option to offer labeled data and the SVM as learning classifier. This hybrid technique currently working on SentBuk application with obtained accuracy value 83.27%.

2.1.2.2 Unsupervised Learning Approaches

Unsupervised learning methods usually rely on a set of predefined POS patterns or a lexicon of sentiment words and phrases [20]. If the average sentiment orientation of these phrases and words is above specific threshold the document is classified as positive and otherwise it is a negative. The selection of theses phrases and words depends on the POS and a lexicon dictionary. The sentiment orientation of these selected phrases and words is calculated using PMI (Pointwise Mutual Information) of the phrase with two sentiment words (one for positive and one for negative). PMI(P,W) measures the statistical dependence between the phrase P and the word W based on their co-occurrence in a given corpus or over the Web (by utilizing Web search queries) [20].

Ortega et al in [44], proposed an unsupervised sentiment analysis system, the data passes through three phases: data preprocessing as normalization, feature extraction, and contextual word polarity using new contextual sentiment classification method based on coarse-grained word sense disambiguation using WordNet [38] and a coarse-grained sense inventory (sentiment inventory) built up from SentiWordNet [6]. Finally tweets are classified using rule-based classifier.

Turney in [59], presented a simple unsupervised learning algorithm for classifying a written review as input and produces a classification output as recommended or not recommended. The proposed system is not especially for tweets data, but tested for a company reviews that offer many services such as automobiles, banks, movies, and travel. Reviews also suffer from informal grammar like tweets. The reviews pass several steps to reach classification. First extract phrases containing adjectives or adverbs as pairs with specific predefined pattern, a POS needed in this step for tagging. Second estimate the semantic orientation of the extracted phrases, using the PMI-IR algorithm with two reference words POOR, EXCELLENT using search engines quires. The third calculate the average semantic orientation of the phrases in the given review and classify the review

as recommended if the average is positive and otherwise not recommended. The accuracy ranges from 84% for automobile reviews to 66%.

2.1.2.3 Semi-supervised Learning Approaches

The scarcity of labeled data in the real world applications of machine learning is one of the most obstruction from using supervised learning algorithms, as labeling is fairly expensive since it requires much human effort and time consuming. Many approaches combine the labeled and unlabeled data, where unlabeled data act as a source that push out labels through unlabeled data. As a result few labeled data can propagate labels through dense of unlabeled data, assuming that closer data points tend to have similar class labels in a manner analogous to k-Nearest-Neighbor (kNN) in traditional supervised [65, 64, 33]. This was the major motivation that led to the development of semi-supervised algorithms which learn from limited amounts of labeled data.

In content-based image retrieval (CBIR) where user can query the system by an image to retrieve similar images, the query image is the only labeled data and many unlabeled data, images, that exist in the database. Another example online web page recommender system; while a user surfing the world wide web he may find an interesting web page, labeled data, and ask the system to similar web pages, abundant unlabeled web pages in the world wide web. In the previous two examples it is hardly or impossible to ask the user to provide another labeled data [64]. There are three main paradigms for semi-supervised learning methods. First is a generative model such as Naïve Bayes classifier and the Expectation Maximization (EM) algorithms is employed to model the labels and parameters estimation. Second is the Co-training where dataset represented by two sets of independent features, then each set is used to train a classifier separately; the prediction of each classifier on unlabeled data are used to help augment the training set of the other classifier. Third is the regularization where the unlabeled data are used to regularize the learning process, for example a graph can be defined on the dataset where nodes represent the data while weighted edges encode the similarity between nodes, then labels smoothly propagate over nodes and edges [64].

The graph based label propagation algorithm is a transductive learning framework that uses few labeled data, seeds, to predict the label of a large amount of data depending on the available seeds and the relation, weighted edges, between the nodes [50]. Most recent focus on such type due to clear mathematical framework and strong performance with suitable model [35].

Adsorption is one of the most recent transductive graph based label propagation

algorithm that can perform multiclass classification that can be scaled and parallelized for large scale data [53]. Talukdar and Crammer in [53, 54] explain extensively how the algorithm setup and work and state the conditions under which it guaranteed to converge, supported with experimental evidence on various real-world datasets demonstrating the effectiveness of the algorithm.

Speriosu et al in [52], presented an approach without the need of annotated training examples. Instead they use label propagation to incorporate labels from a maximum entropy classifier trained on noisy labels and word types extracted from lexicon supported with knowledge from Twitter follower graph. Label propagation algorithms spread label distributions from a small set of nodes seeded with some initial label information called seeds throughout the graph . This classifier will use data from the domain and context which is valuable advantage of this approach. A Graph-based methods such as Modified Adsorption (MAD) algorithm is used to represent the relationships in order to classify tweets. The graph consist of nodes representing tweets, authors and features. Noisy-seed is used which is a combination of three types of seeds, Lexicon-seed created by converting each word from OpinionFinder lexicon, Emoticons-seed, and Annotated-seed from annotated tweets to 100% positive or negative. Each node is connected with an edge to group of seeds that is found in the tweet. Applying this model on different datasets they get an accuracy value from 58.1% to 62.9% without Twitter follower graph, with combination of Twitter follower graph they get better accuracy of value 71.2%.

2.2 Education and Sentiment Analysis

Sentiment analysis as a trending topics that enable all types of institutes which is seeking innovation and concern with its customer satisfaction to master and steering its future and decision making. Education and learning are ranked as the most important field for researches, to enhance the quality of graduates. This demand oblige educator to improve their teaching practice. To the best of our knowledge, there is a paucity of literature in this area [18].

Lin et al in [8], discussed the idea of Affective Computing which they defined as a ”Branch of study and development of Artificial Intelligence that deals with the design of systems and devices that can recognize, interpret, and process human emotions”.

El-Halees in [18], proposed a model to measure the performance of courses based on user-generated contents of academic institutions. The proposed model consist of two main component, feature extraction to extract all features of the course such as topics

resources, books, marks, and teachers; and a classifier to determine the attitude of the student toward the features. The experimental results for applying Naïve classifier to identify polarity concluded by calculation the F-measure equal 77.83%. The author shows also a graphical summarization plotting each feature against its opinion. Some features for example marks show a positive attitudes while books show a negative one.

Haji et al in [7], proposed a conceptual framework for an emotion detection and analysis specially for e-learning system based on the General Text and Language Engineering Infrastructure (GATE), a tool specifically developed for research purposes involving language processing software.

Thomas et al in [57], considered students as a consumer of higher education, thus their satisfaction is important for institution success. They investigated how students' characteristics and experiences affect their satisfaction. They used regression and decision tree analysis with the chi-squared automatic interaction detector (CHAID) algorithm to analyze student opinion data. They concentrated on student satisfactions such as faculty preparedness, social integration, campus services and campus facilities.

Kechaou et al in [28], investigated a supervised SVM based method with three feature selection methods MI (Mutual Information), IG (Information Gain), and CHI statistics (CHI) to pick out discriminating terms for training and classification. After calculating the F-measure for each feature, IG showed better results. The obtained F-measure values rang 72% to 80% using different features.

3

Approach and Methodology

The proposed prototype taps into online microblogs posts related to learning activities. It classifies each post into negative or positive based on the sentiment it contains. We propose an approach that includes the phases described over the next sections, (see Figure 3.1 for an overview).

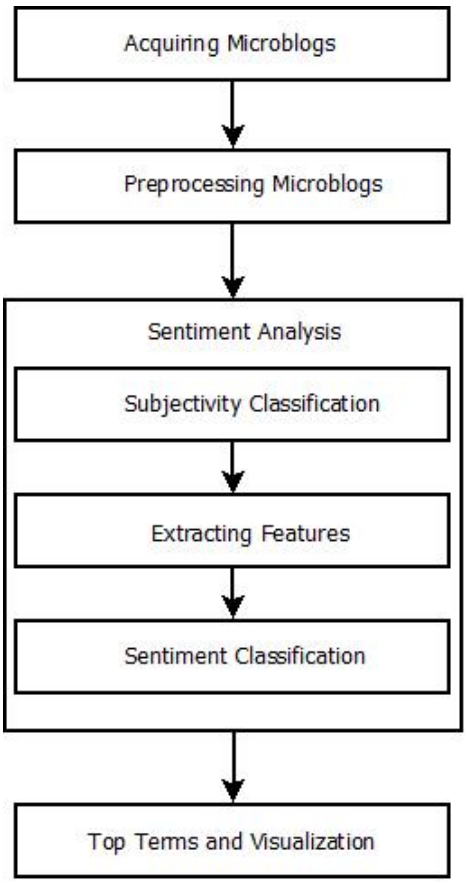


Figure 3.1: Project Framework

3.1 Acquiring and Preprocessing Microblogs

In this phase, microblogs of particular learning courses (e.g. specific accounts or hash tags) are aggregated and acquired from Twitter social network using Twitter API to receive tweets matching a search query [34].

Then preliminary phase, the preprocessing, is required before processing microblogs for further analysis. As we mentioned in Section 2.1.1. This phase includes dividing each microblog (tokenizing) into a sequence of tokens, cleaning, feature reduction, and normalization. Tweets need multiple data manipulations in different levels. Special care for each token in each tweet should be taken to get the cleanest tweet without any loss in meaning and sentiment. The preprocessing phase includes a set of sub-phases, as should be described over the next sub-sections.

3.1.1 Tokenization and Tagging

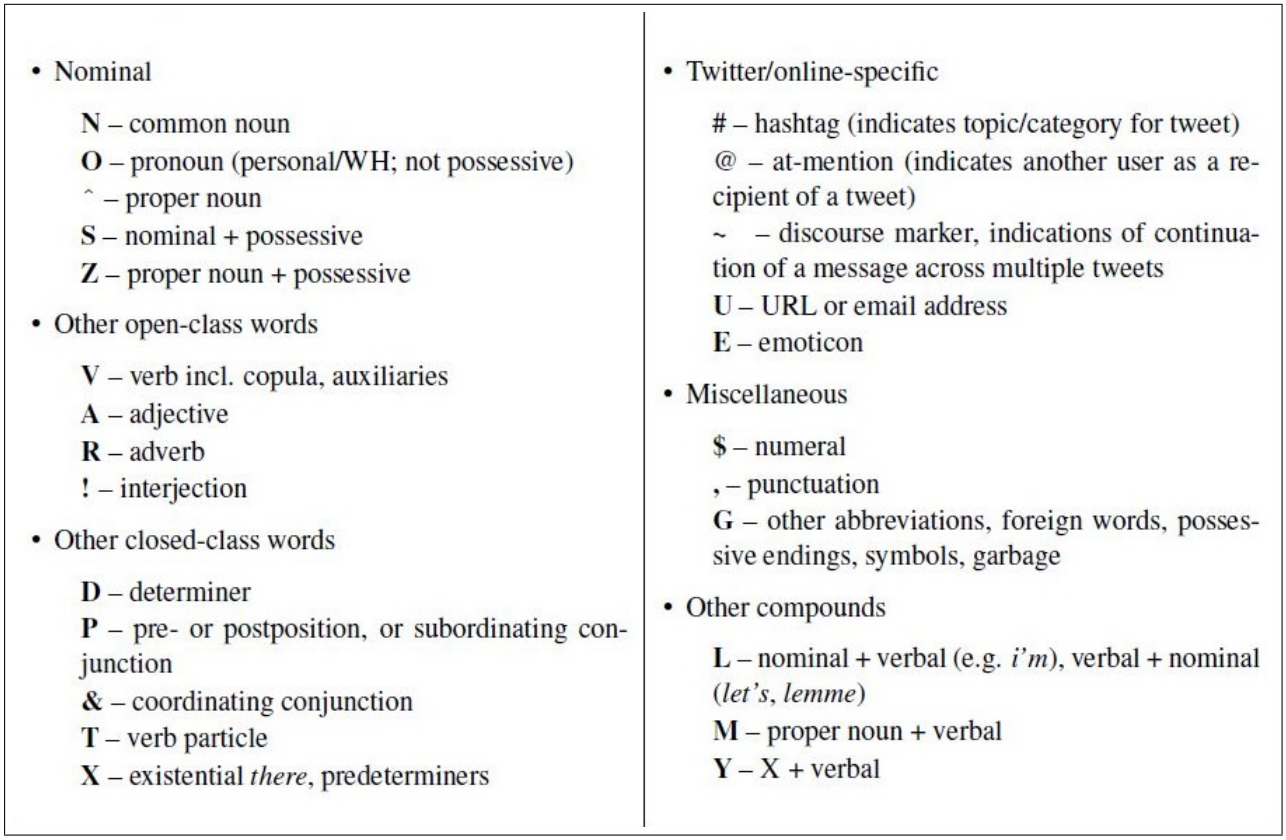
Tokenization is the process that focus on splitting a text into words, phrases, symbols, or other meaningful elements called tokens for further processing. Tagging or also called Part Of Speech annotating (POS) cannot be separated from tokenization during the preprocessing task. Tagging is the process of attaching a suffix representing a part of speech for each token based on both its definition, as well as its context; in simple words identification of tokens as nouns, verbs, adjectives, adverbs, etc.. Figure 3.2 shows the different POS annotations and their descriptions.

Microblogging text has its own special grammar and abbreviations. A tokenizer that is able to recognize each token for example the hashtag, emoticons, URLs and the genre-specific terminology is required. There are different tokenizers for tweets (e.g ArkTweetNLP) that are developed specifically for informal, online conversational text tokenization and tagging that contains many non-standard lexical items and syntactic patterns [46]. For example the tweet:

" RT @anyuser luv Palestine, looooool (: http://website.ps "

is tokenized and tagged as follow:

RT_ RT @anyuser _@ I_^ luv_V Palestine_N ,_G loooool_A (:_E http://website.ps_U



- Nominal
  - **N** – common noun
  - **O** – pronoun (personal/WH; not possessive)
  - **^** – proper noun
  - **S** – nominal + possessive
  - **Z** – proper noun + possessive
- Other open-class words
  - **V** – verb incl. copula, auxiliaries
  - **A** – adjective
  - **R** – adverb
  - **!** – interjection
- Other closed-class words
  - **D** – determiner
  - **P** – pre- or postposition, or subordinating conjunction
  - **&** – coordinating conjunction
  - **T** – verb particle
  - **X** – existential *there*, predeterminers
- Twitter/online-specific
  - **#** – hashtag (indicates topic/category for tweet)
  - **@** – at-mention (indicates another user as a recipient of a tweet)
  - **~** – discourse marker, indications of continuation of a message across multiple tweets
  - **U** – URL or email address
  - **E** – emoticon
- Miscellaneous
  - **$** – numeral
  - **,** – punctuation
  - **G** – other abbreviations, foreign words, possessive endings, symbols, garbage
- Other compounds
  - **L** – nominal + verbal (e.g. *i'm*), verbal + nominal (*let's*, *lemme*)
  - **M** – proper noun + verbal
  - **Y** – X + verbal

Figure 3.2: POS annotations [29]

3.1.2 Cleaning and Features Reduction

Tweets considered as noisy data, many tokens are need to be omitted before further processing. In our approach, we apply the following steps:

- **UserNames**: are used to direct a tweet to a specific users in Twitter. They are mentioned them in the tweet using @UserName. In our approach, we delete the UserName.
- **ReTweet**: is used to forward user other's tweet text by using ReTweet button, RT abbreviation will be add to the new tweet. All RT tagged tweets are deleted in our approach.
- **Discourse marker**: when text is continued across multiple tweets, a discourse marker (~). We simply delete these markers.
- **Numeral**: when text contains numerical values; it is annotated with ($) tag. We delete such tokens.

- **Punctuation**: when text contains punctuation; it is annotated with (,) tag. We delete such symbols.

- **Unknown abbreviations**: if a text contains unknown text such as foreign words, possessive endings, symbols and garbage; all it is annotated with (G) tag. We simply discard such text.

3.1.3 Normalization

As we mentioned before tweets are not like newswire text. Tweets are full of creative abbreviations. Normalization is the process of transforming text into canonical form as follows:

- **URL**: in our approach, if a URL appears in a tweet, it is replaced with its corresponding page title.

- **Slangs**: slangs abbreviation is the theme of micro blogs. A dictionary of these slangs [26] is used to map each slang to its corresponding text

- **Emoticons**: the easiest way to show sentiment and opinions in microblogs is using by emoticons. A dictionary of the most used emotion [5] is used to map each emoticons with its corresponding text, Table 3.1.

Table 3.1: Emoticons [5]

emoticons	Word	emoticons	Word
:/	annoyed	;D	wink
:'(	crying	:-)	happy
>:o	surprise	@_@	amazed
(:	sad	:P	cheeky
:)	happy	8D	laughing
>.<	annoyed	>:(	evil
XD	laughing	:D	laughing
-__-	sleeping	=|	angry
-_-	sleeping	:o	surprise
o.O	surprise		

- **Elongated words**: elongated word contains sequences of many repeated characters. When someone needs to focus on certain feelings it is obvious in micro blogs

Table 3.2: Some of Stop words [52]

a	b	c	d	e	f
g	h	i	j	k	l
m	n	o	p	q	r
s	t	u	v	w	x
y	z	in	of	their	there
an	be	into	on	then	was
and	but	is	or	these	with
are	by	it	such	they	will
as	for	no	that	this	
at	if	not	the	to	

to repeat its letters, for example when one need to express how much he love something he can wrote like looooooove. Keeping such repeated words will affect the character grams features so this will be normalized to three letters to keep the effect of elongated words in the same time not affecting character grams features.

- **Lowercasing letters**: all tweets converted to lowercase to not affect string matching while searching for shared features.

- **Stop Words**: a predefined list of stop words are used to identify their existence in the tweets. In our experiments we studied their effects on the results accuracy, Table 3.2.

3.1.4 Spelling Correction

Tweets may contain incorrect, misspelled or mistyped, words (see Figure 3.3 for sample tweets). If these words remain as they are, this will adversely affect post processing such as subjectivity classification which assumes that all words are correct and fine typed. So we try to correct such words depending on POS pattern matching, see Algorithm 3.1.1.

Figure 3.3: Sample tweets containing misspelled words

Algorithm 3.1.1: SPELLING CORRECTION($suggestionList, backgroundDataset, tweet$)

$suggestionList \leftarrow spellCheck(tweet)$
$generatPOSList \leftarrow POS(tweet)$
for $i \leftarrow 0$ **to** $suggestionList - 1$
  do
$found \leftarrow searchBackgroundDataset(suggestionList[i])$
$suggPOSList \leftarrow POS(BackgroundDataset(found))$
for each $s \in generatPOSList$
  do
for each $s1 \in suggPOSList$
  do
if $s = s1$
  then $counter[i]++$
$correctWord \leftarrow suggestionList[max\ counter[i]]$
return $(correctWord)$

Briefly, the proposed algorithm depends on assumption that the misspelled word has suggestions in a lexicon dictionary. The candidate correct word that has the highest number of hits in a background dataset and which matches the POS pattern of the

misspelled word is selected to replace the misspelled word. Figure 3.4 gives an example of how a misspelled word is corrected.

1. **Applying tokenization and POS tagging** for the tweet: "I_O luv_V orange_N"
2. **Generating token POS patterns** for the misspelled word "luv" as follow:
   (a) OV: if the word appear at the end of tweet.
   (b) OVN: if the word appear at the middle of tweet.
   (c) VN: if the word appear at the beginning of tweet.
3. **Checking spelling**: using a spell checking tool based on lexicon dictionary to check each word (token) in the tweet if it is incorrect, a list of suggested words will be returned.
4. **Searching background dataset**: a newswire dataset that will be used to find how often each suggested word, returned from spell checking, appears in the dataset. In other words, what will be the POS patterns for each suggested word.
5. **Choosing Candidate word**: the word with highest number of hits matching the POS pattern of the misspelled is selected.
6. **Replacing the misspelled spelled token** in the tweet with the selected word.

Figure 3.4: A tweet example with misspelled word "luv": "I luv orange"

3.2 Sentiment Analysis

In this phase the processed tweets are classified into two groups; positive tweets and negative tweets. To achieve this we propose a graph-based semi-supervised approach based on label propagation [53, 65]. Such a method needs only very few labeled examples for training the classifier. Our main concern in this research is to study the microblogs features and their graph model that best reflect the similarity relations among these microblogs. We give in what follows an overview of the label propagation problem setup.

Given fully connected graph $G = (V, E, W)$, where node $v \in V$, an edge $e = (v_1, v_2) \in V \times V$ indicates that the label of the two vertices $v1, v2 \in V$ should be similar and the

weight W_{v_1,v_2} reflects the strength of this similarity. Let $(v_1,l_1)...(v_i,l_i)$ be labeled data, where $L_i = l_1,...l_i$ are the class labels. We assume the number of classes is known, and all classes are presented in the labeled data. Let $(v_{i+1},l_{i+1})...(v_{i+u},l_{i+u})$ be unlabeled data where $L_u = l_{i+1},...l_{i+u}$are unobserved; usually $l \ll u$. The problem is to estimate L_u from V andL_i. Intuitively, we want data points that are close to have similar labels. The edge between any nodes v_1, v_2 is weighted so that the closer the nodes are in local Euclidean distance, the larger the weight W_{v_1,v_2}. In nutshell, the algorithm lets the labels of a node propagate to all nodes through the edges. Larger edge weights allow labels to travel through more easily [53, 65].

Figure 3.5 shows a simple graph with two classes positive and negative of connected tweets, each tweet presented as a node and the similarity between nodes are determined by the connecting weighted edges; the closer the nodes the larger the weight. Few of the graph's nodes are labeled known as seeds (green for positive and red for negative), and the rest are for nodes with unknown labels. By using iterative label propagation algorithm the unknown labels will be determined depending on the propagation of label through edges. This phase has three main sub phases as described in the following subsections.

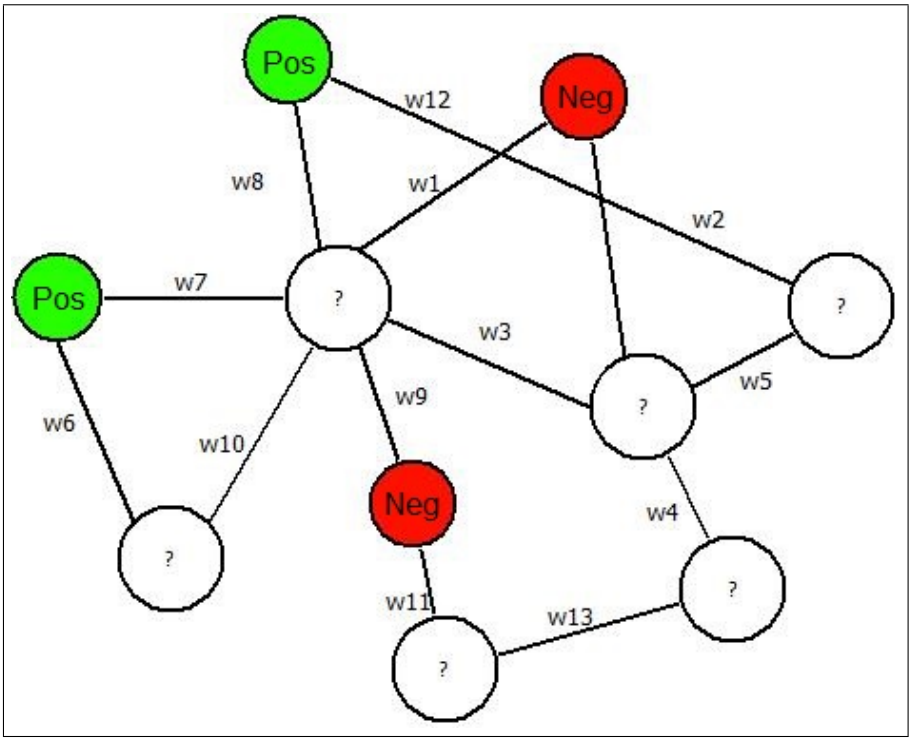


Figure 3.5: Graph-based label propagation example

3.2.1 Subjectivity Classification

This sub phase is required to filter out all objective and neutral microblogs. To do this, we use a simple approach based on lexicon dictionary. Each microblog is examined against the lexicon dictionary to determine whether it contains any sentiment or not. In other words we are searching for subjectivity rather than sentiment polarity. The main goal of this sub phase is to determine if the score returned from the lexicon is a non zero value, as this approach is not domain dependent. The approach we use is shown in Figure 3.6 and described in details in [31]. Briefly each preprocessed tweet is reconstructed after stripping of stop words and hashtags, then passing it to the lexicon. The returned total score will be the sum of all scores for each token in the tweet [31]. As this is a simple method to eliminate the neutral microblogs, it has limitations such as lexicon dictionary usually is a general list for the sentiment of words, it is not optimized for determining the polarity of certain domain [63]. Another inaccuracies seen on lexicon dictionary, that scores may be caused by the reliance on glosses as a source of information for determining term sentiment orientation. [41]

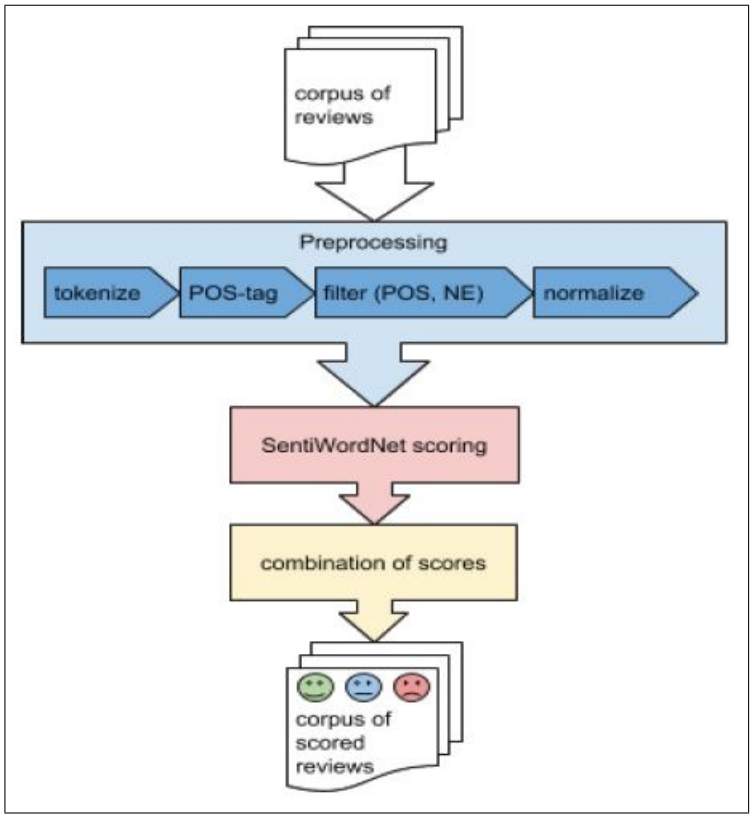


Figure 3.6: Flowchart for opinion mining process applied on reviews using SentiWordNet [31]

3.2.2 Extracting Features

In this sub phase, different kinds of features (e.g. Unigrams, N-char grams, hash tags, POS) are extracted from tweets and then are filtered in order to fed them to the next phase for further processing. Different combinations of features will be selected and examined to identify the best set that are used as a graph model describing the system.

As we consider the semi-supervises approach using label propagation algorithm specially Modified Adsorption algorithm (MAD) [52], we should give attention to two important components; first the graph nodes which is called node features, second the edges connecting these nodes which is called edge features.

3.2.2.1 Node features

The first component for the semi-supervised algorithm we use, Modified Adsorption (MAD), is a representation of data as graph nodes. Each node in the graph model represents a tweet as a vector of features, this representation helps in data regularity. Vector of features fully reflects the main attributes of the tweet and helps in structuring data. Figure 3.7 part (a) shows a sample of vector of features representing a tweet consisting of tokens (Unigrams), N-Char grams, and hashtags; while part (b) shows another representation with POS added as a postfix to each token.

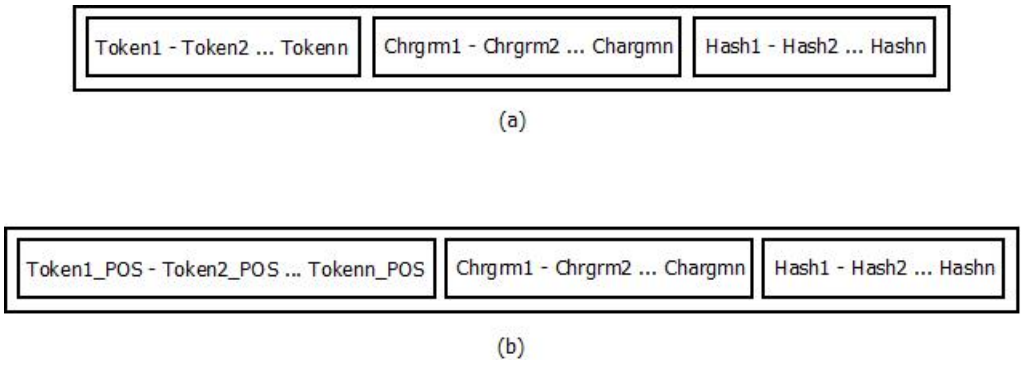


Figure 3.7: Vector of features for a tweet

For example the tweet **"I Like Palestine #MyHome"** can have two representations as shown in part (a) and (b) of Figure 3.8.

3.2.2.2 Edge features

Here we need to focus on how to calculate the weight of edges between the graph nodes. These weights reflect in indirect way the influence of the polarity of one node when it

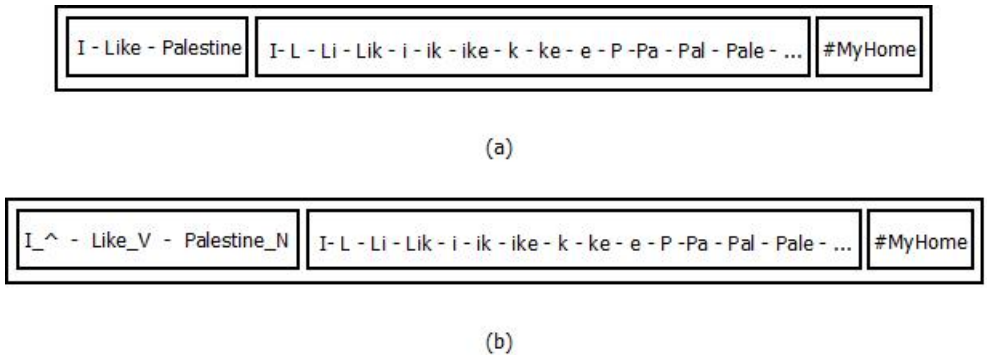


Figure 3.8: A vector of features for a sample tweet

propagates through the edges. The larger the weight of the edge the smoother the polarity is transferred through it.

Through our series of experiments we try to find a good formula to calculate this weight. We propose two formulas:

- **Similarity measure**: first formula depends simply on the similarity between nodes using Simple Matching Coefficient (SMC) [42] which is a statistic used for comparing the similarity and diversity, Equation 3.1. As each node represents a tweet with all of its features. How many shared features between nodes will result in noticeable matching between these two tweets, nodes, thus a large weight value for the edge between them.

$$SMC = \frac{Number\ of\ matching\ attributes}{Number\ of\ attributes} \tag{3.1}$$

Figure 3.9 shows an example of two tweets connected to each other as there are shared features (tok1-tokn2), (hash1), (chrgm1- chrgm2). Based on similarity measure using SMC the resemblance between two tweets is calculated as follow:

$$w_{1-2} = \frac{No.\ of\ shared\ features}{Tweet1\ total\ features} \tag{3.2}$$

$$w_{2-1} = \frac{No.\ of\ shared\ features}{Tweet2\ total\ features} \tag{3.3}$$

The value of w_{1-2} reflect the similarity of$tweet1$ for $tweet2$, while the value of w_{2-1} reflect the similarity of $tweet2$ for $tweet1$.

Since the graph model we use considers only undirected graph, the weight of

an edge between two tweets, $tweet1$ and $tweet2$, is calculated as given in the following Equation 3.4.

$$w(tweet1, tweet2) = Average(w_{1-2}, w_{2-1}) \tag{3.4}$$

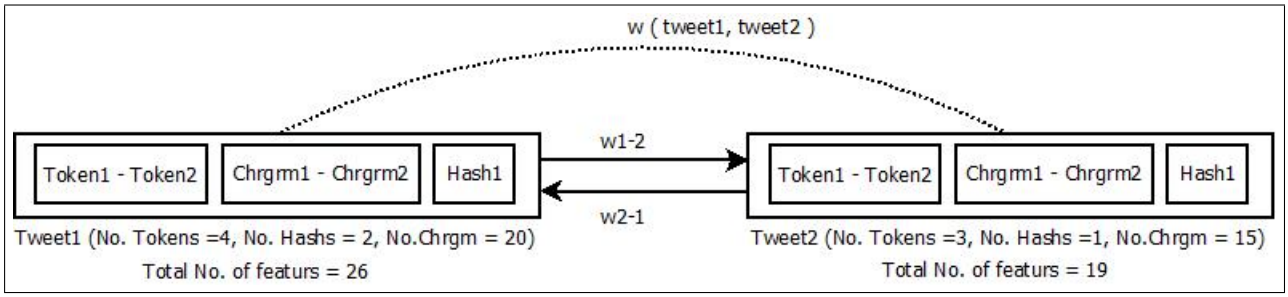


Figure 3.9: Weight based on similarity measure between tweets

- **Proximity measure**: the second formula depends on the concept of distance between nodes using Jaccard coefficient for asymmetric binary variables [42]. Distance represents how far a node from each other, but in our case we need to know how close a node to each other; so the weight between two nodes is the inverse of the distance between nodes. Start by calculating the distance by spreading tweet's features through a contingency table for binary data as shown in Figure 3.10 as an example, then calculate the distance using Equation 3.5. At the end the weight will be the inverse of distance, Equation 3.6.

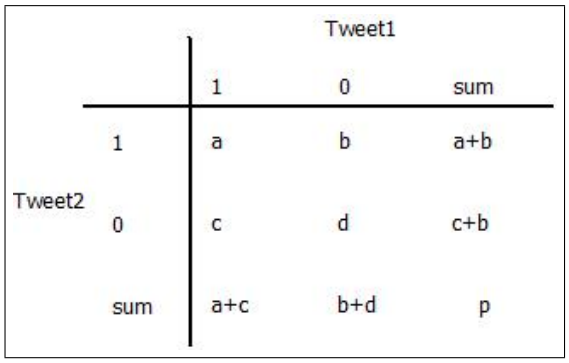


Figure 3.10: Contingency table for binary data [42]

where

a: represents the number of shared features found in both $tweet1$ and $tweet2$

b: represents the number of features found in $tweet2$ and not found in $tweet1$

c: represents the number of features found in $tweet1$ and not found in $tweet2$

$$distance(tweet1, tweet2) = \frac{b+c}{a+b+c} \tag{3.5}$$

$$w(tweet1, tweet2) = 1 - distance(tweet1, tweet2) \tag{3.6}$$

3.2.3 Sentiment Classification

As we consider the semi-supervise approach using label propagation algorithms specially Modified Adsorption algorithm (MAD), there are two important factors that need to be considered while preparing the graph model. First we need to design a graph. We already gave full details about it in Section 3.2.2. Second we need to identify the seeds. Seeds are labeled nodes with a predefined polarity and distributed among other graph nodes. We consider the following variants for seeding the graph:

- **Unigrams seeds**: these seeds represent the most negative and positive tokens found in the dataset. This means that during the subjectivity analysis token, in the tweet except the hash tags tokens is examined against lexicon dictionary to know its polarity. The weight of the edge between any unlabeled nodes and a seed will be 1.0 as this seed is directly found in this tweet.
- **Big grams seeds**: these seeds is about a subset of tweets collected randomly after the subjectivity task applied to the dataset. The weight of the edge between any unlabeled node and a seed will be calculated as described above in Equation 3.4 and Equation 3.6.

Figure 3.11 shows the overall graphical model with all features and seeds. Each tweet represents a node feature in the graph. Shaded nodes are the big gram seeds. SentiWordNet feeds the graph with Unigrams seeds. The dashed edges between nodes features represent the edges features. The solid edges without arrows has the value of 1.0 as it represents the occurrence of the Unigrams seed in the node.

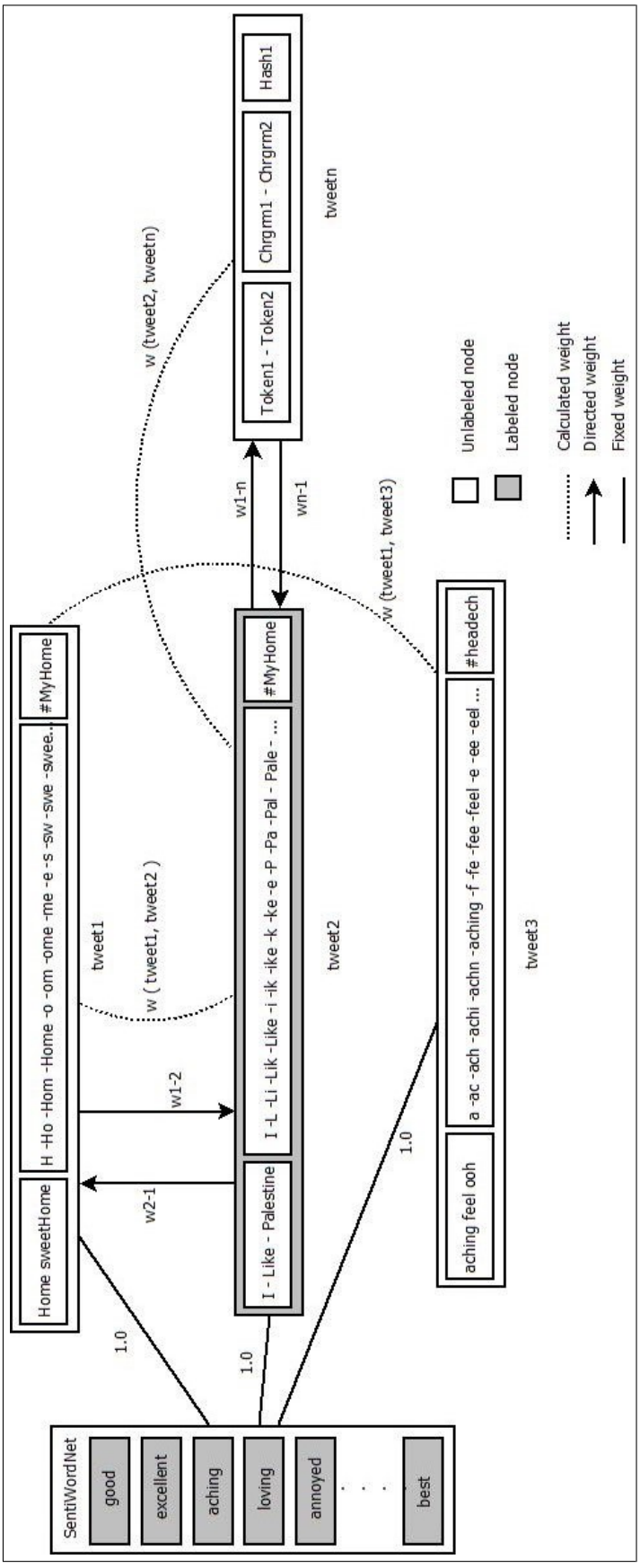


Figure 3.11: The proposed graphical model of our approach

3.3 Top Terms and Visualization

We follow a simple method to identify the top terms of a set of tweets. As we mentioned before, we assume that the topic of each tweet is known in advanced. All tweets that belongs to one topic are aggregated in two groups: positive tweets and negative tweets. We then compute the Term Frequency - Inverse Frequency (TFIDF) [49] of each word in the given group of tweets which represent one topic. The measurement reflect the importance of a word in a group. So the higher the TFIDF value of a word, the more important it is. The top five words in each group are selected. We select the top five words in the positive group and the top five words in the negative group for each topic.

For visualization, we use a simple graphical model. Each node in this model represents one topic. The size of a node reflects the number of tweets that belongs to that topic. The color of the node reflects the overall sentiment on this topic. Digital color can be represented in a different number of ways. The most common way to represent color is via a 6-digit HEX number. Hex is a 6-digit, 24 bit, hexidecimal number that represents Red, Green, and Blue. An example of a Hex color representation is #123456, 12 is Red, 34 is Green, and 56 is Blue. We assign a color to each node based on the following formula.

$$HexColor = \#(TotalNeg,\ TotalPos,\ Zero) \tag{3.7}$$

where:

TotalNeg: total of negative tweets, represents the Red component of the hex color.

TotalPos: total of positive tweets, represents the Green component of the hex color.

Zero : represents the Blue component of the hex color.

4

System Technical Implementation

This chapter explores the technical side of how we accomplish our proposed system and gives preface about the tools and packages we used. Each part of our prototype is programmed and implemented using Java programming language. Minute details about how we implement each specific part in the following subsections.

4.1 Hardware and Software Specifications

In the following subsection a brief details about the hardware specification we used in our experiments, in addition to the software tools and packages used while implementing our sentiment analysis system.

4.1.1 Hardware Specifications

The machine specification we used is a server with two Intel Xeon E5-2650 2.294 GHz processors, 8 cores, with 64 GB physical memory. Supported with a hard disk with 4 TB and Ethernet card with 1 Gbps. To access the server remotely we use VSphere client, which is a tool to configure a host and to operate its virtual machines.

4.1.2 Software Specifications

4.1.2.1 Java and eclipse IDE

Eclipse [1] is an open source integrated development environment (IDE), written mostly in Java, used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in many other programming languages not just Java. Java [2] is an object-oriented functional computer programming language that enable

programmers to develop their applications and not giving concerns on which platform the application will run. As it compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

4.1.2.2 Oracle VM VirtualBox

VirtualBox [3] is freely available as open source cross-platform virtualization software, that enable developer to run multiple guest operating systems on their machine.

4.1.2.3 ArkTweetNLP

One of the most fundamental parts of any linguistic pipeline is part-of-speech (POS) tagging, in order to recognize lexical categories such as verbs, nouns, adjectives, etc... Most POS taggers are trained from treebanks in the newswire domain for formal English. Using such tools for online microblogs will degrade the performance [21]. ArkTweetNLP [5] developed specifically for informal online conversation, is an open source Java software library that provide fast and robust tokenizer and part-of-speech tagger for tweets, its training data of manually labeled POS annotated tweets [29]. ArkTweetNLP considered as the state-of-the-art in POS tagging for both Twitter and Internet Relay Chat (IRC) text [46]. An annotation guideline and how the tokenizer and tagger work is available online [29].

4.1.2.4 SentiWordNet

SentiWordNet [6] is a lexical resource publicly available for research purposes, it is an extension for WordNet [38] the ”dictionary of meaning” combining the functions of dictionaries and thesauruses. SentiWordNet is a freely lexical resource for opinion mining as it associate to each synset of WordNet three sentiment connotation: positive, negative, and objective. The advantage of using synsets instead of terms is to offer different sentiment scores for each sense of one word, because the connotations can differ in one word depending on the sens. Each connotation represented by a score ranged from 0.0 to 1.0 for each synset. This means that the synset may have non zero scores for all the three categories which will indicate the corresponding terms value. See Table 4.1 for an example for synset ”estimable”. [31, 19, 6].

Table 4.1: SentiWordNet scoring example [6]

#	Synset	Sensee	Positive	Negative	Objective
1	estimable	”may be computed or estimated”	0.0	0.0	1.0
2	estimable	”deserving of respect and high regards”	0.75	0.0	0.25

4.1.2.5 JUNTO

The JUNTO Label Propagation Toolkit [48], is a package that provides an implementation for several state-of-the-art label propagation algorithms, Adsorption and Modified Adsorption (MAD) algorithms that described in [53, 54, 56]. Label propagation works on nodes where connected and influence each other based on their connections and the weight of these links.

4.1.2.6 JUNG (Java Universal Network/Graph Framework)

It is an open source Java software library that provides API for modeling, analysis, and visualization of data that can be represented as a graph or network. It is designed to support a variety of representations of entities and their relations, such as directed and undirected graphs, hypergraphs etc. It uses metadata for annotating graphs, entities, and relations. JUNG also provides a visualization framework that makes it easy to interact and explore network data [43, 27].

4.1.2.7 LUCENE

It is an open source Java software library for indexing and searching, it stores each piece of data as a document where document is essentially a collection of fields. LUCENE provides a dynamic document index and supports with highly expressive search API to index and retrieving documents from the index [11].

4.1.2.8 Jazzy (The Java open source spell checker)

It is an open source Java software library that provides API for spelling checking functionality [25].

4.2 Framework Implementation

As we gave details about our proposed approach in Chapter 3 and details about hardware and software specification used in implementing our proposed system in Section 4.1. We

want to put all of them together. All software development done by using Eclipse which is an integrated development environment (IDE). Eclipse includes full support for the Java Platform Standard Edition Version 7, and also supported Java runtime environment JRE7. As JUNTO Toolkit running under Unix operating system, we need a virtualization software such as Oracle VM VirtualBox to install Linux Red Hat. After preparing the appropriate environment we follow the instructions with JUNTO Toolkit to build and run it. Now the development environment is ready with Linux Red Hat as an operating system, Java Eclipse to develop our system components, and JUNTO Toolkit to run MAD algorithm.

4.2.1 Preprocessing Microblogs

In this subtask different packages were used, ArkTweetNLP, Java.net, LUCENE, and Jazzy. There are three main implementation for this subtask:

1. **Tokenize and tag text**: the first step in preprocessing task is to tokenize and tag tweets. This is accomplished by the Java open source ArkTweetNLP [5]. It contains *Twokenize* class, a tokenizer designed for English Twitter text and some other European languages. *Twokenize* class takes text as an input and return a String list of tokens. *Tagger* class is used to map each token in the list to its appropriate tag, Tagger is supported with Twitter POS model with 25-tag tagset [29]. So *Tagger* class takes the list of String tokens as an input and returns a list of *TaggedToken*, pairs of token and its tag.

2. **Dealing with URLs**: Java.net package is used to handle URL mapping from a URL to web page title.

3. **Spelling Correction**: for this task, our work is based on the following sub tasks, with the reference to Algorithm 3.1.1:

    - **Check spelling**: using Jazzy free tool (Java Spell Checker), a pure Java library implementing a spell checking algorithm. This tool checks each word (token) if it is wrong spelled using *SpellChecker* and *SpellDictionary* classes. A list of suggested words will be returned otherwise return zero. Applying it for tokens with length greater than one, has no hashtag POS, and not starting with a capital letters or all caps or containing non literal.
    - **Search a background dataset**: a 20Newsgroups training dataset, newswire dataset, that will be used as a background to find how often each suggested

word, returned from spell checking, appeared in the dataset depending on the POS pattern for each suggested word. Indexing the dataset using LUCENE Java package for indexing and searching, for easily storing and searching the dataset. LUCENE stores each line of the dataset as a document in the index. This is done using set of classes such as *IndexWriter, IndexReader, Analyzer, Document, and TopScoreDocCollector.*

- **Choose the Candidate word**: 1) Search the indexed dataset for suggested words using *IndexSearcher* class and *QueryParser* to compose the search query. 2) The search results will be returned as a list of documents, each *Document* object contains a suggested word, then POS the documents. 3) Extract the POS patterns for each suggested words as appeared in the search results. 4) Compare the extracted patterns with the tweet's patterns. 5) Return the suggested word with max matches, if more than one just pick the first one. 6) Replacing the wrong spelled token in the tweet with the returned candidate suggested word.

4.2.2 Sentiment Analysis

4.2.2.1 Subjectivity Classification

After preprocessing we have a free spelling errors, no slangs or creative phrases, no emoticons. All tweets are now ready to be examined against SentiWordNet [6] to decide whether it contains a sentiment or not. This task accomplished by reconstructing the tweet after removing all hashtags, as it may affect the sentiment of the tweet. Tweets with SentiWordNet score equal to zero are removed as they are considered as neutral.

4.2.2.2 Extracting Features

As we mentioned before in Section 3.2.2 many types of features must be extracted to represent the tweet. Each feature can be used then to look for a link or a relation with other tweets. So after Preprocessing subtask finished we take each tweet and extract its features using *Twokenize*, *Tagg*er, and *TaggedToken* classes supported by ArkTweetNLP [5]. Each tweet represented as a vector of features consisting of:

- Tokens (Unigrams)
- Tokens with POS

- Token's N-Char grams
- Hash tags

Each Unigrams tested against SentiWordNet [6] to get its polarity score as if it is greater than +0.5 it considered as strong positive or less than -0.5 as a strong negative. All these strong negative and positive tokens in all the dataset are collected to feed the label propagation algorithm as node features or labeled nodes that will influence the other unlabeled nodes.

Another type of features, is the edge feature, and how to calculate the edge, link, weight between the graph nodes that will affect the movement and transition of the polarity between all nodes. All calculations based on equations in Section 3.2.2.

4.2.2.3 Sentiment Classification

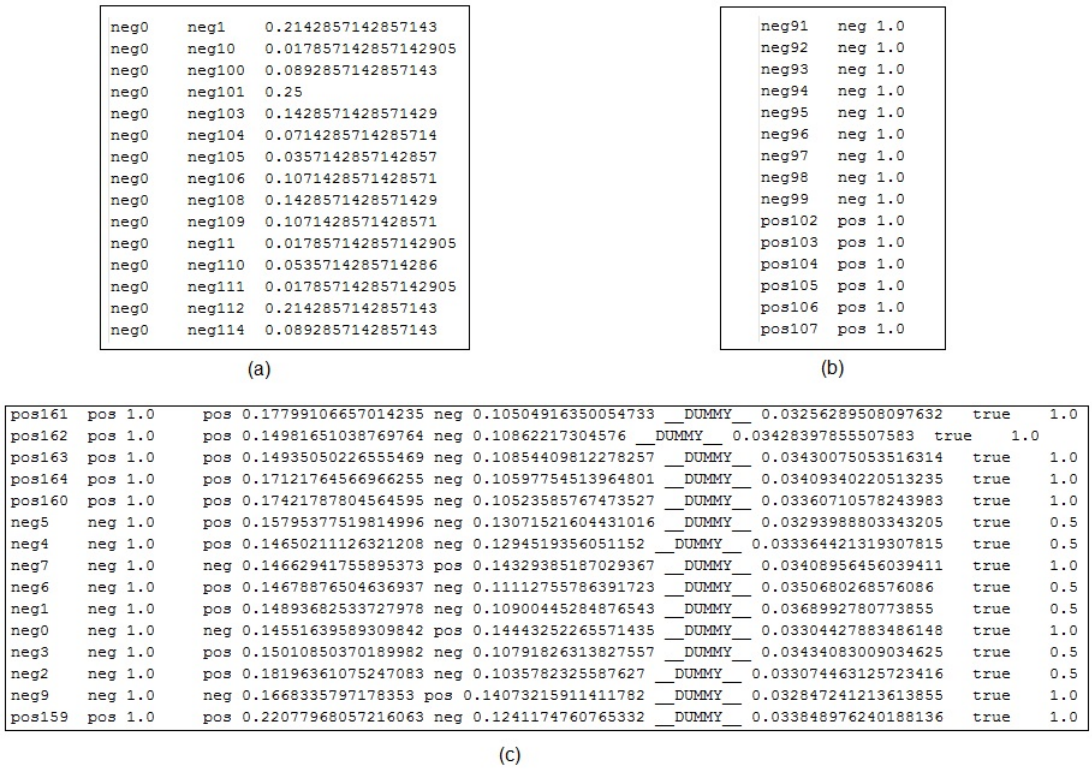
Before starting sentiment classification using JUNTO toolkit [48] some files must be generated and structured in certain way as an input to JUNTO toolkit. First file is the *input_graph* file which contains the names of all nodes in the graph and all its edges with other nodes represented by the value of calculated weight. The second is *seeds* file will represent the list of seeds, labeled nodes. Then we are ready now to run the JUNTO toolkit to obtain the *label_prop_output* file that contains the results to be analyzed. Figure 4.1 shows a snapshot for each file, part (a) represents a sample for the *input_graph* file, part (b) represents a sample for *seeds* file, and part (c) represents a sample for *label_prop_output* file.

4.2.3 Top Terms and Visualization

4.2.3.1 Top Terms

A simple technique used to select tweets that have the top five words with top five TFIDF value in the dataset. We need a tool for optimal indexing and searching dataset to facilitate the process of calculating TFIDF depending on equations described in Section 5.2.0.8. Lucene Java library [11] is used for this purpose as it implements many APIs for indexing, searching, and processing data. The full process we implemented is as follows:

- **Index each tweet** in the dataset as a separate document using *Document* and *IndexWriter* classes. Each document is composed of two fields, one for the path to the text file that store the tweet content and the other for the tweet text.



```
neg0    neg1    0.2142857142857143
neg0    neg10   0.017857142857142905
neg0    neg100  0.0892857142857143
neg0    neg101  0.25
neg0    neg103  0.1428571428571429
neg0    neg104  0.0714285714285714
neg0    neg105  0.0357142857142857
neg0    neg106  0.1071428571428571
neg0    neg108  0.1428571428571429
neg0    neg109  0.1071428571428571
neg0    neg11   0.017857142857142905
neg0    neg110  0.0535714285714286
neg0    neg111  0.017857142857142905
neg0    neg112  0.2142857142857143
neg0    neg114  0.0892857142857143
```

(a)

```
neg91   neg 1.0
neg92   neg 1.0
neg93   neg 1.0
neg94   neg 1.0
neg95   neg 1.0
neg96   neg 1.0
neg97   neg 1.0
neg98   neg 1.0
neg99   neg 1.0
pos102  pos 1.0
pos103  pos 1.0
pos104  pos 1.0
pos105  pos 1.0
pos106  pos 1.0
pos107  pos 1.0
```

(b)

```
pos161  pos 1.0   pos 0.17799106657014235 neg 0.10504916350054733 __DUMMY__ 0.03256289508097632  true   1.0
pos162  pos 1.0   pos 0.14981651038769764 neg 0.10862217304576 __DUMMY__ 0.03428397855507583  true   1.0
pos163  pos 1.0   pos 0.14935050226555469 neg 0.10854409812278257 __DUMMY__ 0.03430075053516314  true   1.0
pos164  pos 1.0   pos 0.17121764566966255 neg 0.10597754513964801 __DUMMY__ 0.03409340220513235  true   1.0
pos160  pos 1.0   pos 0.17421787804564595 neg 0.10523585767473527 __DUMMY__ 0.03360710578243983  true   1.0
neg5    neg 1.0   pos 0.15795377519814996 neg 0.13071521604431016 __DUMMY__ 0.03293988803343205  true   0.5
neg4    neg 1.0   pos 0.14650211126321208 neg 0.1294519356051152 __DUMMY__ 0.033364421319307815  true   0.5
neg7    neg 1.0   neg 0.14662941755895373 pos 0.14329385187029367 __DUMMY__ 0.03408956456039411  true   1.0
neg6    neg 1.0   pos 0.14678876504636937 neg 0.11112755786391723 __DUMMY__ 0.0350680268576086   true   0.5
neg1    neg 1.0   pos 0.14893682533727978 neg 0.10900445284876543 __DUMMY__ 0.0368992780773855   true   0.5
neg0    neg 1.0   neg 0.14551639589309842 pos 0.14443252265571435 __DUMMY__ 0.03304427883486148  true   1.0
neg3    neg 1.0   pos 0.15010850370189982 neg 0.10791826313827557 __DUMMY__ 0.03434083009034625  true   0.5
neg2    neg 1.0   pos 0.18196361075247083 neg 0.1035782325587627 __DUMMY__ 0.033074463125723416  true   0.5
neg9    neg 1.0   neg 0.1668335797178353 pos 0.14073215911411782 __DUMMY__ 0.032847241213613855  true   1.0
pos159  pos 1.0   pos 0.22077968057216063 neg 0.1241174760765332 __DUMMY__ 0.033848976240188136  true   1.0
```

(c)

Figure 4.1: Snapshots for *input_graph*, *seeds*, and *label_prop_output* files

- **Access and search the index** by using *IndexReader* to get a list of distinct words (tokens) appear in the dataset as a prelude to start TFIDF calculations.
- **Start calculating the TFIDF** for each token in the list using many API supported by Lucene such as *TermFreqVector* class that represents a document as a list of pairs of words and its frequency in the document. Then select the top five value.
- **Compose queries** using *QueryParser* class to search the index for each top five words using *IndexSearcher* class.
- **Querying results** returns as *Hit* object that contains the resulting documents.

4.2.3.2 Visualization

For visualization a free Java packages (JUNG) [27] was used for modeling, analysis, and visualization of data that can be represented as a graph or network. First we need to initialize and load the graph. *SparseGraph* class is used to store the nodes of the graph, and the class *Node* contains all information needed about nodes. *Layout*, *VisualizationViewer*, and *Transformer* classes are used for visualization purpose.

4.2.4 Framework Demonstration

Below is a brief demonstration of how our system prototype works:

1. A user enters the course account, he or she is interested in.
2. Our prototype, collects the tweets associated with the entered course.
3. The different topics of the course is given as input to our prototype.
4. The prototype visualizes the results of the sentiment analysis of the collected tweets (see Figure 4.2 for a screen shot of our prototype). The output shows colored circles. Each circle is associated with one topic. The size of the circle reflects the amount of discussion around this topic. The color of the circle is gradient color between red and green which reflects the overall sentiment of that topic.
5. A user can click on a circle, then a window appears that shows two lists; one for positive tweets (green), and one for negative tweets (red), but both are for tweets on the same topic, Figure 4.3.
6. Also another window appears showing the most important words in both lists; red colored words are extracted from negative tweets, and green colored words are extracted from positive tweets, Figure 4.4.

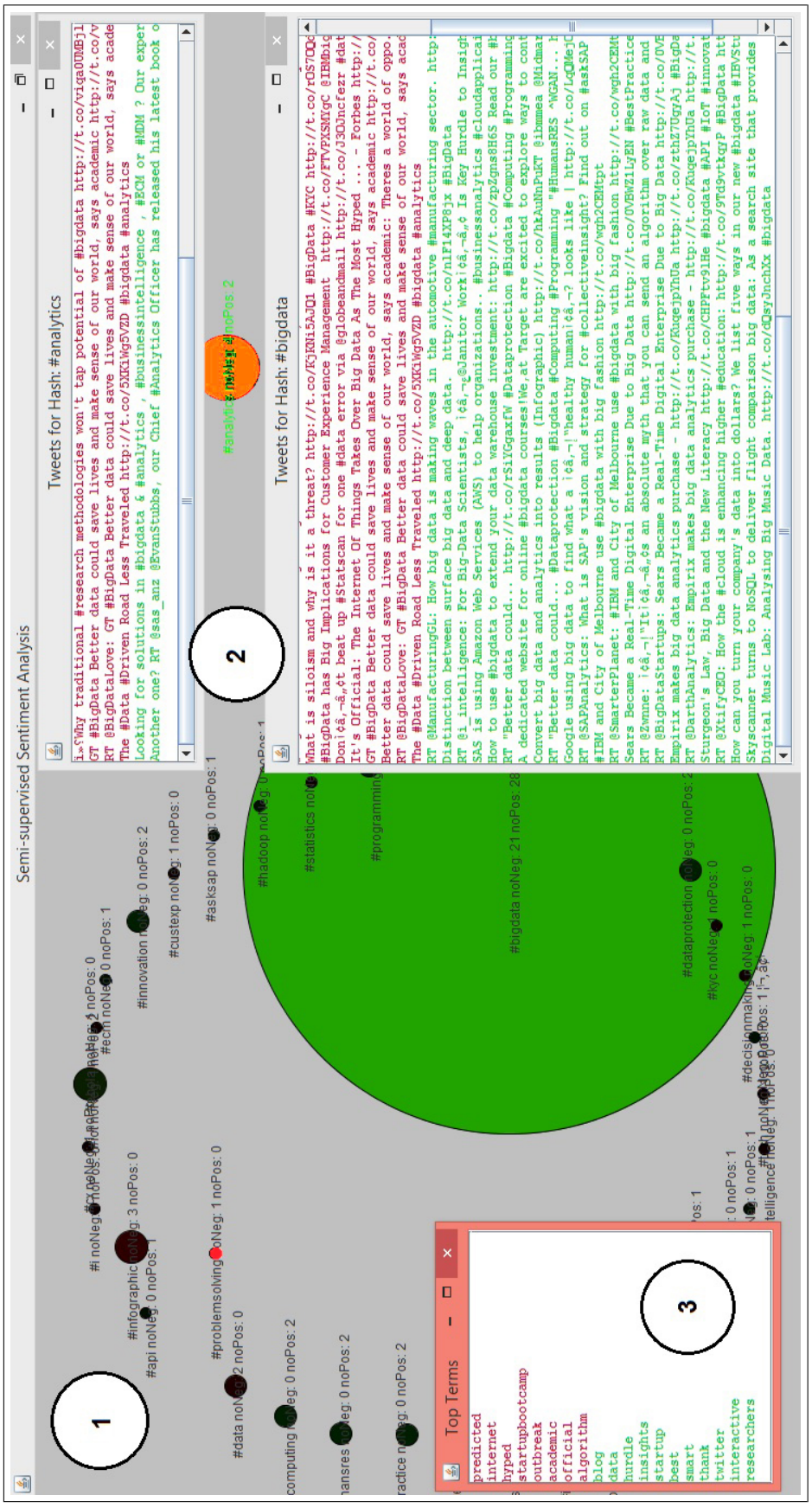


Figure 4.2: Screen shot of our prototype

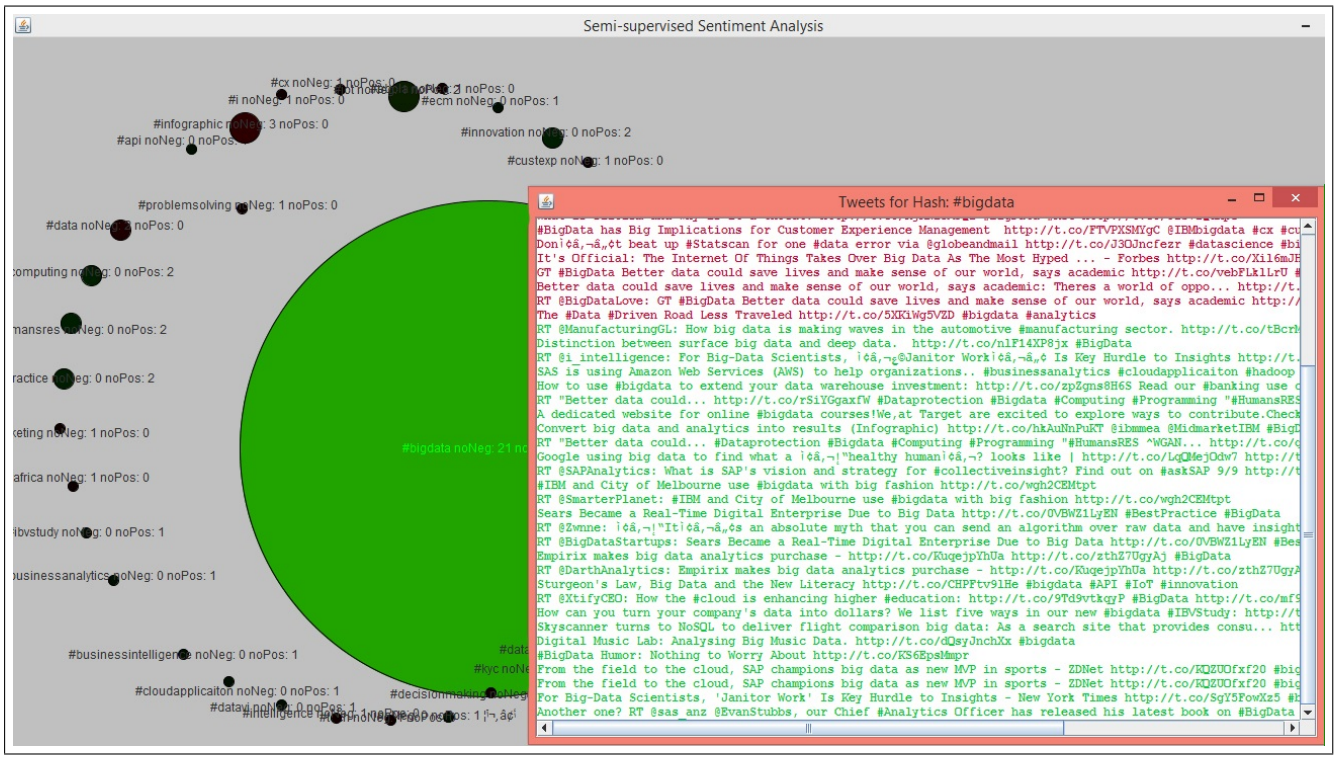


Figure 4.3: Screen shot of tweets belonged to one topic

Figure 4.4: Screen shot of Top terms belonged to one topic

5

System Experiments and Evaluation

In this chapter, we evaluate the proposed approach we introduced for tweets sentiment analysis. Several datasets and measurements are used to show the performance of the system by conducting several comparative experiments.

5.1 Datasets

5.1.0.1 Sentiment140 Dataset

Alec et al [22] as they did sentiment analysis using supervised approach they need a training and testing datasets; the training dataset is collected programmatically using Twitter API with parameters of English as a language, frequency of polling of 2 minutes, time period between April 6, 2009 to June 25, 2009, and query ":(" return tweets that contain positive emoticons, and the query ":(" will return tweets with negative emoticons. After post-processing the data, we take the first 800,000 tweets with positive emoticons, and 800,000 tweets with negative emoticons, for a total of 1,600,000 training tweets. The test data is manually collected, a set of 177 negative tweets and 182 positive tweets are manually marked. Not all the test data has emoticons.

5.1.0.2 Health Care Reform (HCR) Dataset

Michael Speriosu et al [52], created a new annotated dataset based on tweets with hashtag "#hcr" from early 2010. Tweets are manually annotated to (positive, negative, neutral, irrelevant) and separated into training, development and test sets. But we restrict attention only to positive and negative tweets. The training dataset was of total 1498 tweets, and the test dataset is a set of 507 negative tweets and 157 positive tweets.

5.1.0.3 Real Data for Education Field

An automatic collection of tweets using Java application and Twitter API for tweets related to education field. We search for hashtags that used for educational purpose discussing topics and expressing opinions. We found many hashtags but we select "#bigdata", and start pulling tweets related to these hashtags. The raw real dataset collected for "#bigdata" is 1211 tweets.

5.1.0.4 20Newsgroups Training Dataset

The 20 Newsgroups data set was originally collected by Ken Lang [32]. It is a collection of approximately 20,000 newsgroup documents, organized into 20 different newsgroups, each corresponding to a different topic. It has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.

5.2 Measurements

In our experiments we used several measurements and calculations. In what follows, we describe these measurements.

5.2.0.5 Accuracy

Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined in binary classification problem. Accuracy also considered as statistical measure of how well a binary classification test correctly identifies or excludes a condition [42].

$$Accuracy = \frac{True\ cases\ results}{Total\ number\ of\ cases} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.1}$$

5.2.0.6 Precision, Recall, F-measure

Precision and recall [42] are measurements for relevance usually used in pattern recognition and information retrieval with binary classification. Precision, the positive predictive value, is the fraction of retrieved instances that are relevant, while recall, the sensitivity value, is the fraction of relevant instances that are retrieved. High precision means that an algorithm retrieved ultimately more relevant results than irrelevant, while high recall

means that the most of relevant results are retrieved [42].

$$Precision = \frac{TP}{TP+FP} \tag{5.2}$$

Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class, Equation 5.2 [42].

Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class, Equation 5.3 [42].

$$Recall = \frac{TP}{TP+FN} \tag{5.3}$$

where

TP = number of true positive

FP = number of false positive

FN = number of false negative

TN = number of true negative

F-measure is a measure of a accuracy for binary classification problem. As usually precision and recall scores are not discussed in isolation. F-measure is a combination of precision and recall, Equation 5.4, which is a geometric mean of the chance-corrected variants [42].

$$F-measure = \frac{2*Precision*Recall}{Precision+Recall} \tag{5.4}$$

5.2.0.7 Kappa Cohen's Coefficient

Cohen's Kappa measures the agreement between two raters who each classify N items into mutually exclusive categories [10]. Kappa is always less than or equal to 1.0. A value of 1.0 implies perfect agreement and values less than 1.0 imply less than perfect agreement as shown in Table 5.1 [15]. Equation 5.5 shows how to calculate Kappa Coefficient.

$$\kappa = \frac{Pr(a)-Pr(e)}{1-Pr(e)} \tag{5.5}$$

where

Pr(a) = Percentage of agreement

Pr(e) = Probability of random agreement

Table 5.1: Kappa interpretation [15]

Kappa interpretation	Range of values
Poor agreement	Less than 0.20
Fair agreement	0.20 to 0.40
Moderate agreement	0.40 to 0.60
Good agreement	0.60 to 0.80
Very good agreement	0.80 to 1.00

5.2.0.8 TFIDF

Term Frequency – Inverse Document Frequency (TFIDF) [49] is a numerical statistical value that reflect how important a word is to a document in a collection or corpus. The value of TFIDF increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus. TFIDF often used as a weighting factor in information retrieval and text mining such as search engines [49]. TFIDF for the term w can by computed as follow:

TF: Term Frequency, which refers to how frequently a term occurs in a document [49].

$$TF(w) = \frac{Number\ of\ times\ w\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \tag{5.6}$$

IDF: Inverse Document Frequency, which measures how important a term is [49].

$$IDF(w) = \log \frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ w\ in\ it} \tag{5.7}$$

TFIDF: The multiplication of the TF and IDF terms [49].

$$TFIDF(w) = TF * IDF \tag{5.8}$$

5.3 Prototype Evaluation

To evaluate the system we conducted several experiments. Details about these experiments are given in the following subsections.

5.3.1 Experiments Configuration

To configure experiment parameters, an iterative experiment over a **development dataset (DDS)** was held. The development dataset consists of 200 labeled test cases (tweets) of

Sentiment140 training dataset divided to two equal parts for positive and negative tweets. The features extracted from DDS are Unigrams, Unigrams with POS N-Char grams, hash tags, and edges feature. The seeds set was consisting of **10** big gram seeds, **5 as positive** seeds and **5 as negative** seeds. The graphical model for JUNTO was then prepared to start sentiment classification. Multiple experiments with different feature combination are conducted until we accomplished the best results at **0.82** for accuracy, **0.93** for Precision, **0.85** for Recall, and **0.89** for F-measure.

5.3.1.1 Parameters Tuning for MAD

Modified Adsorption algorithm built in JUNTO [48] has a configuration file to adjust some parameters to tune the algorithm output. The spreading of the label distributions can be viewed as a controlled random walk with three possible actions: (1) injecting a seeded node with its seed label, (2) continuing the walk from the current node to a neighboring node, and (3) abandoning the walk. MAD takes three parameters, $\mu 1$, $\mu 2$ and $\mu 3$, which control the relative importance of each of these actions respectively [52, 55].

We set the number of iterations to 100, $\mu 1$ to 1.0, $\mu 2$ to 0.001 and $\mu 3$ to 0.0001. These parameters gave the best result over the DDS.

5.3.1.2 SentiWordNet Polarity Threshold

Another experiment parameter we need to set is SentiWordNet [6] polarity orientation threshold. In our experiments we conducted over DDS; we found that the best polarity orientation threshold for positive words was at +0.5, and for negative words was at -0.5.

5.3.1.3 Experiment setup

We conducted three different experiments using different test datasets to compare our results with different sentiment classification approaches discussed in Section 2.1.2. The experiment details as described below:

1. The first experiment was carried out using Sentiment140 testing dataset to compare our results with a supervised learning approach. We denote this experiment as **Supervised Comparison Experiment**. More details about the dataset and the experiment conducted on it is given in Section 5.3.2.

2. The second experiment was carried out using HCR testing dataset to compare our results with existing a semi-supervised learning approach. We denote this

experiment as **Semi-supervised Comparison Experiment**. More details about the dataset and the experiment conducted on it is given in Section 5.3.3.

3. The third experiment was carried out using collected real educational dataset to examine our approach with a semi-supervised learning approach. We denote this experiment as **Real Data Semi-supervised Experiment**. More details about the dataset and the experiment conducted on it is given in Section 5.3.4.

Each experiment is conducted with a different combinations of features in order to evaluate their effect on sentiment classification. The experiment settings and their different features combinations are shown in Table 5.2.

Table 5.2: Features settings

Features Settings	Hash tags	N-Char gram	Unigrams	Unigrams with POS	Weight (Similarity)	Weight (Distance)	Stop words
Setting1	√	√	√			√	√
Setting2	√	√	√			√	
Setting3	√	√		√		√	√
Setting4	√	√		√		√	
Setting5	√	√	√		√		√
Setting6	√	√	√		√		
Setting7	√	√		√	√		√
Setting8	√	√		√	√		

5.3.2 Supervised Comparison Experiment

In this experiment we aim to compare our approach with a supervised learning approach for sentiment analysis described in [22]. Alec Go et al [22], introduced their contribution in sentiment analysis using machine learning algorithms by applying Naive Bayes, Maximum Entropy, and SVM supervised classifier with emoticons, which are used as noisy labels. Their experiments gave accuracy above 80%. The features extracted are Unigrams, Big grams, Unigrams and Big grams, and Unigrams with part of speech tags. After post-processing the training dataset contains 800,000 tweets with positive emoticons, and 800,000 tweets with negative emoticons, for a total of 1,600,000 training tweets. Which is large amount of noisy data. The test data is manually collected and annotated it contains a set of 177 negative tweets and 182 positive tweets. They reported accuracy values that range from 81.3% - 83.0%, see Table 5.3.

Table 5.3: Alec Go et al experiment summary [22]

Approach	Supervised machine learning
Algorithms	Naive Bayes, Maximum Entropy, and SVM
Training dataset	1,600,000 tweets
Testing dataset	177 negative tweets and 182 positive tweets
Features	Unigrams, Big grams, Unigrams and Big grams, and Unigrams with part of speech tags
Accuracy	81.3% - 83.0%

Our experiment setup: the experiment conducted with the following setup:

1. **Dataset**: we used **Sentiment140 testing dataset** of **359 test cases (tweets)** was used. It consists of 177 negative tweets and 182 positive tweets.
2. **Features**: Multiple features settings are considered as shown in Table 5.2
3. **Seeds**: Different seeds sets are used; Unigrams with 30 positives and 40 negatives, Big grams (tweets) with 5 positives and 5 negatives, and both Unigrams and Big grams seeds.

Results: the results for each feature settings and seeds sets are shown separately in the following Table 5.4.

Table 5.4: Supervised Comparison Experiment results

Features	Seeds Settings	Accuracy	Precision	Recall	F-measure
Setting1	Unigrams	**0.794**	**0.951**	**0.697**	**0.768**
	Big grams	0.584	0.983	0.552	0.707
	Big grams & Unigrams	0.641	0.937	0.594	0.727
Setting2	Unigrams	0.619	0.961	0.576	0.720
	Big grams	**0.752**	**0.956**	**0.692**	**0.795**
	Big grams & Unigrams	0.626	0.959	0.581	0.723
Setting3	Unigrams	**0.694**	**0.951**	**0.647**	**0.769**
	Big grams	0.585	0.983	0.552	0.707
	Big grams & Unigrams	0.641	0.937	0.594	0.727
Setting4	Unigrams	0.622	0.961	0.578	0.722
	Big grams	**0.743**	**0.939**	**0.696**	**0.791**
	Big grams & Unigrams	0.626	0.960	0.581	0.724
Setting5	Unigrams	**0.688**	**0.956**	**0.642**	**0.767**
	Big grams	0.505	0.960	0.508	0.664
	Big grams & Unigrams	0.534	0.948	0.524	0.675
Setting6	Unigrams	0.614	0.961	0.572	0.717
	Big grams	**0.746**	**0.879**	**0.699**	**0.784**
	Big grams & Unigrams	0.664	0.902	0.617	0.733
Setting7	Unigrams	**0.685**	**0.951**	**0.600**	**0.764**
	Bis grams	0.511	0.966	0.511	0.668
	Big grams & Unigrams	0.531	0.948	0.523	0.674
Setting8	Unigrams	0.614	0.9601	0.572	0.717
	Big grams	**0.730**	**0.890**	**0.700**	**0.783**
	Big grams & Unigrams	0.655	0.925	0.606	0.733

Discussion: for the results shown in Table 5.4, we observe the following:

- Our accuracy for different combination of features ranges from 61.8% to 79.94%.
- Edge features depending on distance gives the best results for F-measure $\approxeq$ **0.795** with **setting2** features.
- POS feature has no noticeable effect on results.
- The most affecting node feature is the big gram, may be because of the nature of the dataset as it is a collection of different term queries. So each subset of tweets, related to the same query, influence each other and exchange polarity.
- Also eliminating stop words has a great effect on results.

Although the accuracy of the approach described in [22] is higher than ours, there are number of points to be considered:

- They only mentioned system accuracy; but it is not enough measure for binary classification problem to show the correct behavior of the system like Precision, Recall, and F-measure we already calculated.
- Our obtained precision, recall, and F-measure are relatively good, as the F-measure is around 80%.
- Also compared to the amount of training dataset, we used few labeled examples.
- In addition the seeds we used are from the domain and context which is valuable advantage of our approach.

Summary: Table 5.5 gives an experiment summary.

Table 5.5: Supervised Comparison Experiment summary

Approach	Semi-supervised machine learning
Algorithms	MAD Label propagation
Training dataset	Unigrams: 30 positive and 40 negative , Big grams (tweets): 5 positive and 5 negative
Testing dataset	359 tweets: 177 negative tweets and 182 positive tweets
Features (nodes)	Hashtags, N-Chargram, Unigrams, Big grams, Unigrams and Big grams, and Unigrams with POS
Features (edges)	Edge features
Accuracy	61.8% - 79.94%
F-measure	0.675 - 0.80
Best features setting	Setting2

5.3.3 Semi-supervised Comparison Experiment

In this experiment we aim to compare our approach with a semi-supervised approach for sentiment analysis described in [52]. Michael Speriosu et al [52], introduced their contribution in sentiment analysis using label propagation to incorporate labels from a maximum entropy classifier trained on noisy labels and knowledge about word types encoded in a lexicon, in combination with the Twitter follower graph. Furthermore, they applied their model on different datasets, one of them was the HCR dataset they created. Their experiments gave accuracy ranges between 62.9% - 71.0% depending on different combination of features and seeds they were used.

Table 5.6: Michael Speriosu et al experiment summary [52]

Approach	Semi supervised learning
Algorithm	MAD Label propagation
Training dataset	488 tweets, 43.2% are positive
Testing dataset	396 tweets, 38.6% are positive
Features (nodes)	Unigrams, Big grams, Hashtags, N-Char gram, emoticons.
Features (edges)	Follower-edge, Feature-edges
Accuracy	62.9% - 71.0%

Our experiment setup: the experiment conducted with following setup:

1. **Dataset**: we used **HCR testing dataset** with **396 tweets**; 236 negative tweets and 160 positive tweets.
2. **Features**: multiple features settings were considered as shown in Table 5.2.
3. **Seeds**: different seeds sets were used, Unigrams with 36 positives and 49 negatives, Big grams (tweets) with 4 positives and 4 negatives, and both Unigrams and Big grams seeds.

Results: the results for each feature settings are shown separately in the Table 5.7.

Table 5.7: Semi-supervised Comparison Experiment results

Features	Seeds Settings	Accuracy	Precision	Recall	F-measure
Setting1	Unigrams	0.503	0.905	0.466	0.615
	Big grams	0.563	0.0	NaN	NaN
	Big grams & Unigrams	**0.655**	**0.897**	**0.577**	**0.700**
Setting2	Unigrams	0.464	0.988	0.4500	0.618
	Big grams	0.563	0.0	NaN	NaN
	Big grams & Unigrams	**0.526**	**0.988**	**0.498**	**0.658**
Setting3	Unigrams	0.0	0.0	0.0	0.0
	Big grams	0.563	0.0	NaN	NaN
	Big grams & Unigrams	**0.652**	**0.897**	**0.575**	**0.699**
Setting4	Unigrams	0.464	0.988	0.450	0.618
	Big grams	0.563	0.0	NaN	NaN
	Big grams & Unigrams	**0.526**	**0.988**	**0.498**	**0.658**
Setting5	Unigrams	**0.529**	**0.964**	**0.499**	**0.654**
	Big grams	0.543	0.157	0.437	0.231
	Big grams & Unigrams	0.503	0.582	0.448	0.506
Setting6	Unigrams	**0.490**	**0.981**	**0.483**	**0.650**
	Big grams	0.409	0.726	0.403	0.518
	Big grams & Unigrams	0.449	0.974	0.441	0.607
Setting7	Unigrams	**0.527**	**0.964**	**0.498**	**0.652**
	Big grams	0.546	0.138	0.437	0.209
	Big grams & Unigrams	0.506	0.576	0.449	0.505
Setting8	Unigrams	**0.491**	**0.991**	**0.483**	**0.651**
	Big grams	0.406	0.726	0.401	0.517
	Big grams & Unigrams	0.458	0.9801	0.446	0.613

Discussion: for the results shown in Table 5.7, we observe the following:

- Our accuracy for different combination of features ranges from 52% to 65%.
- Edge features depending on distance gives the best results for F-measure $\approxeq$ **0.70** with **setting1** features.
- POS feature has no noticeable effect on results.
- The most affecting node feature are both the Unigrams and Big grams, may be because of the nature of the dataset as it is a collection of tweets referring the same subject regrading the #hcr hash tag.

Our method has significant additives when compared to the work described in paper [52]:

- The best accuracy we obtained is comparable with the accuracy range obtained by Michael Speriosu et al [52].
- In our approach, we considered fewer and simple features.
- Also compared to the amount of training dataset, we used few labeled examples. With just 8 big grams seeds to reach our best F-measure $\approxeq$ **0.70** with **setting1** features.
- In our approach, we design a much simpler graphical model with fewer nodes and edges. This makes the label propagation algorithm more efficient.

Summary: Table 5.8 gives an experiment summary.

Table 5.8: Semi-supervised Comparison Experiment summary

Approach	Semi-supervised machine learning
Algorithms	MAD Label propagation
Training dataset	Unigrams: 36 positive and 49 negative, Big grams (tweets): 4 positive and 4 negative
Testing dataset	396 tweets, 38.6% are positive
Features (nodes)	Hashtags, N-Chargram, Unigrams, Big grams, Unigrams and Big grams, and Unigrams with POS
Features (edges)	Edge features
Accuracy	52% to 65%
F-measure	0.60 - 0.70
Best features setting	Setting1

5.3.4 Real Data Semi-supervised Experiment

In this experiment we aim to evaluate our system using a real educational dataset. The dataset is for tweets collected based on #bigdata hashtag. This experiment is divided to the two sub-experiments; first one is to compare our results with results obtained from an on-shelf tool, the second one is the golden label experiment.

5.3.4.1 (1) Manual annotation by two recommenders with Kappa Cohen's measurement

As we mentioned before Cohen's kappa is a more robust measure than simple percent to measure the agreement between two raters who each classifies N items into mutually exclusive categories. So our proposed approach will be one of the rater against another an on-shelf tool for sentiment analysis. Repustate (https://www.repustate.com/) is one of the commercial state-of-the-art tools [12].

Our experiment setup:: the experiment conducted with the following setup:

1. **Dataset**: we used **real educational dataset** of **174 tweets**; 39 negative tweets and 135 positive tweets.
2. **Features**: Unigrams, N-char gram, and hashtags were as features.
3. **Seeds**: big grams (tweets) were used as seeds; 2 positives and 3 negatives.

Results: the results are shown in Table 5.9; the calculations are based on Equation 5.5.

Table 5.9: Cohen's kappa results

Percentage agreement Pr(a)	0.873563
Percentage agreement 'by chance' Pr(e)	0.714031
Cohen's kappa	0.557866

Discussion: from the results shown in Table 5.9, there is a moderate agreement, approximating good agreement, between our prototype and the other rater, refer to Table 5.1. Which is a good result.

5.3.4.2 (2) Golden label with Precision, Recall, and F-measure measurements

For the real dataset with unknown label, preliminary we need a gold-standard dataset. In order to create such dataset we tested each tweet against three web sentiment analysis tools for tweets. If a tweet get two results and more for positive then the tweet holds a positive sentiment; doing the same for negative sentiment. Otherwise the tweet is considered neutral and is eliminated. The raw data we used was about 1211 tweets and after going through the different processing steps, dataset decreased to 498 tweets.

The three free web sentiment analysis tools used are:

1. Tweet annotator (http://www.tweenator.com)
2. Semantria (https://semantria.com/demo)
3. Free Sentiment Analyzer (http://www.danielsoper.com/sentimentanalysis)

Our experiment setup: the experiment conducted with the following setup:

1. **Dataset**: we used **real education dataset** of **498 tweets collected for #bigdata hashtag**; 145 negative tweets and 353 positive tweets.
2. **Features**: multiple features settings were considered as shown in Table 5.2
3. **Seeds**: different seeds sets were used. Unigrams with 30 for positive and 40 for negative, Big grams (tweets) with 5 for positive and 5 for negative, and both Unigrams and Big grams seeds.

Results: the results for each feature settings are shown separately in the Table 5.10.

Table 5.10: Real Data Semi-supervised Experiment results

Features	Seeds Settings	Accuracy	Precision	Recall	F-measure
Setting1	Unigrams	0.713	1.0	0.712	0.832
	Big grams	0.625	0.837	0.698	0.761
	Big grams & Unigrams	**0.718**	**1.0**	**0.717**	**0.835**
Setting2	Unigrams	0.711	1.0	0.711	0.831
	Big grams	0.544	0.710	0.670	0.690
	Big grams & Unigrams	**0.716**	**1.0**	**0.715**	**0.834**
Setting3	Unigrams	0.713	1.0	0.712	0.832
	Big grams	0.636	0.851	0.702	0.769
	Big grams & Unigrams	**0.718**	**1.0**	**0.717**	**0.835**
Setting4	Unigrams	0.0	0.0	0.0	0.0
	Big grams	0.548	0.719	0.671	0.694
	Big grams & Unigrams	**0.718**	**1.0**	**0.717**	**0.835**
Setting5	Unigrams	**0.709**	**0.998**	**0.710**	**0.830**
	Big grams	0.230	0.0173	1.0	0.034
	Big grams & Unigrams	0.513	0.509	0.726	0.598
Setting6	Unigrams	**0.709**	**0.998**	**0.710**	**0.830**
	Big grams	0.304	0.035	0.750	0.066
	Big grams & Unigrams	0.685	0.914	0.720	0.806
Setting7	Unigrams	**0.709**	**0.998**	**0.710**	**0.830**
	Big grams	0.298	0.015	1.0	0.029
	Big grams & Unigrams	0.515	0.506	0.731	0.598
Setting8	Unigrams	**0.709**	**0.998**	**0.710**	**0.830**
	Big grams	0.304	0.029	0.834	0.0556
	Big grams & Unigrams	0.685	0.906	0.723	0.804

Discussion: from the results shown in Table 5.10 we can observe the following:

- The accuracy for different combination of features ranges from 70% to 72%.
- As accuracy is not enough measure for binary classification problem, F-measure also calculated to reflect the actual evaluation of the proposed approach.
- POS feature has no noticeable effect on results.
- Edge features depending on distance gives the best results for F-measure $\approxeq$ **0.835** with **setting1** features.
- As we see in results obtained from setting1 with stop word features and or setting2 without stop word features, F-measure not significantly changed. This means using stopwords as features has no effect on enhancing the results. On the other hand, we found that eliminating stopwords has improved the results.
- The most affecting node feature are both big gram and Unigrams.

Summary: Table 5.11 gives an experiment summary.

Table 5.11: Real Data Semi-supervised Experiment summary

Approach	Semi supervised learning
Algorithm	MAD Label propagation
Training dataset	Unigrams: 30 positive and 40 negative Big grams (tweets): 5 positive and 5 negative
Testing dataset	498 tweets: 145 negative and 353 positive
Features (nodes)	Hashtags, N-Chargram, Unigrams, Big grams, Unigrams and Big grams, and Unigrams with POS
Features (edges)	Edge features
Accuracy	70% - 72%
F-measure	0.80 - 0.83
Best features setting	Setting1

6

Conclusion and Future Work

6.0.4.3 Conclusion

As a conclusion our research is an additive contribution to the field of semi-supervised sentiment analysis. In particular, we proposed a sentiment analysis prototype for microblogs posted in learning activities. Most of previous works on sentiment analysis tackled domains such as economic, products, movie reviews, and political domain. There is a paucity of literature in the education domain.

The prototype automatically classified microblogs of learning activities into positive and negative with high precision, high recall, and less costs in terms of learning requirements. Our approach aimed to achieve this objective using a novel combination of features extraction, engineering methods, and using a semi-supervised sentiment classification model based on label propagation algorithm. The costs in terms of learning requirements, considered low when compared to other learning approaches.

We conducted several experiments to evaluate our prototype. An initial demo is produced as beta version.

The results of the experiments conducted to evaluate the model are comparable to existing works. The first experiment was to compare our approach with a supervised learning approach. We obtained an accuracy and F-measure $\approxeq 80\%$. The second experiment was to compare our approach with a semi-supervised learning approach. Our approach has an accuracy $\approxeq 65\%$ and F-measure $\approxeq 70\%$. A third experiment carried on real data from educational domain. We obtained 71% for accuracy and 83.5% for F-measure.

6.0.4.4 Future Work

Enhancing the accuracy and minimizing the resources needed will be our future focus. In particular, we will work to:

- Improve our model to increase the F-measure and accuracy.
- Enhance our model to support different languages.
- Implement a real time interactive framework for tweets sentiment analysis using semi-supervised approach. Such a framework should interact with the users during the preprocessing phase in order to recommend positive and negative seeds.
- Test user satisfaction and the usability of the system in course assessment.

Bibliography

[1] Eclipse - the eclipse foundation open source community. https://eclipse.org/. Accessed: 08-04-2015.

[2] Oracle technology network for java developer. http://www.oracle.com/technetwork/java/index.html. Accessed: 08-04-2015.

[3] Oracle vm virtualbox. https://www.virtualbox.org/. Accessed: 08-04-2015.

[4] ALLEN, I. E., AND SEAMAN, J. Grade change: Tracking online education in the united states, 2013. *Babson Survey Research Group and Quahog Research Group, LLC. Retrieved on 3*, 5 (2014), 2014.

[5] ARCHNA, B., DIPANJAN, D., CHRIS, D., AND ET AL. Twitter natural language processing – noah's ark. http://www.ark.cs.cmu.edu/TweetNLP/. Accessed: 08-03-2015.

[6] BACCIANELLA, S., ESULI, A., AND SEBASTIANI, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC* (2010), vol. 10, pp. 2200–2204.

[7] BINALI, H. H., WU, C., AND POTDAR, V. A new significant area: Emotion detection in e-learning using opinion mining techniques. In *Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on* (2009), IEEE, pp. 259–264.

[8] BUZZI, M. *E-Learning*. ERIC, 2010.

[9] CAMBRIA, E., SCHULLER, B., XIA, Y., AND HAVASI, C. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems 28*, 2 (2013), 15–21.

[10] CARLETTA, J. Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics 22*, 2 (1996), 249–254.

[11] CARPENTER, B. Lucene version 3.0 tutorial. *Draft of: March 31* (2011).

[12] CIELIEBAK, M., DÜRR, O., AND UZDILLI, F. Potential and limitations of commercial sentiment detection tools. In *ESSEM@ AI* IA* (2013), Citeseer, pp. 47–58.

[13] CLARK, I. Formative assessment: Assessment is for self-regulated learning. *Educational Psychology Review 24*, 2 (2012), 205–249.

[14] CREWS, T. B., AND STITT-GOHDES, W. L. Incorporating facebook and twitter in a service-learning project in a business communication course. *Business Communication Quarterly* (2012).

[15] CUNNINGHAM, M. More than just the kappa coefficient: a program to fully characterize inter-rater reliability between two raters. In *SAS global forum* (2009), pp. 242–2009.

[16] DAVE. Twitter for academia. http://academhack.outsidethetext.com/home/2008/twitter-for-academia/. Accessed: 08-03-2015.

[17] DUNN, J. The ultimate guide to using twitter in education. http://www.edudemic.com/twitter-in-education/. Accessed: 08-03-2015.

[18] EL-HALEES, A. Mining feature-opinion in educational data for course improvement. *International Journal of New Computer Architectures and their Applications (IJNCAA) 1*, 4 (2011), 1076–1085.

[19] ESULI, A., AND SEBASTIANI, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC* (2006), vol. 6, pp. 417–422.

[20] FELDMAN, R. Techniques and applications for sentiment analysis. *Communications of the ACM 56*, 4 (2013), 82–89.

[21] GIMPEL, K., SCHNEIDER, N., O'CONNOR, B., DAS, D., MILLS, D., EISENSTEIN, J., HEILMAN, M., YOGATAMA, D., FLANIGAN, J., AND SMITH, N. A. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2* (2011), Association for Computational Linguistics, pp. 42–47.

[22] GO, A., BHAYANI, R., AND HUANG, L. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* (2009), 1–12.

[23] HAN, Q., GUO, J., AND SCHUETZE, H. Codex: Combining an svm classifier and character n-gram language models for sentiment analysis on twitter text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume*

2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (Atlanta, Georgia, USA, June 2013), Association for Computational Linguistics, pp. 520–524.

[24] HLTCOE, J. Semeval-2013 task 2: Sentiment analysis in twitter. *Atlanta, Georgia, USA* (2013), 312.

[25] IDZELIS, M. Jazzy: The java open source spell checker, 2005.

[26] JONES, R. Slang dictionary. http://www.noslang.com/dictionary/y/. Accessed: 08-03-2015.

[27] JOSHUA, O. M., DANYEL, F., AND TOM, N. The java universal network/-graph framework (jung): A brief tour. http://jung.sourceforge.net/presentations/JUNG_M2K.pdf. Accessed: 08-03-2015.

[28] KECHAOU, Z., BEN AMMAR, M., AND ALIMI, A. M. Improving e-learning with sentiment analysis of users' opinions. In *Global Engineering Education Conference (EDUCON), 2011 IEEE* (2011), IEEE, pp. 1032–1038.

[29] KEVIN, G., NATHAN, S., AND BRENDAN, O. Annotation guidelines for twitter part-of-speech tagging version 0.3 (march 2013). http://www.ark.cs.cmu.edu/TweetNLP/annot_guidelines.pdf. Accessed: 08-03-2015.

[30] KOULOUMPIS, E., WILSON, T., AND MOORE, J. Twitter sentiment analysis: The good the bad and the omg! *ICWSM 11* (2011), 538–541.

[31] KREUTZER, J., AND WITTE, N. Opinion mining using sentiwordnet.

[32] LANG, K. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning* (1995), pp. 331–339.

[33] LIU, B. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies 5*, 1 (2012), 1–167.

[34] LLC, T. The Streaming APIs documentation, 2012.

[35] LONG, J., YIN, J., ZHAO, W., AND ZHU, E. Graph-based active learning based on label propagation. In *Modeling Decisions for Artificial Intelligence*. Springer, 2008, pp. 179–190.

[36] MARTIN, J. M., ORTIGOSA, A., AND CARRO, R. M. Sentbuk: Sentiment analysis for e-learning environments. In *Computers in Education (SIIE), 2012 International Symposium on* (2012), IEEE, pp. 1–6.

[37] MESSIEH, N. How to use twitter in the classroom. http://thenextweb.com/twitter/2011/06/23/how-to-use-twitter-in-the-classroom/. Accessed: 08-03-2015.

[38] MILLER, G. A. Wordnet: A lexical database for english. *Commun. ACM 38*, 11 (1995), 39–41.

[39] MOHAMMAD, S. M., KIRITCHENKO, S., AND ZHU, X. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* (2013).

[40] MONTOYO, A., MARTÍNEZ-BARCO, P., AND BALAHUR, A. Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments. *Decis. Support Syst. 53*, 4 (2012), 675–679.

[41] OHANA, B., AND TIERNEY, B. Sentiment classification of reviews using sentiwordnet. In *9th. IT & T Conference* (2009), p. 13.

[42] OLSON, D. L., AND DELEN, D. *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[43] O'MADADHAIN, J., FISHER, D., WHITE, S., AND BOEY, Y. The jung (java universal network/graph) framework. *University of California, Irvine, California* (2003).

[44] ORTEGA, R., FONSECA, A., AND MONTOYO, A. Ssa-uo: unsupervised twitter sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)* (2013), vol. 2, pp. 501–507.

[45] ORTIGOSA, A., MARTÍN, J. M., AND CARRO, R. M. Sentiment analysis in facebook and its application to e-learning. *Computers in Human Behavior 31* (2014), 527–541.

[46] OWOPUTI, O., O'CONNOR, B., DYER, C., GIMPEL, K., SCHNEIDER, N., AND SMITH, N. A. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL* (2013), pp. 380–390.

[47] PANG, B., AND LEE, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* (2008), 1–135.

[48] PARTHA, P. T., AND JASON, B. Junto: The label propagation toolkit. https://github.com/parthatalukdar/junto. Accessed: 08-03-2015.

[49] RAJARAMAN, A., AND ULLMAN, J. D. *Mining of massive datasets*. Cambridge University Press, 2011.

[50] RAO, D., AND RAVICHANDRAN, D. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (2009), Association for Computational Linguistics, pp. 675–682.

[51] SMITH, K. he twitter experiment - bringing twitter to the classroom at ut dallas. http://kesmit3.blogspot.com/2009/04/twitter-experiment-bringing-twitter-to.html. Accessed: 08-03-2015.

[52] SPERIOSU, M., SUDAN, N., UPADHYAY, S., AND BALDRIDGE, J. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP* (2011), Association for Computational Linguistics, pp. 53–63.

[53] TALUKDAR, P. P., AND CRAMMER, K. New regularized algorithms for transductive learning. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 442–457.

[54] TALUKDAR, P. P., AND PEREIRA, F. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), Association for Computational Linguistics, pp. 1473–1481.

[55] TALUKDAR, P. P., AND PEREIRA, F. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), Association for Computational Linguistics, pp. 1473–1481.

[56] TALUKDAR, P. P., REISINGER, J., PAŞCA, M., RAVICHANDRAN, D., BHAGAT, R., AND PEREIRA, F. Weakly-supervised acquisition of labeled class instances

using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), Association for Computational Linguistics, pp. 582–590.

[57] THOMAS, E. H., AND GALAMBOS, N. What satisfies students? mining student-opinion data with regression and decision tree analysis. *Research in Higher Education 45*, 3 (2004), 251–269.

[58] TSYTSARAU, M., AND PALPANAS, T. Survey on mining subjective data on the web. *Data Min. Knowl. Discov.* (2012), 478–514.

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

[60] WIEBE, J. Learning subjective adjectives from corpora. In *AAAI/IAAI* (2000), pp. 735–740.

[61] WIKI, E. Learning management system edutech, 2014.

[62] WU, D., APIDIANAKI, M., CARPUAT, M., AND SPECIA, L. Acl hlt 2011.

[63] YANG, C., BHATTACHARYA, S., AND SRINIVASAN, P. Lexical and machine learning approaches toward online reputation management. In *CLEF (Online Working Notes/Labs/Workshop)* (2012).

[64] ZHOU, Z.-H., ZHAN, D.-C., AND YANG, Q. Semi-supervised learning with very few labeled training examples. In *Proceedings of the national conference on artificial intelligence* (2007), vol. 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 675.

[65] ZHU, X., AND GHAHRAMANI, Z. Learning from labeled and unlabeled data with label propagation. Tech. rep., Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

[66] ZHU, Z., HIEMSTRA, D., APERS, P., AND WOMBACHER, A. Ut-db: an experimental study on sentiment analysis in twitter. Association for Computational Linguistics.