International Journal of Cognitive Informatics and Natural Intelligence
Volume 14 • Issue 2 • April-June 2020

Enhanced Bootstrapping Algorithm for Automatic Annotation of Tweets

Mudasir Mohd, University of Kashmir, Srinagar, India
https://orcid.org/0000-0003-1597-146X

Rafiya Jan, Central University of Kashmir, Srinagar, India

Nida Hakak, Mahareshi Dayanand University, Haryana, India

ABSTRACT

Annotations are critical in various text mining tasks such as opinion mining, sentiment analysis, word sense disambiguation. Supervised learning algorithms start with the training of the classifier and require manually annotated datasets. However, manual annotations are often subjective, biased, onerous, and burdensome to develop; therefore, there is a need for automatic annotation. Automatic annotators automatically annotate the data for creating the training set for the supervised classifier, but lack subjectivity and ignore semantics of underlying textual structures. The objective of this research is to develop scalable and semantically rich automatic annotation system while incorporating domain dependent characteristics of the annotation process. The authors devised an enhanced bootstrapping algorithm for the automatic annotation of Tweets and employed distributional semantic models (LSA and Word2Vec) to augment the novel Bootstrapping algorithm and tested the proposed algorithm on the 12,000 crowd-sourced annotated Tweets and achieved a 68.56% accuracy which is higher than the baseline accuracy.

KEYWORDS

Bootstrapping, Emotion Classification, Emotions, Semantic Similarity

INTRODUCTION

Twitter is leading microblog service used by over 974 million users with 500 million tweets/day, thus is playing an active role in the new form of media. Twitter posts are called tweets and are limited to 280 characters. Users also upload photos and short videos for broadcasting their experience and feelings about daily life (McFedries, 2007). Twitter is acting as an essential communication channel for governments and heads of state to highlight their governance initiatives and interact with their citizens directly. The evolution of Internet and mobile based communications, led to increase in social interaction among multiple users ("social networking sites"), and thus huge data ("Big Data") is equipped depicting the public attitude and acknowlegments related to different events like world events, consumer product events, political and movies events (Salton, 1991). According to the Twitter blog, recently, something remarkable happened on Twitter: #NuggsForCarter was the most retweeted

DOI: 10.4018/IJCINI.2020040103

tweet of the year 2017. A high scholar's call for free nuggets to Wendys became the highest retweeted tweet of all time with 3.24 million retweets[1]. In general, Twitter users now share excessive tweets near about 500 million tweets per day that is about 5,700 Tweets per second, according to mean based mentioned on a later report in Twitter blog[2]. This shows the considerable popularity Twitter is gaining and the role it's playing in changing people's lives. People use Twitter for various reasons. (Java, Song, Finin, & Tseng, 2007) in their study categorize user intentions as: (1) source of information; (2) being social; and (3) retrieving information. (Hakak, Mohd, Kirmani, & Mohd, 2017) have given an excellent summary of the state of work done so far in the area.

Twitter is becoming a reliable media to search for timely information then the web and this information is mined extensively for opinion mining, emotion detection and sentiment polarity by different business and researchers. Automatic affect detection on Twitter is attracting much research since users continuously express their opinions' regarding anything that they are interested in. These opinions include reviews of products, general feelings, etc. Affect detection finds its applications in various applications like (Rodriguez, Ortigosa, & Carro, 2012) monitored how affect and emotional factors determine the outcome of the e-learning environment; (Desmet & Hoste, 2013) showed how affect monitoring on social media can help suicide prevention; (Cherry, Mohammad, & De Bruijn, 2012) used emotion classification to detect depression on social media; (Dadvar, Trieschnigg, Ordelman, & de Jong, 2013) showed how to improve detection of cyberbullying from user content.

Opinion analyzers and emotion detection tools for social media text streams use supervised learning classifiers which rely heavily on the manually annotated corpus. The manually annotated corpus for use in supervised learning is difficult to create and human annotators, who associate different sentences with different categories, traditionally produce annotated corpus. However, this process is arduous and time-consuming and also obtaining an inter-annotator agreement is difficult in such tasks as human judgment is subjective. This research aims to create an auto-annotation tool capable of annotating twitter corpus by analyzing tweets, i.e., to create a bootstrapping algorithm for automatic annotation of the Twitter corpus. Bootstrapping processes lack subjectivity and overlook the inherent semantics of underlying text. Thus, there is a greater need for extending bootstrapping algorithms for achieving better accuracy in the automatic annotation of tweets. For this reason, we propose an extended bootstrapping algorithm for the automatic annotation of tweets.

The proposed enhanced bootstrapping algorithm takes semantics of text as a feature and annotates the corpus using distributional semantic hypothesis. We exploited distributional semantic models for enhancing the bootstrapping algorithm and achieved comparable results. The existing bootstrapping algorithms overlook the semantics of the text and work on the presence of either critical terms in the text or some other statistical features to annotate the sentences and thus are not scalable. The key idea of our proposed enhanced bootstrapping algorithm is thus to extend the bootstrapping process by using semantic models to create any domain annotations and thus have a scalable bootstrapping algorithm.

The proposed algorithm constitutes five important steps: 1) Preprocessing of tweets; 2) Lexicon generation; 3) Enhancement of lexicon of seed words using word2Vec model; 4) Seed extension using and another dictionary-based approaches 5) Using LSA to compute semantic similarity; 5) Using big vectors created using Word2Vec to calculate semantic coherence. The proposed system was evaluated on Kashmir 2016 unrest dataset collected from Twitter. Around 12,000 tweets were manually annotated using crowd-sourcing to check the efficiency of the proposed approach. The results are above the traditional baseline approaches, and thus confirm that the competitive performance of our proposed approach.

Rest of the paper is organized as follows: background, enhanced bootstrapping algorithm in detail, experiments, evaluation and results, discussion and comparative analysis and then conclusions and future work.

BACKGROUND

In this section, we summarize the most recent and relevant corpora developed for emotion and opinion analysis purpose, we discuss the features employed. Also, we present some works where bootstrapping has been employed as a technique for automatic annotation and the results they achieved.

Emotional corpus acts as a means for the supervised learning algorithms for learning the patterns hidden in the underlying document that is used for emotion classification. It serves as labelled training set for the supervised algorithms to learn and infer a function, which can be used to map new example (Rostamizadeh & Talwalkar, 2012).

This prototype analyzed six fundamental emotions viz: sadness, anger, fear, disgust, surprise and happiness. Plutchik and Kellerman (1980) determined the emotion prototype is having eight basic emotion including Ekman's six primary emotions plus two new emotions anticipation and trust. He coordinated these emotions around a wheel. In this wheel, radius depicts the tenor of these emotions. Centric emotions depict the sharp tenor of emotions.

Traditionally emotional corpora are created by manual annotation process, which allows machine learning algorithms to get trained from these human annotations. Trained corpus has sentence level annotations performed by human annotators. Corpora is generally annotated by the six emotional labels proposed by Ekman, there are several such works, like (Alm, Roth, & Sproat, 2005). Performed annotations at sentence level for 185 children stories with six emotional classes; (Aman & Szpakowicz, 2007). Created dataset of blog posts along with emotion intensity values and emotion classes. (Strapparava & Mihalcea, 2008) annotated news headlines with emotion valence and categories; (Balabantaray, Mohammad, & Sharma, 2012) employed sentence annotations of 8,150 tweets acquired from the web. (Hakak & Kirmani, 2018) used supervised learning for mining of opinions of Twitter events.

Additionally, there are other works where corpora are being labelled with other groups of emotions as being proposed by Ekman is found in literature include (Neviarouskaya, Prendinger, & Ishizuka, 2010) used 14 categories of emotions on the corpus of 1,000 sentences of stories; (Mohammad, Zhu, Kiritchenko, & Martin, 2015) annotated 2012 US presidential election twitter dataset with multi-layer emotion, polarity, valence style, and purpose; (Yan & Turtle, 2016) gave EmoTweet 28 in which they used tweet corpus with 28 emotional categories. (De Choudhury, Gamon, Counts, & Horvitz, 2013) employed supervised learning approaches to determine depressive disorders and evaluated physiological attributes like emotion, linguistic style, socialism, languages and prescribed antidepressant medication to design the classifier. Crowd-sourced Tweets are employed for classification and gained 70% accuracy. (Purver & Battersby, 2012) employed supervised learning approaches for emotion analysis. They employed labelled twitter dataset with automatic Ekman annotated (Ekman, 1992) classes viz anger, fear, happiness, sadness, surprise, and disgust and acquired 60% accuracy. (Wilson, Wiebe, & Hwa, 2004) employed a Supervised Learning system to classify text into Objective and Subjective texts. Objectivity defines the tenor (intensity) of the emotions related to the sentence. Subjectivity depicts polarity of a sentence. This approach requires a well-defined sentence prototype to determine the syntactical relations. All these require huge textual emotional corpora along the different types of emotions. However, all these approaches are annotated manually and are thus, highly time-consuming. They are error-prone because of subjective nature.

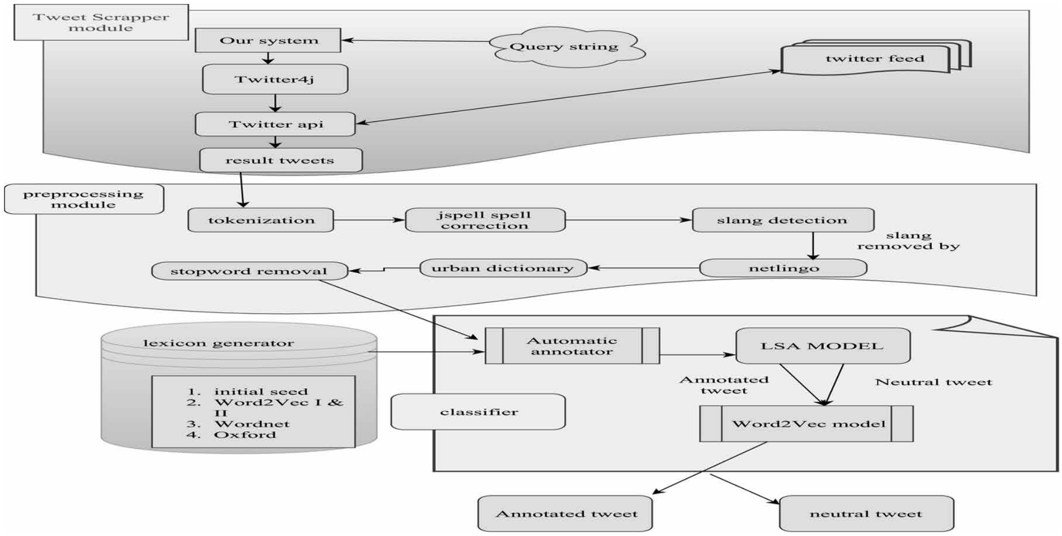
Recent approaches in this area have led to the automatic annotation of text at the sentence level. For example, (Bifet & Frank, 2010) employed Bootstrapping for automatic classification of tweets by enlarging the seed lists using LSA algorithms and Word2Vec models. Their computational performance was also favorable. (Go, Bhayani, & Huang, 2009) uses distant supervision using emoticons to create a positive and negative labelled dataset for the supervised learning algorithms. (Suttles & Ide, 2013) employed Distant supervision by employing Emoticons, Emojis and hashtags for automatic annotation of Tweets using Plutchik's Classes; (Hasan, Rundensteiner, & Agu, 2014) used circumplex model (Russell, 1980) in which emotional states depicting the affective content consists of intensity values and arousal is attained using hashtags of Twitter for automatic annotation of the Twitter corpus. (Qadir & Riloff, 2013) used a bootstrapping system with emotion hashtags. Emotion hashtags are employed for training the supervised classifier. Five emotion hashtags anxiety, anger, joy, sadness, and affection were employed and acquired 68% of accuracy in this classification realm.

As seen in the preceding discussion the focus on automatic annotation has increased drastically, but research is mainly focused on techniques like keyword spotting and presence and absence of emoticons, hashtags, acronyms, and slangs in the text. We have extended the work towards more semantic features, coherence, and domain-specific features of the text. (Jan & Khan, 2018) employed Semantic similarity approach to constructing automatic emotion classifier and achieved promising results with an accuracy of 71.795%. (Hasan, Rundensteiner & Agu, 2019) They collected text using twitter stream API for bootstrapping emotional corpus creation. (Taxer, Becker-Kurz, & Frenzel, 2019) used bootstrapping model to evaluate teacher-student relationships.

In this work, an extended bootstrapping approach has been used to overcome the problem. Our approach is a concoction of supervised learning and unsupervised learning methods and thus leading to an automatic classification process whose effectiveness has been evaluated by results. Bootstrapping process is already been employed in various computational linguistic problems like word sense disambiguation (Thelen & Riloff, 2002) named entity classification (Collins & Singer, 1999) anaphora resolution (Strapparava & Mihalcea, 2008).

1. **Extended Bootstrapping Process:** The Bootstrapping is an automated task used in the contribution of creating an annotated corpora focused on the reduction of both time and cost needed for the development of annotated corpora used for the learning of supervised classifiers. Bootstrapping aims to reduce the need for manual corpora annotations and thus have become a widely researched topic in the area of computational linguistics. Bootstrapping can be used to resolve challenges faced in the computational problems like sentiment analysis, word sense disambiguation, named entity resolution, etc. as all of these problems require labelled data which is too expensive and burdensome to create. In this section, we discussed the proposed extended bootstrapping algorithm which we have developed to annotate the dataset automatically. It is divided into six subsections where the primary task of automatic annotation carried by the extended bootstrapping process is explained. The algorithm receives as input an unlabelled dataset of tweets and a set of "n" classes. The objective of this task is to automatically annotate the unlabelled tweets into any of the "n" classes if the tweet in question is having a sentiment polarity reflected towards any of the "n" classes or neutral if it does not. Figure 1 shows the overall system of extended bootstrapping process diagrammatically. Six subprocesses of our enhanced bootstrapping process are: 1) Twitter Scrapping Module; 2) Preprocessing; 3) Lexicon generation; 4) Extended Bootstrapping Algorithm (EBA):
    a. **Twitter Scrapping module:** In this section, Twitter scrapping module is responsible for mining Twitter and retrieving data as per the query of the system. Data collection module is comprised of the Twitter4j[3] interacting with the Twitter search API[4] with the set of query strings which are probed on the twitter, and the Twitter API returns resulting tweets through twitter4j to our system;
    b. **Preprocessing:** The preprocessing module prepares the tweet for the classifier by performing pre-processing on every individual tweet before passing it to the automatic classifier. Preprocessing stages is made up of the following steps:
        i. Tokenization is performed using the Stanford Core NLP package (Manning, Surdeanu, Bauer, Finkel, Bethard, & McClosky, 2014) to break tweets into sentences and words;
        ii. Slangs and abbreviations are removed from the tweets by using English directories;
        iii. All words are fed to the dictionary module to look for their meaning. If a word is found which does not return useful meaning is fed to the word replacer module to replace it by its proper word. For example, a tweet "@USERNAME u should be gud with ur

Figure 1. Overall methodology



idiotc mind." The words "u," "gud," "ur" and "idiotc" do not fetch any meaning from the dictionaries hence these words are passed to the word replacer module and are replaced by "you," "good," "your," and "idiotic" words respectively. The dictionaries used are Wordnet and JSpell. In word replacer module we have used SMS dictionary[5], Netlingo[6] and the urban dictionary[7];

iv. Rt prefixes Retweets, i.e., tweets which were earlier sent by someone else, usually such tweets, rt, or RT and hence were dropped from the dataset;
v. Stop words are removed next from the tweets using TF-IDF feature, Stanford, and wiki;
vi. URLs and usernames are also stripped from the tweets;
vii. Next lemmatization is applied to words to obtain their stems. Lemmatization is done using Stanford core NLP package (Manning, Surdeanu, Bauer, Finkel, Bethard, & McClosky, 2014);
viii. Special charters are also stripped from the tweets;
ix. All the text is changed to lowercase characters;
x. Then the refined tweet is supplied to the automatic classifier;

c. **Lexicon Generation:** For all the n-classes we created an exhaustive set of seed words. These seed words represent the primary expression of sentiment expressed by the individual classes. They were chosen by the domain experts of the sentiments that were evaluated and is the only manual intervention part of the extended bootstrapping process. Choosing the right set of seed words is essential as the whole lexicon building process is entirely dependent on it:
   i. Initial seed set;
   ii. Seed extension using Word2Vec;
   iii. Seed extension using WordNet;
   iv. Seed extension using Oxford thesaurus.

1. **Initial Seed Set Generation:** For all the n-classes we created an exhaustive set of seed words. These seed words represent the primary expression of sentiment expressed by the individual classes. They were chosen by the domain experts of the opinion that were evaluated and was the only manual intervention part of the extended bootstrapping process. Choosing the right set of seed words is essential as the whole lexicon building process is primarily dependent on it;

2. **Seed Extension Using Word2Vec:** Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) is a two-layer neural network that processes text. It produces as output, a set of vectors for the text corpus as input. Vectors produced by Word2Vec are feature vectors of words, to be probed, in some corpus. Corpus acts as a domain for the production of vectors and is subject sensitive. Word2Vec algorithm is trained as a vector space representation of terms by exploiting two layers of the neural network. Word2Vec has two architectures: CBOW (Continuous Bag-Of-Words) and Skip gram models. These architectures characterize how neural networks determine the word representations of each word. CBOW predicts the current word according to its context while Skip gram determines the context of a word according to a given word. We employed Skip gram model for our seed extension;
3. **Seed Extension Using Wordnet:** WordNet (Miller, 1995) is a lexical database for English language. It is organized into synsets that group verbs, nouns, adjectives, and adverbs. Each synset expresses a distinct concept. Synsets are linked to each other through conceptual relationships. WordNet interlinks specific sense of words and not just strings of letters. The primary relationship among words in WordNet is synonymy. WordNet has 117,000 synsets linked to each other. Our proposed algorithm probes words in WordNet for their synonyms. Each word in the lexicon created in the preceding phase is looked up in WordNet to retrieve its synonyms, and these synonyms are then added to the class of word which leads to its inclusion. Figure 2 shows the method of seed extension by WordNet;
4. **Seed Extension Using Oxford:** Oxford Thesaurus (D'Alessandro, 2004) is the most extensive thesaurus in the world with more than 600,000 synonyms and antonyms, compiled by the English department of the University of Glasgow. Our proposed algorithm use oxford dictionary API[8] for retrieving synonyms of seed words from Oxford thesaurus. Oxford thesaurus provides a semantically linked collection of related words. We probed for seed words developed in the preceding stage in the Oxford dictionary for synonyms. Synonyms retrieved are added to the lexicon as they are not already present in the lexicon. If a word W_i belonging to class X, then synonyms retrieved for the word Wi from Oxford thesaurus are added to the class X. The process of seed extension by Oxford thesaurus is explained in Figure 3.

Oxford thesaurus is explained in Figure 3 Weights are assigned to each seed word in the lexicon. Weights are given such that each seed word is treated as being of different importance in the classification process. Seed words with a higher weight mean it's more important than seed words

Figure 2. Seed extension process using WordNet

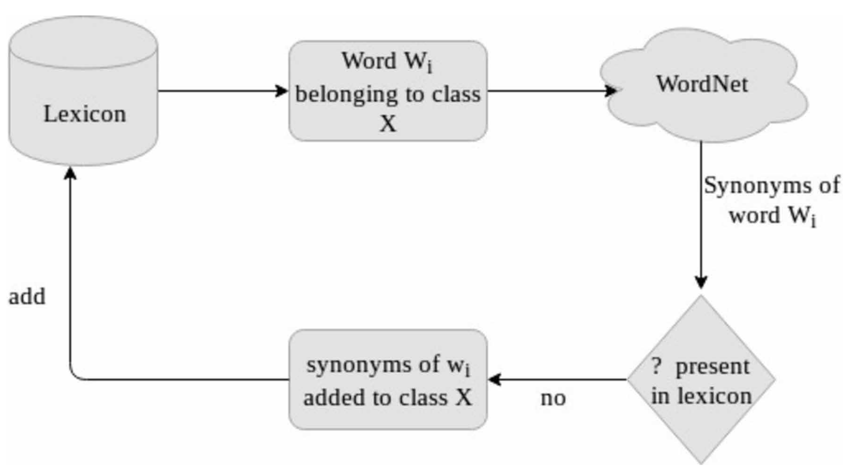
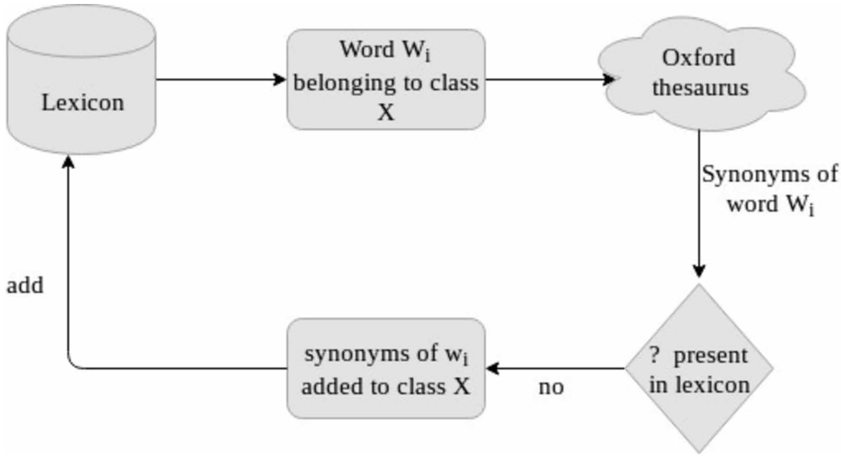


Figure 3. Seed extension process using Oxford



with lower weight in the classification process. Weights assignment to each seed word is according to its relevance to the classification process. Seed words belonging to Initial seed set are given highest weight as they are chosen manually followed by seed words added by Word2Vec I and Word2Vec II as these are derived from the corpus. Then seed words are retrieved by WordNet and then followed by Oxford. Table 1 shows the weight assigned to different words by our seed extension process.

Extended Bootstrapping Algorithm (EBA)

The proposed Enhanced bootstrapping algorithm (EBA) classifies twitter feeds into different classes using the following

1. Enhanced Sentiment Classifier (ESC)
2. Normalized Latent Semantic Analyser (NLSA)
3. Big-vector Approach using Word2Vec (BVW)

Using EBA, we classify tweets into n-different classes and obtain an automatically annotated tweet corpus. Figure 4 shows the extended bootstrapping process.

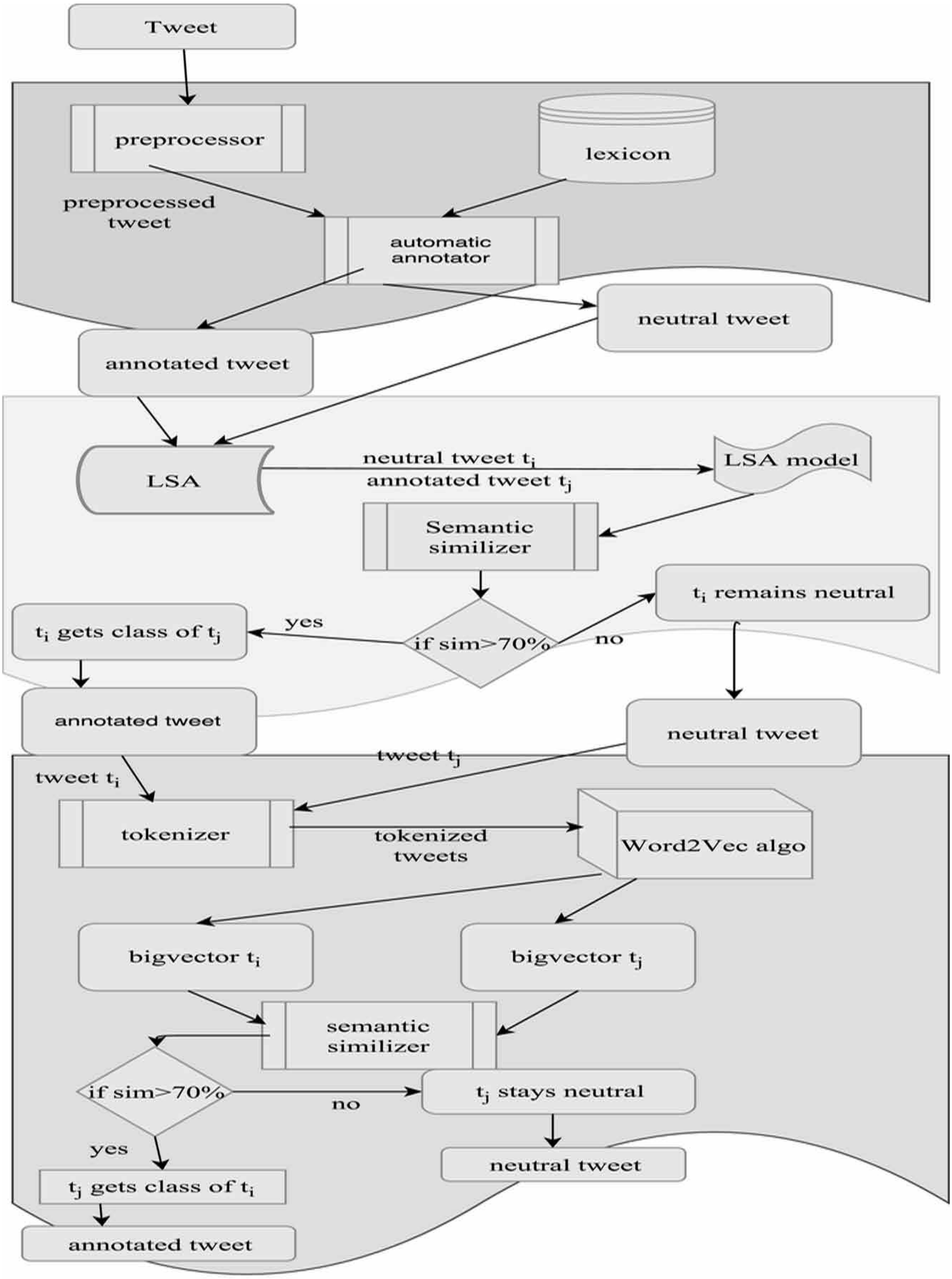
Enhanced Sentiment Classifier (ESC)

The enhanced bootstrapping classifier is used to map tweets to classes. It uses lexicon created in the preceding section to automatically classify tweets into one of the different classes, if tweet describes sentiment towards any of the class or neutral otherwise. It measures the degree of match between tweet

Table 1. Weight assignment to different stages of lexicon generation

Seed Extension Used	Weight
Initial seed set	5
Word2Vec I	4
Word2Vec II	3
WordNet	2
Oxford	1

Figure 4. Extended bootstrapping process



and seed words. Seed words with their corresponding weights represent the categories and tweets represent the documents. These categories are then equated with documents using the Scoring function. Vector Space Model (VSM), determines the vector representation of categories and documents and Cosine Similarity is employed to evaluate the semantic coherence. Then we used the Scoring function to evaluate the label of the tweet:

$$Score\left(t, V_{[n]}\right) = sim_{vsm}\left(id_c, t\right) = \cos\left(id_c^{\rightarrow}, t^{\rightarrow}\right)$$

where id_c is seed word, t is preprocessed tweet, $\cos$ is cosine similarity, $id_c^{\rightarrow}$ is a vector of seed word, $t^{\rightarrow}$ is a vector of tweet and $V_{[n]}$ is a vector of n elements.

We used a scoring function Score (t_i, $V_{[n]}$), whose arguments are pre-processed tweet and a vector of n-elements all initialized to zero. The scoring function maps the tweet to the vector having maximum value for the class to which this tweet belongs by comparing words in the tweet with the lexicon, if the tweet does not belong to any of the classes, i.e., its having neutral opinion then tweet is labelled as neutral with its class determined as neutral. The lexicon is divided into n-sets of classes with different words belonging to each set.

Let T be the set of tweets t defined as: T= {t_1, t_2, t_3, t_4, …, t_n}.and Let W be the set of words w in each tweet t defined as: W = {w_1, w_2, w_3 …, w_m}.

Let C_1 denote set of words in lexicon belonging to i.e.:

$$C_1 = \left\{w \mid w \in \text{class}_1\right\}$$

Let C_2 denote set of words in lexicon belonging to i.e.:

$$C_2 = \left\{w \mid w \in \text{class}_2\right\}$$

Let C_3 denote set of words in lexicon belonging to i.e.:

$$C_3 = \left\{w \mid w \in \text{class}_3\right\}$$

Let C_n denote set of words in lexicon belonging to i.e.:

$$C_n = \left\{w \mid w \in \text{class}_{n\wedge}\right\}$$

The class for the tweet t is calculated as:

Let LX denote set of all words in the lexicon, i.e.,
LX = {Set of all words in lexicon}.
Score(t_i; $V_{[n]}$) =
$V_{[1]} \leftarrow V_{[1]} + W(w_a)$; $(w_a \in W) \wedge (t \in T) \wedge (w_a \in C_1)$
$V_{[2]} \leftarrow V_{[2]} + W(w_b)$; $(w_b \in W) \wedge (t \in T) \wedge (w_b \in C_2)$
$V_{[3]} \leftarrow V_{[3]} + W(w_c)$; $(w_c \in W) \wedge (t \in T) \wedge (w_c \in C_3)$

⋮
⋮
$V[n] \leftarrow V[n] + 1; (w_g \in W) \wedge (t \in T) \wedge (w_g \notin LX)$

where w_a, w_b,w_c, w_d, w_e, w_f and w_g are words belonging to set of words W and t is a tweet from the set of tweets T. $W(w_i)$ is the weight of the word w. ESC is an improvement over by employing enhanced lexicon and pre-processed text (Liu, Li, Lee, & Yu, 2004). Stanford core NLP is used to tokenize tweet into words, and these words are probed for detecting the class of tweet. If a word is found belonging to some class in the lexicon, the scoring function of our ESC algorithm adjusts the vector $_{[n],}$ by adding its corresponding entry in the vector and the weight of the word to which it matched. The max function of our ESC algorithm returns the maximum value from the vector$_{[n]}$ that represents the class assigned to the tweet from any of the n-representative classes plus neutral. Thus, the tweet is assigned the class of the maximum words to which the representative words belonged. If it does not contain any sentiment, then the tweet is given a neutral class. If the max function returns more than one entry, then we use the conflict resolution function of the ESC algorithm. The conflict resolution works by probing the vector$_{[n],}$ words representing entries from the vector$_{[n]}$ are probed for their distance from the subject. Class of the word which is at a minimum distance from the subject is assigned to the tweet:

$\text{Class}(t_i, V_{[n]}) = \max(v_{[n]}); (t_i \in T)$

Algorithm: ESC Enhanced Sentiment Classifier

```
Input: Lexicon, Tweet set.
Output: Set of tweets with their classes
Lx ←Lexicon
C1 ← w ∈ class1
C2 ← w ∈ class2
C3 ← w ∈ class3
⋮
Cn ← w ∈ classn
T = {t1, t2, t3, t4 ..., tn}   /* the set of all pre-processed tweets*/
W = {w1, w2, w3, w4 ...wn}, /* the set of words in a tweet */

For Each ti ∈ T do
       {
       V [7] ← {0, 0, 0, …0} /* vector assigned to 0  */

For each wj ∈ Wi do
       {
                if ∃ wn ∈ C1 ∧  wn = wj then
                  {
V1←V1+W(n)
}
     If ∃ wn ∈c2 ∧ wn = wj then
                  {
V2←V2 + W(n)
}
```

```
⋮
                    If ∃ wn ∈ Cn-1 ∧  wn = wj then
                  {
Vn-1←Vn-1 + W(n)
}
         else
{
Vn←Vn+1;
}
    }
  If max(V[n] = 1) then
             {
class(ti)←max(V[n])
  }

else
{
entry{i}←max(V[n]); subject ←ti

For Each I ∈ entry do
{
dis{j}←distance(subject, wi)
}
}
Class(ti)←dis;
}
```

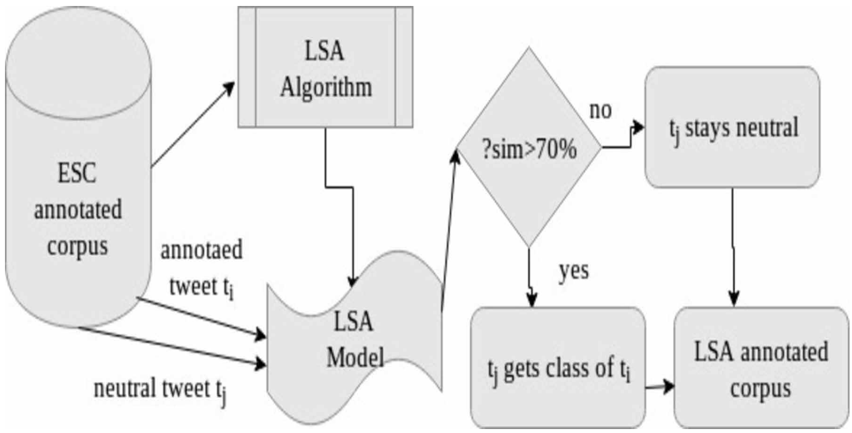
Normalized Latent Semantic Analyzer (NLSA)

Measuring the degree of match between categories and documents (tweets) in the vector space model is severely affected by feature sparsity. This sparsity problem is reduced by employing Latent Semantic Analysis (LSA) (Dumais, 2004). In this step, neutral tweets generated By the ESC algorithm are mapped to the classes using LSA which is used to analyze the relationship between a set of documents and their corresponding terms. LSA creates a learned model by using training corpus which can be used to determine the similarity of documents against the model designed by LSA. We have used the entire annotated tweet set generated by the ESC algorithm as the corpus for LSA. To do this, a similar approach used by (Gliozzo, Strapparava, & Dagan, 2009) is used, they have used latent semantic spaces to estimate the similarity between words and documents. We have used LSA to find similarity between annotated tweets and unannotated tweets, i.e., neutral tweets generated by ESC algorithm and only those annotations are taken where similarity score is higher than 70%. Figure 5 illustrates this concept.

Big-Vector Approach Using Word2Vec (BVW)

In this step, neutral tweets generated by ESC algorithm and NLSA are mapped to classes using BVW, BVW is a novel feature which uses Word2vec (Mikolov, Chen, Corrado, & Dean, 2013) algorithm and cosine similarity (Suttles & Ide, 2013) measure to annotate tweets. Using BVW, we can annotate the tweets that were not annotated so far and were given a neutral label. BVW uses Word2Vec to generate vectors of all words of a pre-processed tweet that are then mapped into Big-vectors. Big-vectors are created by combining all vectors of all words of a tweet into a single vector called big vector. Figure 7 shows the Big-vector formation for tweet t_i. Once Big-vectors are formed we are using them for the annotation process. Big-vectors cover the semantic information of the tweet and are used to map the

Figure 5. LSA annotation



semantic similarity between tweets. Big-vectors are thus semantically rich bag of words representation of individual tweets. Cosine similarity is then employed to compute semantic coherence between tweets: annotated tweets Big-vectors and neutral tweets Big-vectors and are then matched using cosine similarity for annotating neutral tweets. If the cosine similarity of the two paired Big-vectors is greater than 80%, then the neutral tweet is given the class of tweet to which it matched. Figure 8 shows the process of BVW. All neutral tweets generated in the preceding section are matched against every individual annotated tweet through their Big-vector representations, and wherever semantic similarity is greater than 80%, the neutral tweet gets the class of annotated tweet with which it matched. Figure 7 shows the Big-vector formation, and Figure 6 shows annotation using BVW.

Experiments

In this section, we discuss the experiments performed to test our enhanced bootstrapping algorithm. We downloaded data from Twitter and used our proposed algorithm to annotate the data automatically. Data downloaded was about Kashmir unrest in 2016 (details in the next section) to automatically annotate the unrest data. Six classes were used: Op1, Op2, Op3, Op4, Op5 and Op6 to test our proposed bootstrapping approach. Although classes can be any generic classes like Ekman 's six classes of emotions (Ekman, 1993) but since data is about Kashmir unrest, we have prepared data for mining opinion during the uprising and hence these classes.

Datasets

The dataset was downloaded from twitter using streaming API[9] and twitter 4j[10]. This dataset was downloaded to mine the Kashmir Unrest 2016, also known as Burhan aftermath[11]. Kashmir unrest 2016 is the series of violent protests that happened with the killing of Hizbul Mujahidin (HM)[12] commander Burhan Wani, Wani was killed on 8th July 2016. After his killing, a series of violent protests started in Kashmir Valley. We started downloading the event on Twitter using the Twitter streaming API. The Dataset was downloaded from 12 July 2016 to 31 December 2016. The dataset is composed of 4,928,436 tweets. The words that were used as the query in the twitter streaming API were: Kashmir Unrest, Kashmir crisis, Kashmir blind spot, Kashmir, Kashmir killings, free Kashmir, azaadi, stop killings in Kashmir, go India go back, save Kashmir, Kashmir bleeds. The downloaded dataset contains non-English tweets which were eliminated from it and resulting dataset has 4,072,133 tweets. The sample tweets downloaded are shown in Table 2. Figure 8 shows the frequency of tweets per day of the event during the event. Figure 9 gives the timeline and rate of tweets in log_{10} .

Figure 6. Big-vector formation

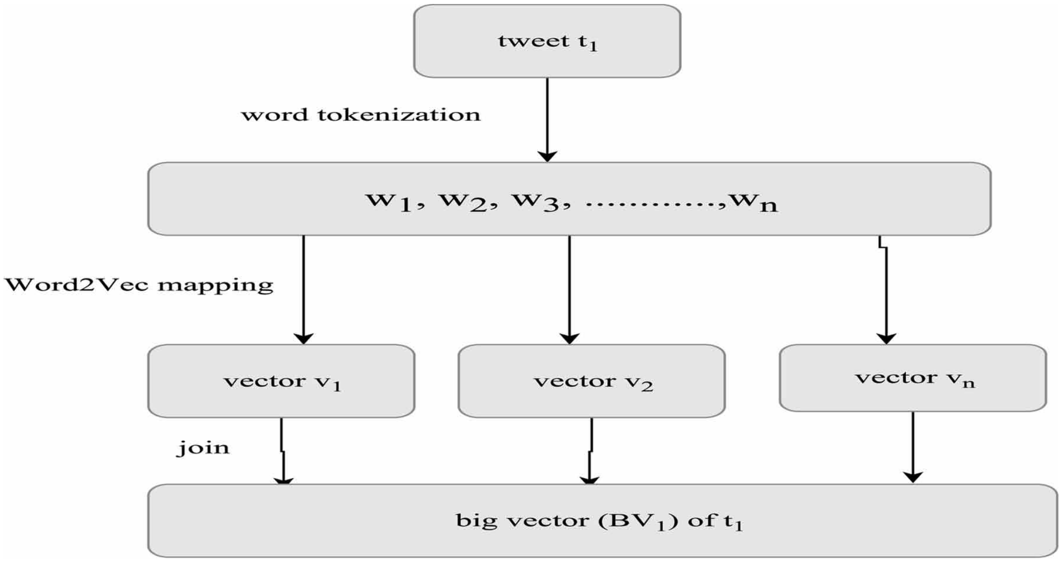
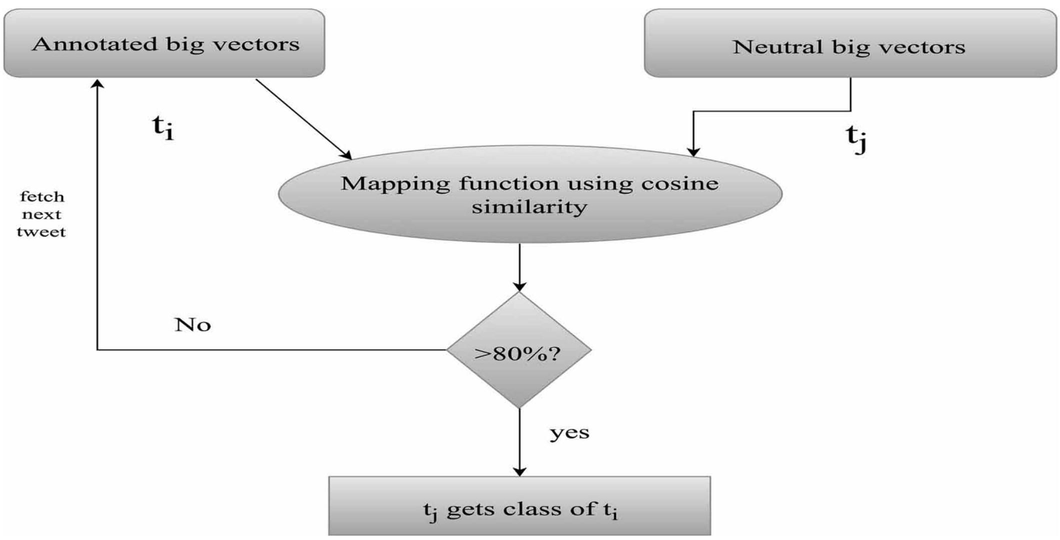


Figure 7. Annotation using BVW



Lexicon Generation Process

For all the six classes viz Op1, Op2, Op3, Op4, Op5, and Op6, we created an exhaustive seed sets about all the six classes. Seed sets were created using the domain experts for Kashmir conflict and refined several times monitoring the twitter feed. Table 3 summarizes the different classes and the seed words in them and most important seed words used for different classes.

Next, we applied Word2Vec for seed extension of the initial seed set. We used the downloaded twitter dataset as the corpus, as our classification is data specific and domain dependent. Thus for learning of Word2Vec, the complete data set was given as corpus and was probed for the words in the initial seed set to achieve seed extension of our initial seed set. The seed extension by Word2Vec was

Figure 8. Rate of tweets per day during unrest in 2016

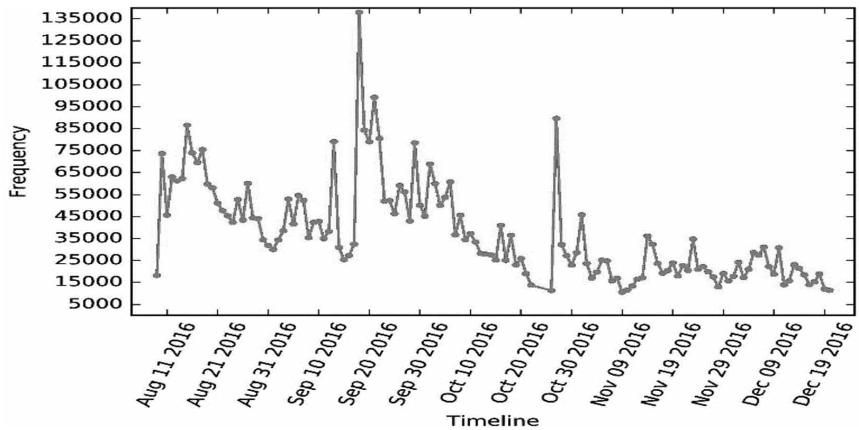
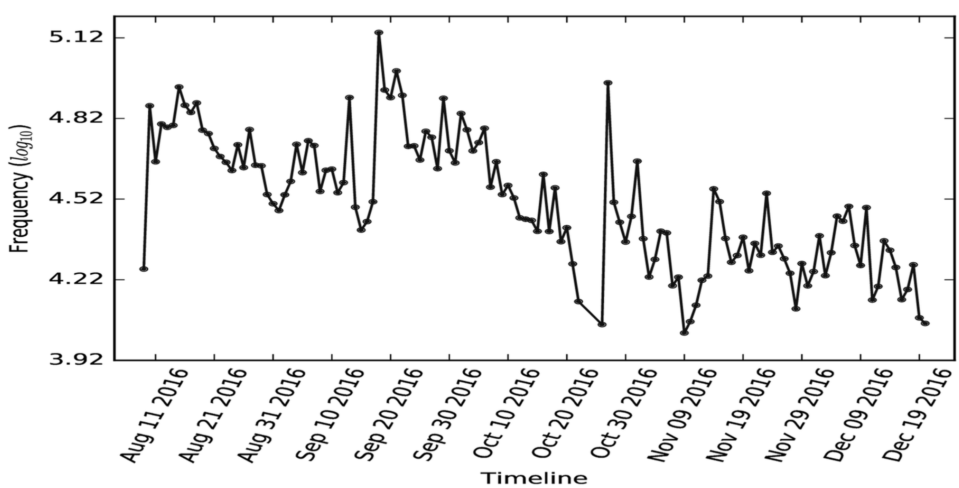


Table 2. Sample tweets in dataset

Serial Number	Tweet
1	{"TweetedOn":" Sat Oct 22 16:08:55 CEST 2016","location":"Jammu And Kashmir","tweet":"RT @user: Eyewitness: Srinagar, Kashmir https://t.co/ZyFqcegjRV https://t.co/Rcm1EqCwdS, "tid":789830899420372992 "
2	{"TweetedOn ":"Sat Oct 22 16:07:12 ","tweet":"RT @username: Break the silence. Stand with Kashmirhttps://t.co/spscfc0fKp","tid":789830468418011140.}
3	{"TweetedOn ":"Sat Oct 22 16:06:36 ","location":"Los Angeles, California, USA","tweet":"Eyewitness: Srinagar, Kashmir https://t.co/ZyFqcegjRV https://t.co/Rcm1EqCwdS","tid":789830318085636098.}
4	{"TweetedOn ":" Sat Oct 22 14:33:42 ","tweet":"Fond of selfies, 14-yr old Sayaar killed by pellets on September 10 Young #Kashmirawaitsjustice @PMOIndia @amnesty https://t.co/wvI6vwpFZB","t id":791256487951757312.}
5	{"TweetedOn ":" Sat Oct 22 14:32:09 ","tweet":"RT @ username: Excellent question from my Justice colleague @ username MP in #PMQs on human rights abuses in Kashmir & campaignfor\u2026", tid":791256097902452736.}

Figure 9. Timeline of Kashmir unrest (time versus frequency in log10)



done twice to achieve seed extension I (we call it Word2Vec I) and then to produce seed extension II (Word2vec II). The output of Word2Vec I was given as input to the Word2Vec II to produce seed extension using Word2Vec. Statistics of seed, the extension is presented in Table 4. Word2Vec is used twice to obtain two seed extensions of the seed set. We stopped after two phases because further extensions were leading to an error in classification. Word2Vec lessens the bias which can lurk in the process of annotation. Skip-gram model of Word2Vec algorithm is employed for annotation. Unified Corpus obtained as enhanced seed lexicon is fed to Word2Vec model for training. To further enrich the seed words we used WordNet (Miller, 1995). Each word in the seed set created in the preceding phase is looked up in the WordNet to retrieve its synonyms, and these synonyms were added to appropriate classes. Figure 10 shows the archetype of enhanced seed lexicon using WordNet. In this Step, each word in the lexicon is explored to determine synonyms by employing the WordNet, and synonyms are having sense similar to the probed word is added to the lexicon belonging to the class to which examined word belonged. Figure 10 shows an example of the seed extension using WordNet. The

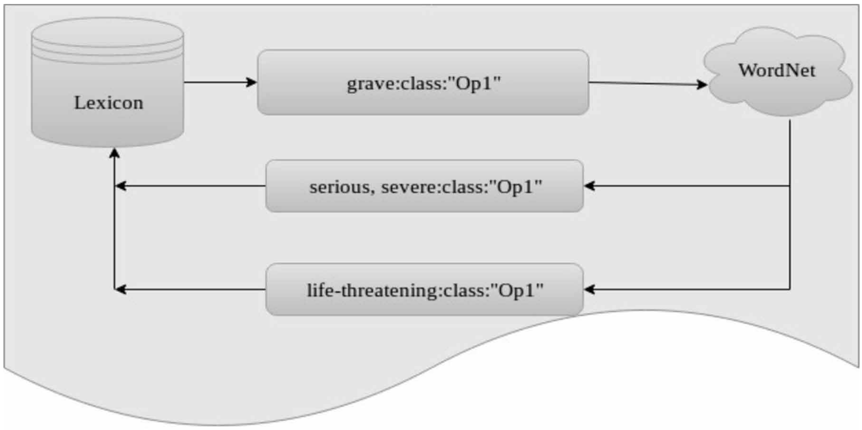
Table 3. Sample of seed words in lexicon

Class	Seed Words	# Seed Words
Op1	India, Kashmir, prison, war, school, human, right, violate, blood, burn, pellet, blind, child, police, kill, child, grave, crpf, beat, force, fire, gun, curfew, Muslim, injury, force, occupy, shell	29
Op2	Geelani, unrest, education, effect, mask, youth, damage, problem, young, terrorist, dirty, illiterate, direction, religion, separatist, stone, pellet, throw, militant, evacuate, want, state, Kashmir, dismiss, local	26
Op3	Kashmir, bomb, unrest, Nawaz, trouble, Taliban, isis, terrorist, Balochistan, illegal, fuel, unrest, terror, state, train, camp, pok, blackday, destabilize, claim, Pakistan, ceasefire, violate, Nawaz	25
Op4	Nehru, support, Burhan, Geelani, Kashmir, referendum, dispute, resolve, issue, protest, separatist, love, silent territory, uno, Burhan, hero, martyr, demand, freedom, Kashmir, develop, Congress, Abdullah, Azadi	28
Op5	Islamic, poster Pakistan, valley, beauty, flag, banayaga, love, zindabad, support, Jinnah, Jeeva, zindabad, Muslim, raise, issue, peace, peaceful, Pakistani, fake, strike, China, support, Kashmiri	26
Op6	PDP, Indian, bjp, Modi, game, part, India, Kashmir, with, celebrate, Diwali, legal, accession, Hindustani, discipline, win, army, job, accede, home, protect, love, Modi	26

Table 4. Word2Vec seed extension distribution

Classes	# Seed Words After Enhancement
Op1	231
Op2	137
Op3	97
Op4	225
Op5	40
Op6	98

Figure 10. WordNet seed extension



word "grave" in the lexicon has a class assigned to it as "Op1," the word is probed for synonyms in WordNet and retrieved most suitable synsets of it: "serious," "severe" and "life-threatening". These synsets are supplemented to the lexicon and are labelled with the class of "grave," i.e., Op1. After this process, the lexicon is enhanced by 4693 words more, and results in a lexicon of 6331 words.

All the words in the lexicon were explored for synsets in the Oxford American Writer Thesaurus (D'Alessandro, 2004) and all suitable synonyms were collected and added to the lexicon. Figure 11 shows an example of the seed extension using the Oxford Dictionary for the word "religion." When Oxford Dictionary was probed for the word 'religion', belonging to the Op2 class, following synonyms were retrieved: church, creed, denomination, affiliation body, faith, belief, divinity, faith, community, following, theology, sect, cult, worship, teaching, doctrine, religious group, persuasion. Before adding these to the lexicon, the process checks if the synonyms are already present in the lexicon. The class annotation to the retrieved synonyms is the same as that of the probed word. During the process if synonyms had more than one annotation, they were dropped, i.e., all those words which are found to be at the intersection of the opinion classes are dropped. For example, the class assigned to the synonym "religious group" had both annotations: "Op4" and "Op1," so synonym "religious group" was dropped and was not added to the seed set lexicon. After this process, seed set lexicon is enhanced with 3304 more words, and emerging a lexicon with 9635 words. Table 5 summarizes the count of words in the lexicon after every stage of seed extension.

EVALUATION AND RESULTS

In this section, we describe experiments of the extended bootstrapping algorithm and automatic annotation and show the results. For our experiments, we have used twitter dataset described in the previous section. Dataset had pre-processed tweets with retweets and foreign language tweets removed. Experiments were performed using random 12,000 tweets from the dataset. Crowd-sourcing was employed to obtain human annotations for creating a test set to test our system (Mohammad & Turney, 2013). We used the crowd-sourcing method to manually annotate our chosen 12,000 random tweets as done by (Machedon, Rand, & Joshi, 2013). Among the 12,000 tweets, different tweets sets are assigned to several groups of people such that we had at least three judgments for the tweets of 12,000 set. Maximum voting was implemented, and the resulting dataset had 9,818 tweets where all three judges have agreed on the opinion. Thus, we had an efficient test set of 9,818 tweets to test the effectiveness of the automatic sentiment classifier. Table 5 and Table 6 summarize the no. of tweets belonging to each class in the manually created training set. Figure 12 shows the distribution of tweets.

The evaluation methodology that we used assesses our approach in two dimensions. On one side we are evaluating the usability of the lexicon built using our extended bootstrapping algorithm, and on the other aspect, we are assessing the correctness of our classifier with that of the manual classification, employing an agreement measure. Confusion matrices, accuracy, F1-score, Cohen's kappa, precision, and recall are used to compute the performance of proposed classification algorithm. We have also compared the results of our classification algorithm with the results of other similar techniques and found our results outperforming them, thus confirming the superiority of our proposed method.

Confusion matrices are suitable to compute the effectiveness of the multiclass classifier. Table 7 presents the design of the confusion matrix. The elements in the diagonal positions of our confusion matrix show the correct classifications, i.e. correct predictions of the classifier also known as true positives represented by the tp_Op_i with respect to the false classifications of our classifier. False negatives in the confusion matrix are in the corresponding column of the class. Both false negatives and false positives are misclassifications of our classifier.

Table 8 shows the number of tweets classified using different sub-processes of our classification algorithm.

Figure 11. Oxford Thesaurus seed extension

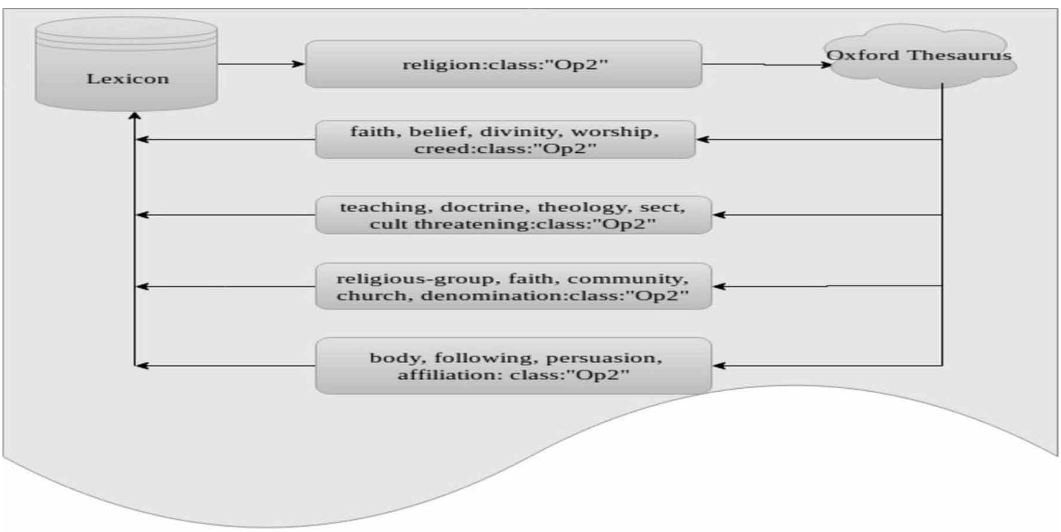


Table 5. Distribution per class in the crowd-sourced dataset

Classes	# Instances of Manually Annotated Corpus
Op1	2544
Op2	1638
Op3	1562
Op4	2534
Op5	171
Op6	837
Neutral	532

Table 6. Seeds after each stage of lexicon generation

Approach	#of Seed Words Retrieved	Total in Lexicon
Initial seed	160	160
Word2vec-I	668	828
Word2vec-II	810	1638
WordNet	4693	6331
Oxford Dictionary	3304	9635

Figure 12. Distribution of tweets

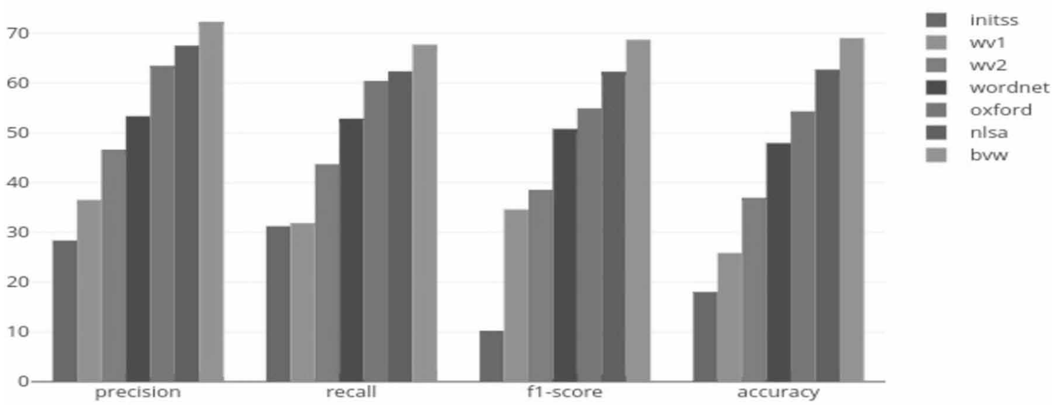


Table 7. Confusion matrix

		Predicted Class	
Actual class		Opi	Opj
	Opi	tp_Opi	e_ $OpiOpj$
	Opj	e_ $OpiOpj$	tp_Opi

Precision is determined by the ratio of correct classifications of the classifier and the sum of correct classifications and false predictions with respect to the class. Mathematically precision of class Op_i:

$$PrecisonOp\_i = \frac{tp\_Op\_i}{tp\_Op\_i + fp\_Op\_i}$$

Recall is determined by the ratio of true predictions upon the sum of true predictions and the predictions that are wrongly classified as false towards a class i. Recall of class Op_i :

Table 8. Distribution per class in the crowd-sourced dataset

	Classes							
	Manual	**Initial Seed Set**	**Word2vec I**	**Word2Vec II**	**WordNet**	**Oxford**	**LSA**	**BVW**
Op1	2544	658	930	1058	1471	1707	1941	2231
Op2	1562	496	640	749	1049	1093	1156	1334
Op3	1638	682	858	891	928	1008	1264	1438
Op4	837	556	1895	1865	1436	1398	1376	1378
Op5	171	254	215	209	182	166	182	162
Op6	2534	1470	1832	2208	2232	2402	2402	2249
Neutral	532	5702	3448	2838	2520	2044	1326	826
Total	9818	9818	9818	9818	9818	9818	9818	9818

$$Recall\ Op\_i = \frac{tp\_Op\_i}{tp\_Op\_i + fn\_Op\_i}$$

F1-score of a classifier is the weighted harmonic mean of the test's precision and recall. Mathematically:

$$\mathrm{F1\_score} = 2* \frac{\mathrm{Precision} - \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Accuracy for the classifier is the ratio of correct predictions upon all the predictions of the classier. Mathematically:

$$\mathrm{Accuracy} = \frac{tp\_OpC}{total}$$

There tp_OpC is the sum of all true positives and total is the total no of classifications.

Cohen's kappa (Cohen, 1960) we have used to measure the effectiveness of our classifier against manually created training set. Mathematically:

$$K = \frac{p_0 - p_e}{1 - p_e}$$

where p_0 is the observed agreement and p_e is the expected agreement.

We have used to leave one out validation. The classification algorithm runs on the dataset processing each tweet and classifies the tweet into one of the six classes or neutral of it does not depict any opinion.

The evaluation results of precision, recall, f1-score and Cohen's kappa for each process are shown in Tables 9- 16. Figure 13 shows the graph for precision, recall and F1 score of our system Table 17 gives the comparison of our classifier with the other techniques.

DISCUSSION AND COMPARATIVE ANALYSIS

The results of our proposed Enhanced approach and its comparison with other similar techniques are discussed in this part. We acquired an accuracy of 68.56% and Macro Average of Precision: 71.14%, Recall: 67.09% and F1_score: 68.04%. The agreement between the results of our proposed Enhanced approach and manual annotation was found to be having micro-average of 58%. The results confirm the competitive competence of our enhanced bootstrapping approach and thus the usability of our proposed system for creation of automatically annotated test data for the supervised classifiers.

Using the initial seed set as a starting point, which was created by domain experts, we created a lexicon of 9635 words. We used both domain-specific words and dictionary-based words to extend our seed set lexicon. The domain-specific word enhancement was done by employing Word2Vec model, and dictionary-based extension was done using WordNet and Oxford thesaurus.

CONCLUSION AND FUTURE WORK

This paper presents a novel approach for scalable and semantically rich automatic annotation of the twitter corpus for Supervised Learning algorithms. We employed our extended bootstrapping algorithm to create the corpus for opinion mining using supervised learning algorithms. Moreover, we presented a method for creation of lexicon utilizing both domain specific models using Word2Vec algorithm and dictionary-based extension using Oxford thesaurus and WordNet. We confirmed the effectiveness of our automatic annotator by using crowd-sourced annotated dataset. The evaluation metrics confirm the contribution of our proposed approach for annotation process since the results of our extended bootstrapping Algorithm and agreement metrics are promising. This shows our automatic annotation algorithm can be used as an alternative for manual annotation. Thus, our proposed algorithm can be efficiently used for the creation of training set for the usage of supervised learning algorithms without much human intervention. Our main conclusion is that:

1. Existing state-of-art in the field is limited and one dimensional. Thus, there is a need to enhance the bootstrapping algorithms to capture the features like domain dependent features;
2. We employed comprehensive evaluation metrics to evaluate the correctness of our approach;
3. The results of the extended bootstrapping algorithm are sound enough to be used as an alternative way of creating corpora for opinion analysis and emotion detection Tasks for Twitter data;

Table 9. Confusion matrix and results on initial seed set

Initial Seed Set Lexicon	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	f1-Score
Op1	197	127	0	96	9	200	29	7.74	29.93	12.29
Op2	170	142	75	9	0	82	18	9.09	28.62	13.79
Op3	110	112	251	0	0	146	63	15.32	36.80	21.63
Op4	47	32	19	252	27	123	56	30.10	45.32	36.17
Op5	41	27	16	8	96	49	17	56.14	37.49	45.17
Op6	483	420	46	0	25	514	30	20.28	34.96	25.66
Neutral	1496	702	1231	518	14	1420	319	59.96	5.59	10.22
Macro-avg								28.37	31.24	10.22
Accuracy								18.03%		

Table 10. Confusion matrix and results of Word2Vec I

Word2Vec I as a Lexicon	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	f 1-Score
Op1	325	45	33	43	27	436	21	12.77	36.02	18.85
Op2	85	205	94	36	9	168	43	13.12	32.03	18.61
Op3	298	71	326	21	7	98	37	19.90	37.99	26.11
Op4	519	427	0	541	0	391	26	64.63	28.54	39.59
Op5	48	27	4	12	86	10	28	50.29	40.00	45.55
Op6	343	416	292	44	24	696	17	27.46	37.99	31.87
Neutral	926	371	898	140	18	735	360	67.66	10.44	18.08
Macro-avg								36.52	31.85	34.60
Accuracy								25.86%		

Table 11. Confusion matrix and results of Word2Vec II

Word2Vec II as a Lexicon	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	f1-Score
Op1	640	96	138	47	5	46	36	27.12	65.21	38.30
Op2	96	342	60	5	4	232	10	21.89	45.66	29.59
Op3	21	17	548	15	0	273	17	33.45	61.50	43.35
Op4	268	320	295	617	21	330	14	73.71	33.08	45.66
Op5	47	22	9	23	92	16	0	36.62	42.02	39.13
Op6	510	109	465	118	36	928	42	36.62	42.02	39.13
Neutral	912	656	123	12	13	709	413	77.63	14.55	24.50
Macro-avg								46.31	43.71	38.56
Accuracy								36.97%		

Table 12. Confusion matrix and results of WordNet

WordNet Seed Set Lexicon	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	f1-Score
Op1	963	146	125	62	23	127	25	37.85	65.46	47.96
Op2	156	568	143	14	17	132	19	36.36	54.14	43.50
Op3	18	6	713	7	0	157	27	43.52	76.83	68.48
Op4	161	205	73	698	5	286	8	83.39	48.60	61.41
Op5	37	16	19	8	95	7	0	55.55	52.19	53.81
Op6	602	228	119	16	15	1235	17	48.73	53.33	51.82
Neutral	607	393	446	32	16	590	436	81.95	17.30	28.56
Macro-avg								53.35	52.83	50.79
Accuracy								47.95%		

Table 13. Confusion matrix and results of Oxford

Oxford as Lexicon	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	f1-Score
Op1	1256	26	194	0	21	192	18	49.37	73.57	59.08
Op2	43	739	75	20	14	189	13	47.31	67.61	55.56
Op3	48	4	812	6	0	127	11	49.57	80.55	61.37
Op4	97	198	79	782	5	216	21	93.43	55.93	69.97
Op5	26	12	23	0	103	2	0	60.23	62.04	61.12
Op6	547	267	73	7	18	1479	11	58.36	61.57	59.92
Neutral	527	366	382	22	10	329	458	86.09	21.87	17.43
Macro-avg								63.48	60.44	54.93
Accuracy								57.33%		

Table 14. Confusion matrix and results of Automatic Annotation by NLSA

Normalized LSA	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	f1-Score
Op1	1463	56	199	10	26	215	19	57.50	73.48	64.51
Op2	31	838	65	108	0	102	12	53.64	72.49	61.65
Op3	139	0	940	0	0	180	17	57.38	74.36	64.77
Op4	63	192	74	808	0	220	19	96.53	58.72	73.02
Op5	21	14	20	5	109	9	4	63.74	59.89	61.75
Op6	507	209	73	15	13	1562	23	61.64	65.02	63.28
Neutral	320	267	267	17	4	246	438	82.33	33.03	47.14
Macro-avg								67.53	62.37	62.30
Accuracy								62.72%		

Table 15. Confusion matrix and results of Automatic Annotation by BVW

BVW	Confusion Matrix							Results		
	Op1	Op2	Op3	Op4	Op5	Op6	Neutral	Precision	Recall	F-Score
Op1	1700	55	214	109	9	30	18	65.8	79.62	72.05
Op2	50	1010	70	99	3	94	18	63.8	75.15	69.01
Op3	140	1	1098	1	1	168	29	67.27	76.35	71.52
Op4	70	192	80	795	2	219	20	75.71	57.69	65.48
Op5	21	14	0	5	109	9	1	76.22	68.55	72.18
Op6	500	211	80	20	15	1599	24	69.85	65.29	67.49
Neutral	100	100	90	21	4	170	420	79.34	46.4	58.55
Macro-avg								71.14	67.09	68.04
Accuracy								68.56%		

Table 16. Cohen's Kappa score

Class	Kappa Score						
	Initial Seed	Word2Vec I	Word2Vec II	WordNet	Oxford	LSA	BVW
Op1	-0.27	-0.17	-0.01	0.16	0.31	0.42	0.54
Op2	-0.08	-0.03	0.07	0.24	0.37	0.44	0.57
Op3	-0.01	0.03	0.201	0.32	0.39	0.48	0.61
Op4	0.23	0.61	0.71	0.81	0.92	0.96	0.96
Op5	0.55	0.49	0.52	0.54	0.59	0.63	0.63
Op6	-0.07	0.02	0.14	0.30	0.43	0.48	0.51
Neutral	0.57	0.67	0.76	0.80	0.85	0.81	0.82

Figure 13. Precision, recall and F1 score of our proposed bootstrapping algorithm

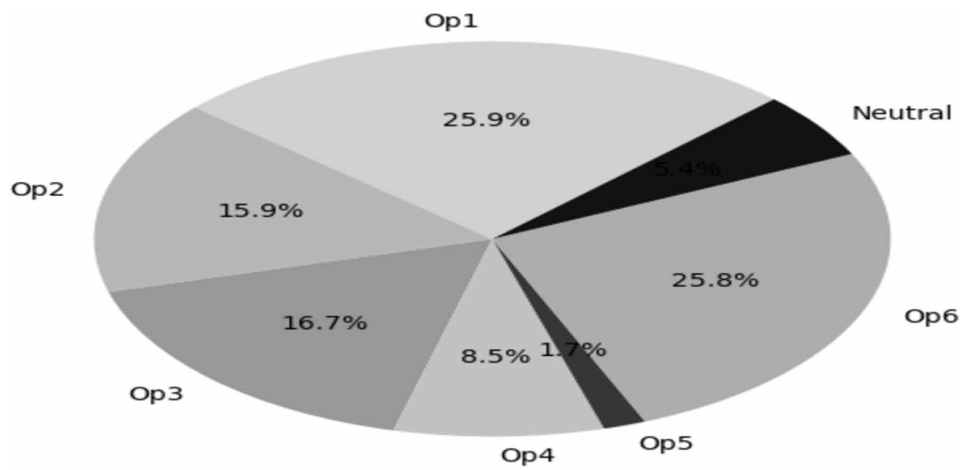


Table 17: Comparative analysis

Technique	Accuracy
Extended Bootstrapping algorithm	68.56%
Canales, Strapparava, Boldrini, & Martnez-Barco, 2016	59.5%
Go, Bhayani, & Huang, 2009	65.2%

4. The significance of our lexicon generation technique for the automatic annotation task results in better performance. Hence, our proposed algorithm allows us to consider bootstrapping as an efficient way to automate the annotation task for the creation of corpora to be used by supervised learning algorithms, without manual annotations which are laborious and time consuming.

Our future research will be focused on using other semantic similarity models like GloVe: Global Vectors for Word Representation (Pennington, Socher, & Manning, 2014) and testing our approach on other corpora.

REFERENCES

Alm, C. O., Roth, D., & Sproat, R. (2005, October). Emotions from text: machine learning for text-based emotion prediction. *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 579-586). Association for Computational Linguistics. doi:10.3115/1220575.1220648

Aman, S., & Szpakowicz, S. (2007, September). Identifying expressions of emotion in text. *International Conference on Text, Speech and Dialogue* (pp. 196-205). Springer. doi:10.1007/978-3-540-74628-7_27

Balabantaray, R. C., Mohammad, M., & Sharma, N. (2012). Multi-class twitter emotion classification: A new approach. *International Journal of Applied Information Systems*, *4*(1), 48–53. doi:10.5120/ijais12-450651

Bifet, A., & Frank, E. (2010, October). Sentiment knowledge discovery in twitter streaming data. *International conference on discovery science* (pp. 1-15). Springer. doi:10.1007/978-3-642-16184-1_1

Canales, L., Strapparava, C., Boldrini, E., & Martnez-Barco, P. (2016, October). Exploiting a bootstrapping approach for automatic annotation of emotions in texts. *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 726-734). IEEE. doi:10.1109/DSAA.2016.78

Cherry, C., Mohammad, S. M., & De Bruijn, B. (2012). Binary classifiers and latent sequence models for emotion detection in suicide notes. *Biomedical informatics insights, 5*, BII-S8933.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46. doi:10.1177/001316446002000104

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Academic Press.

Dadvar, M., Trieschnigg, D., Ordelman, R., & de Jong, F. (2013, March). Improving cyberbullying detection with user context. *Proceedings of the European Conference on Information Retrieval* (pp. 693-696). Springer.

D'Alessandro, P. (2004). *The Oxford American writer's thesaurus*. Oxford University Press.

Desmet, B., & Hoste, V. (2013). Emotion detection in suicide notes. *Expert Systems with Applications*, *40*(16), 6351–6358. doi:10.1016/j.eswa.2013.05.050

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science & Technology*, *38*(1), 188–230. doi:10.1002/aris.1440380105

Ekman, P. (1993). Facial expression and emotion. *The American Psychologist*, *48*(4), 384–392. doi:10.1037/0003-066X.48.4.384 PMID:8512154

Gliozzo, A., Strapparava, C., & Dagan, I. (2009). Improving text categorization bootstrapping via unsupervised learning. *ACM Transactions on Speech and Language Processing*, *6*(1), 1–24. doi:10.1145/1596515.1596516

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, 1(12).

Hakak, N. M., Mohd, M., Kirmani, M., & Mohd, M. (2017, July). Emotion analysis: A survey. *Proceedings of the 2017 International Conference on Computer, Communications and Electronics (Comptelix)* (pp. 397-402). IEEE. doi:10.1109/COMPTELIX.2017.8004002

Hasan, M., Rundensteiner, E., & Agu, E. (2014). Emotex: Detecting emotions in twitter messages.

Java, A., Song, X., Finin, T., & Tseng, B. (2007, August). Why we twitter: understanding microblogging usage and communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis* (pp. 56-65). ACM. doi:10.1145/1348549.1348556

Liu, B., Li, X., Lee, W. S., & Yu, P. S. (2004, July). *Text classification by labeling words* (Vol. 4). AAAI Press.

Shamma, D. A., Kennedy, L., & Churchill, E. F. (2009, October). Tweet the debates: understanding community annotation of uncollected sources. *Proceedings of the first SIGMM workshop on Social media* (pp. 3-10). ACM. doi:10.1145/1631144.1631148

Liu, B., Li, X., Lee, W. S., & Yu, P. S. (2004, July). *Text classification by labeling words* (Vol. 4). AAAI Press.

Machedon, R., Rand, W., & Joshi, Y. V. (2013). Automatic classification of social media messaging using multi-dimensional sentiment analysis and crowdsourcing.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60). Academic Press. doi:10.3115/v1/P14-5010

McFedries, P. (2007). Technically speaking: All a-twitter. *IEEE Spectrum*, *44*(10), 84–84. doi:10.1109/MSPEC.2007.4337670

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), 436–465. doi:10.1111/j.1467-8640.2012.00460.x

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. In Adaptive computation and machine learning. MIT Press.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41. doi:10.1145/219717.219748

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), 436–465. doi:10.1111/j.1467-8640.2012.00460.x

Mohammad, S. M., Zhu, X., Kiritchenko, S., & Martin, J. (2015). Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, *51*(4), 480–499. doi:10.1016/j.ipm.2014.09.003

Neviarouskaya, A., Prendinger, H., & Ishizuka, M. (2010, August). Recognition of affect, judgment, and appreciation in text. *Proceedings of the 23rd international conference on computational linguistics* (pp. 806-814). Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). doi:10.3115/v1/D14-1162

Rodriguez, P., Ortigosa, A., & Carro, R. M. (2012, July). Extracting emotions from texts in e-learning environments. *Proceedings of the 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)* (pp. 887-892). IEEE. doi:10.1109/CISIS.2012.192

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*(6), 1161.

Salton, G. (1991). Developments in automatic text retrieval. *science, 253*(5023), 974-980.

Suttles, J., & Ide, N. (2013, March). Distant supervision for emotion classification with discrete binary values. *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 121-136). Springer. doi:10.1007/978-3-642-37256-8_11

Thelen, M., & Riloff, E. (2002, July). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing* (Vol. 10, pp. 214-221). Association for Computational Linguistics. doi:10.3115/1118693.1118721

Salton, G. (1991). Developments in automatic text retrieval. *science, 253*(5023), 974-980.

Strapparava, C., & Mihalcea, R. (2008, March). Learning to identify emotions in text. *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1556-1560). ACM. doi:10.1145/1363686.1364052

Thelen, M., & Riloff, E. (2002, July). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing* (pp. 214-221). Association for Computational Linguistics. doi:10.3115/1118693.1118721

Plutchik, R., & Kellerman, H. (1980). *Theories of Emotion* (Vol. 1). Academic Press.

Wilson, T., Wiebe, J., & Hwa, R. (2004, July). Just how mad are you? Finding strong and weak opinion clauses. In AAAI (Vol. 4, pp. 761-769). AAAI Press.

Jan, R., & Khan, A. A. (2018). Emotion Mining Using Semantic Similarity. *International Journal of Synthetic Emotions*, *9*(2), 1–22. doi:10.4018/IJSE.2018070101

Purver, M., & Battersby, S. (2012, April). Experimenting with distant supervision for emotion classification. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 482-491). Association for Computational Linguistics.

De Choudhury, M., Gamon, M., Counts, S., & Horvitz, E. (2013). Predicting depression via social media. *ICWSM*, *13*, 1–10.

Qadir, A., & Riloff, E. (2013). Bootstrapped learning of emotion hashtags# hashtags4you. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis* (pp. 2-11). Academic Press.

Yan, J. L. S., & Turtle, H. R. (2016). EmoCues-28: Extracting Words from Emotion Cues for a Fine-grained Emotion Lexicon. *Emotion and Sentiment Analysis*, 40.

Hasan, M., Rundensteiner, E., & Agu, E. (2019). Automatic emotion detection in text streams by analyzing Twitter data. *International Journal of Data Science and Analytics*, *7*(1), 35–51. doi:10.1007/s41060-018-0096-z

Taxer, J. L., Becker-Kurz, B., & Frenzel, A. C. (2019). Do quality teacher–student relationships protect teachers from emotional exhaustion? The mediating role of enjoyment and anger. *Social Psychology of Education*, *22*(1), 209–226. doi:10.1007/s11218-018-9468-4

Hakak, N., & Kirmani, M. (2018). Opinion mining of twitter events using supervised learning. *International Journal of Synthetic Emotions*, *9*(2), 23–36. doi:10.4018/IJSE.2018070102

ENDNOTES

[1] https://blog.twitter.com/official/enus/topics/events/2017/-nuggsforcarter-is-now-the-most-retweeted-tweet-of-all-time.html.
[2] http://blog.twitter.com/2013/happy-birthday-twitter.html
[3] http://twitter4j.org/en/
[4] https://api.twitter.com/1.1/search/tweets.json
[5] http://smsdictionary.co.uk/abbreviations
[6] http://www.netlingo.com/
[7] www.urbandictionary.com
[8] https://developer.oxforddictionaries.com/
[9] https://dev.twitter.com/streaming/overview
[10] http://twitter4j.org/en/
[11] https://en.wikipedia.org/wiki/2016-17-kashmirunrest
[12] Hizbul Mujahidin is a terrorist organization running in Kashmir