STM JOURNALS

Journal of Artificial Intelligence Research & Advances
ISSN: 2395-6720 (Online)
Volume 5, Issue 2
www.stmjournals.com

Opinions Mining of Twitter Events Using Spatial-Temporal Features

Mudasir Mohd*, Rana Hashmy
Department of Computer Science, University of Kashmir, Srinagar, Jammu and Kashmir, India

Abstract

Micro-blogs are a powerful tool to express an opinion. Twitter is one of the fastest growing micro-blogs and has more than 900 million users. Twitter is a rich source of opinions as users share their daily experience of life and respond to specific events using tweets on twitter. In this paper, an automatic opinion classifier capable of automatically classifying tweets into different opinion expressed by them is developed. Also, manually annotated corpus for opining mining to be used by supervised learning algorithms is designed. Opinion classifier uses semantic, lexical, domain dependent and context features for classification. Results obtained confirm competitive performance and robustness of the system. Classifier accuracy is more than 75%, which is higher than the baseline accuracy.

Keywords: *Opinion mining, supervised learning, Twitter events, Twitter corpus*

****Author for Correspondence*** E-mail: mudie.mohammad@gmial.com

INTRODUCTION

Opinion mining is used to refer to the task of automatically determining the opinion expressed in text, phrases, sentences or any piece of writing. However, more generally it is used to determine one's attitude towards a particular event or reaction to an event. Here, attitude means qualitative opinion, feeling or reaction to some situation that triggered the event. Opinion measurement and sentiment analysis for Twitter data are attracting much research both in academia and industry. Millions of users of Twitter are posting tweets about their daily life, and write opinions on a variety of issues. Twitter is easily accessible through smartphones and other web services and thus is a preferred media for communication. Hence internet users are shifting from traditional communication tools like blogs and mailing lists to Twitter. More specifically, as Twitter users are expressing their opinions about several issues including religious matters, political views, and reviews of e-services, products or even movie reviews, Twitter has become a viable source for opinion measurement and detection. Such data is extremely potent for the industry for feedback and political parties to frame their policies and strategies.

Twitter posts are 240 characters long messages called tweets. Java *et al.* in their study suggest that users use Twitter for following reasons: (1) for information source, (2) to be in touch with family and friends and (3) seeking information about trends happing worldwide [1]. Opinion mining involves data mining and Natural Language Processing (NLP) techniques to uncover hidden information and opinions from social web's substantial textual sources. Opinion mining from the text written in natural languages is challenging as it requires a deep understanding of explicit and implicit, regular and irregular, syntactic and semantic rules of language. Therefore, opinion mining is a challenge for NLP researchers for utilizing tools of NLP for efficient and effective opinion mining systems and thus leading to substantial practical impact.

Most of the companies are using opinion mining systems to create and automatically maintain reviews written by their customers for their popular products. Opinion mining can also be used by companies to improve customer relationship. Opinion mining also finds its applications in the recommender systems used by e-commerce sites. Moreover, opinion mining can play an essential part in the antispam policy drafting. Political parties also use opinion mining to know if the people support their decisions and programs or not and use feedback to frame their policies and strategies for future.

In this paper, we discuss opinion mining of Twitter event using state-of-art NLP techniques and machine learning tools. We show how to use Twitter for the corpus of opinion detection system. We have collected a corpus of 4,928,436 tweets about event namely Kashmir Unrest 2016 downloaded from the twitter from 10, July 2016 to 31, December 2016. We then preprocess tweets cleaning noise and performing other text transformations. Then the state-of-art linguistic analysis is performed using NLP techniques and then built an opinion classifier using supervised learning algorithms. We have also developed a rich opinion corpus using crowd-sourcing.

CONTRIBUTIONS

The contributions of our research are:

1) We present a method for efficient preprocessing of text that remove and replace slangs and can correct misspelled words.
2) We have built a vibrant opinion mining corpus for supervised learning algorithms.
3) We have performed state-of-art NLP techniques for linguistic analysis of text to uncover explicit and implicit, regular and irregular, and syntactical and semantic language rules.
4) Built a supervised opinion classifier capable of mining opinions in Twitter.
5) Performed evaluation experiments that confirm competitive performance and robustness of our system.

ORGANISATION

Rest of the paper is organized as: Next part of the paper discusses related works in the field of opinion analysis and sentiment analysis. Then we present our automatic opinion mining algorithm, discuss experiments and evaluations and lastly, the conclusion and future work.

RELATED WORKS

Since the growth of social media, research in the field of opinion mining and sentiment analysis has grown many folds. Opinion mining is done at two levels: document level opinion mining and sentence level opinion mining. Turney used the average semantic orientation of phrases to calculate documents polarity [2]. He used pointwise mutual similarity to compute semantic orientation between documents and extracted phrases. Turney and Littman used cosine similarity and latent semantic analysis at document level for opinion analysis [3]. Dave *et al.* used term frequencies on uni-gram, bi-gram, and tri-gram to classify reviews on Amazon [4]. Das and Chen used financial documents to classify sentiment polarity [5]. Pang *et al.* used state-of-art machine learning algorithms to classify movie reviews for their sentiment polarity [6]. Pang *et al*. also used subjectivity with minimum graph cuts along with machine learning to detect sentiment in movie reviews as document-based opinion classification [7]. Our work is entirely different from these approaches as we have used sentence level opinion classification with state-of-art machine learning algorithms that are feature classifiers.

Zhuang *et al.* used sentence level classification using high-frequency keywords and high-frequency opinion keywords for classification of movie reviews [8]. They used dependency graphs to identify feature-opinion pairs, but they used a fixed set of keywords and thus their system was limited. Popescu *et al.* used relaxation labeling approach for finding semantic orientation for words since their approach used only higher frequency keywords than the threshold value in their experimental set [9]. Sentence level opinion mining also suffers from the drawbacks like data sparsity (fewer data about theme), large datasets not available, noisy datasets, and misspelled words. Moreover, traditional approaches like TF-IDF (term frequency-inverse document frequency), bag of words do not perform well in sentence level opinion mining.

Most of the work on opinion classification, sentiment analysis and emotion analysis at the sentence level is focused on classifying tweets using machine learning classifiers [10–13]. All these approaches employ supervised learning classifiers trained from the manually annotated corpus for opinion mining task. Emotion lexicons and rule-based strategies, however, do not produce satisfactory results when used for opinion mining at the sentence level.

Several works have suggested extending emotion lexicons with distribution semantic algorithms like Word2Vec for classification [14, 15]. All these works focus on sentiment analysis, opinion mining or emotion detection at the fine-grained or coarse-grained level, however, they lack subjectivity and domain dependent features. In this work, we will be using domain dependent features and show how they help in improving the efficiency of the classifier. In this paper, we have built an opinion corpus that can be used by other classifiers. We propose novel domain dependent features useful for opinion mining purpose.

METHODOLOGY

In this section, a novel opinion classifier is proposed, built using supervised classifier. The supervised classifier is built using lexical, contextual, semantic, domain dependent and morphological features. We have used pre-processing to normalize the dataset by reducing noise and correct misspelled words. Our classifier clarifies tweeter feed into six opinion classes, viz.; anti-India, pro-India, anti-Kashmir, pro-Kashmir, anti-Pakistan, pro-Pakistan and neutral depending on the opinion described in the tweet. We have built an opinion corpus of 9818 tweets that is useful for several opinion related tasks.

DATASET

We used twitter streaming API and twitter4j to download our dataset for opinion mining task. Dataset was downloaded to automatically for mining opinion during Kashmir unrest 2016. Violent protests erupted in the Kashmir valley (J&K India) during summer of 2016 due to the killing of Hizbul Mujahidin commander, Burhan Wani. Burhan got killed on 8th July 2016. On 12th July 2016, we started downloading the tweets about the event until 31st December 2016. The dataset had 4,928,436 tweets. The downloaded dataset has non-English tweets as well we removed them and kept English only tweets, re-tweets were also discarded. Figure 1 shows timeline of event where the frequency is plotted against date. Table 1 shows a sample of tweets downloaded.

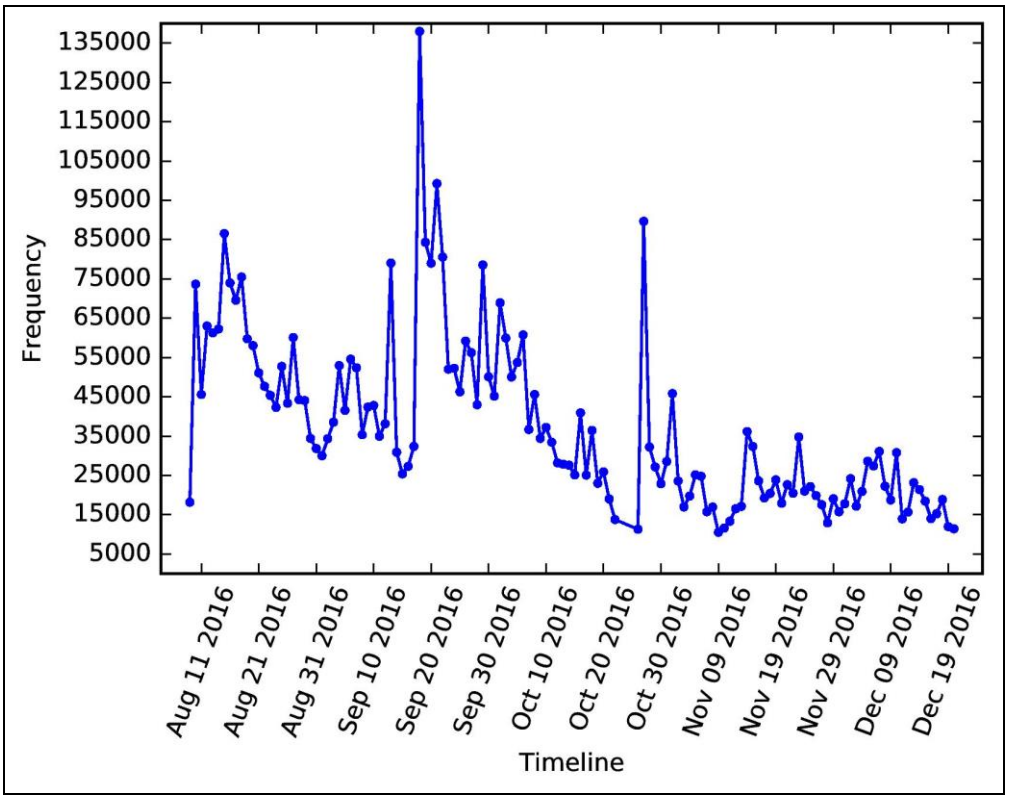

Fig. 1: *Timeline of the Event.*

Table 1: *Sample Tweets.*

Serial Number	Tweet
1	Created At: Wed Oct 26 14:28:10 CEST 2016, location: Bangalore, tweet: RT @Username: 19 schools burnt down in Kashmir in 3 months. Dear @username, pls confirm if schools were useless because they didn't have a\u2026", lang: en, rtweet: true, tid:791255097766350849, username: username
2	Created At: Wed Oct 26 14:28:11 CEST 2016, location: India, tweet: Burning schools is new way of celebrating\#Diwali\#kashmir, lang: en, rtweet: false, tid: 791255101570555905, username: username
3	Created At: Wed Oct 26 14:28:12 CEST 2016, tweet: RT @maulinshah9: @username During the unnatural alliance of\#PDP_BJP in the state and NDA in the Center, \#Kashmir has faced maximum days of \u2026, lang: en, rtweet: true, tid: 791255105563570176, username: username
4	{"**tweet**": "@saysdiyanag @CatchNews This one statement made me worry more about his wife than the jawans in Kashmir. So much power", "**tid**": 791255101570556914}
5	{"**tweet**": "@TimesNow i get stunned when u still trying to get consensus over Kashmir..it is never going to take place..the Q is why gov. (1)", "**tid**": 763044533558874113}
6	{"**tweet**": "Anyone watching Times Now? Better not. Hell break loose over Kashmir & la ilaaha illalla.Where are these debates taking us2. Shameful indeed.", **"tid",** 763046521180790784}

Automatic Opinion Classification

Our automatic opinion classifier is developed using supervised classifier. We have used multiclass Support Vector Machine ($SVM^{multiclass}$) [17]. Our automatic classification algorithm consists of following parts:

- Twitter scrapping module,
- Pre-processing,
- Lexicon generation,
- Feature selection,
- Generation of opinion corpus, and
- Automatic classifier.

We discuss each of the sub-processes in following subsections. Figure 2 shows overall functioning of our system.

Twitter Scrapping Module

We made a scraper for collecting tweets related to the unrest. Scrapper was developed using twitter4j and twitter streaming API. Figure 3 shows the architecture of our scrapping module.

Our scrapping module receives a set of query string words which are probed to Twitter for retrieving relevant tweets.

Preprocessing Module

Our preprocessing module removes noise from tweets. Tweets are unstructured and noisy which need to be cleaned for processing before passing to the supervised classifier for classification. Our preprocessing module uses following steps to clean the tweet:

- Stanford core NLP package is used to tokenize tweets into words [18].
- We use English dictionary modules to remove slangs and abbreviations from the tweet. All words of the tweet are fed to dictionary module for retrieving their meaning. All words that have no proper meaning returned are passed to the word replacer module for replacement with proper meanings. The dictionary module is used WordNet and Jspell [19]. The word replacer module uses Netlingo and urban dictionary.
- Stemming from each word in the tweet is done using potter stemmer [20].
- Stop words removed.
- Usernames and URLs removed.
- The case of tweet changed to lowercase.

Lexicon Generation

We built an opinion lexicon for feature extraction to be used by the opinion classifier. The lexicon contains seed words representing opinion mined. For all the classes of opinion that are to be mined in the Twitter dataset, we built lexicon. Lexicon was initially developed using initial seed for all classes and later on extended by using distributional semantic similarity models like Word2vec [21]. Word2vec algorithm is a semantic distribution algorithm which allows us to capture domain dependent features. Word2vec gives vectors of each word, where vectors are the semantic extensions of the word probed. Word2vec was trained using the entire dataset downloaded and thus enabled us to capture domain-specific extensions of the word in the initial seed sets. Table 2 shows seed words in the initial seed set and Table 3 shows the distribution of words in the lexicon after seed extension using Word2Vec.

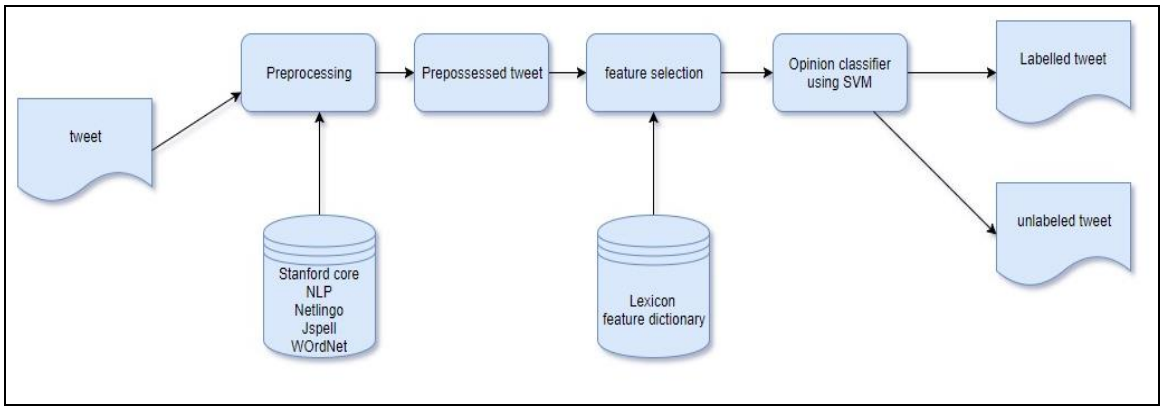

Fig. 2: *Overall Working of the System.*

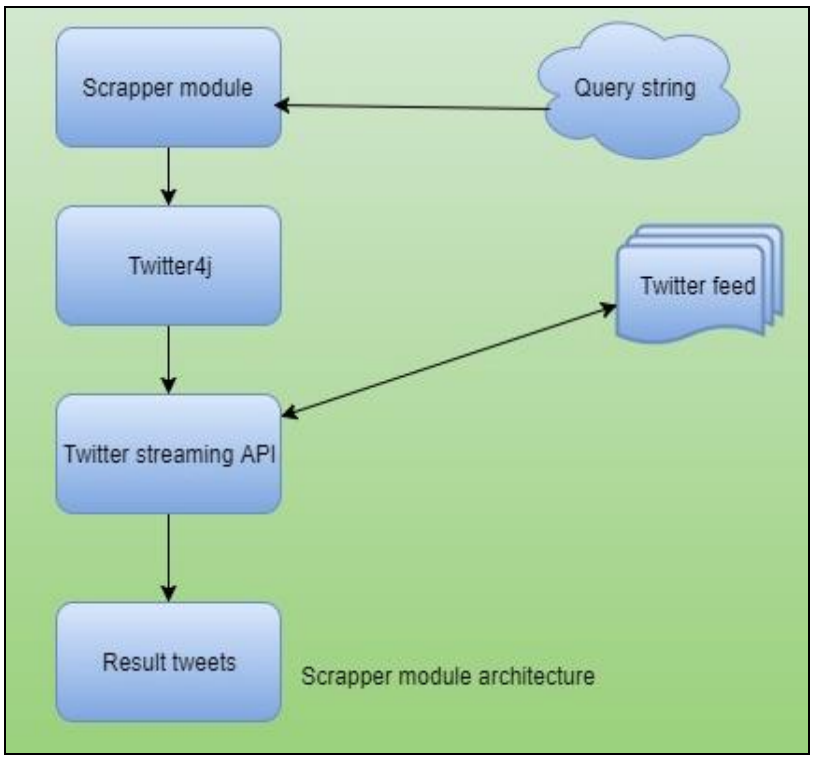

Fig. 3: *Scrapping Module Architecture.*

Table 2: *Seed Words in Lexicon.*

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

Feature Selection

Features were extracted using java. We used following features for our classification:

- *Unigrams*: Unigrams are nouns, adjectives, noun verbs from the corpus.
- *Bigrams*: Bigrams are the randomly selected two unigrams.
- *Trigrams*: Trigrams are the three words. The word after and before the middle word depends on the middle word. The middle word is chosen from the seed set, after reading the tweet if the seed word exists there we choose words before and after the seed word in the tweet to form the trigram.
- *Adjective*: Adjectives are extracted from the corpus using the Stanford NLP toolkit.
- *POS-Bigrams*: Using the same NLP toolkit we extracted adjectives and proper nouns from the corpus to generate the POS-bigrams.
- *Seed words*: Seed words in the lexicon chosen as features.

Opinion Corpus

Any corpora annotations are reliable only when we have multiple annotations for the same tweet by multiple judges. We used five annotators to annotate our corpus. A total of 12,000 tweets randomly selected from the dataset were chosen for annotation. Annotators did not receive any training but were shown samples to make them understand what kinds of annotation were required. Each annotator has to label a tweet with a category among the six opinion classes namely: Anti-India, Anti-Kashmir, Anti-Pakistan, Pro-Kashmir, Pro-India, Pro-Pakistan, or Neutral if no opinion is found. Thus opinion corpus has seven labels, six opinion classes plus neutral. From the 12,000 tweets, we choose only those tweets where at least three annotators have agreed, and thus our opinion corpus has 9,818 tweets. Table 4 shows the distribution of opinion classes in the opinion corpus.

Table 3: *Distribution after Extension.*

Class	#Seed Words after Extension
Anti-India	231
Anti-Kashmir	137
Anti-Pakistan	97
Pro-Kashmir	225
Pro-Pakistan	40
Pro-India	98

Table 4: *Distribution of Opinion Classes in the Manually Annotated Corpus.*

Class	# of Instances in Manually Annotated Opinion Corpus
Anti-India	2544
Anti-Kashmir	1638
Anti-Pakistan	1562
Pro-Kashmir	2534
Pro-Pakistan	171
Pro-India	837
Neutral	532
Total	9818

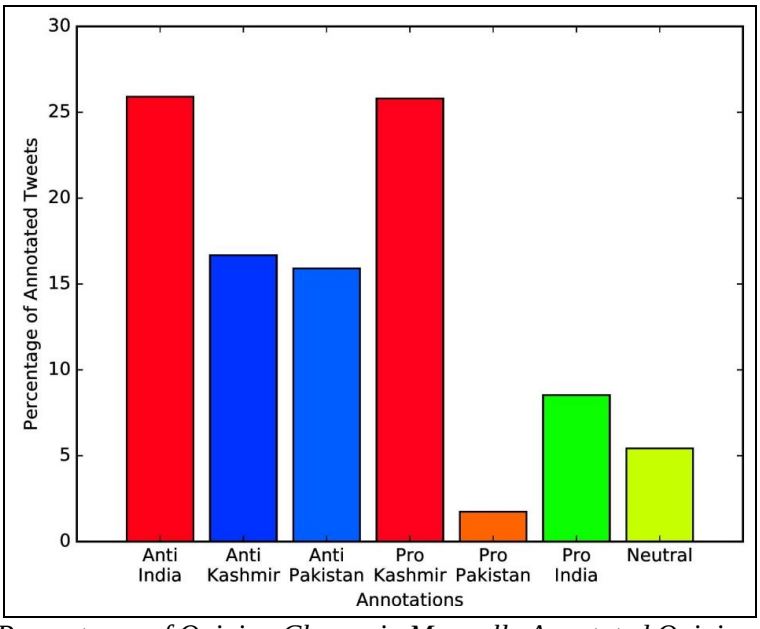

Fig. 4: *Percentages of Opinion Classes in Manually Annotated Opinion Corpus.*

Automatic Classifier

Our automatic opinion classifier uses Support vector machine ($SVM^{multiclass}$) for classification. It processes input as vectors in a feature weight combination. We transform features generated in the preceding phase into vectors for classification. We have used different combinations of features to test our classification.

$SVM^{multiclass}$ uses multiclass formulization described by Crammer and Singer [22], but it optimizes it with an algorithm that is fast in linear case.

For a training set (x_1,y_1) ... (x_n,y_n) with labels y_i in [1..k], it finds the solution of the following optimization problem during training:

$\min 1/2\Sigma_{i=1..k} w_i*w_i + C/n \Sigma_{i=1..n} \xi_i$
for all y in [1..k]: $[x_1 \bullet w_{yi}] >=$
$[x_1 \bullet w_y]+100*\Delta(y_1, y)-\xi_1$
for all y in [1..k]: $[x_n \bullet w_{yn}] >=$
$[x_n \bullet w_y]+100*\Delta(y_n,y)-\xi_n$

C is the usual regularization parameter that trades off margin size and training error. $\Delta(y_n, y)$ is the loss function that returns 0 if y_n equals y, and 1 otherwise.

EVALUATION AND RESULTS

In this section, we describe the experiments of our automatic opinion classifier and show the results. For our experiments, we have used downloaded dataset as discussed in the previous section. Dataset was normalized using pre-processing and foreign language tweets, and re-tweets were removed. Experiments were performed on a manually annotated dataset of 9,818 tweets having majority opinion of at least three annotators.

Confusion matrices, accuracy, F_1-score, precision, and recall were used to evaluate the proposed opinion classifier. Confusion matrices are a good indicator of measuring the efficiency of the multi-class classifier. The general structure of a confusion matrix is shown in Table 5. The diagonal elements represent the true positives (correctly classified) of the classification process, represented by tp in the table and other elements in the corresponding row represent the false positives. Elements of the corresponding column represent the false negatives. Misclassifications are both false positives and false negatives.

Precision is the ratio of true-positives to the sum of false negatives and true positives. Mathematically, the precision of class A is defined as:

$$precison\ A = \frac{tpA}{tpA + fpA}$$

Where, tp_A is the true positives of class A and fp_A is the false positives of all classes corresponding to class A.

The recall is the ratio of true positives to the sum of true positives and false negatives. Mathematically, recall of class A is defined as:

$$recall\ A = \frac{tpA}{\text{tpA} + \text{fnA}}$$

Where, tp_A is the true positives of class A and fn_A is the false negatives of all classes corresponding to class A.

The harmonic mean of both precision and recall gives F-score of the classifier. Mathematically F_1-score of classifier is:

$$\text{F1} - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Accuracy is the ratio of correct classifications to the total classifications. Mathematically,

$$\text{Accuracy} = \frac{tpc}{total}$$

The tpc is the sum of all true positives and total is the total no. of classifications. Table 6 shows the confusion matrix and results of our classifier. Figure 4 shows the graph of our classifier.

Table 5: *General Structure of Confusion Matrix.*

		Predicted class			
		A	**B**	**C**	**D**
Known class	A	tp_A	eAB	eAC	eAD
	B	eBA	tp_B	eBC	eBD
	C	eCA	eCB	tp_C	eCD
	D	eDA	eDB	eDC	tp_D

Table 6: *Confusion Matrix and Evaluation.*

	Confusion Matrix							Results		
	AI	**AP**	**AK**	**PI**	**PP**	**PK**	**Neutral**	**Precision**	**Recall**	**F1-Score**
AI	1658	56	220	119	10	26	19	66.23	75.52	70.57
AP	49	1008	65	108	0	102	12	64.53	75.56	69.41
AK	139	0	1102	0	0	180	17	67.27	76.63	71.64
PI	63	192	74	810	0	220	19	96.77	58.70	73.13
PP	21	14	0	5	110	9	0	64.32	67.90	66.06
PK	493	209	73	15	13	1623	23	64.04	66.27	65.13
neutral	80	97	84	15	12	185	442	83.08	53..21	65.09
Macro-average								72.32	67.72	68.71
Accuracy								75.05%		

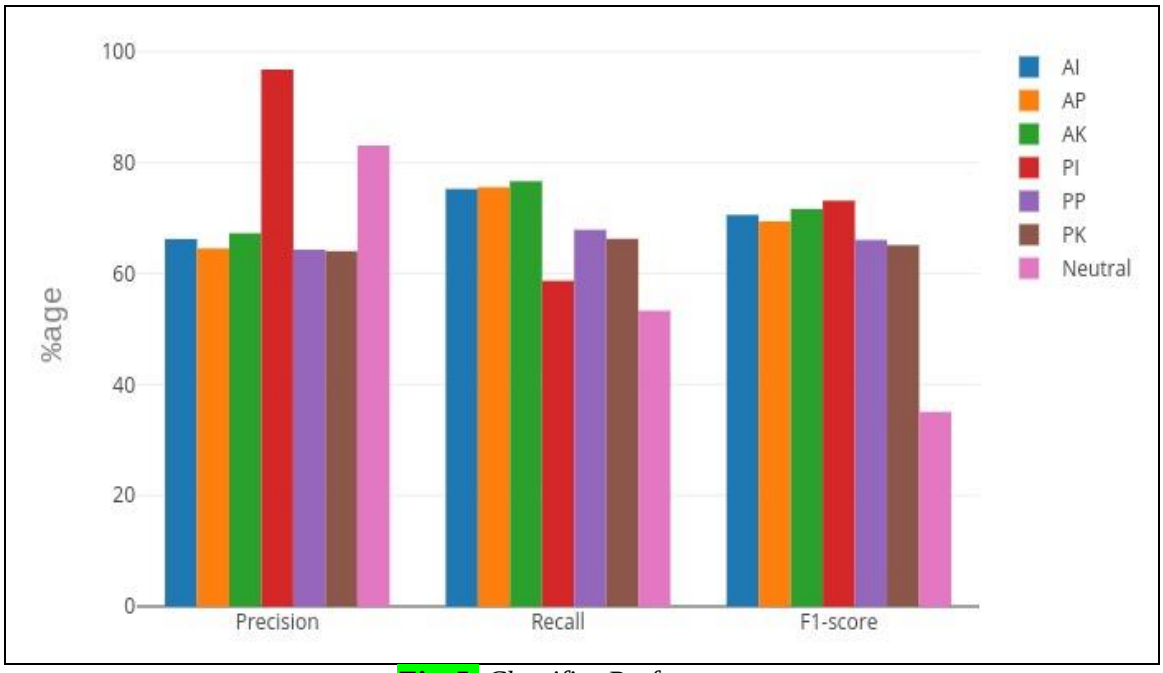

Fig. 5: *Classifier Performance.*

CONCLUSION AND FUTURE WORK

This research is about the opinion mining using supervised classifiers, use of extensive features to improve the classifier accuracy. This research aims to build an efficient opinion corpus that is usable for opinion mining purpose. The supervised classifier is trained on well-crafted opinion corpus, and results confirm the competitive performance and robustness of the system. Our proposed algorithm achieved an overall accuracy of 75.05%. With the micro-average precision of 72.32%, recall of 67.72% and f1-score of 68.71% which is higher than the baseline accuracy.

In our future work, we will use more lexicographical, and topic modulation features to improve the performance of our supervised classifier. Usage of Glove for lexicon building will also be explored.

REFERENCES

1. Java A, Song X, Finin T, *et al.* Why We Twitter: Understanding Microblogging Usage and Communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, ACM. Aug 2007; 56–65p.
2. Turney Peter D. Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics. 2002.

3. Turney Peter D, Littman Michael L. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Trans Inform Syst (TOIS)*. 2003; 21(4): 315–346p.
4. Dave Kushal, Steve Lawrence, Pennock David M. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *Proceedings of the 12th International Conference on World Wide Web*, ACM. 2003.
5. Das Sanjiv, Mike Chen. Yahoo! for Amazon: Extracting Market Sentiment from Stock Message Boards. *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*. 2001;
6. Pang Bo, Lillian Lee, Shivakumar Vaithyanathan. Thumbs up?: Sentiment Classification Using Machine Learning Techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*, Association for Computational Linguistics. 2002.
7. Pang Bo, Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics. 2004.
8. Zhuang Li, Feng Jing, Xiao-Yan Zhu. Movie Review Mining and Summarization. *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ACM. 2006.
9. Popescu Ana-Maria, Bao Nguyen, Oren Etzioni. OPINE: Extracting Product Features and Opinions from Reviews. *Proceedings of HLT/EMNLP on Interactive Demonstrations*, Association for Computational Linguistics. 2005.
10. Kouloumpis Efthymios, Theresa Wilson, Moore Johanna D. Twitter Sentiment Analysis: The Good the Bad and the OMG! *ICWSM*. 2011;
11. Saif Hassan, Yulan He, Harith Alani. Semantic Sentiment Analysis of Twitter. *International Semantic Web Conference*. Berlin, Heidelberg: Springer; 2012.
12. Sarlan Aliza, Chayanit Nadam, Shuib Basri. Twitter Sentiment Analysis. *Information Technology and Multimedia (ICIMU), 2014 International Conference on*. IEEE; 2014.
13. Balabantaray Rakesh C, Mudasir Mohammad, Nibha Sharma. Multi-Class Twitter Emotion Classification: A New Approach. *International Journal of Applied Information Systems (IJAIS)*. 2012; 4(1): 48–53p.
14. Canales Lea, *et al*. Exploiting a Bootstrapping Approach for Automatic Annotation of Emotions in Texts. *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. IEEE; 2016.
15. Mikolov Tomas, *et al*. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv: 1301.3781*. 2013.
16. Hall Mark, *et al*. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explor Newsletter*. 2009; 11(1): 10–18p.
17. Joachims Thorsten, Thomas Finley, Chun-Nam John Yu. Cutting-Plane Training of Structural SVMs. *Mach Learn*. 2009; 77(1): 27–59p.
18. Manning Christopher, *et al*. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2014.
19. Miller George A. WordNet: A Lexical Database for English. *Commun ACM*. 1995; 38(11): 39–41p.
20. Van Rijsbergen Cornelis J, Stephen Edward Robertson, Porter Martin F. *New Models in Probabilistic Information Retrieval*. London: British Library Research and Development Department; 1980.
21. Mikolov Tomas, *et al*. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv: 1301.3781*. 2013.
22. Crammer K, Singer Y. On the Algorithmic Implementation of Multi-class SVMs. *JMLR*. 2001; 2(3/1): 265–292p.

Cite this Article
Mudasir Mohd, Rana Hashmy. Opinions Mining of Twitter Events Using Spatial-Temporal Features. *Journal of Artificial Intelligence Research & Advances*. 2018; 5(2): 36–44p.