Unsupervised Learning Method for Sorting Positive and Negative Reviews Using LSI (Latent Semantic Indexing) with Automatic Generated Queries

Sheikh Muhammad Saqib[1], **Fazal Masud Kundi**[1], **Shakeel Ahmad**[2]

[1]Institute of Computing and Information Technology Gomal University. Pakistan.
[2]Faculty of Computing and Information Technology in Rabigh (FCITR) King Abdul Aziz University (KAU) Jeddah Saudi Arabia

Summary
Companies having different products rely on their customers' reviews of their products. After purchasing a product, a customer will post some reviews on the website. Before purchasing the product, another customer will read the feedback from these reviews before making a decision. It is very important for companies to analyse such reviews, whether they are negative or positive, to enhance the quality of their product. Researchers are now working on separating the negative or positive comments by means of sentiment scores. A sentiment analysis can be performed through supervised learning or unsupervised learning, where each method requires a lot of pre-processing work for the analysis. This paper presents a strategy for sentiment classification using Latent Semantic Indexing (LSI). The purpose of LSI is to rank documents with respect to a given query. However, in this study, a mechanism was provided to generate positive and negative queries automatically. These queries were then used to obtain negative and positive scores so that a decision could be made on the basis of these scores. This method was not only aimed at separating the positive and negative reviews, but also at providing ranked lists of positive or negative comments. These lists are very important for companies to carry out significant reviews from the top of the negative list, and shining reviews from the top of the positive list. The sorted lists of positive/negative reviews based on the LSI scores generated by the positive/negative queries were checked manually, and were proved to be highly satisfactory, while the precision of the sentiment analysis was 0.67, which could be increased by a little bit of tuning of the given reviews. The MCC value also showed that this method was acceptable.

Key words:
Reviews, Sentiment Score, Sentiment Analysis, Pre-Processing, Supervised and Unsupervised Learning, Latent Semantic Indexing

1. Introduction

From this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1].

Nowadays, the Internet is available for everyone, i.e. through smart phones. This medium is being used for shopping, making hotel reservations, booking tickets, and also for people to obtain information before going to an important place. By using the reviews of different people, a decision can be made on which place is better. So, online reviews are very important for quality-related issues because negative opinions about a product on the web can change the minds of 80% of customers with regard to their purchasing decision [1].

How can a company improve the quality of a product, place, etc. from the huge amount of reviews? Many studies have been carried out with regard to sentiment analysis, which is about determining the sentiment orientation of a review or comment [2][3][4][5]. Sentiment orientation means that a positive opinion will be an exact positive, and a negative opinion will be an exact negative [6]. The view, assessment or feeling of a person towards a product [7], aspect [8], or service is known as a sentiment [9][10][11]. Such a feeling, which is either positive or negative, can be assigned a score. Most of the work in sentiment analysis is based on binary classification, which means that reviews or blogs are divided into "positive" and "negative" classes [12][13]. The classification of text sentiments can be done in two ways, i.e. through machine learning and score-based approaches [14][15]. Machine learning uses training data [16], while the other method uses several attributes of an entity to determine the scores. In the score-based approach, opinions can be oriented as positive or negative [17][18]. The work of [19] used a combined approach of SentiWordNet and lexical resources to determine the scores for slangs. A lexicon-based approach for extracting sentiment orientations of opinions has been used for scoring [17][20][21]. Studies by [22][23] used lists of positive and negative words to determine the polarity of a sentence by creating a training matrix and random forest classifier based on supervised learning. A sentiment analysis can be performed using different methods [24][25][26][27], with each method having an improved accuracy with respect to the previous one. Although a lot of work is involved in sentiment orientation [28] with the use of adjectives, frequent nouns and noun phrases, sentiment shifters, handling of 'but' clauses, decreased and increased quantity of an opinionated item; high, low,

Manuscript received January 5, 2018
Manuscript revised January 20, 2018

increased and decreased quantity of a positive or negative potential item; desirable or undesirable facts; deviations from the norm or a desired value range; and the production and consumption of resources and waste, etc., these are very important for determining the polarity of a document or sentence [29][30][21]. However, a large amount of online data is generated every day with unprecedented speed and size. Most of the available information on the Internet is in text and unstructured forms, i.e. online reviews, blogs, chats, and news. An aspect-based sentiment analysis, which can be carried out by using only particular aspects [21][31][32][33], requires less effort compared to a sentiment analysis of an object with respect to all aspects. Reviews are rated according to a particular object, so there should be a direct method to determine whether a review is positive or negative. LSI (Latent Semantic Indexing) is better for such a purpose [34]. LSI [35] has been used for the clustering of documents and for concept representations. An extended method based on LSI is able to filter unwanted emails in Chinese and English [36]. Here, the author proposed a framework using the LSI method with a little bit of pre-processing work to determine the polarity of a review. According to the experimental results, the sorted positive and negative lists were highly satisfactory, while the sentiment analysis achieved a precision and accuracy of 0.67 and 0.64, respectively, where these deficiencies could be removed by handling the ‘but’ clauses.

This study made the following key contributions:

- A method was proposed for generating an automatic positive query (PosQ) and negative query (NegQ) using a lexicon of positive and negative words, which is necessary for Latent Semantic Indexing, i.e., there was no necessity to provide the queries as input.
- A list of positive /negative reviews, which were closely related to the PosQ/NegQ, i.e. the most positive and most negative reviews, was generated.
- All the reviews were separated or classified into negative and positive polarities using the LSI scores.

2. LSI-Based Sentiment Analysis of Reviews

The reviews and queries make up the input for the LSI algorithm. The queries are generated automatically. After the processing using LSI, the output is scored. On the basis of these scores, a decision will be made as to whether the review or comment is negative or positive.

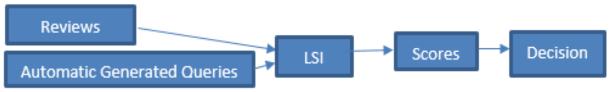

Fig. 1 LSI-Based Sentiment Analysis

Basically, two types of queries will be generated, namely, positive queries and negative queries.

2.1 Positive Query and Negative Query

In Fig-2, reviews with a list of positive words will generate a positive query (PosQ), while reviews with a list of negative words will generate a negative query (NegQ). Scores through the LSI will be calculated twice. The first score is a positive score (PosScore) with a PosQ, and the second is a negative score (NegScore) with a NegQ.

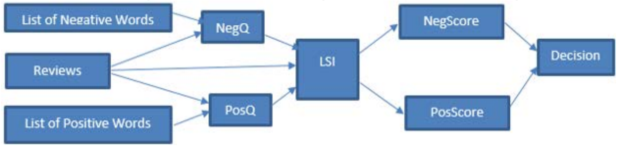

Fig. 2 LSI with Negative and Positive Queries

LSI, which was proposed by Deerwester in 1990, is an efficient information retrieval algorithm [37]. Basically, in LSI, there is a cosine similarity measure between the coordinates of a document vector and the coordinates of a query vector. If this value is 1, it means the document is 100% closer to the query, if it is 0.5, it means the document is 50% closer to the query, and if it is 0.9, it means the document is 90% closer to the query.

The major point now is finding the coordinates of each document and query. A singular value decomposition (SVD) can determine the points or coordinates of a document and query. Through the SVD, three matrices, S, V and U, which will be used for further processing, can be determined by a matrix. To determine the values of such variables, the SVD requires a matrix. The matrix consists of rows and columns containing integers, while here the inputs are different text documents. A feature matrix can be obtained by calculating the frequencies of each word. This means that first, a feature matrix is created from all the documents, and then, the SVD is calculated. After this, the supporting variables, S, V and U, will be calculated by using NumPy (Numeric Python). The coordinates of all the documents will be determined from S, and these coordinates will be merged with the query to obtain the query coordinates. Finally, a cosine similarity function will be applied to these coordinates to find the documents that are closest to the query [34].

$$q = \mathrm{q}T\ \mathrm{U}_{\mathrm{K}}\ \mathrm{S}_{\mathrm{K\text{-}1}} \quad (1)$$

Then, the score of each review, d with respect to a query, q can be determined by using the product equation:

$$Sim(q,d) \quad = \frac{q.d}{|q|\ |d|} \qquad (2)$$

A decision will be made on the basis of the NegScore and PosScore. The calculated values of both scores will be between 0 and 1. A review will be considered to be positive if the PosScore is greater than the NegScore. Otherwise, the review will be negative.

2.2. Process for the Creation of NegQ and PosQ

First, the reviews of an object will be divided into words or chunks. Next, intersections of these words with a List of Positive Words (LPW) will be made. Then, this list will be updated with the PosQ to form the PosQ Union. This process will be carried out for all the reviews, as shown in Table-1, where the PosQ for Review1 was "a d" and then with Review2 it was "a d g". Since there were two reviews, so PosQ would be "a d g".

Table 1: Creating a Positive Query with Two Reviews

A=Review1: a b c d e
B=Review2: d e f g h
C=List of Positive words: a d I j k m g
D= A Intersection C: a d
PosQ = PosQ Union D: a d
D= B Intersection C: d g
PosQ = PosQ Union D: a d g

The process is the same for the negative query, NegQ; with the difference being that here a List of Negative Words (LNW) will be used. The following figure shows the creation of a positive query, PosQ, and a negative query, NegQ.

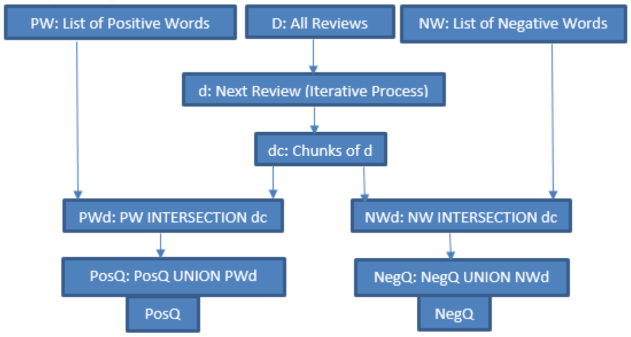

Fig. 3 Process for the Creation of Positive and Negative Queries

From Fig-3 above, it is clear that the PosQ, NegQ, and chunks of reviews will be passed through the LSI to calculate the PosScore and NegScore, respectively. A decision will be made on the basis of these scores. Eq-3 shows all the reviews, and Eq-4 depicts the tokens. Some pre-processing will be carried out such as replacing the word "bad" for "not good", so a review can have only positive and negative words and no negations, as calculated in Eq-5. Eq-6 is used for filtering the tokens, i.e. removing all the stop words.

$$S = \mathrm{a}_{x=1}^{n} S_x \qquad (3)$$

$$(x) = \mathrm{a}_{i=1}^{n} T_i \qquad (4)$$

$$FT(x) = \mathrm{U}_{i=1}^{n}\{Antonyme\ (T_i)\ ,\quad if\ T_{i-1} \notin Negations \qquad (5)$$

$$FT(x) = \mathrm{U}_{i=1}^{n}\{T_i\ ,\quad if\ T_i \notin SW \qquad (6)$$

where x = 1, 2, 3…n, SW means stop words, S represents the total number of reviews, T(x) represents the tokens of the xth review, and FT(x) represents the filtered tokens of the xth review.

The positive query and negative query can be generated as:

$$Pos_Q = \mathrm{a}_{x=1}^{n}\{FT(x)_i\ , \mathrm{a}_{i=1}^{n} \qquad if\ FT(x)_i\ a??\ LPW \qquad (7)$$

$$Neg_Q = \mathrm{a}_{x=1}^{n}\{FT(x)_i\ , \mathrm{a}_{i=1}^{n} \qquad if\ FT(x)_i\ a??\ LNW \qquad (8)$$

where x = 1, 2, 3…n and $FT(x)_i$ means the ith chunk of the xth review. Pos_Q contains those words from all the reviews that belong to the LPW, and Neg_Q contains those words that belong to the LNW.

2.3. Sorted Lists of Negative and Positive Reviews

If there are already positive and negative reviews, then the most positive to the less positive list can be easily determined through the PosQ or the most negative to the less negative list through the NegQ using latent sematic indexing, where the documents closest to a given query can be found. Suppose, there are already known positive reviews. From Eq-6, $FT(x)$ are the filtered tokens of each xth positive review. Then the LSI score, $LSI(Score)_x$ of each positive review based on the LSI can be found through the positive query, Pos_Q.

$$LSI(Score)_x = \mathrm{a}_{x=1}^{n}(LSI_x\ (FT(x), Pos_Q)) \qquad (9)$$

Suppose, there are already known negative reviews. From Eq-6, $FT(x)$ are the filtered tokens of each xth negative review. Then, the LSI score, $LSI(Score)_x$ of each negative review based on LSI can be found through the negative query, Neg_Q.

$$LSI(Score)_x = \mathrm{a}_{x=1}^{n}(LSI_x\ (FT(x), Neg_Q)) \qquad (10)$$

2.4. Sentiment Analysis Based on Positive and Negative Queries

If all the reviews are a mixture of positive and negative reviews, the positive score and negative score of each xth review can be calculated using the PosQ and NegQ, respectively. If the positive LSI score is greater than the negative LSI score, it means that the review is closest to the positive query and can be considered as positive, and vice versa. The calculation is according to Eq-11 below, and the process is shown in Fig-4.

$$R_{decision}(x) = \mathrm{a}_{x=1}^{n}\begin{cases} P_x\ , if\ (LSI_x\ (FT(x), Pos_Q) > LSI_x\ (FT(x), Neg_Q)) \\ N_x\ , else \end{cases} \qquad (11)$$

where $R_{decision}(x)$ is the decision of the xth review. It is positive if its positive score based on the LSI is greater than its negative score based on the LSI.

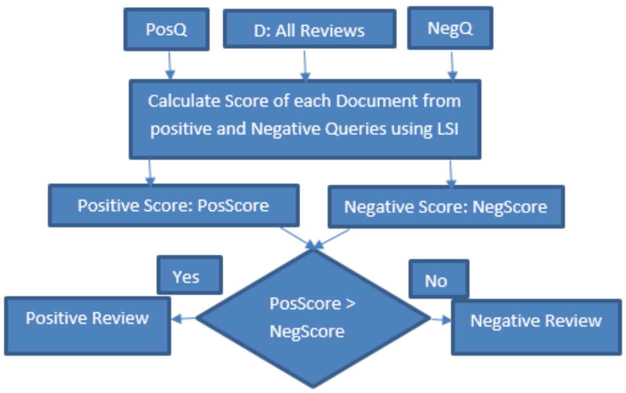

Fig. 4 Decision on the Basis of Positive and Negative Scores

3. Algorithm for Proposed Framework

The algorithm for such a framework required reviews about an entity, a list of positive words and a list of negative words as the input. The processing for the sentiment analysis was done by means of LSI to determine the sentiment orientation of the reviews as the output, as shown in Table-2.

Table 2: Algorithm for LSI-Based Sentiment Analysis

```
Input: Online Reviews/Blogs,
List of Positive Words (PW)
List of Negative Words (NW)
Output: Positive or Negative Review
Function LSISentiment(AllReviews)
1-    D=AllReviews
2-    For d in D
TokenDocument=Token(d)
PosQ=PosQ UNION GenerateQry(TokenDocument,PW)
NegQ=NegQ UNION GenerateQry(TokenDocument,NW)
End For
3-    TokenDocument=Token(D)
4-    PosScoreAllDoc=LSIScore(PosQ, TokenDocument)
5-    NegScoreAllDoc=LSIScore(NegQ, TokenDocument)
6-    If PosScore > NegScore Then
d is positive
else
d is negative
End If
End Function
Function Token(Document)
Return (Chunks of Document)
End Function
Function GenerateQry (TokenDocument,W)
Return (TokenDocument INTERSECTION W)
End Function
Function LSIScore(Query, AllDocuments)
1.    Make Frequency Matrix from AllDocuments
fMat=FrequencyMatrix(AllDocuments)
2.    Make Query Matrix
qMat=QueryMatrix(Query)
3.    Decompose Frequency Matrix in U,S,V using SVD from
      USVT
4.    Determine V from VT
5.    Find UK,Vk and SK
6.    UK = Extracting first two column of U
7.    VK = Extracting first two column of V
8.    SK= Extracting first two column and row of S
9.    Each row of V relates to Coordinates of Document
10.   Find Coordinates of Query from q = qTUkSk-1
11.   First find SK inverse from SK→8
 Each row of inversed SK is the coordinates of each document.
12.   Second q transpose from Query Matrix →2
13.   UK is already determined → 6
14.   Now, find q = qTUkSk-1
15.   q has coordinates of query
16.   Find dot product of q with each document coordinates
      (→11)
 Return (Score of all Documents)
 End Function
```

From Table-2, it is clear that this algorithm consists of different modules.

Tokenizing Review: This module will divide the whole review into different words or chunks.

Generating Queries Module: This module will generate a positive query and negative query by passing a list of positive words (PW) and negative words (NW).

Scoring Module: This module will determine the score of each review with respect to the positive query (PosQ) and negative query (NegQ) using the LSI method.

Decision Module: This module will compare the positive score and negative score of each review to determine whether the review is positive or negative.

4. Results and Conclusion

The major purpose of Latent Semantic Indexing (LSI) is to have a cosine similarity measure between the coordinates of a document vector and the coordinates of a query vector. If this value is 1, it means the document is 100% closer to the query; if it is 0.5, it means the document is 50% closer to the query; and if it is 0.9, it means the document is 90% closer to the query. In this framework, automatic queries are generated, i.e. a negative query and a positive query. Then, a positive score is calculated for the positive query, and a negative score for the negative query of each review. In this way, it can be determined whether a review is very closely related to being positive or negative. Hence, sorted lists can be obtained of known positive and negative reviews based on the LSI scores. Finally, it can be determined whether the review as a whole is negative or positive by comparing their scores. This study used 450 reviews about a hotel. There were 225 positive reviews labelled accordingly from P[0] to P[224], and 225 negative reviews appropriately labelled from N[0] to N[224], and different attempts were made to determine the maturity level of this work. A sample listing of the said datasets is presented in Table-3, where P[0], P[1], P[2] were three selected positive reviews from P[0] to P[224], and N[7], N[9], N[22] were three selected negative reviews from N[0] to N[224].

Table 3: Sample reviews from prepared datasets

Known Classification	***Labels***	***Hotel Reviews***
Positive	P[0]	“It's located in the suburb of Casalotti at the city limits of Rome. I would not recommend it for those who do not want to spend time travelling back and forth to the tourist areas of Rome using public transportation.
	P[2]	“I experienced very good service at any time during the day or late at night. The rooms were always clean and nicely set-up.
	P[3]	“I spent a week in the hotel and I could appreciate the staff friendliness and service. The environment is perfect, a mix of tradition with modern services.”
Negative	N[7]	“Warning! While planning our honeymoon we contacted Cellini based on Rick Steves and Tripadvisor reviews. All seemed well - until we emailed to re-confirm our reservation that they suddenly lost. Interesting that we were quoted a discount rate from the Rick Steves book and then suddenly forgotten about during high travel season.
	N[9]	“my room was extremely hot (in January!) and it's been impossible lower the temperature (the AC only works in summer). I also found it was overpriced for what it offered.”
	N[22]	“The staff were terrible. Hotel is kind of a misnomer. The reason there isnt a picture is because the "hotel" takes two buzzers to get in and you are inside a large, non-descript building. If you werent looking for it, you'd never find it.”

First, all the known positive reviews and PosQ were passed through the LSI algorithm to find the score, where P[122] had the highest score of 0.999998787218472, and P[39] had the lowest score of -0.235799445788549.

The first 10 highest positive reviews are shown in Table-4 below.

Similarly, all the known negative reviews and NegQ were passed through the LSI algorithm to find the scores, where N[37] had the highest score of 0.999988332474444 and N[86] had the lowest score of 0.218629554518072.

The contents of the highest and lowest reviews are shown in the following table.

Table 4: First 10 Positive Reviews.

SNO	***Labels***	***LSI Score With PosQ***	***Names on Hard Drive***
1	P[122]	0.999999	1398.pos
2	P[222]	0.999995	1498.pos
3	P[103]	0.999994	1379.pos
4	P[152]	0.99999	1428.pos
5	P[169]	0.999979	1445.pos
6	P[133]	0.999977	1409.pos
7	P[132]	0.999974	1408.pos
8	P[144]	0.999942	1420.pos
9	P[164]	0.999936	1440.pos
10	P[172]	0.999934	1448.pos

Table 5: LSI Score of Highest and Lowest Positive Reviews

Review with Score	Reviewed Text
Highest Score Review (LSI Score 0.999998787218472)	“We had very high expectations for our evening at Rosemary’s restaurant in Las Vegas. It was a 20-minute drive from the Bellagio and well worth the trip. They were unable to seat us for our reserved time, and we were offered drinks from the bar. When we were seated, the chef prepared us a special starter course. They were so apologetic and attentive that we could not be upset by the wait. The food was outstanding. One of the best meals I have eaten. The service was phenomenal. So often you pay a high price, and have just ordinary service. Not here. I highly recommend Rosemary’s. The manager gave us his card for our next visit, and assured us of special service. We will definitely return.”
Lowest Score Review (LSI Score -0.235799445788549)	“Stayed two nights at this hotel in September, and loved this hotel. The rooms were comfortable but of a reasonable size. Some noise could be heard from other rooms at times but this is a common experience in European hotels. Breakfast was excellent, and the staff very obliging and friendly. This hotel is some distance from the centre of Rome and it took between 40 mins. and an hour to reach the centre of Rome. A pleasant stay overall.”

The first 10 highest negative reviews are shown in the following table.

Table 6: First 10 Negative Reviews.

SNO	*Labels*	*LSI Score with NegQ*	*Names on Hard Drive*
1	N[37]	0.999988	0038.neg
2	N[89]	0.997311	0090.neg
3	N[175]	0.993838	1451.neg
4	N[95]	0.989034	0096.neg
5	N[130]	0.988721	1406.neg
6	N[76]	0.983676	0077.neg
7	N[7]	0.982207	0008.neg
8	N[206]	0.979232	1482.neg
9	N[109]	0.973316	1385.neg
10	N[41]	0.972075	0042.neg

The sorted positive and negative lists of reviews were manually checked and found to be almost correct. If mixed reviews were obtained, the unknown polarity of each review was found, i.e. whether positive or negative, by calculating the LSI score of each review based on the PosQ and NegQ before a decision was made. Out of the 224 positive reviews, it was determined that 188 could be considered as being true positives, i.e. actually positive, and 36 were considered as false negatives, i.e. actually positive. Out of the 224 negative reviews, it was determined that 103 could be considered as true negatives, i.e. actually negative, and 121 were considered as false positives, i.e. actually negative. The following table depicts the PosScore and NegScore scores of 5 positive reviews and 5 negative reviews out of 450 reviews.

Table 7: Sentiment Orientation Based on LSI Positive and Negative Scores

Known Polarity	**Labels**	**PosScore**	**NegScore**	**Decision**
Positive	P[0]	0.264822878	0.248676714	TP
	P[1]	0.490042528	0.475411854	TP
	P[2]	-0.121751247	-0.138315626	TP
	P[3]	0.075604627	0.058936242	TP
	P[4]	0.319988926	0.304116985	TP
Negative	N[0]	0.655928	0.669328287	TN
	N[1]	0.654399	0.667822994	TN
	N[2]	0.888122	0.896202709	TN
	N[3]	0.669228	0.682415227	TN
	N[4]	0.442012	0.457990916	TN

After the experiments, the obtained results is showing in Table-8. The results with respect to the positive sensitivity was 0.83 and with respect to the negative sensitivity was 0.45, thereby indicating that with respect to the positive, the result was satisfactory, while with respect to the negative, greater efforts were required, the reason being that most of the negative comments had a lot of positive words with the "but" clause. For example, in the review, "Its service is good and nice but it is bad for a living place", there were two positive words and single negative word. As such, this review was very close to being a PosQ rather than a NegQ. Therefore, the sensitivity can be improved by handling the "but" clause and doing some pre-processing based on word sense disambiguation by deleting irrelevant and duplicate reviews from the dataset, removing stop words, replacing repeated spaces with single space characters, converting each word into a lower and singular word, and lemmatization [5,22], so that the word can be found in a lexicon of positive and negative words.

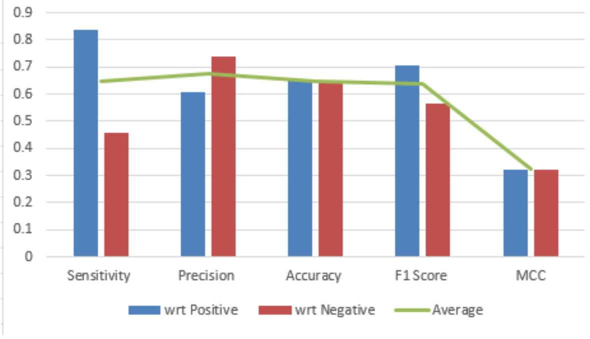

Fig. 5 Graphical Representation of Results

The MCC (Matthews Correlation Coefficient) is a correlation coefficient between the observed and predicted binary classifications; it returns a value of between −1 and +1. A coefficient of +1 represents a perfect prediction, 0 is no better than a random prediction, and −1 indicates total disagreement between the predicted and observed values. An MCC value that is greater than 0 and less than +1 indicates that the proposed method is acceptable. The precision of 0.67 and accuracy of 0.64 can also be increased based on the above-mentioned issue of pre-processing.

Table 8: Experimental Results.

Measures	**wrt Positive**	**wrt Negative**	**Average**
Sensitivity	0.8393	0.4598	0.64955
Precision	0.6084	0.741	0.6747
Accuracy	0.6496	0.6496	0.6496
F1 Score	0.7054	0.5675	0.63645
MCC	0.3233	0.3233	0.3233

References

[1] "3 Reasons Urgent Care Facilities Should Care About Online Reviews," 31 july 2017, 2017. [Online]. Available: https://resources.reputation.com/reputation-com-blog/3-reasons-urgent-care-facilities-should-care-about-online-reviews.

[2] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," Expert Syst. Appl., vol. 72, pp. 221–230, 2017.

[3] F. Wu, Y. Huang, and Z. Yuan, "Domain-specific sentiment classification via fusing sentiment knowledge from multiple sources," Inf. Fusion, vol. 35, pp. 26–37, 2017.

[4] wang maoquan, C. Shiyun, X. yufei, and Z. lu, "EICA at SemEval-2017 Task 4: A Simple Convolutional Neural Network for Topic-based Sentiment Classification," Proc. 11th Int. Work. Semant. Eval., pp. 737–740, 2017.

[5] D. Bollegala, T. Mu, and J. Y. Goulermas, "Cross-Domain Sentiment Classification Using Sentiment Sensitive

Embeddings," in IEEE Transactions on Knowledge and Data Engineering, 2016, vol. 28, no. 2, pp. 398–410.

[6] B. Liu, "Sentiment Analysis and Opinion Mining," Synth. Lect. Hum. Lang. Technol., vol. 5, no. 1, pp. 1–167, 2012.

[7] J. Jin, P. Ji, and R. Gu, "Identifying comparative customer requirements from product online reviews for competitor analysis," Eng. Appl. Artif. Intell., vol. 49, pp. 61–73, 2016.

[8] L. Shu, H. Xu, and B. Liu, "Lifelong Learning CRF for Supervised Aspect Extraction," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 148–154.

[9] F. e-M. K. Khan, B.B. Baharudin, A. Khan, "Mining opinion from text documents," Adv. Res. Comput. Commun. Eng., vol. 3, no. 7, pp. 217–222.

[10] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008.

[11] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, "A Review of Feature Extraction in Sentiment Analysis," J. Basic. Appl. Sci. Res, vol. 4, no. 3, pp. 181–186, 2014.

[12] S. Wang, D. Li, Y. Wei, and H. Li, "A Feature Selection Method Based on Fisher ' s," Springer-Verlag Berlin Heidelb., pp. 88–97, 2009.

[13] A. Esmin and S. Matwin, "Hierarchical Classification Approach to Emotion Recognition in Twitter Hierarchical Classification Approach to Emotion Recognition in Twitter," 2012, no. March 2016.

[14] W. Y. and L. H. A. Wang S, Li D, Song X, "A feature selection method based on improved fisher's discriminant ratio for text sentiment classification," Expert Syst. Appl., vol. 38, no. 7, p. 8696–8702., 2011.

[15] L. C. H. and C. H. Chen LS, "A neural network based approach for sentiment classification in the blogosphere.," J. Informetr., vol. 5, no. 2, pp. 313–322, 2011.

[16] T. Brychcín, M. Konkol, and J. Steinberger, "Machine Learning Approach to Aspect-Based Sentiment Analysis," in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 817–822.

[17] F. M. Kundi, A. Khan, S. Ahmad, and M. Z. Asghar, "Lexicon-Based Sentiment Analysis in the Social Web," J. Basic. Appl. Sci. Res, vol. 4, no. 6, pp. 238–248, 2014.

[18] S. Muhammad and F. Masud, "MMO: Multiply-Minus-One Rule for Detecting &amp; Ranking Positive and Negative Opinion," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 5, pp. 122–127, 2016.

[19] M. Z. A. Fazal Masud Kundi, Shakeel Ahmad, Aurangzeb Khan, "Detection and Scoring of Internet Slangs for Sentiment Analysis Using SentiWordNet Fazal," Life Sci. J., vol. 11, no. 1, pp. 66–72, 2014.

[20] J. Fiaidhi, O. Mohammed, S. Mohammed, S. Fong, and T. Kim, "Opinion Mining over Twitterspace : Classifying Tweets Programmatically using the R Approach," Digit. Inf. Manag. (ICDIM), Seventh Int. Conf. on. IEEE, pp. 313–319, 2012.

[21] A. Jeyapriya and C. S. K. Selvi, "Extracting aspects and mining opinions in product reviews using supervised learning algorithm," in 2nd International Conference on Electronics and Communication Systems, ICECS 2015, 2015, pp. 548–552.

[22] D. K. Kirange, R. R. Deshmukh, and M. D. K. Kirange, "Aspect Based Sentiment analysis SemEval-2014 Task 4," Asian J. Comput. Sci. {&} Inf. Technol., vol. 4, no. 8, pp. 72–75, Aug. 2014.

[23] Deepak Kumar Gupta and Asif Ekbal, "Supervised Machine Learning for Aspect based Sentiment Analysis," in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 319–323.

[24] T. Shaikh and D. Deshpande, "A Review on Opinion Mining and Sentiment Analysis," in IJCA Proceedings on National Conference on Recent Trends in Computer Science and Information Technology, 2016, no. 2, pp. 6–9.

[25] A. Alghunaim, M. Mohtarami, S. Cyphers, and J. Glass, "A Vector Space Approach for Aspect Based Sentiment Analysis," Proc. NAACL-HLT 2015, pp. 116–122, 2015.

[26] M. Cuadros, S. Sebastian, G. Rigau, E. H. Unibertsitatea, and S. Sebastian, "V3: Unsupervised Aspect Based Sentiment Analysis for SemEval-2015 Task 12," no. SemEval, pp. 714–718, 2015.

[27] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4 : Sentiment Analysis in Twitter," Proc. 11th Int. Work. Semant. Eval., vol. 3, no. 4, pp. 502–518, 2017.

[28] V. S. Rajput, "Overview of Use of Natural Language Processing in Sentiment Analysis based on User Opinions," Int. J. Adv. Res. Comput. Sci. Softw. Eng. An, vol. 6, no. 4, pp. 594–598, 2016.

[29] S. M. Street and B. Liu, "Identifying Noun Product Features that Imply Opinions," Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol., no. 2009, pp. 575–580, 2011.

[30] S. S. Htay and K. T. Lynn, "Extracting product features and opinion words using pattern knowledge in customer reviews," Sci. World J., vol. 2013, 2013.

[31] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: Mining Customer Opinions from Free Text," in Proceedings of the 6th international conference on Advances in Intelligent Data Analysis, 2005, pp. 121–132.

[32] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06, 2006, pp. 43–50.

[33] S. Gojali and M. L. Khodra, "Aspect based sentiment analysis for review rating prediction," in 4th IGNITE Conference and 2016 International Conference on Advanced Informatics: Concepts, Theory and Application, ICAICTA 2016, 2016.

[34] S. M. Saqib, K. Mahmood, and T. Naeem, "Comparison of LSI algorithms without and with pre-processing : using text document based search," Accent. Trans. Inf. Secur., vol. 1, no. 4, pp. 44–51, 2016.

[35] A. Huang, D. Milne, E. Frank, and I. H. Witten, "Clustering Documents using a Wikipedia-based Concept Representation," in PAKDD 2009: Advances in Knowledge Discovery and Data Mining, 2009, pp. 628–636.

[36] Q. YANG, "SUPPORT VECTOR MACHINE FOR CUSTOMIZED EMAIL FILTERING BASED ON IMPROVING LATENT SEMANTIC INDEXING," in , Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 2005, pp. 18–21.

[37] G. J. Phadnis N, "Framework for document retrieval using latent semantic indexing," Int. J. Comput. Appl., vol. 94, no. 14, 20