A Systematic Approach to Retrieve Semantically Similar Documents

Tariq Hussain Laghari, Syed Sajjad Hussain and Manzoor Hashmani

***Abstract*—Finding required information on any particular topic give rise information retrieval (IR) problems. Information retrieval techniques resolved these issues but these techniques are unable to extract opinions from unstructured text.Hence, for this problem opinion mining techniques are utilized. Opinion mining extracts opinions from unstructured data source efficiently but it extracts opinions with polarity either negative or positive. The single topic or product where one opinion is similar to another, opinion mining is unable to detect semantic similarity between opinions which are semantically similar. There are other tools available too (Turnitin, Viper, PlagScan) for retrieving the text or documents on the basis of text syntax.However, a system is required which retrieves text with respect to semantics of the text or opinions despite of text syntax. In this study we proposed a system to retrieve text on basis of semantic similarity among text or opinions from a collection of texts (entire corpus). This system utilizes Latent Semantic Indexing to find the semantic similarity in Recognizing Textual Entailment (RTE)dataset. Features of text dataset (RTE) are term frequencies obtained from term document matrix.Implementation of our proposed system showed convincing results by retrieving semantically similar documents from RTE-3 dataset. We provide multiple queries and system retrieved documents similar to each query respectively. In addition we analysed input parameters affecting the output results which can be understood from variation in evaluation matrices that are Precision and Recall.**

I. Introduction

In this innovative era of digital media, advanced internet services and communication medium everyone can access desired information in no time. The desired information includes text, images, audio and video, most widely desired information is text. Apart from that, rapid advancement of internet technology, communication bandwidth, and data storage motivates the stakeholders to add rich contents on websites/ web portals. This exponential growth of contents results into the tremendous rise in information volume on internet. Let us keep in view some facts of information volume growth. Murray of Cyveillance [1] assessed contents on web as 2.1 billion pages and growing with the rate of 7.3 million web pages a day (July 2000). The indexed documents on the web is above 5.04 billion pages (May 12, 2014). These are figures for general content on internet but focusing on the text content, every single minute 510 comments are posted on most popular social media website (Facebook). This rapid increase in volume of text on internet ascends the information retrieval issue. The reason for that is, the available text information is unstructured in nature. Unstructured text is the text written in free text style i.e. blogs, wikis, reviews, social media comments etc. Due to this property of text, retrieving unstructured text information is problem namely information retrieval problem. These problems are resolved by different techniques and approaches of information retrieval. The techniques used for information retrieval are efficient but there are some limitations in them. There are different domains where text retrieval is needed on the basis of semantic similarity of text documents. There are several tool which retrieves the documents on the basis of text syntax (Turnitin, Viper, PlagScan). However, these tools are found to be deficient in retrieving semantically similar text documents.

The recent advances in the world of communication media and internet, connected the whole world on internet where users can communicate with each other, send and receive data or information with each other. Apart from that every user on daily basis provides feedback on any content or information present on the internet. These feedbacks can be pages liked, ratings provided on any product, content liked and disliked on social networking sites. All these feedbacks are kind of user's emotions or feeling or sentiments which they share on the internet or among their groups. In other manners by which the people or end users share their knowledge or information is in the form of text. This text content is found in many categories i.e. comments on any subject or topic in social sites, comments on any product (usually by consumers sharing pros and corns), text information on blogs, reviews on movies, games, software. In general all these categories of user added text information represents their sentiments or opinions on any particular topic, subject, product, content. This is how every individual is author on social forums and also gets remarks on shared information. This is the most growing trend of today's generation and opinions are becoming most important textual information.Opinion Mining is also termed as Subjectivity Analysis or Sentiment Analysis because it deals with the textual information a person (end user) shared on web in the form of opinions. Actually these opinions reflect the emotions / sentiments of the user on particular topic. Specifically this term is defined as:

- Opinion Mining / sentiment analysis is a kind of natural language processing which determines the intentions of writer towards a particular topic / subject or product.

  OR

  Opinion Mining / sentiment analysis is the study of recognizing the stated opinion on a specific subject / topic.

Opinion mining approaches are the basic techniques and following them the other tasks are performed which are coved by this domain including Opinion Aggregation and newly growing sub-domain contradiction analysis. Summarizing the

product reviews provided by the users on any product is the aggregation of opinions present in the form of reviews and attracted by several researchers [2], [3], [4] and [5]. Finding opinionative sentences task is found beneficial for overall Opinion mining by some researchers [6], [3], [7], [8], [9], and [10]. Classifying neutral opinions is witnessed as an effective method [11].

II. PROPOSED METHODOLOGY

According to goals and aims of this research work we focused on the suitable approach to be utilized for semantically similar text retrieval. The basic aim is to use a systematic approach to retrieve the text which is semantically or meaning wise similar present in the entire corpus of opinionative text. So, particularly this task is to retrieve opinions having similar semantics not to extract and classify the opinions. Due to this reason, this research work proposes an approach that specifically focuses on the semantics or meanings of the text not on the syntax of the text.

A. Architecture of Proposed System

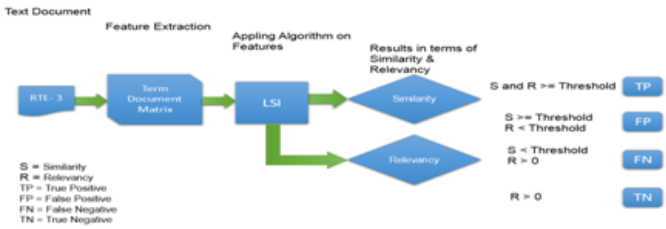

Fig. 2.1: Work Flow of Semantic Text Retrieval

In this research we propose to use an efficient technique which specifically deals with text retrieval of semantically similar text fragments known as Latent Semantic Indexing. This is a indexing and retrieval technique to find underlying or hidden concepts behind the words or terms. For the purpose of evaluating the capability of this system we use the text dataset which is mainly used for textual entailment task. Components of this system are defined briefly and figure 2.1 shows the system flow.

B. Dataset Description

We selected the dataset from the challenges named Recognizing Textual Entailment. There are number of challenges held but for this system we selected RTE-3 dataset.

1) Recognizing Textual Entailment (RTE) Challenges Overview: RTE challenges provide a platform where different groups around the world participates to develop recognizing textual entailment systems i.e. systems that are used to find entailments between the text. Several applications of Natural Language Processing (NLP) that performs Textual entailment tasks are: Question Answering (QA), Summarization, Machine Translation and Paraphrasing, Information Extraction (IE), Queries based Information Retrieval (IR) [12]. The primary focus of RTE challenges is to treat the Textual entailment task individually, the reason for that is to develop a system specifically focusing on the process of recognizing or finding textual entailments. This research is focused on evaluating a system which can be utilized for contradiction finding so text data needed for this task is found in RTE data sets and one of the data set is chosen to evaluate contradiction analysis systems.

2) Recognizing Textual Entailment (RTE-3) Dataset: This dataset consists of:

- 800 pairs of text for development.
- 800 pairs of text for testing.

After the success of RTE-2 and interest of researchers, the next RTE-3was introduced with some new features. Each pair consists of a brief text ”T” up to a paragraph and a short sentence hypothesis ”H”. Unlike previous datasets of RTE this RTE-3data set contains longer texts and makes this dataset more realistic scenario based. Another advancement of RTE-3 dataset is annotations according to 3-way decision in terms of entailment as shown in figure 2.2:

- ”YES” (entails)
- ”NO” (contradicts)
- ”UNKNOWN” (doesn’t entail but is not a contradiction).

The pair annotated ”YES” represents that text t entails hypothesis h which means topic of text t and hypothesis h is similar and does not contradicts each other. The pair annotated ”NO” represents that text t does not entails hypothesis h which means topic of text t and hypothesis h is similar but text t and hypothesis h contradicts each other. The pair annotated ”UNKNOWN” represents that text t neither entails hypothesis h nor contradicts hypothesis h which means topic of text t and hypothesis h is not similar so text t and hypothesis h neither entails nor contradicts each other.The basic reasons for choosing RTE-3 dataset are:

- The length of text ”T” in all pairs was increased for more realistic evaluation prior to previous RTE-1 and RTE-2 datasets.
- The longer text ”T” were marked ”L”.
- RTE-3 dataset is annotated according to all three possible outputs and all four NLP tasks: i.e. Question Answering (QA), Information Extraction (IE), Summarization, and Queries based Information Retrieval (IR).

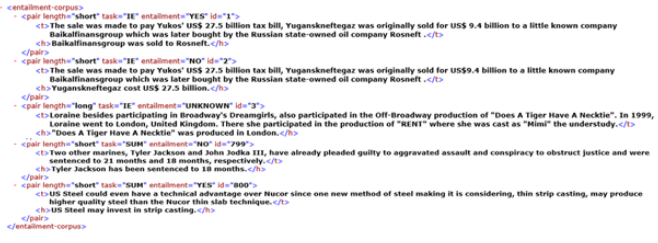

Fig. 2.2: Sample RTE-3

C. Feature Extraction

The proposed algorithm uses a matrix as feature and that matrix is known as Term Document Matrix which comprise of rows as words or terms and columns as sentences or documents.RTE-3 dataset is available in .xml extension file with annotations, in .txt extension file it has just text sentences. Columns represent documents (sentences of rte-3) and rows represent terms (words of rte-3). Each element of Term Document Matrix is term frequency of a particular term/word corresponding to the dictionary / collection of words found in entire text file. This matrix is generated by a MATLAB toolbar named TMG for preprocessing of text documents which create a sparse matrix called index to generate term document matrix [13].

D. Algorithms

The algorithm named Latent Semantic Indexing is applied to matrix obtained as feature. LSI is an indexing method and uses Singular Value Decomposition (SVD) for the decomposition of feature matrix. In addition this system also calculates the relevancy between the query terms and document terms in corpus.

1) Latent Semantic Indexing (LSI) Description: Latent Semantic Indexing is an indexing and retrieval technique used to recognize and analyse meaning/semantics/concepts present in an unstructured collection of text.Latent Semantic Indexing (LSI) is a technique to analyse text documents for the purpose of finding the concepts/meaning/semantics of underlying in text documents. The reason behind the use of LSI is the capability to mine the conceptual content of text by forming relations among the terms occurring in related situations. The fundamental difficulty in finding relevant documents from search words is to match the semantics hidden in the term/words of unstructured text. LSI resolves this issue by matching and comparing terms/words and text documents in a concept space. Text documents are also denoted as bags of words, in which the frequency of terms/words is more significant than the arrangement of words.The patterns of terms/words occurring collectively in text documents are called concepts in LSI. For example:

- ”Jaguar”, ”car”, ”speed” may generally occur in text about sports cars.
- ”Jaguar”, ”animal”, ”hunting” may refer to the concept of jaguar the animal.

According to LSI semantic similarity is found in the terms/words sharing a mutual context in text documents. LSI uses Singular Value Decomposition for the mapping of terms to concepts.

2) Latent Semantic Indexing (LSI) Working: LSI was basically designed for Information Retrieval task which is used to retrieve documents relevant to the query but this task also includes keyword matching and weighted keyword-matching and also vector based representation according to frequency of words or terms in a document. Using this vector based approach, LSI finds out the semantic similarity among the text by generating vector representations of text and comparing them. LSI generates vector based representations of text with the help of SVD. Singular Value Decomposition (SVD) is algebraic matrix operation used to rank and re-arrange dimensions in vector space. SVD arrange the dimensions in decreasing order of significance of dimensions. Vector space shows the most significant dimension in first and least significant dimension in last and some of less significant dimensions are also eliminated. The reason for this elimination is that the top most significant dimensions are actually used for finding the semantic similarity among the text. The arrangement of the dimensions which are in reduced form helps finding similarity because term or words which occur in similar context get the similar values of similarity for ranking. There may exist some differences but main steps performed by LSI are:

- The text data also called a text dataset should have the text in the form of documents and must be separate. Separation here means a paragraph of a text in text dataset is taken as a document where there is a space of more than one line between the text sentences. This actually shows the paragraph’s text is related with text with in a paragraph.
- The next step involves the task to generate a matrix representing co-occurrence of terms and documents also called as Term-Document Matrix. This matrix contains rows and columns where terms or words are represented as rows and documents or paragraphs are represented as columns. The values in the cells of this matrix represents frequency of a particular term in a particular document. Suppose m is number of terms and n is number of documents in complete text dataset then this matrix shows that every document have m-dimensional vector and every term have an n-dimensional vector.
- Term frequency weighting techniques can be applied to every cell value in the matrix. Weighting is used to decrease the influence of commonly occurring terms or words in the complete text dataset. Mostly log entropy is a technique used for weighting.
- In the next step SVD is applied with the parameter k. This parameter K determines the required number of dimensions. SVD generates three matrices and if they are multiplied they results in the original matrix but this result is obtained is SVD is calculated with all dimensions. In recent implementations SVD is applied with value of k that covers most significant dimensions because calculating it with all dimensions need high memory.
- Three matrices are obtained by last step, first with vector consisting of k dimensions for every document, second with vector consisting of k dimensions for every term or word and third consisting of k singular values. First and second matrices vector spaces are different from original one and the third matrix containing singular values is utilized for transformation of vector spaces.
- The document vectors have LSI representation of documents for the purpose of information retrieval task. The query is converted to pseudo doc in vector space of document by joining the vectors of query terms and dividing them by singular values. The comparison of vectors takes place by calculating cosine among them

or distance metrics can also be used instead. The similar vectors in query and document vector space shows the similarity in meaning.

E. Results

The results obtained in terms of similarity and relevancy have the values of similarity and relevancy of every sentence as compared to query sentence.

Similarity Values of Multiple Quires: As mentioned in the dataset description the dataset of RTE-3 has 1600 documents or sentences and 6714 terms or words so we evaluated this system providing ten random quires and there arrangement is mentioned below. At each query similarity values are calculated for all 1600 documents but the graphs show the similarity of 100 documents. The reason for that is displaying similarity of 1600 documents makes the graph congested and difficult to understand. X axis of each graph represents number of sentences which are actually 1600 but only 100 documents slot is displayed.Y axis represents similarity values of 100 sentences with the query.

1) Similarity Values of 1^{st} Sentence with 1-100 Sentences:

- 1^{st} Query is 1^{st} Sentence

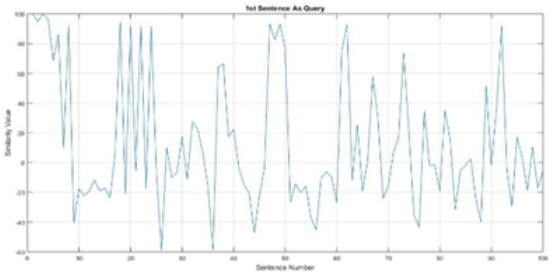

Fig. 2.3: Similarity Plot of First Query

  - The graph in figure 2.3 shows similarity of 1^{st} Sentence with 1 to 100 sentences and 1^{st} sentence is semantically similar with 3rd sentence and nine more sentences. The same was manually verified too.

2) Similarity Values of 200^{th} Sentence with 200-300 Sentences:

- 2^{nd} Query is 200^{th} Sentence

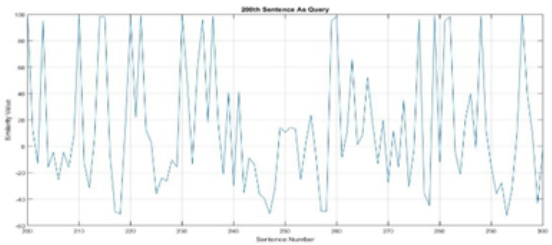

Fig. 2.4: Similarity Plot of Second Query

  - The graph in figure 2.4 shows similarity of 200^{th} Sentence with 200 to 300 sentences and 200^{th} sentence is semantically similar with 17 more sentences. The same was manually verified too.

3) Similarity Values of 400^{th} Sentence with 400-500 Sentences:

- 3^{rd} Query is 400^{th} Sentence

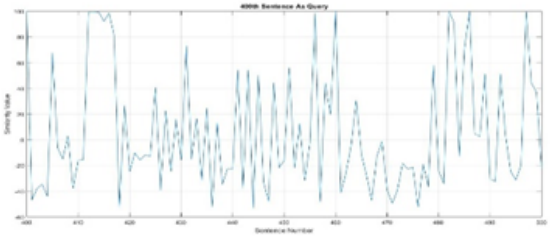

Fig. 2.5: Similarity Plot of Third Query

  - The graph in figure 2.5 shows similarity of 400^{th} Sentence with 400 to 500 sentences and 400^{th} sentence is semantically similar with 11 more sentences. The same was manually verified too.

4) Similarity Values of 623^{nd} Sentence with 600-700 Sentences:

- 4^{th} Query is 623^{nd} Sentence

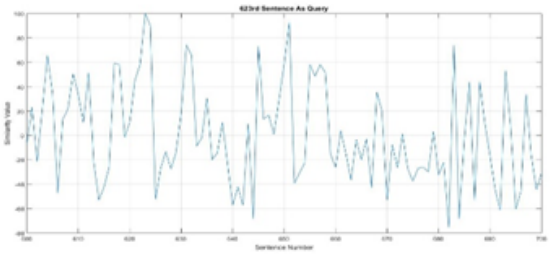

Fig. 2.6: Similarity Plot of Fourth Query

  - The graph in figure 2.6 shows similarity of 623^{nd} Sentence with 600 to 700 sentences and 623^{nd} sentence is semantically similar with 2 more sentences. The same was manually verified too.

5) Similarity Values of 813^{th} Sentence with 800-900 Sentences:

- 5^{th} Query is 813^{th} Sentence

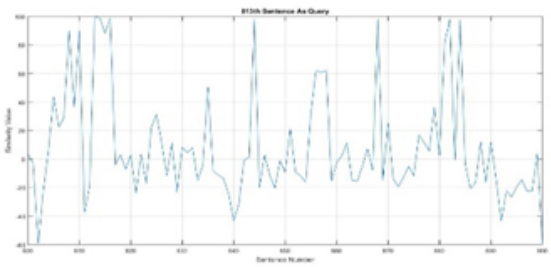

Fig. 2.7: Similarity Plot of Fifth Query

– The graph in figure 2.7 shows similarity of 813^{th} Sentence with 800 to 900 sentences and 813^{th} sentence is semantically similar with 8 more sentences. The same was manually verified too.

6) Similarity Values of 1003rd Sentence with 1000-1100 Sentences:

- 6th Query is 1003rd Sentence

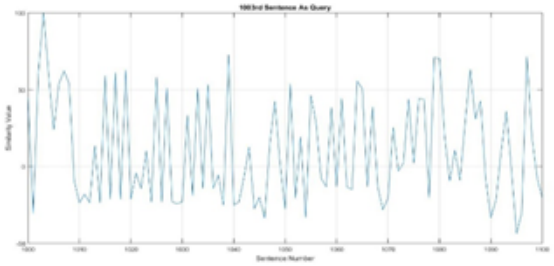

Fig. 2.8: Similarity Plot of Sixth Query

– The graph in figure 2.8 shows similarity of 1003rd Sentence with 1000 to 1100 sentences and 1003rd sentence is somewhat semantically similar with 4 more sentences. The same was manually verified too.

7) Similarity Values of 1200th Sentence with 1200-1300 Sentences:

- 7th Query is 1200th Sentence

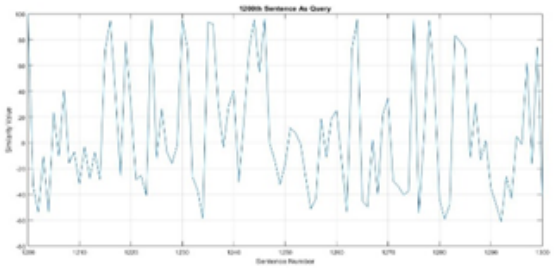

Fig. 2.9: Similarity Plot of Seventh Query

– The graph in figure 2.7 shows similarity of 813^{th} Sentence with 800 to 900 sentences and 813^{th} sentence is semantically similar with 8 more sentences. The same was manually verified too.

8) Similarity Values of 1348th Sentence with 1300-1400 Sentences:

- 5^{th} Query is 813^{th} Sentence

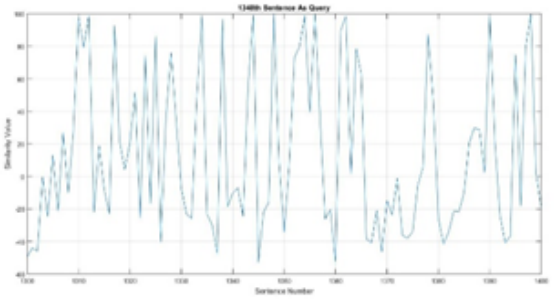

Fig. 2.10: Similarity Plot of Eigth Query

– The graph in figure 2.9 shows similarity of 1200th Sentence with 1200 to 1300 sentences and 1200th sentence is semantically similar with 11 more sentences. The same was manually verified too.

9) Similarity Values of 1436th Sentence with 1400-1500 Sentences:

- 9th Query is 1436th Sentence

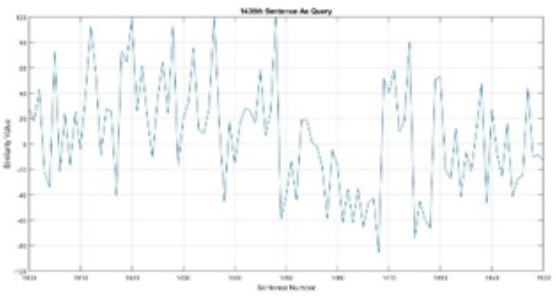

Fig. 2.11: Similarity Plot of Nineth Query

– The graph in figure 2.11 shows similarity of 1436th Sentence with 1400 to 1500 sentences and 1436th sentence is semantically similar with 5 more sentences. The same was manually verified too.

10) Similarity Values of 1598th Sentence with 1500-1600 Sentences:

- 10th Query is 1598th Sentence

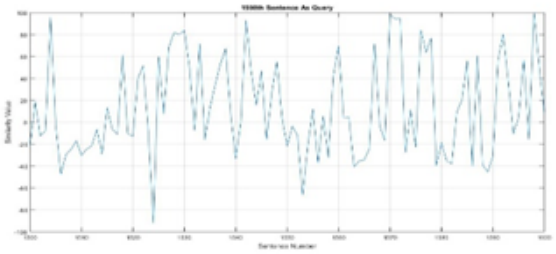

Fig. 2.12: Similarity Plot of Tenth Query

– The graph in figure 2.12 shows similarity of 1598th Sentence with 1500 to 1600 sentences and 1598th sentence is semantically similar with 10 more sentences. The same was manually verified too.

F. Summary of LSI Results

As defined and displayed in above we implemented Latent Semantic Indexing technique to retrieve semantically similar documents from RTE-3 dataset by providing multiple quires to the system. Similarity results obtained at each query are shown in graphs and it can be clearly observed that every single query is semantically similar with at least two and at most seventeen documents among hundred documents slot. It is obvious that if the similarity results of all documents could be displayed then the much more number of similar documents can be shown in graphs. The evaluation of the results is carried out by retrieving all similar documents so the values of Precision and Recall shows the performance of the system at large scale.

III. EVALUATION

For the purpose of evaluation of results these values are passed through the threshold to obtain the components of confusion matrix i.e. True Positive, False Positive, True Negative, False Negative. Putting these matrices in Precision and Recall formula we get the Precision and Recall values.

A. Confusion Matrix

True Positive is the number of positive instances classified correctly and False Negative is the number of positive instances classified incorrectly. False Positive is the number of incorrectly classified positive instances but actually they are negative instances. True Negative is the number of negative instances classified correctly. They are described as follows according to the evaluation of our proposed system and shown in table 3.1.

- True Positive: Sentences with Similarity and Relevancy equal/above threshold.
- False Positive: Sentences with Similarity equal/above threshold And Relevancy below threshold.
- False Negative: Sentences with Similarity below threshold And Relevancy above zero.
- True Negative: Sentences with Relevancy above zero.

Confusion Matrix	Classified Positive	Classified Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

TABLE **I**: Confusion Matrix

B. Precision and Recall

Precision: Fractional form of documents retrieved which are relevant to the user's required information. This is stated as the number correct documents divided by the number of all documents retrieved. Expressed as:

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Recall: Fractional form of documents which are relevant to the query that are retrieved successfully. This is stated as the number correct documents divided by the number of relevant documents. Expressed as:

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

According to Confusion Matrix defined above and displayed in table 3.1. The formula for Precision and Recall is:

$$Precision(p) = \frac{TP}{TP + FP} \tag{1}$$

$$Recall(r) = \frac{TP}{TP + FN} \tag{2}$$

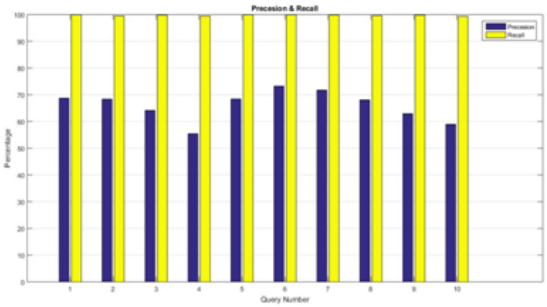

Fig. 3.1: Precision and Recall Plot of Ten Quires

The X-axis of the graph in Figure 3.1 represents the number of query and the Y-axis of this graph represents the Precision and Recall matrix values of 10 random queries respectively. Each query number have two bars so the first bar represents Precision value and the second bar represents Recall value. These values are obtained by the evaluation of the similarity results and relevancy results of 10 random quires with all 1600 documents respectively. The Precision and Recall values are calculated after the calculation of Confusion Matrix. Precision and Recall values of each query varies depending on semantically similar documents retrieved.

IV. Conclusion

In this study we have analysed the proposed system for semantic text retrieval by submitting ten different and random quires to the system and the results showing the similarity between quires and documents of dataset shows that system is capable of retrieving similar documents. The Precision and Recall values at each query submitted to the system has variation reflects the effect of input query on these values. Reason of this variation is number of documents matching with query in terms of semantics and this is the input parametric effect on the output results. Latent Semantic Indexing algorithm is found effective in retrieving the text documents that are semantically similar. The system proposed in this research work is capable of successfully retrieving the semantically similar text. However, it may not be effective for recognizing textual entailment task as it involves the verification of facts and figures that are present in pair of text. It is therefore concluded that using LSI can retrieve semantically similar text and in the domain of opinion mining LSI algorithm can be widely utilized for the retrieval of opinions that are semantically similar expressed on a single topic or product. This way the users expressing similar opinions can also be pointed. The main target of our proposed system is the domain of "Opinion Mining". However, there can be many other applications of this system, for example: Effective Plagiarism Detection, More Relevant Document Retrieval and Relevance and Contradiction of News-feeds.

References

[1] B. H. Murray and A. Moore, "Sizing the internet," *White paper, Cyveillance*, p. 3, 2000.

[2] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the web," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 341–349.

[3] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 519–528.

[4] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the web," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 342–351.

[5] G. Carenini, R. T. Ng, and E. Zwart, "Extracting knowledge from evaluative text," in *Proceedings of the 3rd international conference on Knowledge capture*. ACM, 2005, pp. 11–18.

[6] J. Wiebe, T. Wilson, and M. Bell, "Identifying collocations for recognizing opinions," in *Proceedings of the ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, 2001, pp. 24–31.

[7] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271.

[8] J. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated texts," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2005, pp. 486–497.

[9] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 2005, pp. 347–354.

[10] E. Riloff, J. Wiebe, and W. Phillips, "Exploiting subjectivity classification to improve information extraction," in *Proceedings of the national conference on artificial intelligence*, vol. 20, no. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1106.

[11] M. Koppel and J. Schler, "The importance of neutral examples for learning sentiment," *Computational Intelligence*, vol. 22, no. 2, pp. 100–109, 2006.

[12] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, "The third pascal recognizing textual entailment challenge," in *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*. Association for Computational Linguistics, 2007, pp. 1–9.

[13] D. Zeimpekis and E. Gallopoulos, "Design of a matlab toolbox for term-document matrix generation," in *Proceedings of the Workshop on Clustering High Dimensional Data, SIAM*. Citeseer, 2005.

Tariq Hussain Laghari tariq_hl89@yahoo.com
Faculty of Engineering, Sciences and Technology, Iqra University, Karachi.

Syed Sajjad Hussain shrizvi@iqra.edu.pk
Faculty of Engineering, Sciences and Technology, Iqra University, Karachi.

Manzoor Hashmani mhashmani@iqra.edu.pk
Faculty of Engineering, Sciences and Technology, Iqra University, Karachi.