International Journal of Engineering & Technology, 7 (3.34) (2018) 775-780

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Interactive Intelligent Software System and NLP Techniques for Document Processing

Prashant G Desai [#], Sarojadevi H [*], Niranjan N Chiplunkar [#]

[#] Research Scholar, Department of Computer Science & Engineering, PrincipalN.M.A.M.I.T., Nitte - 574110, Karnataka, India
[*] Professor, Department of Computer Science & Engineering,N.M.I.T., Bengaluru - 560064, Karnataka, India
**Corresponding Author Email: [1] prashanth_desai@yahoo.com ,[2]hsarojadevi@gmail.com, , [3]niranjannchiplunkar@rediffmail.com*

Abstract

The text written within the documents in different formats contains valuable information. Since the quantum of this kind of unstructured text to be processed is very large, a lot of research has taken place towards finding an intelligent system which helps in discovering the valuable information. The proposed research has developed a software system with the objective of processing natural language text and producing results of importance. This paper presents two new algorithms for document processing. The first algorithm interacts with users to find shorter answers using the query submitted by the user. The results show a precision of 80%. The second algorithm is based on the concept of a template prepared and input by the human. It is employed for representing the original document in a concise format. The experimental results obtained and evaluated with the help of metrics from within the domain demonstrate that an accuracy of 73% can be achieved.

Keywords: *Artificial Intelligence; Conversation software; Dialogue; Summarization; Template based algorithm*

1. Introduction

The technology of writing intelligent computer programs can be defined as artificial intelligence. The need of writing intelligent programs is relevant in the present scenario since the information around us has been digitized. It is extremely critical to extract matters of significance from the digitized information. An area of research in the artificial intelligence domain is that of natural language processing (NLP). This area of research helps in developing the algorithms which offer human-computer interaction for discovering information of high quality. It is very easy for human beings to interpret the sentences provided in natural language. However, the computers may not understand the natural language text. Hence, they must be trained with large quantity of dataset before being used.

An NLP system accepts a text document as input and discovers intelligent information from it. The system is considered to be useful, if the discovered information is highly accurate. The text present inside the documents cannot be processed as it is. The reason is that it may be composed of unwanted words, symbols etc. called as noise. To remove this noise from the text, pre-processing is performed. Once the text is ready for subsequent steps, the NLP based techniques are applied to derive knowledge of interest from these documents. A simplified architecture for natural language processing is shown in fig. 1.

Therefore, the concept of NLP is considered as a multidisciplinary field [1]. The decisive goal of text processing or text mining is to evolve at information that was previously not known. A few of the applications of text mining are as listed below.

- Information retrieval
- Document classification
- Question answering systems
- Automatic text summarization systems
- Expert systems
- Web searching engines
- Web crawlers

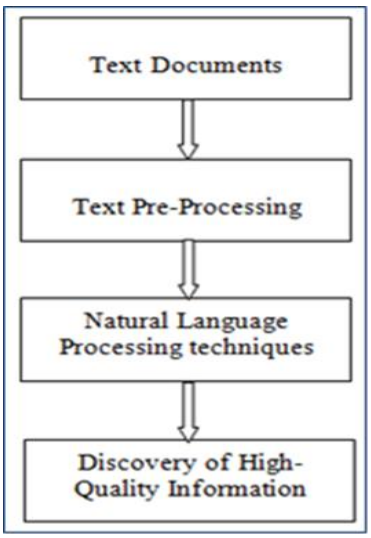

Fig. 1: Architecture of Natural Language Processing

In this paper, we present the NLP techniques for extracting information from the text documents belonging to technical

domain. The remaining part of the paper is structured as follows: Related work is discussed in section II, Section III illustrates the methodology adopted for the design of the algorithms. The experimental results and their evaluation are discussed in section IV. And section V provides remarks on conclusion and presents scope for future work.

2. Related Work

The concept of natural language processing is adopted in identifying the terms which represent symptoms, the disease names and names of genes present in biomedical text [2]. The parser developed by name "AZ-NOUN-PHRASER" is responsible for collecting nouns and phrases consisting of nouns. These words are called as entities. The finite state automaton (FSA) is used to recognize the relations between the extracted entities. The relations established are saved in the database. The developed system is evaluated with an input of 26 abstracts containing 237 sentences. The algorithm implemented was able to find 296 relevant relations from a total of 330 relations. In other words a precision of 90% has been achieved by the experiments conducted.

A hashing technique called Locality-Sensitive-Hashing is used to provide the same bucket in case of a collision between two objects [3]. With the help of a hash-key, every object is associated with "n" number of co-clusters. The candidate co-cluster is created using the objects having hash-key in common. The set of features is defined in the next step for each co-cluster. Then co-clusters having a document set and an associated feature set is obtained.

One way of dealing with processing of research papers is to translate from regular document structure into XML formats [4]. The sections and other relevant details of the research papers are used for generating the rules of extraction and stored in the rule base. These rules are then used to convert a document into XML format. The algorithm aims at representing the unstructured text into a structured format.

The process of teaching is assisted with an expert decision making system called Intelligent Tutoring system [5]. A set of rules made of if-then-else statements are used to acquire and represent the knowledge. Then the decisions are made by the expert system with the help of Neurules as representation of knowledge formalism and the respective inference mechanism. Neurules are rules based on neurocomputing and symbolic specification thereby being hybrid in nature.

The framework is designed in such a way that, the knowledge is extracted from a group of documents [6]. The domain is identified with the help of a module that infers the domain. The ontology class, corresponding attributes and their values are extracted. These domain specific ontologies are then stored by creating an instance.

The research papers under scrutiny are accompanied by a set of key words and a section by name abstract in the beginning [7]. The algorithm is tasked with processing these key words and words which are part of abstract from the documents. The words identified in the process are used for specifying a domain. In this work a method called "Term Frequency Inverse Document Frequency", TF-IDF for domain keyword extraction is implemented and evaluated.

A clustering method based on node positions is adopted for text mining [8]. The system proceeds by creating nodes for each document. The nodes are marked with high pointer having all headings & subheadings of the document and the low pointer the concerned text of the subheading. The percentage of match is displayed from the existing dataset else a new node is generated.

ion [9] is used for modeling the topic from Wikipedia text and the micro text from Twitter posts. R-Package is adopted in the training process and subsequently topic models are created for text from both Wikipedia and Twitter text.

The research in [10] focuses on discovering a pattern set and evaluation of term support. Then the documents are assigned with a weight to decide the relevance.

3. Methodology and Design

This section discusses the methodology adopted in designing the algorithm and the system in detail. The system is developed using Java programming language and other related software tools.

3.1 Document Pre-Processing Steps

It is essential to transform the text from input document into a format that is good for processing. Cleaning is done in this phase by removing unwanted symbols and stemming is accomplished in this phase.

3.2 Document Representation using Vector Space Model

The words from within the document are represented in the form of vector space model which treats every single unique word as a dimension. The approach is known as bag-of-words approach [11]. In this method, a matrix is created where, columns represent the terms and the rows represent documents. The entries in the matrix are the frequency of words (tf) for a document 'd'. The term frequency for a given document 'd'dtf is represented by

$$d_{tf} = (tf_1, tf_2, tf_3, ...tf_n)$$

3.3 Tokenization

With the help of punctuation mark "dot" to identify end of a sentence and white spaces as a measure to identify the boundary of a word, whole text is broken down into smallest units called tokens.

3.4 Data cleaning

The text obtained after tokenization contains unwanted symbols, text (symbols being translated into text form) which are to be removed. Hence, during this step of data cleaning, unwanted elements are removed from the text. It is assumed that the text is free from spelling and grammatical mistakes.

3.5 Stemming

In order to make comparison and information extraction, more inclusive and efficient, words are condensed to their basic form using Porter Stemmer algorithm [12].

3.6 Removal of Stop Words

While writing the text, words such as articles, pre-positions pronouns etc. are used to prepare meaningful sentences. However, they are not required while applying the algorithms for extracting information since they occur more frequently without representing theme of the document. These words termed as "stop words" are removed at the time of applying the algorithm

3.7 Language Understanding

Every token of the text is tagged with an appropriate Part-Of-Speech (POS). Here a part-of-speech such as verb, pronoun, noun,

preposition, adjective, adverb etc is assigned to every token.

3.8 Dialogue Management Module

The dialogue management module is designed to converse with user in a natural language. Human beings provide the request using sentences formulated in English. This module is built on the concept of training. In other words, the algorithm aims at accumulating knowledge at the time of training. This accumulated knowledge is then applied during conversation with the user. The user can converse with the system for the purpose of seeking short answers and also for obtaining summary of a given document. The two methods are: 1. Query based interaction method and 2. Automatic text summarization. These methods are described in the sections below.

3.9 Query based Human-Computer Interaction

The query based HCI interface provides short responses to the user queries. This section provides an overview of the process of answering the queries. The architecture of the module is shown in fig. 2.

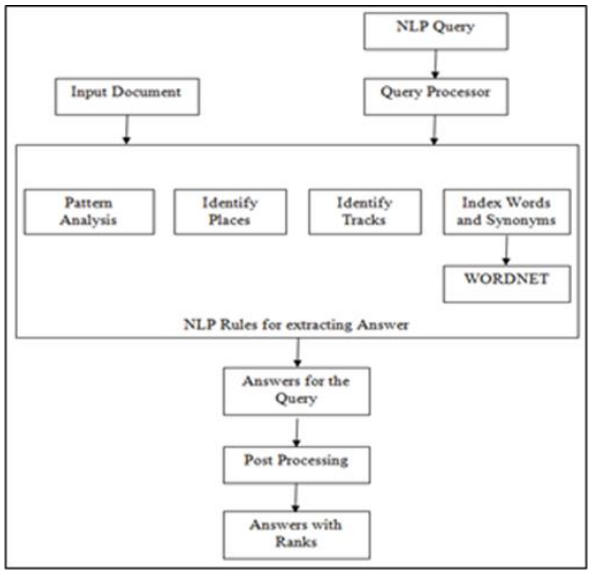

Fig. 2: Architecture of Human-computer Interaction module

The algorithm for query based interaction takes two inputs: Input document and Query from the human being.

In order to identify the dates, the regular expressions are used. The regular expressions are prepared by considering the writing style of dates such as dd-MON-YYYY, dd-MM-YY, dd-MM-YYYY and many such commonly used patterns. Tracks are the fields of interest under which research articles are published. The names of such areas or fields are combination of several technical words. The steps of the algorithm, used to mark out the technical words from within a sentence, presented below

1. Sentence is tokenized into words.

2 Obtain all technical words from a sentence

3. If several technical words are appearing in sequence then such a sequence is identified as a valid track or subject of interest.

The procedure is repeated for all sentences of the input document. These modules which function as rules for identifying the dates and tracks have been presented in our previous work [13]. The improvement to the earlier work is the addition of training module for identifying places. The training process allows the user to annotate the features from the training set into places, person names, and locations. This annotated information is then applied for identifying named entities. Another important feature added for answering the queries is the use of index words. The extraction of index words from the user query, for answering, is the highlight of this module. The algorithm used for answering the queries is discussed below.

1. User selects an input document from where information is required.
2. The option of important dates can be used to obtain important dates.
3. The places option is used for getting information of places.
4. If some other information is required, then a query formulated is submitted to the system

a. Get the user request in natural language text.

b. All stop words are removed from the text.

c. The remaining words are considered to be the index words.

d. The words are reduced to their root form to make the information extraction more inclusive.

e. Synonyms of the index words are obtained using WORDNET since synonyms of the index words might be present in the text of the document.

f. The system responds with multiple answers. For every answer a rank is assigned using:

g. Rank of the Answer=count of index words appearing both in answer and the question + Boosting Factor

Where Boosting Factor= Index words count present in same sequence

3.10 Template based Automatic Text summarization

There are two methods in summarizing a document. 1. Summarization based on abstraction 2. Summarization based on extraction

The first method uses the concept of rewriting the original sentences without affecting the actual meaning while the second method adopts extracting the original sentences based on the significance attached by the summarizing algorithms. The research presented here adopts the second method of text summarization, i.e., extraction based text summarization.

The important sentences from the document are considered as the summary of the document. This section discusses an algorithm called template based algorithm for identifying and extracting the important sentences from the document. The POS tags such as “Adjectives”, “Noun”, and “Adverbs” etc indicate the significances of the words. Therefore, while identifying the sentences, emphasis is given to these POS tags and patterns prepared using these POS tags. Several POS patterns are prepared as part of template. The template also takes into consideration other user requirements such as whether to include dates, places, person names and knowledge from a specific domain. The steps followed in the design of algorithm for summarizing the document are given below.

1. User chooses input document to be summarized.
2. The option “important dates” is chosen to include dates.
3. The option “Places” is chosen to include places.
4. The option “Person names” is selected to include person names.
5. Desired pattern of POS tags is created.
6. Several patterns can be prepared as in step-5.
7. Complete template is created by using the step-2 to step-6.

8. Based on the prepared template summarization of input document is done

a. Obtain sentences of input document.

b. POS tags are assigned to every sentence.

c. Any sentence having a match with POS tag pattern present in template is added to the summary.

d. Sentences having Person names, Dates, Places are also added to the summary.

e. Sentence containing information about a specific selected domain added to the summary.

9. User is presented with the final summary.

3.11 Tools Used

The development of the algorithm proposed is implemented with the help of several software tools. The list below contains the tools used.

- Programming with Java
- NetBeans IDE
- Microsoft SQL Server
- WordNet

4. Results and Discussions

This section reports the experimental set up for finding the results, the metrics used for evaluating the results obtained. The two algorithms discussed in section III: query based human-computer interaction and template based automatic text summarization produce different types of results. Therefore, the relevant metrics are used for evaluating these algorithms separately.

The input for both algorithms would be a technical research document having English sentences. It is assumed that the text is free from spelling mistakes, grammatical errors, written entirely in single column format. The images, tables and charts have been exempted from processing.

4.1 Metrics used for evaluation of Query Based Human-Computer Interaction and results

The metric Mean Reciprocal Ranking (MRR) has been used [14] for evaluating the outputs obtained from this module. This metric evaluates a system that produces several answers. MRR can be computed using the formula below:

$$MRR = \frac{1}{|QUERY|}\sum_{i=1}^{|QUERY|}\frac{1}{Answer's\ Rank_i}$$

Where QUERY is the count of sample queries

Answer's rank$_i$ is the position of rank of first relevant answer in query "i"

The algorithm produces several answers for a query. It is because of the presence of key words or index words and their synonyms obtained from the WORDNET, in multiple sentences. A cut off value 5 is used, to consider the generated answers. That is, top 5 answers are used in the process of evaluation. Table I documents the results obtained using MRR metric.

Table I: results of MRR Metric

Sl. No.	Query	Correct Answer's Rank	Inverse Rank
1	architecture algorithm	1	1/1=1
2	define conversational agent	3	1/3=0.33
3	define expert system	2	1/2=0.5
4	modules of the proposed work	1	1/1=1
5	System architecture	3	1/3=0.33
6	sections of paper	2	1/2=0.5
7	information retrieval system	1	1/1=1
8	what is knowledge base	1	1/1=1
9	indexing process	1	1/1=1
10	Query formulation process	3	1/3=0.33
Total of inverse rank			6.99
Count of Queries			10
MRR			0.69

When the system produces several correct answers, the MRR metric does not define what action has to be taken. Therefore, the evaluation has to be performed using another metric by name Mean Average Precision (MAP). The MAP performs calculation of precision at the position of each correct answer in a list of answers. MAP is calculated using the below mentioned equation.

$$M.A.P. = \frac{\sum_{q=1}^{Q} Avg\ \Pr(q)}{Q}$$

Table II shows the results computed with the help of MAP measure.

Table II : Results of MAP Metric

Sl. No.	Query	Correct Answer's Position	Computing M.A.P.	Final Value of MAP
1	Architecture Algorithm	1,2,3,4	1/1+2/2+ 3/3+4/4= (1+1+1+1)/4	1
2	Define Conversational Agent	1,3	1/1+2/3= (1+0.66)/2	0.83
3	Define Expert System	2	1/2	0.5
4	Modules Of The Proposed Work	1,2	1/1+2/2= (1+1)/2	1
5	System Architecture	3	1/3	0.33
6	Sections Of Paper	2	1/2	0.5
7	Information Retrieval System	1	1/1	1
8	What Is Knowledge Base	1	1/1	1
9	Indexing Process	1,2,3,4	1/1+2/2+ 3/3+4/4= (1+1+1+1)/4	1
10	Query Formulation Process	1,3	1/1+2/3= (1+0.66)/2	0.83
Average MAP Of All Queries				0.8

It can be derived from the results tabulated in table 1 & table 2 that, when all relevant answers are taken into consideration, the accuracy of the algorithm is enhanced to 80%.

4.2 Metrics for Evaluating Summary

We have used four different similarity metrics [15] for evaluating the generated summary. The metrics selected here also indicate the accuracy of knowledge extracted. The list below provides similarity metrics used for evaluating the summary.

- Cosine similarity
- Euclidian distance
- Jaccard coefficient
- Pearson Correlation Coefficient

4.3 Cosine Document Similarity

The cosine similarity between system generated summary & manually created summary is obtained using the formula given below:

$$SIMIL(a,b) = \frac{a \ . b}{||a|| \ . \ ||b||} = \frac{\sum_{i=0}^{n-1} a_i \ . b_i}{\sqrt{\sum_{i=0}^{n-1} (a_i)^2} \cdot \sqrt{\sum_{i=0}^{n-1} (b_i)^2}}$$

Where, a_i i^{th} term appearing in document a

b_i i^{th} term appearing in document b

4.4Euclidian Distance between the Documents

The calculation of Euclidian distance is performed using the formula below

$$ED(\vec{t_x}, \vec{t_y}) = (\sum_{t=1}^{n} | W_{t,x} - W_{t,y} |^2)^{1/2}$$

$\vec{t}_x$ Vector created for terms of "document x"

$\vec{t}_y$ Vector created for terms of "document y"

$W_{t,x}$ Term Weight in "document x"

$W_{t,y}$ Term Weight in "document y"

The Euclidian distance metric indicates how distinct two documents are with each other. Therefore, to obtain the similarity of two documents using this metric, the value of Euclidian distance must be subtracted from 1.

4.5 Jaccard Coefficient Document Similarity

The Pearson Correlation Coefficient measure is computed by the formula given below

$$SIMIL_{PCC}(\vec{t_x}, \vec{t_y}) = \frac{n\sum_{i=1}^{n} W_{i,x} \times W_{i,y} - TermFq_x \times TermFq_y}{\sqrt{\left[n\sum_{i=1}^{n} W_{i,x}^{\ 2} - TermFq_x^{\ 2}\right]\left[n\sum_{i=1}^{n} W_{i,y}^{\ 2} - TermFq_y^{\ 2}\right]}}$$

Where $TermFq_x = \sum_{i=1}^{n} W_{i,x}$ $TermFq_y = \sum_{i=1}^{n} W_{i,y}$

$W_{i,x}$ *Weight for term i in input document x*

$W_{i,y}$ *Weight for term i in input document y*

4.6 Results of Summary Evaluation

The proposed template based algorithm obtains the summary of the given document, based on the inputs supplied by the user in the form of a template. The summary generated would contain more sentences, if fewer inputs are provided inside the template. The table III lists the contents of a template developed by user, for summarizing the document.

Table III : User Inputs Prepared as Template

Sl. No.	Sequence of POS Tags
1	NNP NN VB
2	NN DT JJR VBZ
3	DT JJ FW NNP
4	DT VB NNPS
5	Consider Locations Name
6	Consider dates

The text from document which matches the entries from within the template is extracted into the summary. Here, the input document has 226 sentences. The template constructed is provided to the algorithm for summarizing the input document as depicted by figure 3.

Fig. 3: Template Based Summary

The algorithm uses the input patterns and also other criteria from the template. The resulting sample summary shown in fig. 3 contains 136 sentences, from 226 original sentences belonging to the chosen document. In other words, approximately 61% of the original sentences of the input file are accounted for summarization. The percentage of retaining the sentences in the summary is purely conditioned on the components of template prepared by the user. In other words, if the ingredients of template prepared, are less, then more sentences are retained in summary. On the other hand, if there are more criteria involved while preparing a template, then there will be less number of sentences in the resulting summary.

The automated summary is compared with the summary generated by the human beings. There are ten manually created summaries used for comparing the automated summary. The human beings create such manual summaries depending on their understanding of the text. The process of obtaining similarity values is done using four metrics as discussed in the beginning of this section. Fig. 4 shows ten different cosine similarity values for the experiment conducted. The average of all ten values is recorded.\

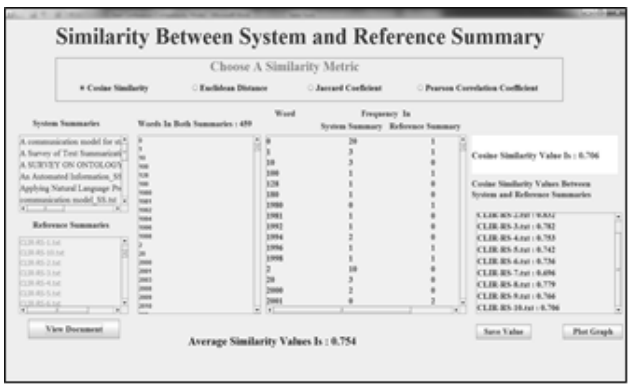

Fig. 4: Ten different similarity values of cosine similarity metric

In the same way the results obtained from the rest of similarity

metrics are also recorded. Table IV documents the average similarity value of all metrics.

TABLE IV : Results of Similarity Metric

Sl. No.	Similarity Measure Used	Average Similarity Value
1	Cosine Similarity	0.754
2	Euclidian Distance	0.704
3	Jaccard Coefficient	0.706
4	Pearson Correlation Coefficient	0.714

The metrics of evaluation except Euclidian Distance display the percentage of similarity between two different input documents. The distance between input documents is calculated by Euclidian Distance which indicates how different one document is from other.

5. Conclusion and Future Work

Over the past 6 decades natural language processing has been a very challenging area. Since there is no specific location of the important information within the document, the challenge is to identify which information is significant. The proposed research work has designed a conversational software system with the objective of extracting information from the natural language text .The algorithms designed for developing the conversational software system exploit the significance of POS tags which actually provide the meaning for the text , synonyms of the index words and the sequence of index words. The paper discusses two algorithms: Query based human-computer interaction and Template based automatic text summarization. The first algorithm is able to successfully produce the small responses while the second highlights and extracts the important sentences in the form of summary.

The work discussed here concentrates on the technical documents as input. This work can be extended to several other prominent domains such as medical documents, documents related to intelligence agencies, etc. The two algorithms designed are based on the concept of extracting existing sentences from the input document. But the research needs to be done in order to rewrite the original sentences without changing the actual meaning. In other words, multiple sentences from the input document can be combined together and represented by a single sentence without deviating from the original meaning.

References

[1] S.Brindha, K.Prabha and S.Sukumaran, "The comparison of term based methods using text mining", Proceedings of International Journal of Computer Science and Mobile Computing , Vol. 5, Issue. 9, September 2016, pg.112 – 116

[2] Gondy Leroy, Hsinchun Chen and Jesse D. Martinez, "A shallow parser based on closed-class words to capture relations in biomedical text", Proceedings of Journal of Biomedical Informatics 36,2003, pp145–158

[3] Fabrıcio Olivetti de Franca, "Scalable Overlapping Co-Clustering of Word-Document Data", Proceedings of 11th International Conference on Machine Learning and Applications,2012, pp 464-467

[4] Jianguo Chen and Hao Chen "A Structured Information Extraction Algorithm for Scientific Papers based on Feature Rules Learning", Proceedings of journal of software, vol. 8, no. 1, january 2013, pp55-62

[5] Ioannis Hatzilygeroudis and Jim Prentzas, a "Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities" Proceedings of Expert Systems with Applications 26 , 2004, pp- 477–492

[6] Raghu Anantharangachar, Srinivasan Ramani and S Rajagopalan, " Ontology Guided Information Extraction from Unstructured Text", Proceedings of International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.1, January 2013, pp 19-36

[7] R. Jayanthi and S. Sheela, "Domain Extraction From Research Papers", Proceedings of Journal of Science and Technology (JST) Volume 2, Issue 4, April 2017, pp42-50

[8] Yogendra Singh Rajput and Priya Saxena, "A Combined Approach for Effective Text Mining using Node Clustering", Proceedings of International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 4, April 2016, pp321-324

[9] Zhou Tong and Haiyi Zhang,"A text mining research based on LDA topic modeling", Proceedings of Computer Science & Information Technology, 2016, pp 201–210

[10] Ning Zhong, Yuefeng Li and Sheng-Tang Wu , "Effective Pattern Discovery for Text Mining", Proceedings of IEEE transactions on knowledge and data engineering, Vol. 24, No. 1, January 2012, pp 30-44

[11] D. Dubin, "The most influential paper Gerard Salton never wrote", Library Trends 52(4), 2004, pp. 748–764

[12] "PorterStemmer", http://www.tartarus.org/~martin/PorterStemmer

[13] Prashant G Desai, Sarojadevi, and Niranjan N Chiplunkar, "Rule-Knowledge Based Algorithm for Event Extraction", Proceedings of International Journal of Advanced Research in Computer and Communication Engineering, ISSN : 2319-5940, Vol.4, Issue-1, January 2015, pp 79-85 , Impact Factor: 1.770

[14] David Hawking and Nick Crasswell, "Measuring Search Engine Quality", Proceedings of Journal of Information Retrieval, 2000, pp1-27

[15] Anna Huang, "Similarity Measures for Text Document Clustering", Proceedings of the New Zealand Computer Science Research Student Conference, 2008, pp 49-56