Performance Analysis of Semantic Similarity from Wikipedia using AUC

S. S. Arya
Assistant Professor
Department of Computer Science and Engineering
New Horizon College of Engineering, Bangalore

S. Shanmuga Priya
Senior Assistant Professor
Department of Computer Science and Engineering
New Horizon College of Engineering, Bangalore

Abstract- The semantic similarity measurement between words or phrases play vital role in Natural Language Processing and Information retrieval tasks such as Word Sense Disambiguation, Query expansion etc. Similarity can be computed from a thesaurus such as WordNet or statistics from a large corpus. This paper presents a Wikipedia based measurement of semantic similarity using LIBSVM. The words from the snippets obtained from Wikipedia are processed using stemming algorithm and the stop words are removed from the document. TF-IDF provides a weighing scheme that gives the attribute- value representation of word pairs in documents. The resulting feature vectors are used in SVM classification. We used LIBSVM in RapidMiner data mining tool for experimental evaluation. Our method was evaluated using WordSim353 similarity dataset. LIBSVM is trained to classify between synonymous word pairs and non-synonymous word pairs. The evaluation result shows that our method has higher accuracy and AUC results than the existing methods. Our experiment also proves that AUC in general is a better measure than accuracy which is to be considered in machine learning applications.

Keywords – Web Mining, Machine Learning, Snippets, Support Vector Machine, LIBSVM, Area Under Curve

I. Introduction

Data available on the Internet are co-related by various semantic relations. Identifying the semantics helps in accurate data retrieval which helps many web mining applications. Measuring semantic similarity plays a vital role in computational linguistics. It’s more challenging for real time applications. Numerous definitions of semantic measures are widely proposed. However it’s commonly accepted that semantic measures aims at determining the likeness of different instances based on their semantics or meaning. The word pair monkey, banana is more closely related than monkey, phone. The aim of semantic measurement between words is to provide the machine the ability to compare the instances based on their meaning. In semantic similarity approach the meaning of target pair of words is determined by training the machine with a set of dataset called benchmark dataset. If two words are similar based on some similarity measures, the meaning of target instances is deemed similar to the benchmark instances. Carefully designed and controlled datasets help in exploring different types of semantic relations.

Measurement of semantic similarity between words is based on a thesaurus such as WordNet or statistics from a large corpus. Manually maintaining WordNet [12] is impossible since semantic similarity between words changes over time and domain. The term apple is frequently associated with computer on the web. The meaning of apple as a computer is not listed in WordNet or dictionaries. The web is a huge, dynamic, heterogeneous collection of information which can be considered as a live corpus. Search engines provide useful information relevant to the user query from the web. Each search engine has its own method for filtering information based on data mining techniques. Even popular search engines sometimes fail to provide the relevant web pages for user queries. Since the web contents are very vast and not reliable, we confine our research to Wikipedia contents [2]. We made use of Wikipedia to measure the semantic similarity between words. It is impossible to analyze each document separately. Snippets and page counts are the two important measures for analyzing the web documents. In this paper we made use of snippets derived from the Wikipedia.

Snippets are small text documents which gives an extracted summary about the page contents. When the user searches for a word in Wikipedia, it gives information regarding the application, history, examples, technology etc. Wikipedia provides relevant information without trawling through a bunch of Google searches. A sample snippet for the search key ‘automobile’ is

"A car or automobile is a wheeled, self-powered motor vehicle used for transportation. Most definitions of the term specify that cars are designed to run primarily on roads, to have seating for one to eight people, to typically have four wheels, and to be constructed principally for the transport of people rather than goods."

The snippet downloaded from Wikipedia cannot be used as such since it contains many ambiguous and semantic unrelated words which must be pre-processed. Stemming algorithm can be used for data processing. Stop words [3] have to be removed from the text documents. After extracting the feature vector from the snippets using TF-IDF [4], the semantic similarity is measured with a machine learning approach called Support Vector Machine (SVM). The existing system makes use of cosine similarity and TF-IDF. The proposed system gives better result than the existing system in terms of accuracy and Area Under Curve (AUC) [18].

II. RELATED WORK

Taxonomy based measures were considered as a great source for measuring semantic similarity between words. [6] proposed an edge counting based method by making use of WordNet. This method succeeded in finding similarity measures close to human ratings. Wei He et al. proposed WNOntoSim a hybrid approach for measuring semantic similarity between ontologies based on WordNet. They used WordNet to calculate semantic similarity at the elementary level and constructed contexts of node [5]. Since the attributes provided by WordNet are sometimes less than needed for accurate word sense disambiguation. WordNet is not a complete solution since manually maintaining the thesaurus is very difficult. An alternative way to measure similarity is based on the notion of information content proposed by P.Resnik et al. [7]. Word similarity is based on their content. The limitation with this approach is that they have not taken into acscount of word sense disambiguation.

Support Vector Machines are well suited for text classification [13].The goal of text classification is to categorize the documents into a fixed number of predefined categories. The documents are first converted into an attribute-value representation. SVM technique is based on Structural Risk Minimization technique. Results show that SVM shows accurate classification over the existing methods such as Rocchio's algorithm, Bayes classification-Nearest Neighbor algorithm, C4.5 algorithm. They have given theoretical and empirical evidence that SVM is very well suited for text categorization. SVMs do not require manual parameter tuning. Mohan Kumar et.al [16][15] has discussed the application of SVM in mammogram. The experimental analysis shows that SVM is the accurate classification tool.

Danushka Bollegala et al. proposed a web search engine based approach [2] for measuring semantic similarity. They calculated semantic similarity between words using page counts and snippets retrieved from search engines. Similarity scores such as Dice, Jaccard, Overlap, and PMI are calculated from page counts and lexico syntactic patterns [8] are extracted from snippets. The combined similarity score is given as feature vector for classification using Support Vector Machine. Arya et.al [9] has proposed that SVM has the limitation of handling latent variables. So they made use of Latent Structural Support Vector Machine (LS-SVM) which shows accurate similarity scores than SVM. They have taken into account the rank of web pages as hidden or latent variables. But LS-SVM is well suited for image classification. Lakshay Sahni et al. proposed a combined approach of Web search engine based snippets and page counts and the Taxonomy measures from WordNet. But the limitation with these methods is that the snippets generated on the search engines are mainly dependent on the ranking algorithms. The resultant snippet may not be relevant to the user search key.

Inorder to enhance the topic specific web crawling Pesaranghader, A et al. introduced LinCrawler to distinguish topic sense-related links from the others [10]. By making use of this approach relevant snippets can be retrieved for the user query. Shirakawa, M et.al proposed Wikipedia base measurement using Naïve Bayes algorithm [17]. Lu Zhiqiang et al. proposed an approach for measuring semantic similarity between words using contents from Wikipedia [1]. They calculated the semantic similarity between words using TF-IDF and cosine similarity. The experimental results show that the similarity scores obtained from the Wikipedia contents are more accurate than the Google snippets. The snippets provided by the Google websites may be irrelevant to the user query. In this approach the snippets are taken from Wikipedia after analyzing the contents which are reliable than any other source of internet. But the cosine similarity technique proves to be inefficient in case of long documents having vectors with small scalar products and a high dimensionality. The order of the terms in the document plays no role in vector space representations. So we proposed the Support Vector machine learning which is proved efficient than any other techniques [13].

III. METHODOLOGY

A. Outline

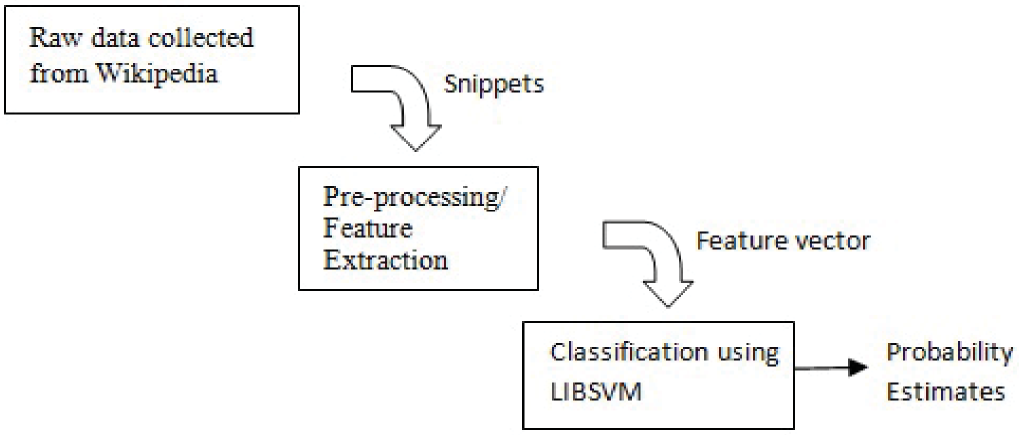
Raw data collected from Wikipedia
Snippets
Pre-processing/ Feature Extraction
Feature vector
Classification using LIBSVM
Probability Estimates

Fig 1: Outline of proposed work

An outline of our proposed work is shown in fig 1. The proposed system exploits snippets retrieved from Wikipedia. Wikipedia is a widely available, open source encyclopedia which is a great tool for learning and researching information. The information from Wikipedia may be irrelevant and redundant. The presence of noise data makes the web mining tasks more complex. Snippets provide a sample content of search engines' result pages. The proposed system makes use of snippets. It makes more enticing for users to click on and thereby enhancing the search listing. Snippets reduce the result navigation cost of end users. The text document from snippets should be pre-processed as it may contain many ambiguous or semantically unrelated words.

The representation of the input words plays an important role in the accuracy of machine learning systems. The documents which are strings of characters have to be converted into suitable representation before further processing. We have used stemming algorithm [20] to delete stop words from the text documents. Stemming is a process of linguistic normalization in which words are normalized to their stem or base form. It converts the variant forms of a word into a common form. The words such as 'connecting', 'connected', 'connectives', 'connections' etc are normalized into the word 'connect'. The words of each retrieved snippet document are stemmed similarly. Stop words are removed once the stemming algorithm is applied. Stop lists in paper [] are used to process the data from snippets. The processed information is then stored in database.

The words extracted from the database have to be converted into feature vector representation suitable for machine learning system. Selecting the relevant features and encoding them for machine learning systems have an impact on the machine learning system's ability to extract a good model. The interesting work in building a classifier is to select a set of relevant features and deciding how to represent them suitable for machines. Although it's possible to get good performance by using a simple and obvious set of features, the accuracy of machines can be improved by carefully designing te features. We made use of Term Frequency- Inverse document Frequency (TF-IDF) [4] which provides a weighing scheme to show how relevant a word is in a document. This leads to an attribute-value representation of words in text document. The feature term frequency (w_i , d) denotes that the word w_i occurs d times in the document. The words in the documents are converted into feature vectors only if they are not stop words such as not, and, or etc. Term frequency is the number of times a word appears in a document divided by the total number of word in that document (1). Inverse Document Frequency is the logarithm of number of documents in the corpus divided by the number of documents where the specific term appears.

For a term i in document j,

$$w_{i,j} = tf_{i,j} * \log\frac{N}{df_i} \qquad (1)$$

$tf_{i,j}$ = number of occurences of i in j
df_i = number of documents containing i
N = number of documents

The resulting feature vector is used in machine learning system. The existing system made use of cosine similarity. Cosine similarity is often used in information retrieval by using the Vector Space model. The documents are represented as vectors of words by using term frequency. Cosine similarity approach is easy to implement, but it is not a good metric as the reduction is not substantial. Researchers have proved that Support Vector Machine (SVM) is considered as the suitable machine learning approach for text classification due to the following advantages [13].

i. High dimensional input space
ii. Few irrelevant features
iii. Document vectors are sparse
iv. Most text categorization problems are linearly separable

The proposed system makes use of SVM to calculate the similarity scores of word pairs from the feature vector representations. SVM is a powerful method for classification and regression. SVM takes a set of training data as input in which each input is marked as belonging to one of two class labels. The training algorithm builds a model that assigns data in the training set into one category or the other. Formally it constructs a hyper plane which can be used for classification, regression or other actions. The proposed system makes use of LIBSVM [11] implemented in java as it supports internal multiclass learning and probability estimation after applying the learned model on a classification dataset. LIBSVM by default uses rbf kernel which maps samples into a higher dimensional space non-linearly. It can handle the case when the relation between attributes and the class label is non linear. In our experiments we have used rbf kernel and the feature vectors of the corresponding word pairs are converted into column separated value (CSV) file which is then used as testing dataset into SVM. It classifies the word pairs and will return the similarity score in the range of 0-1. The similarity scores are compared with the WordSim353 similarity dataset to calculate the correlation.

B. System Architecture

In order to present the complete idea, we depict the system architecture of the proposed system in Fig 2.

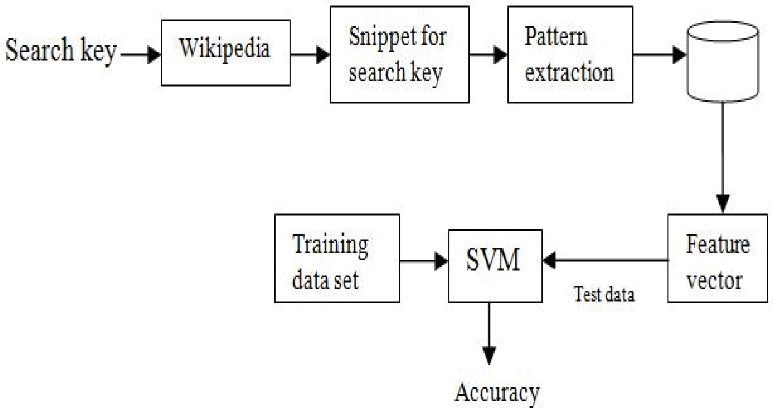

Fig. 2: System Architecture

The word pair is given as a search key in Wikipedia. Wikipedia provides information relevant to the search key such as definition, variations, examples, history, references, external links etc. Snippets are extracted from Wikipedia contents. Snippets cannot be used as such since the words may be semantically irrelevant or ambiguous. The words are preprocessed before classification. The commoner morphological and inflectional endings from words can be removed using the stemmer algorithm []. The processed words are stored in the database. The string of characters has to be transformed into a feature vector representation suitable for the learning algorithms. We used Term Frequency- Inverse document Frequency (TF-IDF) approach which is a weighting factor in web mining applications.

The corresponding feature vector for each word pair is given as training dataset in SVM. The similarity measure is then calculated.

IV. DATA ANALYSIS AND RESULTS

The performance of the proposed system is evaluated by using RapidMiner [14], an open source predictive analysis platform used in data mining applications. In order to evaluate our system, we selected the word pairs from WordSim353 dataset which contains 353 word pair human similarity ratings. The degree of similarity of each pair is accessed on a 0-10 scale. SVM is trained using features extracted from synonymous and non-synonymous word pairs in WordSim353. The work flow in RapidMiner is in fig 3.

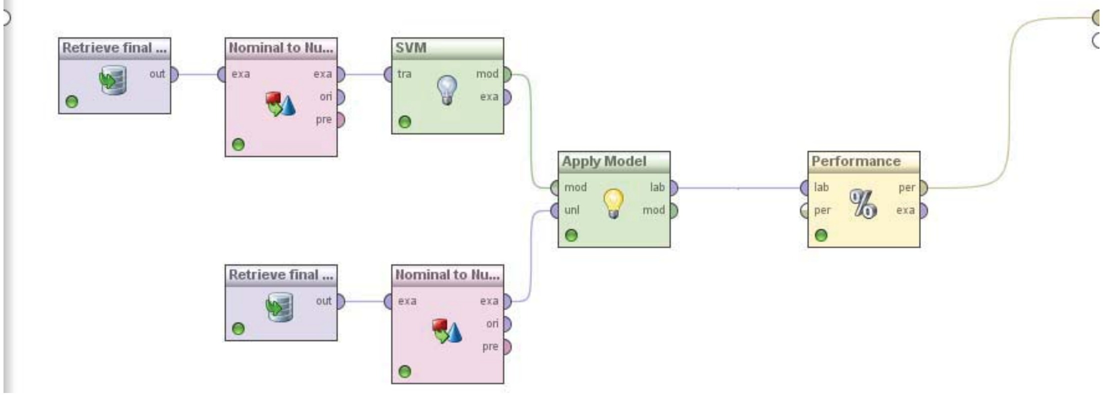

Fig 3: Work Flow of the proposed system in RapidMiner

The training dataset extracted from the feature vector is given to LIBSVM which is integrated software for support vector multi-class classification. LIBSVM supports C-SVC and nu-SVC SVM types for classification. RapidMiner cannot handle nominal attributes. So the dataset attributes are converted into numeric values. The SVM model is first trained on the given data related to the data is learned by the model. Then that model is applied on the testing dataset for class label prediction. In our experiments we have used the class label values synonymous and non-synonymous. We used Gaussian kernel for the experiments. The parameters C and gamma were determined by testing the maximum accuracy in the 2D grid formed by different values of C and gamma. The other SVM parameters were set by the default values in the toolkit. The performance operator in the data mining tool evaluates measures such as precision, recall, accuracy and AUC (Area Under Curve). These values are listed in table 1.

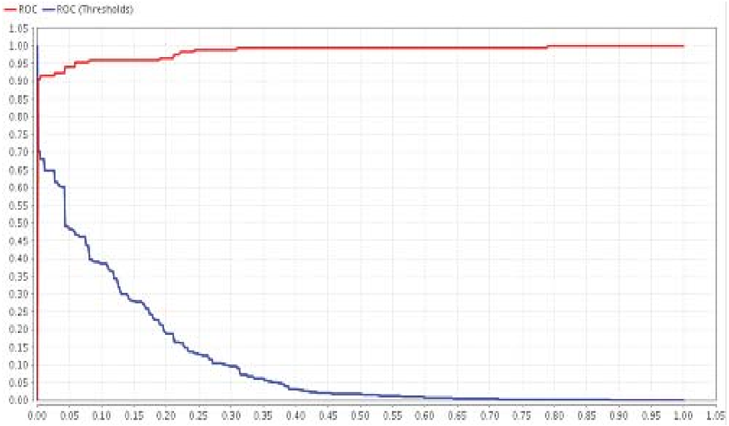

Fig 4: AUC results

The average predictive accuracy of our experiments is 94.62 % and the average predictive AUC is 96.4 %. AUCs provide a better measure than the accuracy as ranking is more important in data mining applications and AUC reflects ranking much more accurately and directly than accuracy [18]. AUC results using LIBSVM is shown in fig 4.

CLASS LABEL	AUC	PRECISION	RECALL	F-SCORE
SYNONYMOUS	0.943	94.15	95.68	94.90
NON-SYNONYMOUS	0.985	95.15	93.45	94.29

Table 1: F-score

The Create Lift Chart operator creates a lift chart based on a Pareto plot of the discretized confidence values of the given testing dataset and the trained model. The lift chart measures the performance effectiveness of LIBSVM by calculating the ratio between results obtained with a model and the results obtained without a model. The result obtained without a model is based on some random dataset used by the tool kit. Fig 5 shows a visualization of the discretized confidences together with the counts for the synonymous values.

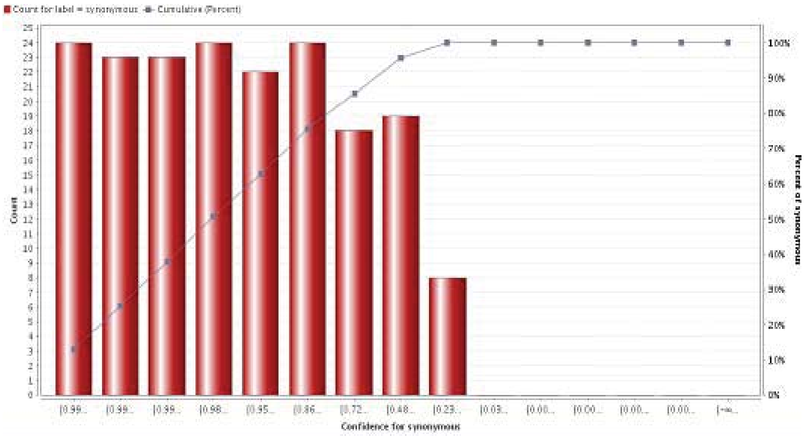

Fig 5: Lift chart of discretized confidences with the count of synonymous values

V.CONCLUSION

In this paper, we have presented an approach for measuring semantic similarity between words using Wikipedia. The existing system made use of cosine similarity approach and TF-IDF. Our proposed work measured similarity using LIBSVM which provides an accurate measurement for text classifiers. Words are selected from Wikipedia snippets and processed using stemming algorithm. The resulting feature vectors from TF-IDF are used in LIBSVM. We made experimental analysis on accuracy and AUC. The results prove that the proposed similarity measurement using LIBSVM shows more accuracy than the cosine similarity method. The average predictive accuracy is 94.62 % and the average predictive AUC is 96.4 %. Our experimental results also prove that the AUC, in general is a better measure than accuracy and many data mining algorithms should be re-evaluated in terms of AUC's.

REFERENCES

[1] Lu Zhiqiang, Shao Werimin and Yu Zhenhua, " Measuring Semantic Similarity between Words Using Wikipedia" IEEE Proc. International Conference on Web Information Systems and Mining, 2009

[2] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka, "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words" IEEE ,2011.

[3] Christopher Fox, 1989, "A stop list for general text", ACM SIGIR Forum, Volume 24, Issue 1-2 (Fall 89/Winter 90) Pages: 19-21

[4] Akiko, Aizawa. An information-theoretic perspective of tf–idf measures. National Institute of Informatics, 2002.

[5] Wei He; Xiaoping Yang; Dupei Huang, "WNOntoSim: A Hybrid Approach for Measuring Semantic Similarity between Ontologies Based on WordNet," in Web Information Systems and Applications Conference (WISA), 2011 Eighth , vol., no., pp.73-77, 21-23 Oct. 2011

[6] Dou Hao; Wanli Zuo; Tao Peng; Fengling He, "An Approach for Calculating Semantic Similarity between Words Using WordNet," in Digital Manufacturing and Automation (ICDMA), 2011 Second International Conference on , vol., no., pp.177-180, 5-7 Aug. 2011

[7] P.Resnik, “Using Information Content to Evaluate Semantic Similarity in taxonomy”, Proc. 14th Int’l Joint Conf. Artificial Intelligence, 1995

[8] J.Pei , J.Han, B.Mortazavi-Asi,J.Wang, H.Pinto, Q.Chen, D.Dayal, and M.Hsu, “ Mining Sequential Patterns by Pattern-Growth: The Prefix span Approach”, IEEE Trans. Knowledge and Data Eng., 2004,vol. 16, no. 11, pp. 1424-1440.

[9] Lavanya, S.; Arya, S.S., "An approach for measuring semantic similarity between words using SVM and LS-SVM," in Computer Communication and Informatics (ICCCI), 2012 International Conference on , vol., no., pp.1-4, 10-12 Jan. 2012

[10] Pesaranghader, A.; Mustapha, N.; Pesaranghader, A., "Applying semantic similarity measures to enhance topic-specific web crawling," in Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on , vol., no., pp.205-212, 8-10 Dec. 2013

[11] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[12] George A. Miller, “WordNet: A Lexical Database for English”.

[13] Thorsten Joachim’s, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”.

[14] Markus Hofmann, Ralf Klinkenberg, “RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series),” CRC Press, October 25, 2013.

[15] Mohan Kumar S , et al. "Classification of Micro Calcification And Categorization Of Breast Abnormalities - Benign and Malignant In Digital Mammograms Using SNE And DWT", Karpagam Journal of Computer Science , Volume-07, Issue-05, July-Aug, 2013- Page Numbers: 253 to 259, ISSN No: 0973-2926

[16] Mohan Kumar S , et al. "The Performance Evaluation of the Breast Mass classification CAD System Based on DWT, SNE AND SVM", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 10, October 2013, Page Numbers: 581-587, ISSN 2250–2459

[17] Shirakawa, M.; Nakayama, K.; Hara, T.; Nishio, S., "Wikipedia-Based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes," in Emerging Topics in Computing, IEEE Transactions on , vol.3, no.2, pp.205-219, June 2015

[18] Jin Huang; Ling, C.X., "Using AUC and accuracy in evaluating learning algorithms," in Knowledge and Data Engineering, IEEE Transactions on , vol.17, no.3, pp.299-310, March 2005

[19] Mohan Kumar S , et al. "Classification of Microclacification in digital mammogram using SNE and KNN classifier", International Journal of Computer Applications, IJCA 03,2013, ISSN: 0975 - 8887

[20] Lovins, Julie Beth (1968). "Development of a Stemming Algorithm".Mechanical Translation and Computational Linguistics 11: 22–31.]