Research Journal of Applied Sciences, Engineering and Technology 8(23): 2320-2328, 2014
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 18, 2014 Accepted: October 17, 2014 Published: December 20, 2014

An Efficient Technique to Implement Similarity Measures in Text Document Clustering using Artificial Neural Networks Algorithm

[1]K. Selvi and [2]R.M. Suresh
[1]Sathyabama University,
[2]Sri Muthukumaran Institute of Technology, Chennai, India

Abstract: Pattern recognition, envisaging supervised and unsupervised method, optimization, associative memory and control process are some of the diversified troubles that can be resolved by artificial neural networks. Problem identified: Of late, discovering the required information in massive quantity of data is the challenging tasks. The model of similarity evaluation is the central element in accomplishing a perceptive of variables and perception that encourage behavior and mediate concern. This study proposes Artificial Neural Networks algorithms to resolve similarity measures. In order to apply singular value decomposition the frequency of word pair is established in the given document. (1) Tokenization: The splitting up of a stream of text into words, phrases, signs, or other significant parts is called tokenization. (2) Stop words: Preceding or succeeding to processing natural language data, the words that are segregated is called stop words. (3) Porter stemming: The main utilization of this algorithm is as part of a phrase normalization development that is characteristically completed while setting up in rank recovery technique. (4) WordNet: The compilation of lexical data base for the English language is called as WordNet Based on Artificial Neural Networks, the core part of this study work extends n-gram proposed algorithm. All the phonemes, syllables, letters, words or base pair corresponds in accordance to the application. Future work extends the application of this same similarity measures in various other neural network algorithms to accomplish improved results.

Keywords: Artificial neural networks, natural language processing, porter stemming, similarity measure, wordnet

INTRODUCTION

Clustering is vital because it enables to rapidly access relevant documents in the web for given pair of words document. The amount of time taken for retrieving the documents should not be more than a second in order to confirm whether the documents what the user is searching for is the correct document. Algorithms and software are needed by the web search engines for fulfilling the query from the users and return relevant documents precisely on time. Thus we understand that, besides the conventional clustering algorithms there is a pressing need for a new efficient document clustering algorithms.

The alarming rise of the World Wide Web is a huge impediment for researchers to collect similar documents over the internet. While choosing the relevant documents from the huge volumes of search results returned to a plain query, search engine is getting perplexed. This problem can be averted by clustering similar web documents and assisting the user in identifying the relevant data effortlessly and successfully.

In order to provide services essential for the user, majority of the internet web documents are widely accessible. These types of documents do not have secret/classified or sensitive data. Currently, because of replication, identical information exists in more than one document. Evading such document replication is one way of providing privacy of those documents. In doing so, the privacy of individual copyrights of documents is protected.

Two documents are considered to be similar if the similarity measure is 1 and if the similarity measure is to some extent or not fully they are considered as different. Offline and online techniques are the two categories of document clustering tasks. Offline technique involves gathering a fixed, formerly compiled collection of documents whereas online techniques function on an incrementally compiled set of documents.

To accomplishing a good document clustering the best features have to be chosen. In this study, the choosing of the best similarity measure and appropriate algorithms for document comparison are considered.

The mathematical model and visual representation of neural connection in a human brain is called as Artificial Neural Networks (ANN). It is signified by a topology of associations between various elements which are nodes or neurons or perceptions shown in layers. The connection among neurons of the layers is conveyed by Connection strength utilizing matrices for

Corresponding Author: K. Selvi, Sathyabama University, Chennai, India

computation purpose. The connections are otherwise the fluid in which the neurons are placed in the brain. The ANN topology is trained by Algorithms developed by various researchers and is applied in a computer program. Final weights are a set of representative numbers acquired and store in the database for future utilization as the data is presented to the program.

LITERATURE REVIEW

Latent Semantic Kernels (Reddy *et al*., 2014) is one of the early advances in this vein. It is proposed in the Information Retrieval community which is a kernel-based extension to the renowned Latent Semantic Indexing (LSI) (Manna *et al*., 2009). A kernel matrix is calculated over text documents in LSK and the Eigen-decomposition of this matrix is utilized to compute a new estimation of lesser rank basis for the space. Capturing semantic ideas can be instinctively thought of as the scope of the new basis which is approximately related to co-varying subsets of the dimensions in the original space. This approach varies in the primary respects from our work although there possibly will be a few superficial comparables. Making a new kernel function, without utilizing an existing kernel matrix to deduce semantic dimensions is the key aim of this method. In this method, it is not needed to compute an expansion for a known text snippet except when we would like to estimate the kernel function. It is not required to explicitly compute a complete kernel matrix over few set of existing text snippets nor its Eigen-decomposition. In fact, the kernel we present here can perhaps be utilized to make the kernel matrix on which the Eigen-decomposition is performed and is wholly complimentary to work on LSK.

The work of Kongsorot and Horata (2014) is an approach more similar to that considered here. A kernel for establishing the resemblance of individual terms based on the compilation of documents that these terms appear in has been done in their work. And, they learn a Semantic Proximity Matrix that records the connection of individual terms by fundamentally measuring the connection in the documents that contain these terms. However, the kernel we consider in our study is striving to establish resemblance on the whole text snippets and not just between single terms. In contrast to the earlier work, the kernel between snippets can be computed on-line precisely as desired and does not necessitate performing an optimization over the whole compilation of documents. Use of cross-lingual method to study a se-mantic representation for a document has been attempted to address in earlier works. Here, one begins with a mass of document pairs, in which each pair is the same document written in two different languages. In order to establish amalgamation of related words in one language that associate well with combinations of words in the other language, a correlation examination is then carried out between the corpora in each language. And by this means, word associations in a specified language are learnt. Apparently, this approach does not necessitate such paired corpora (Reddy *et al*., 2014).

Due to the assumption that documents are independent from each other, they are represented as "Bag of Words" (BOW) in the usual text classification algorithms. Though these approaches make the models simpler, it disregards the semantic information between terms of each document. A new technique to determine semantic resemblance based on higher-order reliance between documents is developed in this study. A kernel for Support Vector Machines (SVM) algorithm is proposed using these dependencies which is called Higher-Order Semantic Kernel. Many tests have been undertaken with an intend to present comparative performance of Higher-Order Semantic Kernel. The experiments utilizing Higher-Order Semantic Kernel on many popular datasets prove that categorization performance is remarkably enhanced in contrast to the first-order methods (Altlnel *et al*., 2013).

Tremendous progress in science and technology made it feasible for the amassing of bulk of data. However, the bigger challenge is to procure valuable data from this enormous amount of data. Efficient Data mining techniques have been developed to triumph over this challenge. A relative study on the consequence of a diverse similarity measures in clustering documents in the same data set is carried out in this study. Cosine similarity, Euclidean distance, Correlation Coefficient and Jaccard Coefficient are the four similarity measures based on which document clustering is done. In terms of purity, Correlation and Jaccard demonstrates enhanced performance while Euclidean demonstrates the poorest performance in majority of the cases. Document recovery will be more competent and time taken will be less, if vital word finding can be integrated with the document clustering (Kavitha Karun *et al*., 2013).

One of the most tedious processes is determining similarity among words by utilizing a search engine based solely on page counts. Search engines do not acknowledge the position of words in a document; it rather considers a document as a bag of words. So this study proposes a transformation function for web measures in order to determine semantic resemblance between two specified words. It only utilizes page counts returned by Web search engines and makes use of the document's title attributes. There is an accomplishment of the correlation coefficient close to 71%. And investigational outcome on benchmark datasets prove that the proposed approach outdo similarity measures defined over snippets alone. Making use of this novel approach on named entities and enhancing resemblance score results using the

transformation function will be part of future work (Hamani and Maamri, 2013).

In this study, identifying (KWS) the main word in handwritten documents is approached here by means of Word Graphs (WG) acquired by utilizing segmentation free handwritten text detection technology. It is based on N-gram Language Models and Hidden Markov Models. With respect to techniques that disregard word contexts or which rely on image-matching with pre-segmented segregated words, the Linguistic perspective remarkably enhances KWS performance. But setting side by side to other KWS approaches that straightly work on the original images (where demands are usually exceedingly high) the WG-based KWS can be remarkably quicker. This study examines the exchange between WG size and KWS information-recovery performance. The outcome obtained proves that, without being devoid of the brilliant KWS performance accomplished with huge WGs, the small and computationally low cost WGs can be utilized (Toselli and Vidal, 2014).

An unsupervised process forming its basis solely on detecting the similarity association between documents with the yield as a set of clusters is called Text clustering. A commonality measure is defined in this research to detect resemblance between two text files which is used as a similarity measure. Any existing recurrent item finding algorithm such as apriori or fp-tree is applied to the initial set of text files to decrease the dimension of the input text files. For all the documents, a document feature vector is formed and then a vector is formed for all the stationary text input files. Thus the Proposed algorithm has the input as similarity matrix and output as a set of clusters compared to other clustering algorithms that establish in advance the count of clusters (Reddy *et al*., 2014).

Depending on the significance of the words or concepts present in it, every document has a definite orderly ranking structure. For the investigation of discourse structures within a document, represented by hierarchical document signatures, this study develops document computing procedures. A series of data describing a definite case is called a signature. Within the series, the position of the individual data is set and holds local value semantics. A kind of fuzzy signature, called hierarchical document signature is used in this study. This is to modularize a formless document in a pecking order, from document order to sentence order, sentence order to attribute order and finally to word order. Occurrence of words is utilized as the information of the lowest module to locate the resemblance among the subsequent superior module. This is accomplished by combining the signature values giving sentence pair coherence (Manna *et al*., 2009).

A combined method that is derived from the edge-based notion by adding the information content as a decision factor. They take into account the fact that edges in the taxonomy may have unequal link strength, so link strength of an edge between two adjacent nodes is determined by local density, node depth, information content, and link type. The similarity between two words is simply the summation of edge weights along the shortest path linking two words (Jiang and Conrath, 1998).

A theoretically well-motivated measure that is similar to Resnik's information content. Lin's modification consisted of normalizing by the combination of information content of the compared concepts and assuming their independence. As introduced above, different methods use different information sources and, thus, result in different levels of performance (Lin, 1998).

In this study the commonly used information sources measured semantic similarity between two queries using snippets returned for those queries by a search engine. For each query, they collect snippets from a search engine and represent each snippet as a TF-IDF-weighted term vector. Each vector is L2 normalized and the centroid of the set of vectors is computed. Semantic similarity between two queries is then defined as the inner product between the corresponding centroid vectors. They did not compare their similarity measure with taxonomy-based similarity measures (Sahami and Heilman *et al*., 2006).

The probability for the number of words to be encountered by a handwriting or speech detection system is very huge. And to gather data to consistently calculate the parameters of statistical grammars for the entire probable words is actually not achievable. Alteration of a pre Determined State-Transducer (FST) n-gram that facilitates the formation of a static transducer is presented. It is likely to decrease the number of actual occurrences of the character-level grammar by incorporating paths in the "LG" transducer which consists of lexicon and n-gram making it more generative with respect to the n-grams observed in the corpus. And the resulting transducer fits the memory of practical machines. For handwriting detection, this model is assessed by utilizing the RIMES and the IAM data bases. Its outcome on the vocabulary size is studied here and proves that this model is viable with up to date solutions (Messina and Kermorvant, 2014).

This study proposes a technique of efficient text analysis and modeling with an ability to produce constant valence ratings at the sentence level commencing from word and multi-word term valence ratings. In order to competently combine the valence of single-word and multi-word terms, a back-off algorithm is applied. A term detection criterion is particularly utilized to choose the suitable n-gram terms, commencing with bigrams and potentially backing off to unigrams. Utilizing semantic resemblance estimates computed on a huge web corpus, lexicon expansion method produces the term affective ratings. The

proposed framework gives up to date results with an accomplishment of 75% accuracy (Malandrakis *et al*., 2013).

For Single Layer Feed forward neural Networks (SLFNs) Extreme Learning Machine (ELM) is a popular algorithm. But a number of the responsibilities that ELM focuses are single-label, where an instance of the input set is connected with one label. This study proposes a novel method for preparing ELM that will be competent of more than one label categorization utilizing the Canonical Correlation Analysis (CCA). CCA-ELM is the name of the new method involving 4 steps in the training process. Any correlation between the input characters and the set of labels using CCA is calculated in the initial step. The succeeding step plots the input space and label space to the new space. Then ELM is utilized to categorize and the ultimate step is to plot to the original input space. Thus compared to the other algorithm that uses similar standard CCA, experimental results prove that CCAELM can enhance ELM for categorization on more than one label learning and its recognition performances are superior (Kongsorot and Horata, 2014).

Remarkable increase in the amount of data captured and made accessible to scientists for exploration pose a huge challenge for finding the most appropriate data. This study studies the idea of varying relevance and therefore ranked search and whether it applies to numeric data-which means, whether data sets are like documents for Information Retrieval techniques and assessment measures to employ. A user study is presented which shows that dataset resemblance reverberate with users as a basis for relevance and therefore, for ranked search. A prototype implementation of ranked search over datasets is calculated with a second user study and shows that ranked search enhances a scientist's ability to locate required and accurate data (Megler and Maier, 2014).

Keyword Search (KWS) from speech is an information recovery task which consists of detecting all probable positions of a given query term in a large speech data set. This study presents a new lower-bound to the well known TWV performance metric for contemporary KWS systems. This lower bound is maximized in a semi-supervised kernel method for rescoring a KWS system. Earlier work does not clearly optimize the TWV. The approach is partially supervised as a manifold regularization term. There is a 4.8% progress in MTWV over a manifold-regularized kernel logistic regression baseline. With respect to the ASR posterior scores, the produced confidence achievements are corresponding and their MTWV is enhanced by an impressive 1.8% (Audhkhasi *et al*., 2014).

Techniques using n-grams as inputs have established to be successful over a broad range of file and data type classification but there are mixed results of the usefulness of unigrams and bigrams as inputs independently. Support Vector Machines (SVMs) composed of unigrams and bigrams and complexity and other byte frequency-based measures, as inputs are used in this study. Compared to previous reports, remarkable results have been accomplished through the use of unigrams and bigrams connected in series as input and a linear kernel SVM. This is the first study to use n-grams linked in series as the solitary input. The study also proved that, over fitting and bad generalization features are the consequence of excess varieties of characteristics as inputs. Microsoft Office 2010 files, file system data, base64, base85, URL encoding, flash video, M4A, MP4 and JSON records have been rarely or not studied at all in the past and this study incorporated all these types. An open source software tool called Sceadan-Systematic Classification Engine for Advanced Data Analysis is the approach that triumphs (Beebe *et al*., 2013).

This study proposes a new scheme called the sequence profiling based on N-gram models for the detection of stochastic discrete event system. Neither any system models nor any system-precise knowledge are involved in this method. From the target system, the information essential for orderly profiling is only event logs. N-gram models are building from a uncomplicated statistical analysis from event logs in the usual circumstances. In order for the diagnoser to approximately calculate whether any faults has happened or not, N-gram model is used as the base. The wildcard characters in the short sequences used in the N-grams is introduced to cover behavior not related to the faults. This leads to elimination of the effect by subsystems which may not be associated to faults. Application to fault diagnosis of a more than one processor system clearly demonstrates the efficiency of the proposed approach (Hiraishi *et al*., 2013).

PROPOSED METHODOLOGY

Documents are files consisting of two characters called ASCII and non ASCII. RFP, certificate, thesis, white paper financial statement, Nondisclosure agreement, Mutual nondisclosure agreement, summons, license, application forms, user-guide, brief, mock-up, paper, journal, Invoice, quote, Proposal, gazette, Packing slip, Manifest, Spread sheet, Waybill, Bill of Lading, script are some of the illustration categories created using templates. By using different types of editors the documents can be computerized. It commences from basic editors which use only ASCII characters to sophisticated editors which embed graphics. Special characters represent the graphics in a document.

Processing of documents where removal of information should be avoided taking care of word pair search is part of similar documents search. The measures of similarity will be perfect when the

documents are appropriately processed before and the value of search will be highest.

Processing of documents before is the initial step. The documents should be changed into a depiction suitable for applying the learning algorithms during this preprocessing stage. Vector space model is the most popularly used method for document representation.

Each document can be changed into a vector D where each vector contains more than two dimensions. Each dimension of a vector is a unique representation of a document. The numerical values of the vector are almost same when the contents of two documents are almost similar. Each feature of the vector will represent a definite term which is a single word or phrase. A phrase can have more than one word. By using statistical or Natural Language Processing (NLP) techniques the phrases can be extracted. Extraction of phrases for the frequently visible words in the document is through the NLP statistical methods.

The Training Based Artificial Neural Networks algorithms involve the following steps.

Part I: Document preprocessing:

Step 1: Search space provides a set of words.

Step 2: Each document is changed to vectors of numerical values in the search space after preprocessing. This process can be averted if in the searching folder, vectors of numerical values are previously available subsequent to the available documents.

Step 3: For calculating, the inappropriate data are removed during this process and the bag of representation of words which are distinct single words is formed.

Step 4: The below mentioned task are carried out after the Bags of words are indexed:

- White space and punctuations called Tokenization divide the string into tokens.
- Each token is stemmed into its root form by changing a noun into its singular form and eliminating words like articles, conjunctions or pronouns.

Stop word elimination and scoring: The remaining words are coded into numerical values.

Step 5: Converting vectors into a matrix is formed. In this matrix presence of absence of words corresponding to a document is indicated using 1 or 0. On the other hand, frequency of words is represented in fractions in the matrix by normalization.

A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat" and "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words "fishing", "fished" and "fisher" to the root word, "fish". On the other hand, "argue", "argued", "argues", "arguing" and "argus" reduce to the stem "argu" (illustrating the case where the stem is not itself a word or root) but "argument" and "arguments" reduce to the stem "argument".

Part 2: Training the ANN algorithm: In Part 2, training patterns are formed from the matrix of numbers formed in Step 5 of Part 1 depending upon the type of algorithm used for training the topology of ANN. Supervised algorithm, unsupervised algorithm and recurrent algorithm are the three types of algorithm that can be used for training the ANN topology.

Training patterns with input features representing a document is used in this algorithm and a labelling feature that corresponding to the pattern is utilized. Thus, the training pattern should consist of input features and target output. Counter propagation algorithm, back propagation algorithm, radial basis function and time delayed network are the examples of supervised learning algorithms.

Algorithm used:

```
PorterStemmer portstem = new
PorterStemmer ();
string str = "";
WHILE ((str = br.readLine ())! = null)
{
IF (str.trim (), StartWith (sen.trim ()))
{
IF (!str.startWith ("@"))
{
    Double sim = new
    Ngram ().getSimilarly (str, sen, 4);
    String [] tem = str.spit ("--");
IF (tem.length>1)
{
     FLOAT m = Float.parseFloat (tem [1]);
    System.out.point ("Similarity: "+sim
      +" " + sen + "--" + srt);
IF (sim>0.55)
{
RETURN m;
}}}}
```

Based on different weight updating rules the algorithms are developed. Errors are calculated in the forward process of the ANN and weight updating is carried out during the reverse or recurrent process in each weight updating rules. The connections among the nodes between layers are represented by matrices in the training process of the ANN algorithms. The matrices are initialized with random numbers in majority of the algorithms.

At the final stage of the training process, the matrices contain final weight values for mapping inputs and outputs. Retrieving of the documents corresponding to word pair, final weights are used for processing with the vector corresponding to word pair during the testing of the ANN. Final weights are obtained from the pattern itself without any initialization of the weight matrices in another technique.

Part 3: Testing the ANN for document clustering: Believing that the documents are processed before and the vector representation exists, the word pair is changed into vector and processed with concluding weights procured from an ANN algorithm. The outputs in the output layer of the ANN are then utilized for similarity measures. Rather than using the similarity measures, the outputs of ANN can be as well interpreted whether the documents recovered or collected are appropriate to the words.

Part 4: Document similarity measure: A similarity measure between two documents must be ascertained in order to use a clustering or classification algorithm. Jaccard Coefficient, Euclidean Distance, Pearson Correlation Measure, Cosine Similarity and Multi-viewpoint-Based Similarity Measure are some of the existing similarity measures.

RESULTS AND IMPLEMENTATION

The algorithm used in our proposed method is implemented in Java.

N-grams: Word prediction algorithm that utilizes probabilistic methods to envisage subsequent word after examining N-1 words is known as N-Grams. Hence, we comprehend that the probability of computing the subsequent word is strongly associated to computing the possibility of a sequence of words.

Simple unsmoothed N-grams: Delegating equal probability to each word can be the uncomplicated probabilistic model for word prediction. Suppose, if N words are in a language, then 1/N would be the probability of any word succeeding another word. Actually, some words are more frequent than the others in languages and this fact is not taken into consideration in this approach which is a huge setback.

The process of assigning the probability of a word Wi following the word wi-1 is the probability of the word Wi, itself which is an enhancement from the above described model. Suppose, word “the” occurs 7% in Brown corpus and “rabbit” and the rate of occurrence is 1/10.000. So, for any word, 7% would be the probability of the subsequent word being “the”. Though in some contexts, occurrence of the “rabbit” after a word is much more probable than occurrence of “the”. For instance, “rabbit” following the word “white” seems much more logical than the word “the” following “white”.

Markov assumption: By closely examining the above proposal we can comprehend that, in certain contexts it is more probable for some words to follow a word. It would be accurate to identify all the words up to the word that we are trying to predict but identifying the entire history would be inefficient. The reason being we can encounter infinitely many sequences of sentences and the history we know would have certainly not occurred before. Therefore, we will approximate the history by only a few words.

According to Markov assumption (Bigram), we can envisage the probability of the upcoming word by solely looking at the last word encountered. We need to understand the difference between Trigram which means looking last two words in the past and N-gram which means looking N-1 words in the past. Bigram to Trigram and to N-gram can be generalized. So, $P(word_n | word_1^{n-1}) \approx P(word_n | word_{n-N+1}^{n-1})$ would be the general equation for the conditional probability of the subsequent word in a sequence where $Word_1^{n-1}$ represents word sequence $Word_1$, $Word_2$, ..., $Word_{n-1}$.

Maximum Likelihood Estimation (MLE) is the most uncomplicated way to approximate the probabilistic. It is based on taking counts from the corpus and standardizing them to lie in the interval (0, 1). For example to compute the bigram probability of word y following x is to count the bigrams c (xy) from the corpus and normalize it with the number of bigrams that starts with x.

N-grams algorithm:

```
LIST<result>processString (srring c, int n)
{
    LIST<result>t = new ArrayList<result> ();
    String spacer = "";
    FOR (int i = 0; i<n-1; i++)
    {
    Spacer = Spacer+ "%";
    }
    c = spacer+c+Spacer;
    FOR (int i = 0; i< c.lenght (); i++)
    {
      IF (i< = (c.lenght () -n))
      {
          IF (contains (c.substring (i, n+1)) >0)
          {
              t.get (i).setTheCount (results.
                    get (i).getTheCount ()+1);
          }
          ELSE
          {
             t.add (new result (c.substring
                                (i, n+i), 1));
           }
       }
    }
    RETURN t;
}
```

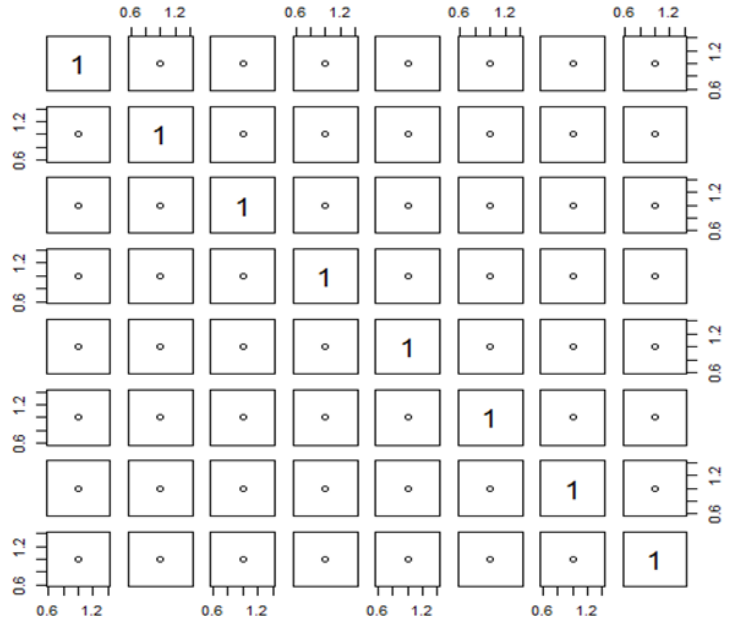

Fig. 1: Vector tf-idf chart

$$P\left(\frac{Wordn}{Wordn-1}\right) = \frac{C(Wordn-1 * Wordn)}{C(Wordn-1)}$$

In the denominator, C ($Word_{n-1}$) represents the count of bigrams starting with $Word_{n-1}$ because the bigrams starting with $Wordn_{-1}$ is equal to the number that $Wordn_{-1}$ occurs in our corpora. The general equation for estimating probability for a MLE N-gram is:

$$\frac{Wordn}{Word\frac{n-1}{n-N+1}} = \frac{C(Word\frac{n-1}{n-N+1}Wordn)}{C(Word\frac{n-1}{n-N+1})}$$

Figure 1 describes the tf-idf. Either in a compilation or corpus, the numerical statistic that is meant to reflect the significance of a word to a document is known as term frequency-inverse document frequency (tf-idf). In the process of information retrieval and text mining, it is commonly utilized as a weighting factor. The value obtained is directly dependent upon the frequency of a word occurring in the document. However, it is counterbalanced by the occurrence of the word in the corpus, which enables to manage the fact that some words occurs more frequently than others.

Search engines generally utilize alterations of the tf-idf weighting plan as an essential tool in order to score and categorize a document's applicability given a user enquiry. Besides that, it can also be utilized in a wide range of disciplines for processing stop-words.

Suppose, an uncomplicated way to find the document most relevant to the query "the brown cow" in a set of English text documents is to commence the process by removing documents. Although those documents that do not contain all three words "the", "brown" and "cow" are removed, many documents are still left. So, in order to further differentiate them, the "term frequency" is counted and sums them all together.

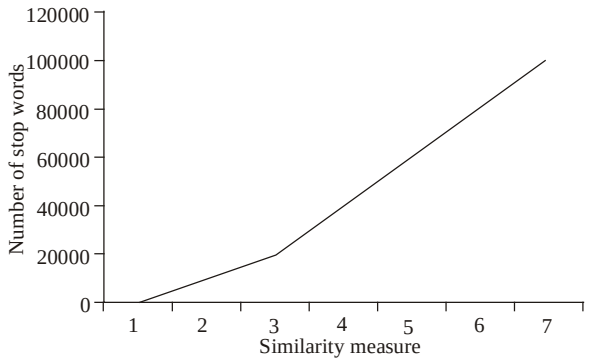

Fig. 2: No. of stop words vs. similarity

Table 1: Similarity between the two documents

No. of stop words	Similarity
1000	0.20
10000	0.40
20000	0.50
40000	0.55
60000	0.61
80000	0.65
100000	0.68

Figure 2 and Table 1 shows that precision increases as the similarity increases with increase in the number of stop words. Thus, we comprehend that in order to enhance the performance of the result it is essential to boost the stop words. Suppose we have a set of English text documents and wish to determine which document is most relevant to the query "the brown cow". A

simple way to start out is by eliminating documents that do not contain all three words "the", "brown" and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document and sum them all together; the number of times a term occurs in a document is called its term frequency.

CONCLUSION

In order to determine the challenges of pattern recognition, predicting supervised and unsupervised method, optimization, associative memory and control process and the artificial neural networks is intended to use. The outputs of ANN can be as well interpreted whether the documents recovered or gathered are appropriate to the words. Unsupervised processes form its basis only on detecting the similarity association between documents with the productivity as a set of clusters. A commonality measure is defined in this research to detect resemblance between two text files which is used as a similarity measure. Any existing recurrent item finding algorithm such as apriori or fp-tree is applied to the first set of text files to decrease the dimension of the input text files. The main part of this study work extends n-gram proposed algorithm, based on Artificial Neural Networks and matches all the phonemes, syllables, letters, words or base pairs as per the application. Similarity of the text based on the stop word will give the efficiency, so this system will act like ANN and automatically based on our definition the systems will produce output. Future work extends the application of this same similarity measures in various other neural network algorithms to accomplish improved results.

REFERENCES

Altlnel, B., M.C. Ganiz and B. Diri, 2013. A novel higher-order semantic kernel for text classification. Proceeding of the International Conference on Electronics, Computer and Computation (ICECCO, 2013), pp: 216-219.

Audhkhasi, K., A. Sethy, B. Ramabhadran and S.S. Narayanan, 2014. Semi-supervised term-weighted value rescoring for keyword search. Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP, 2014), pp: 7869-7873.

Baleia, J., P. Santana and J. Barata, 2014. Self-supervised learning of depth-based navigation affordances from haptic cues. Proceeding of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC, 2014), pp: 146-151.

Beebe, N.L., L.A. Maddox, L. Lishu and S. Minghe, 2013. Sceadan: Using concatenated N-gram vectors for improved file and data type classification. IEEE T. Inf. Foren. Sec., 8(9): 1519-1530.

Bollegala D., Y. Matsuo and M. Ishizuka, 2006. Disambiguating personal names on the web using automatically extracted key phrases. In Proc. of the 17th European Conference on Arti¯cial Intelligence, pp: 553-557.

Hamani, M.S. and R. Maamri, 2013. Word semantic similarity based on document's title. Proceeding of the 24th International Workshop on Database and Expert Systems Applications (DEXA), pp: 43-47.

Hiraishi, K., M. Yoshimoto and K. Kobayashi, 2013. Diagnosis of stochastic discrete event systems based on N-gram models with wildcard characters. Proceeding of the IFIP/IEEE International Symposium on Integrated Network Management (IM'2013), pp: 1383-1388.

Jiang J. and D. Conrath, 1998. Semantic similarity based on corpus statistics and lexical taxonomy. In Proc. of the International Conference on Research in Computational Linguistics ROCLING X.

Kavitha Karun, A., P. Mintu and K. Lubna, 2013. Comparative analysis of similarity measures in document clustering. Proceeding of the International Conference on Green Computing, Communication and Conservation of Energy (ICGCE, 2013), pp: 857-860.

Kongsorot, Y. and P. Horata, 2014. Multi-label classification with extreme learning machine. Proceeding of the 6th International Conference on Knowledge and Smart Technology (KST, 2014), pp: 81-86.

Liangboonprakong, C. and O. Sornil, 2013. Classification of malware families based on N-grams sequential pattern features. Proceeding of the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA, 2013), pp: 777-782.

Lin D., 1998. An Information-Theoretic Definition of Similarity, Proc.15th Int'l Conf. Machine Learning (ICML), pp: 296-304.

Malandrakis, N., A. Potamianos and S. Narayanan, 2013. Continuous models of affect from text using n-grams. Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP, 2013), pp: 8500-8504.

Manna, S., B.S.U. Mendis and T. Gedeon, 2009. Hierarchical document signature: A specialized application of fuzzy signature for document computing. Proceeding of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp: 1083-1088.

Mclean D., Y. Li, and Z.A. Bandar, 2003. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources, IEEE Trans. Knowledge and Data Eng., 15(4): 871-882.

Megler, V.M. and D. Maier, 2014. Are datasets like documents? Evaluating similarity-based ranked search over scientific data. IEEE T. Knowl. Data En., 99: 1.

Messina, R. and C. Kermorvant, 2014. Over-generative finite state transducer n-gram for out-of-vocabulary word recognition. Proceeding of the 11th IAPR International Workshop on Document Analysis Systems (DAS, 2014), pp: 212-216.

Rada R., H. Mili, E. Bichnell and M. Blettner, 1989. Development and Application of a Metric on Semantic Nets. IEEE Trans. Systems, Man and Cybernetics, 19(1):17-30.

Reddy, G.S., T.V. Rajinikanth and A.A. Rao, 2014. A frequent term based text clustering approach using novel similarity measure. Proceeding of the IEEE International Advance Computing Conference (IACC, 2014), pp: 495-499.

Resnik P., 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. Proc. 14th Int'l Joint Conf. Aritificial Intelligence.

Selvi K., R.M. Suresh, 2012. Measure Semantic Similarity between words Using Fuzzy Formal Concept Analysis. In Proc. of Int'l conference On Computer Science and Engineering ICCSE2012-IRNet, pp: 31-34.

Sahami, M. and T. Heilman, 2006. A web-based kernel function for measuring the similarity of short text snippets. Proc. of 15th International World Wide Web Conference.

Taeho, J., 2013. Application of table based similarity to classification of bio-medical documents. Proceeding of the IEEE International Conference on Granular Computing (GrC), pp: 162-166.

Toselli, A.H. and E. Vidal, 2014. Word-graph based handwriting key-word spotting: Impact of word-graph size on performance. Proceeding of the 11th IAPR International Workshop on Document Analysis Systems (DAS, 2014), pp: 176-180.

Yu, K., J. Yunde and F. Yun, 2014. Interactive phrases: Semantic descriptionsfor human interaction recognition. IEEE T. Pattern Anal., 36(9): 1775-1788.