Content based Document Classification using Soft Cosine Measure

Md. Zahid Hasan[1], Shakhawat Hossain[2], Md. Arif Rizvee[3], Md. Shohel Rana[4]
Department of Computer Science & Engineering, Daffodil International University, Dhaka, Bangladesh[1, 3, 4]
Department of Computer Science & Engineering, International Islamic University Chittagong, Chittagong, Bangladesh[2]

***Abstract*—Document classification is a deep-rooted issue in information retrieval and assumed to be an imperative part of an assortment of applications for effective management of text documents and substantial volumes of unstructured data. Automatic document classification can be defined as a content-based arrangement of documents to some predefined categories which is for sure, less demanding for fetching the relevant data at the right time as well as filtering and steering documents directly to users. For recovering data effortlessly at the minimum time, scientists around the globe are trying to make content-based classifiers and as a consequence, an assortment of classification frameworks has been developed. Unfortunately, because of using conventional algorithms, almost all of these frameworks fail to classify documents into the proper categories. However, this paper proposes the Soft Cosine Measure as a document classification method for classifying text documents based on its contents. This classification method considers the similarity of the features of the texts rather than making their physical compatibility. For example, the traditional systems consider 'emperor' and 'king' as two different words where the proposed method extracts the same meaning for both of these words. For feature extraction capability and content-based similarity measure technique, the proposed system scores the classification accuracy up to 98.60%, better than any other existing systems.**

Keywords*—Classification; similarity; feature extraction; cosine similarity; soft cosine measure; content; document

I. Introduction

Document classification refers to the way of storing similar documents together. This is considered as a major challenge for Information Retrieval in light of the fact that getting the right documents at the right time is hardly possible if the records are not legitimately classified and sorted out. Document classification can be done intellectually or automatically. Content-based document classification is an automatic arrangement of documents where the system goes through the entire content and groups them in light of the extracted features. There are several well-established algorithms for solving the content-based document classification problems. But none of these algorithms are fit for dealing with the similitude of the two words meaning the same. So, a much complex algorithm is required for considering the closeness of features in a vector space model (VSM) [1]. The framework proposed in this paper utilizes the Soft Cosine Measure for classifying text documents, which finds out the likeness of features of two documents. The proposed framework utilizes both Term Frequency (TF) [2] and Term Frequency-Inverse Document Frequency (TF-IDF) [3] to find the most important words in a text with the goal that, no vital term is missed. As a result, the framework is capable of providing with the most precise outcomes. The performance accuracy of the proposed system is 11.2% superior to the most accurate document classifiers like Cosine Similarity [4]. The proposed system is tested 103 times and almost every time it classified documents correctly.

The paper is composed as follows: In Section 2, an investigation into the previous document classification frameworks is presented. Based on that investigation, a new classification framework, Soft Cosine Measure is proposed in Section 3. A data processing technique for the proposed classification method is described in Section 4. Section 5 describes the feature extraction techniques and Section 6 demonstrates the feature vector construction procedure. In Section 7, the implementation of Soft Cosine Measure for content-based document classification is outlined. A numerical study is provided in Section 8 and a detailed analysis of the system outcomes is done in Section 9. The paper is finished up in Section 10.

II. Background Study

The evolution of document classification has started a long ago but still, it's a far away from getting saturated. Researchers have been applying various mathematical models to boost a sophisticated document classifier and, in that consequence, a number of documents classification frameworks have been established.

C. Goutte, L. Versoud, E. Gaussier, Eybens used Probabilistic Hierarchical Model for text categorization [5]. Probabilistic Hierarchical Model deals with classifying objects into similar categories where an object may coexist in multiple categories. For that purpose, the object-categories are organized in a hierarchical process where a clear dependency among the categories is visible. The final classification task is accomplished by providing some labels to the objects.

Evgeniy Gabrilovich and Shaun Markovitch argued to use C4.5 for text categorization [6] rather than traditional algorithms like Support Vector Machine (SVM). They showed that C4.5 is more capable of handling Redundant Features than SVM for categorizing texts into the preferred classes. Evgeniy Gabrilovich and Shaun Markovitch used Aggressive Feature Selection technique for developing a more sophisticated text categorization model. De Mello R.F., Senger L.J. and Yang L.T. (2005) introduced an Artificial Neural Network for content-based text classification [7]. The artificial neural network is much enough intelligent to cluster documents without any previous domain knowledge. It uses document features to organize documents into proper clusters.

Y. H. Li and A. K. Jain conducted an experiment over Naive Bayes Classifier, Nearest Neighbor Classifier, Decision Trees and a Subspace Method [8] to find out the best-fit algorithm for document classification. The authors found that Naive Bayes and Subspace Method performed better than the other two classifiers on their data sets. They used the downloaded dataset for their experiments and their classification accuracy was approximately 83%. Authors also conducted an experiment to compare the performance accuracy among the Bayesian classifier, decision tree (ID3) and nearest neighbor with respect to the binary feature vector.

Shreyatakhatri proposed an improved K-means algorithm for Document Clustering [9]. In their experiment, they found that their proposed algorithm's accuracy was much better compared to current algorithms regarding time complexity and F-measure. On the other hand, Janani Balakumar proposed Enhanced Bisecting K-means (EBK) algorithm for clustering text document [10]. EBK is presented as an improved version of the Bisecting K-means clustering algorithm, capable of handling the limitations of Bisecting K-means algorithm while clustering documents into the assigned classes.

In 2017 S. Adinugroho, Y. A. Sari, M. A. Fauzi, and P. P. Adikara proposed a hybrid algorithm for clustering text documents correctly [11]. Their proposed algorithm consists of two prominent algorithms: Latent Semantic Indexing (LSI) and Pillar Algorithm. S. Adinugroho, Y. A. Sari, M. A. Fauzi, and P. P. Adikara used LSI to extracts features and Pillar Algorithm to select seeds. A year back, in 2016, P. Bafna, D. Pramod and A. Vaidya proposed a new document clustering method, TF-IDF along with fuzzy k-means algorithm [12] to achieve the maximum clustering accuracy. The authors carried out a number of experiments on several datasets to verify the clustering accuracy of their proposed methodology. For that purpose, an approach for removing clamorous as well as less important data was followed by the authors. Hierarchical agglomerative clustering approach was applied in this experiment to optimize the system's performance.

Some researchers believe that Cosine Similarity secures the maximum accuracy in case of text document clustering. Lailil Muflikhah and B. Baharudin claimed that their proposed method performs better than all other clustering algorithms [13]. They calculated the performance of their system with f-measure some 0.91 and entropy about 0.51. They also claimed that the performance of their proposed system remains the same even if the system is applied to a huge amount of data. Baoli Li and Liping Han conducted a very deep analysis on cosine similarity for clustering text documents. Based on the experimental results, Baoli Li and Liping Han stated that cosine similarity is not always fit for document clustering tasks. Then they proposed Distance Weighted Cosine Similarity [14] for classifying text documents.

M. L. Aishwarya and K. Selvi proposed an intelligent similarity measure technique [15] to solve document clustering problems. They used the concept of the Neural Network algorithm to develop their intelligent similarity measure approach. These researchers suggested Echo State Neural Network and Radial Basis Function to cluster documents if the tainting dataset contains irrelevant documents.

Unfortunately, none of these classifiers could overcome some common drawbacks which lead the scientist to develop a content-based document classification framework, a well more intelligent classification model, equipped for classifying any text documents just by experiencing its content.

III. Methodology

A. Soft Cosine Measure

Soft Cosine Measure, a new concept in classification tasks, considers the pairs of features [16] to discover the similitude between two word vectors in a vector space model (VSM) [17]. Although Soft Cosine Measure has derived from the Cosine Similarity, there is a major distinction between these two concepts. Cosine Similarity ordinarily considers the cosine of the angle between two non-zero vectors to discover the similitude between two documents [18] where Soft Cosine Measure calculates comparability between the features extracted from those documents. For, two N-dimension vectors α and β the Soft Cosine Similarity can be calculated as follows:

$$\text{Soft Cosine } (\alpha,\beta\ ) = \frac{\sum_{i.j}^{N} s_{ij}\alpha_i\beta_j}{\sqrt{\sum_{i.j}^{N} s_{ij}\alpha_i\alpha_j}\sqrt{\sum_{i.j}^{N} s_{ij}\beta_i\beta_j}};$$

Where, $s_{i,j}$= similarity ($feature_i$, $feature_j$)

If, $s_{i,i}$=1 and $s_{i,j}$= 0 for i ≠ j then,

$$\text{Soft Cosine } (\alpha,\beta\ ) = \frac{\sum_{i.j}^{N} \alpha_i\beta_i}{\sqrt{\sum_{i.j}^{N} \alpha_i\alpha_i}\sqrt{\sum_{i.j}^{N} \beta_i\beta_i}}$$

$$= \frac{\sum_{i=1}^{N} \alpha_i\beta_i}{\sqrt{\sum_{i=1}^{N} {\alpha_i}^2}\sqrt{\sum_{i=1}^{N} {\beta_i}^2}}$$

$$= \frac{\alpha\ .\ \beta}{\|\alpha\|\|\beta\|} = \text{Cosine Similarity.}$$

So, when there is no similarity between the features of the objects, Soft Cosine Measure becomes proportional to the regular Cosine Similarity formula.

B. Data Processing

For obtaining a better classification model, the proposed framework needs to be trained with a legitimate set of pre-processed data. And to process data properly, the proposed system encounters a couple of steps as stated in Fig. 1.

C. Removing Punctuation

Removing punctuation is a vital task in natural language processing. There are quantities of approach to expel punctuation from a text. In the proposed system, a regular expression is used to dispose of all the punctuations.

D. Converting String into Lower Case Letter

For processing the data in the most convenient way, all the letters in a textual content are converted into the lower-case form.

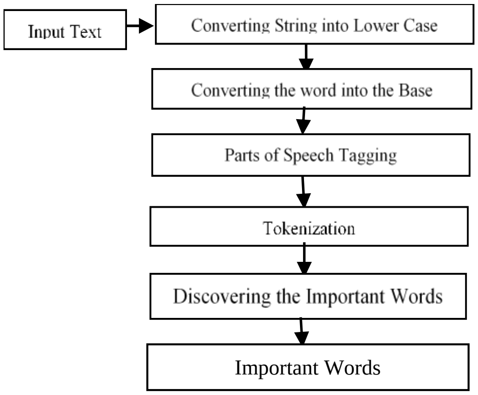

Input Text
Converting String into Lower Case
Converting the word into the Base
Parts of Speech Tagging
Tokenization
Discovering the Important Words
Important Words

Fig. 1. Data Processing Diagram.

E. Converting the Word into the base form

A critical task in natural language processing is to convert all the words into their base form. This causes a framework to comprehend words regardless of whether they are in various structures. For instance, Table I represent some words in their base forms.

To convert the words into their base form, Streamer Porter Algorithm [19], [20] is used in this system.

F. Parts of Speech Tagging

The system proposed in this paper has used Latent Analogy [21] for tagging parts of speech. Only the noun and the verbs are used to train the system.

G. Tokenization

The process of breaking up the content into distinct meaningful units is recognized as tokenization. Tokenization is an important task for this system as the system makes the vector with some particular words or tokens.

H. Discovering the Important Words

All the words in a text file are not equally important for a specific purpose. So, researchers have taken this task as a challenge to find the important words out from a text document. As a result, various algorithms have been developed to discover the vital words from a text document. However, the proposed framework utilizes two most proficient algorithms to choose the important words: (TF) and TF-IDF as shown in Fig. 2.

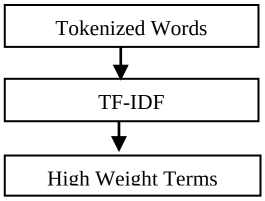


Fig. 2. Important Words Acquisition.

TABLE I. WORDS IN VARIOUS FORMS AND THEIR BASE FORM

Word in different Form	Base Word
ran, run, running, runs, runner	run
good, better, best	good

I. Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency [2] is the calculation of how many times each word appears in a text document. Term Frequency (TF) in documents can be calculated by using the logarithmic scale.

$$tf_{t,d} = \begin{cases} \log\left(1 + f_{t,d}\right), & if\ f_{t,d} > 0 \\ 0, & otherwise \end{cases}$$

where, t defines a term, d is the document and $tf_{t,d}$ is the frequency of the term in the documents.

Inverse Document Frequency [22], [23] is the calculation that determines whether a term is common or rare across all the documents. Inverse Document Frequency can be easily calculated by the following formula.

$$idf_{t,D} = \log\frac{|D|}{|d \in D: t \in d|}$$

where, d is a document and D is the set of documents.

Term Frequency-Inverse Document Frequency [3] calculates the high weighted terms in a set of documents.

$$tf - idf_{t,d,D} = tf_{t,d} * \ idf_{t,D}$$

where, t is a term, d is a document and D is a set of documents.

The implementation of TF-IDF mimics the following algorithm:

```
1. Set d ←Text Document, t← a specific term; t € d
2. Set n ←Number of times term t appears in d,
3. Set m← Total number of terms in d;
4. Compute X ← n ÷ m;
5. Set N← Total number of documents and M← Number
   of documents with t;  M € N
6. Compute Y← N ÷ M;
7. Set K ←Term Frequency Inverse Document Frequency
8. Compute K← X × Y;
9. Print K;
```

IV. FEATURE EXTRACTION

The basic convenience of Soft Cosine Measure over Cosine Similarity is its capability of computing similarity between two documents by using their inner features regardless of whether they have any physical comparability or not. For doing that, Soft Cosine Measure discovers features of each vital term in all the documents in a document set. For that purpose, it uses a dictionary that exposes all the possible words with the same meaning for a given word. This process is utilized at both the training and system handling time. The entire procedure is graphically spoken to in the accompanying segment.

A. Feature Extraction During System Training

For preparing the framework, vital terms are gathered from a substantial amount of datasets which indicates a solid plausibility of having duplicate terms. So, it becomes essential to extract feature at data processing time to remove redundant high weight terms. The imperative terms are principally put away in a cluster and the accompanying algorithm is utilized to evacuate all the redundant data.

```
1. start
2. set i to 0
3. if A[i+1]==A[i]
     3.1 remove A[i+1]
     3.2 else look for A[i] in lexicon and put all possible
         features of A[i] into B[j]
         3.2.1  set j to 0
         3.2.2  if A[i+1]==B[j]
                3.2.2.1  remove A[i+1]
         3.2.3  else set j to j+1;
     3.3 set i to i+1;
4. stop
```

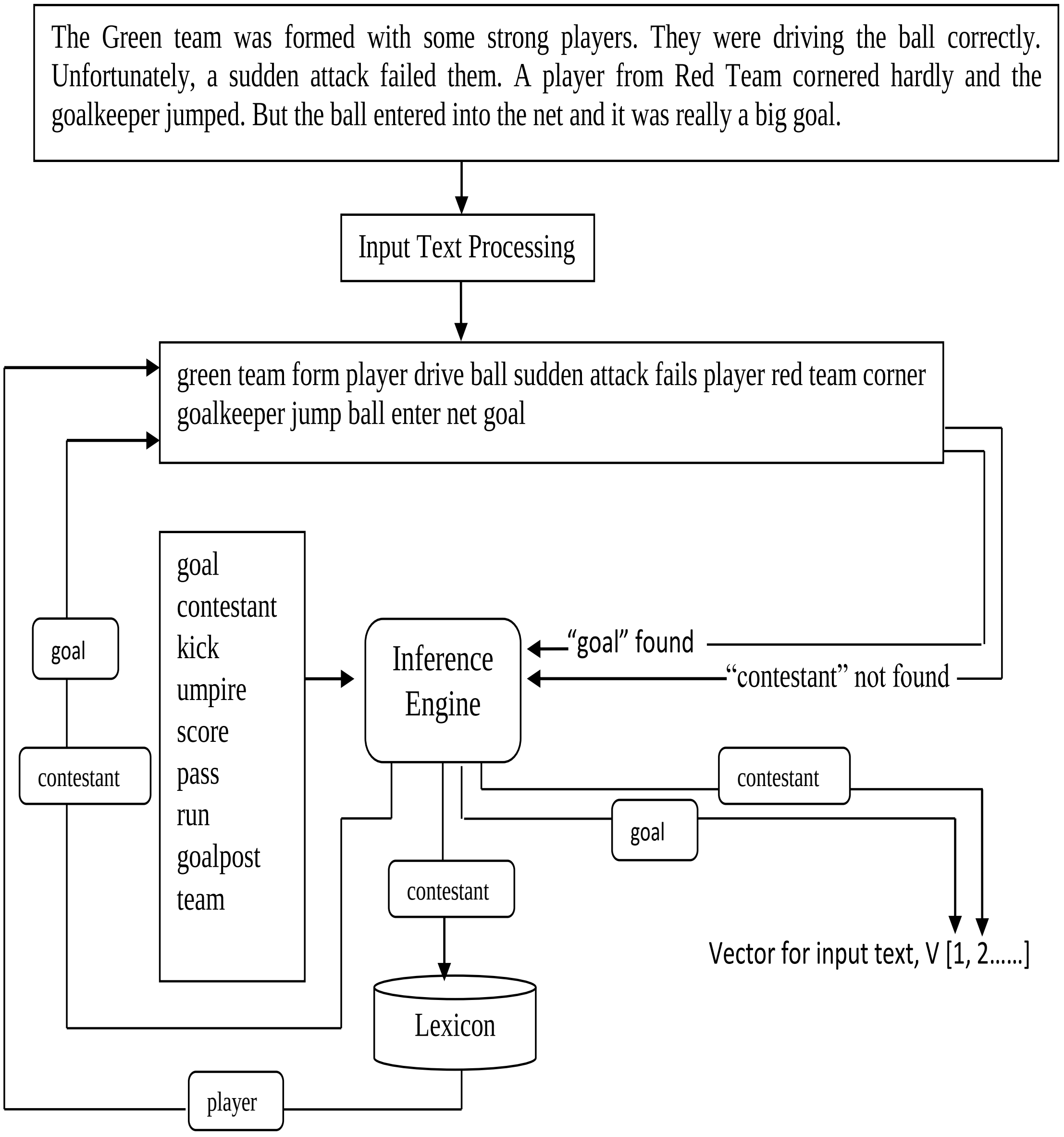
B. Feature Extraction During Similarity Scoring

Feature extraction during classification is the key operation that enables the proposed framework to perform better than all other existing frameworks. This task is performed in accordance with the following algorithm (Fig. 3).

C. Feature Vectors Construction

In the wake of finding the essential words, the framework figures the Term Frequency for those words and builds the final vectors with the resultant numeric quantities. The following algorithm demonstrates the vector construction process.

```
1. start
2. declare high weight terms as $T_{i\,;\,1\le i\le n}$
3. declare the document as $d_{i\,;\,1\le i\le n}$
4. compute term frequency, TF for $T_1$ in document, $d_1$
5. extract features for $T_1$ in $d_1$
   5.1 add all TF for $T_1$
   5.2 rerun the sum
6. repeat step 4 for n times
7. store the result in a vector, V
8. repeat step 3 and 4 for n times
9. end
```



The Green team was formed with some strong players. They were driving the ball correctly. Unfortunately, a sudden attack failed them. A player from Red Team cornered hardly and the goalkeeper jumped. But the ball entered into the net and it was really a big goal.

green team form player drive ball sudden attack fails player red team corner goalkeeper jump ball enter net goal

Fig. 3. Feature Extraction Procedure.

V. Implementation of the System

Soft Similarity or Soft Cosine Measure classifies text documents in view of the content it bears. For that, only a cosine angle between the features of the trained data and the input data is estimated. The architectural design of how Soft Cosine Measure classifies the text documents is presented in Fig. 4.

Based on the similarity scores, performed by different text documents, the system classifies the documents into some predefined classes. A very straightforward algorithm for the proposed system is given below.

```
1.  start
2.  scan the input text, T
3.  process T
4.  extract the feature of T from the lexicon, D
5.  make a feature vector, $V_i$
6.  find the similarity score between $V_i$ and $V_t$ ; {$V_t$ | $V_t$∈V; |V|= n} where,
      V= trained data vector and n= number of classes
7.  store the score in a list, L
8.  repeat Step 6 and 7 for n times
9.  find the biggest score from L
10. make the final decision to put T into $S_t$;  {$S_t$ | $S_t$∈ S; |S|=|V| } S= set of classes.
11. end
```

Soft Cosine Measure considers the features in VSM which makes it equipped for figuring the likenesses between two documents regardless of whether they have any common word or not. It uses a lexicon to extract the features that are actually taken into account to quantify the similarity between the meaning of two words rather than the words themselves.

In the above figures, a very transparent comparison between Cosine Similarity and Soft Cosine Measure is stated. Fig. 5(a) exhibits a very big difference between Hi and Hello where in Fig. 5(b), Hi and Hello are considered as the words with the same meaning.

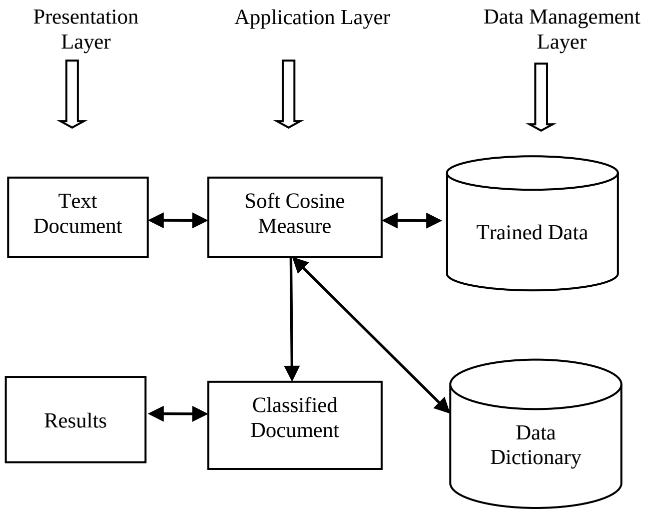
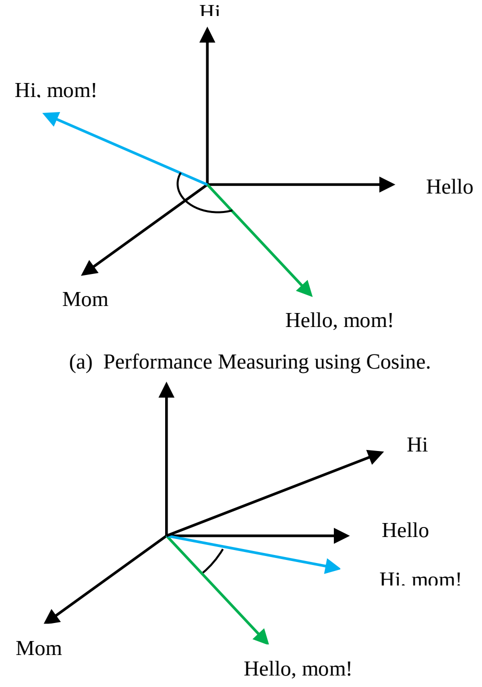


Fig. 4. System Architecture for Document Classification.



(a) Performance Measuring using Cosine.

(b) Performance Measuring using Soft Cosine.

Fig. 5. Comparison between Cosine Similarity and Soft Cosine Measure.

VI. Experimental Results

A number of experiments have been conducted over the proposed system and each of these experiments confirms that Soft Cosine Measure performs better than Cosine Similarity. The similarity scores of Cosine Similarity and Soft Cosine Measure between two sample documents are presented in Table II.

TABLE II. Similarity Score Comparison between CS and SCM

Document 1	Document 2	Cosine Similarity Score	Soft Cosine Measure Score
Every Mom is the most amazing person for her children - she's their heroine. Mom always knows how her child feels and can help with any problem. Mom can make the most complicated braid and explain fractions; Mom can help to wake her child up in the morning and hug her tightly when she's sad.	Each Mom is the most stunning individual for her kids - she's their champion. Mother dependably knows how her kid feels and can help with any issue. Mother can make the most entangled mesh and clarify divisions; Mom can get her kid up toward the beginning of the day and embrace her firmly when she's pitiful.	0.5041	0.9997

They system is tested with different types and sizes of documents to observe its performance under different circumstances. Fig. 6 shows that Soft Cosine Measure performs 49.56% better than the Cosine Similarity. This experiment has been done with 50 distinct documents and for every document Soft Cosine Measure performs around 45% better than Cosine Similarity.

However, the proposed system has been tested for 114 times with 103 different documents to classify into five categories. Each time the system has classified 101 documents correctly which secures its accuracy rate up to 98.06% (Table III).

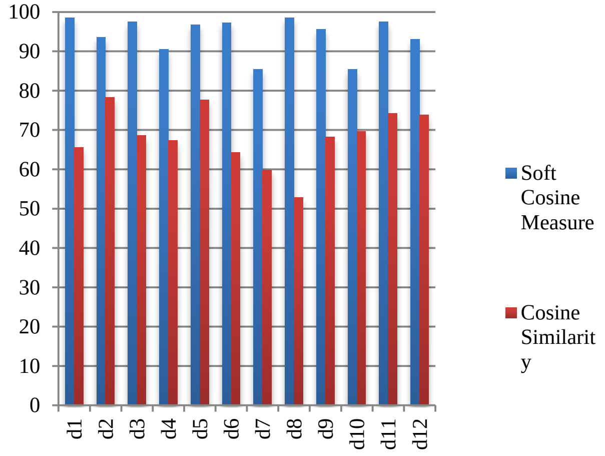
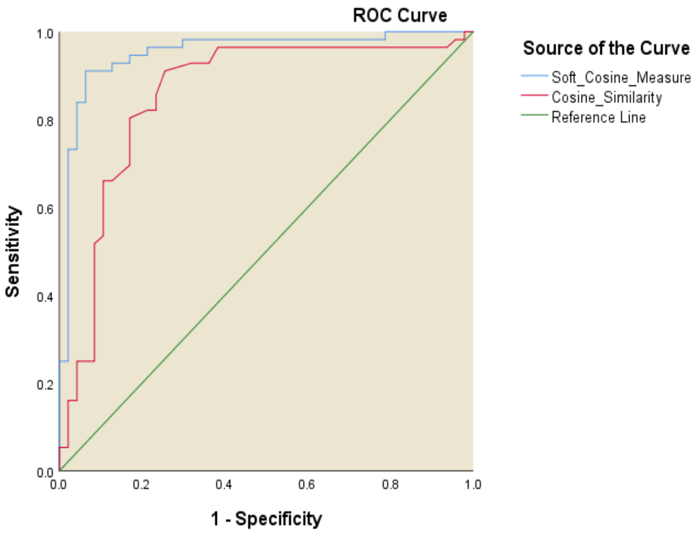


Fig. 6. Similarity Scores of Soft Cosine Measure and Cosine Similarity for 12 different Documents.

TABLE III. DOCUMENT CLASSIFICATION RESULTS OF SOFT COSINE MEASURE

Documents	System Selected Category	Actual Category	Remark
Doc1	History	History	✓
Doc 2	History	History	✓
Doc3	Literature	Literature	✓
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
Doc48	Comics	Comics	✓
Doc49	Politics	History	X
Doc50	Science	Science	✓
Doc51	Science	Science	✓
Doc52	Literature	Literature	✓
Doc53	History	Politics	X
.	.	.	.
.	.	.	.
.	.	.	.
.			.
Doc100	Comics	Comics	✓
Doc101	History	History	✓
Doc102	Literature	Literature	✓
Doc103	Politics	Politics	✓



Diagonal segments are produced by ties.

Fig. 7. ROC Curves Illustrate the Classification Accuracy of Soft Cosine Measure and Cosine Similarity.

Fig. 7 shows the ROC (Receiver Operating Characteristics Curve) curves where the blue curve represents the classification accuracy of Soft Cosine Measure and the red curve depicts the accuracy of Cosine Similarity. The AUC (area under the curve) of Soft Cosine Measure is 0.925 and AUC of Cosine Similarity is 0.840. So, it turns out to be evident that, the classification accuracy of Soft Cosine Measure is superior to the Cosine Similarity.

VII. RESULT ANALYSIS

Though document classification is a very important task in natural language processing for its extended use case, it is yet a big challenge to come across the most intense precision in document classification. Researchers have tried numerous strategies to locate the most extreme precision in content-based document classification. According to a contemporary research outcome [24], it has been clear to researchers that Support Vector Machine (SVM) provides the maximum accurate result than any other methods in document classification which is estimated 90.26%. But another recent research demonstrates that cosine similarity classifies content-based document more efficiently than SVM and its accuracy reaches up to 93.9% [4]. However, this research with adequate evidence clarifies that Soft Cosine Measure performs better than Cosine Similarity. A very straightforward comparison among different classification accuracy is given in Table IV.

TABLE IV. CLASSIFICATION ACCURACY OF DIFFERENT METHODS

Methodology	Accuracy (%)
SVM	90.26
Decision Tree	76.99
K Nearest Neighbor	84.60
Naive Bayes	84.70
Cosine Similarity	93.90
Soft Cosine Measure	**98.60**

VIII. Conclusion

Soft Cosine Measure is a state-of-the-art mathematical model that considers the features in a vector space model to quantify the comparability between two text documents. The proposed system uses this mathematical model to construct a content-based document classification framework. To classify any document the system considers the edge between the component vectors of the given documents and the readied data. The system secures its precision rate up to 98.6% which is vastly improved than some other existing framework. However, an improved feature extraction technique can increase the performance of Soft Cosine Measure up to 100%.

References

[1] D.L. Lee, HueiChuang,K.Seamons,"Document ranking and the vector-space model", IEEE Software ,Volume: 14, Issue: 2, Mar/Apr 1997 ,DOI: 10.1109/52.582976

[2] Mikio Yamamoto, Kenneth W. Church, "Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus",Computational Linguistics archiveVolume 27 Issue 1, March 2001,Pages 1-30, doi:10.1162/089120101300346787

[3] Rung-Ching Chen, Jui-Yuan Liang, Ren-Hao Pan, "Using recursive ART network to construction domain ontology based on term frequency and inverse document frequency" Expert Systems with Applications, Volume 34, Issue 1, January 2008, Pages 488-501

[4] Radhamothukuri, Nagaraju.M, DivyaChilukuri, "Similarity Measure For TextClassification", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 5, Issue 6, November - December 2016 ,ISSN 2278-6856

[5] C. Goutte, L. Versoud, E. Gaussier, Eybens, “Method forMulti-class, multi-label categorization using probabilistic hierarchical modeling”, U.S Patent 7 139 754 B2, Nov 21, 2006

[6] EvgeniyGabrilovich and Shaun Markovitch, “Text Categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5”, Proceedings of 21st International Conference on Machine Learning, 2004.

[7] de Mello R.F., Senger L.J., Yang L.T. (2005) Automatic Text Classification Using an Artificial Neural Network. In: Ng M.K., Doncescu A., Yang L.T., Leng T. (eds) High Performance Computational Science and Engineering. IFIP—The International Federation for Information Processing, vol 172. Springer, Boston, MA.

[8] Y. H. Li A. K. Jain,"Classification of Text Documents", The Computer Journal, Volume 41, Issue 8, 1 January 1998, Pages 537–546, https://doi.org/10.1093/comjnl/41.8.537

[9] Shreyata khatri1 ,Dr. KanwalGarg, "Document Clustering Using Improved K-Means Algorithm"International Journal of Engineering Research and General Science Volume 4, Issue 3, May-June, 2016 ISSN 2091-2730.

[10] BALAKUMAR, Janani; VIJAYARANI, S.. An Improved Bisecting K-Means Algorithm for Text Document Clustering. International Journal of Knowledge Based Computer System, [S.l.], p. 32-37, dec. 2016. ISSN 2321-5623.

[11] S. Adinugroho, Y. A. Sari, M. A. Fauzi and P. P. Adikara, "Optimizing K-means text document clustering using latent semantic indexing and pillar algorithm," 2017 5th International Symposium on Computational and Business Intelligence (ISCBI), Dubai, 2017, pp. 81-85.,doi: 10.1109/ISCBI.2017.8053549

[12] P. Bafna, D. Pramod and A. Vaidya, "Document clustering: TF-IDF approach," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 61-66. doi: 10.1109/ICEEOT.2016.7754750

[13] L. Muflikhah and B. Baharudin, "Document Clustering Using Concept Space and Cosine Similarity Measurement," 2009 International Conference on Computer Technology and Development, Kota Kinabalu, 2009, pp. 58-62. doi: 10.1109/ICCTD.2009.206.

[14] Li B., Han L. (2013) Distance Weighted Cosine Similarity Measure for Text Classification. In: Yin H. et al. (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2013. IDEAL 2013. Lecture Notes in Computer Science, vol 8206. Springer, Berlin, Heidelberg

[15] M. L. Aishwarya and K. Selvi, "An intelligent similarity measure for effective text document clustering," 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), Kovilpatti, 2016, pp. 1-5.doi: 10.1109/ICCTIDE. 2016.7725342

[16] Sidorov, Grigori; Gelbukh, Alexander; Gómez-Adorno, Helena; Pinto, David. "Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model" Computación y Sistemas. 18 (3): 491–504. doi:10.13053/CyS-18-3-2043 Retrieved 7 October 2014.

[17] ThomasMikolov et al. Efficient Estimation of Word Representations in Vector Space ,arXiv:1301.3781v3 [cs.CL] 7 Sep 2013

[18] Li B., Han L. (2013) Distance Weighted Cosine Similarity Measure for Text Classification. In: Yin H. et al. (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2013. IDEAL 2013. Lecture Notes in Computer Science, vol 8206. Springer, Berlin, Heidelberg

[19] M.F. Porter, (1980) "An algorithm for suffix stripping", Program, Vol. 14 Issue: 3, pp.130-137, https://doi.org/10.1108/eb046814

[20] Atharva Joshi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 2016, 266-26

[21] Jerome R. Bellegarda, Part-of-Speech Tagging by Latent Analogy IEEE Journal of Selected Topics in Signal ProcessingVolume: 4, Issue: 6 Dec. 2010 ;DOI: 10.1109/JSTSP.2010.2075970

[22] Church K., Gale W. (1999) Inverse Document Frequency (IDF): A Measure of Deviations from Poisson. In: Armstrong S., Church K., Isabelle P., Manzi S., Tzoukermann E., Yarowsky D. (eds) Natural Language Processing Using Very Large Corpora. Text, Speech and Language Technology, vol 11. Springer, Dordrecht, https://doi.org/10.1007/978-94-017-2390-9_18

[23] Stephen Robertson, (2004) "Understanding inverse document frequency: on theoretical arguments for IDF", Journal of Documentation, Vol. 60 Issue: 5, pp.503-520, https://doi.org/ 10.1108/ 00220410410560582

[24] Choudhury, S., Batra, T., Hughes, C., & LEMMATIZER, L. (2016). Content-based and link-based methods for categorical webpage classification.