SIMILARITY MEASURE FOR TEXT CLASSIFICATION

Radha mothukuri, Nagaraju.M, Divya Chilukuri

Department Of Computer Science and Engineering,
QIS Institute of Technology, Ongole.

Abstract

Text processing plays very important role in information retrieval, data mining and web search. Text classification can efficiently enhance the text processing capability by automatically sorting out them according to defined collection of categories. Measuring the similarity between documents is an important operation in the text processing field. In this work, a similarity measure is proposed to compute the similarity between two documents with respect to a feature. The proposed measure takes the following three cases into account: The feature appears in both documents, the feature appears in only one document, and the feature appears in none of the documents. For the first case, the similarity increases as the difference between the two involved feature values decreases. Furthermore, the contribution of the difference is normally scaled. For the second case, a fixed value is contributed to the similarity. For the last case, the feature has no contribution to the similarity. The proposed measure will be extended to gauge the similarity between two sets of documents. The effectiveness of this measure will be evaluated on several real-world data sets for text classification problems.

Keywords*:***-**text mining, classification, similarity measure,accuracy.

1. INTRODUCTION

The fast growth and dynamic change of online information have provided us a very large amount of information and lead to information overload. Text classification is an important tool for organizing documents into categorizations by applying statistical methods. By this way, the utilization of the documents can be expected to be more effective. As a result, the situation of information overload may be alleviated. The aim of document categorization is to assign a number of appropriate categories to a textual document based on the content. This categorization process has many applications such as document routing, dissemination, or filtering. A large number of techniques have been developed for text classification, including Naive Bayes, Nearest Neighbor, neural networks, regression, rule induction, and Support Vector Machines.

1.1 Text Classification

Text classification is the task of automatically classifying a set of text documents into different categories from a predefined set. If a document belongs to exactly one of the categories, it is a single-label classification task; otherwise, it is a multi-label classification task. TC uses several tools from Information Retrieval (IR) and Machine Learning (ML) [3] and has received much attention in the last years from both researchers in the academia and industry developers.

The stages of text classification are

- Document collection
- Preprocessing
- Feature Selection
- Feature weighting
- Classification algorithms
- Performance measure

1.1.1Documents Collection

It is a first step of classification process. Different types(formats) of document like html, pdf, doc, web content etc are collected.

1.1.2 Preprocessing

Data pre-processing comprises six sub-components including document conversion function word removal, word stemming, feature selection, dictionary construction, and feature weighting. The functionality of each component is described as follows:

- Document converting: converts different types of documents such as XML, PDF, HTML, DOC format to plain text format.
- Function word removal: removes topic-neutral words such as articles (a, an, the), prepositions (in, of, at), conjunctions (and, or, nor), etc. from the documents.
- Word stemming[16]: standardizes word’s suffixes (e.g., labeling **--** label, introduction introduct).

1.1.3 Feature Selection

Feature selection studies how to select a subset or list of attributes or variables that are used to construct models describing data. Its purposes include reducing dimensionality, removing irrelevant and redundant features, reducing the amount of data needed for learning, improving algorithms’ predictive accuracy, and increasing the constructed models’ comprehensibility.

Dimension reduction techniques can generally be classified into Feature Extraction (FE) approaches and Feature Selection (FS). FS algorithms select a subset of the most representative features from the original feature space.

1.1.4 Feature Weighting

In this step, the document is transformed from the full text version to a document vector. The Extracted text documents are converted into Boolean weighting by using the indexing technique of Term Frequency – Inverse Document Frequency. TF–IDF[21] is the product of two statistics, term frequency and inverse document frequency. The term frequency tf (*t*, *d*), the simplest choice is to use the raw frequency of a term in a document, i.e. the number of times that term *t* occurs in document *d*. The raw frequency of *t* by f (*t*, *d*), then the simple tf scheme is tf (*t*, *d*) = f (*t*, *d*). The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

Id (t, D) = log$|D||\ d\epsilon D\ :\ t\epsilon d\ |$

|D|: cardinality of D, or the total number of documents in the corpus.

Then TF–IDF is calculated as

$$\text{tf - idf}\ (t, d, D) = \text{tf}\ (t, d) \times \text{idf}\ (t, D)$$

1.1.5 Classification Algorithms

Documents can be classified by three ways: Unsupervised (unlabelled), Supervised (labelled) and Semi supervised. Automatic text classification have been extensively studied and rapid progress is seen in this area.

Some classification approaches are Bayesian classifier, Decision Tree, K-nearest neighbor(KNN)[19] , Support Vector Machines(SVMs)[22] and Neural Network.

1.1.6 Performance Measure

This is the last stage of text classification. It evaluates the effectiveness of a classifier, in other words, its capability of taking the right categorization decisions. Classification performance is measured using both recall and precision. In this case, recall is the proportion of the correct documents that are assigned to a category by the algorithm. Precision is the proportion of documents assigned to a category that belong to that category.

II.PROPOSED SYSTEM

Measuring the similarity between documents is an important operation in the text classification approach. Usually, the dimensionality of a document is large and the resulting vector is sparse, i.e., most of the feature values in the vector are zero. Such high dimensionality and sparsity can be a severe challenge for similarity measure which is an important operation in text processing algorithms. To overcome this problem in this work a similarity measure SMTP(Similarity Measure for Text Processing) is proposed to compute the similarity between documents. This similarity measure is applied in knn based text classification on several real world datasets.

III.System Architecture

The architecture of the proposed system is shown in Figure 1.

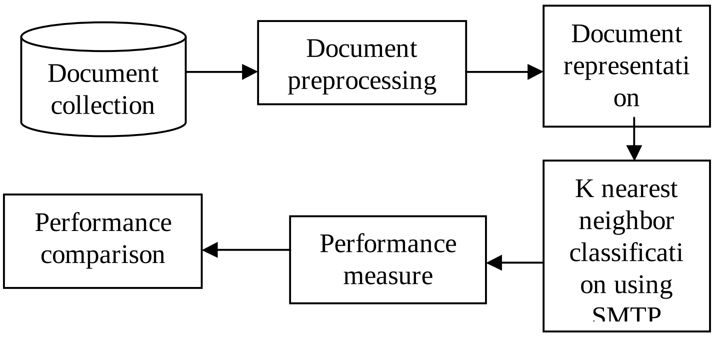


Fig 1.System Architecture

IV.WORKFLOW

The work flow of the system indicates step by step process of implemented work.

- **Step1:** Initially the dataset can be loaded.
- **Step2:** Then, preprocess the dataset by removing the stop words and by applying stemming. Porter stemming algorithm is used for stemming the word.
- **Step3:** Represent the preprocessed documents in vector format by computing term frequencies or tfidf values.
- **Step4:** Now apply the k nearest neighbor classification algorithm using different similarity measures and classify the documents.
- **Step5:** Compute the classification accuracy for each similarity measure.
- **Step6:** Finally compare the results of proposed measure with existing measures.

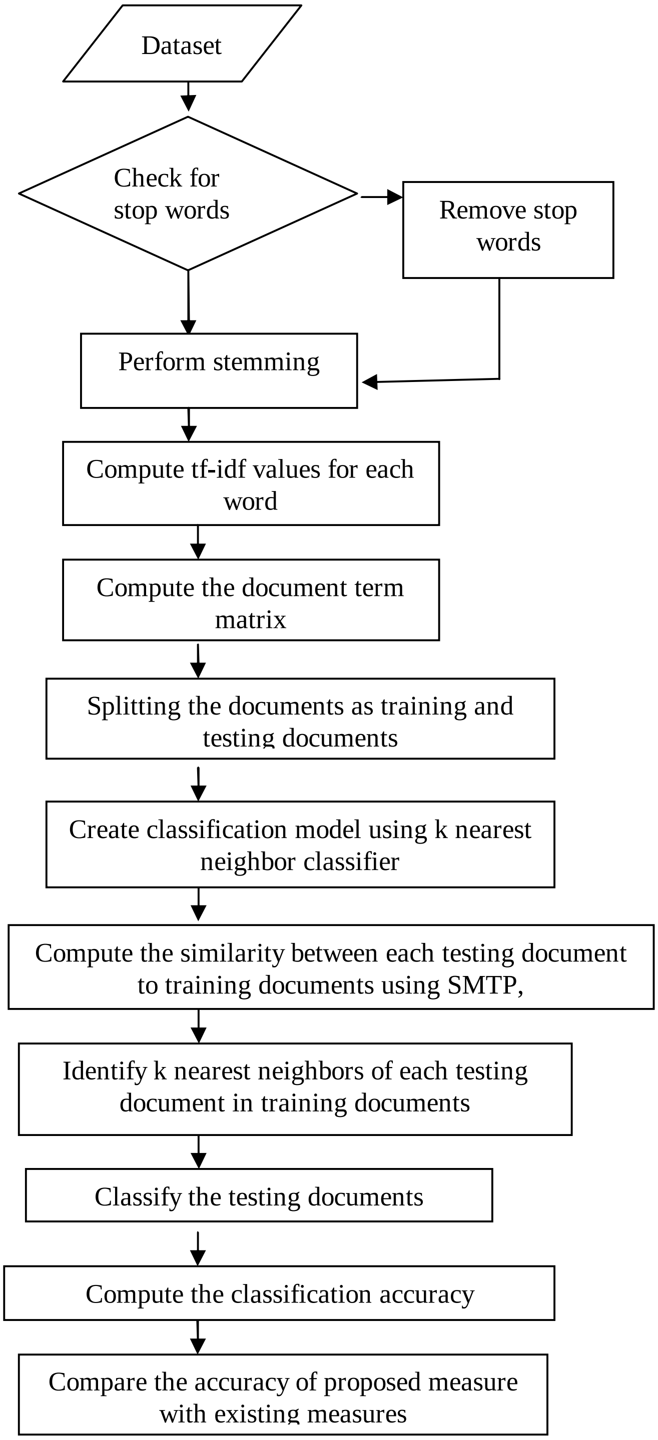

Dataset
Check for stop words
Remove stop words
Perform stemming
Compute tf-idf values for each word
Compute the document term matrix
Splitting the documents as training and testing documents
Create classification model using k nearest neighbor classifier
Compute the similarity between each testing document to training documents using SMTP,
Identify k nearest neighbors of each testing document in training documents
Classify the testing documents
Compute the classification accuracy
Compare the accuracy of proposed measure with existing measures

Fig 2: Workflow of the proposed system

V.ALGORITHMS

Algorithm for stop words removal:

Input: D(d1,d2,….dn) // Given dataset

Output: Doc(d1,d2…dn) // Documents without stopwords

Steps:

1. Remove punctuation
1.1 Take out!@#$%^&*():"{}[]';<>+_= etc.
2. Detect words in document
3. Remove the, to, am, you,…(stop words list)
4. Return document set without stop words

Algorithm for word stemming:

After removing the stop words from the documents stemming is performed. For stemming the words porter stemming algorithm is used.

Input: D(d1,d2,…dn) // Given dataset

Output: Doc(d1,d2…dn) // Document vectors with stemmed words

Steps:

1. Gets rid of plurals and -ed or -ing suffixes
2. Turns terminal y to i when there is another vowel in the stem
3. Maps double suffixes to single ones: -ization, -ational, etc.
4. Deals with suffixes, -full, -ness etc.
5. Takes off -ant, -ence, etc.
6. Removes a final -e
7. Return document vectors with stemmed words.

Algorithm for TFIDF weighting

Input: D(d1,d2..dn) // Document vectors with features

Output: D(d1,d2…dn) // Document vectors with feature weights

Steps:

1. Calculate the term frequency TF(t)
2. Calculate the document frequency DF(t)
3. Calculate the inverse document frequency IDF(t)=log[|D|/DF(t)]
4. Finally compute the TFIDF of a feature t TFIDF(t)=TF*IDF
5. Return document vectors with feature weights.

Algorithm: KNN Classifier

Input: D(d1,d2..dn) // Document vectors with class labels

K // k nearest neighbors

Output: C(c1,c2..cn) // Predicted class labels

Steps:

1. Load the dataset and class labels.
2. Randomly split the dataset into training(train_data) and testing documents(test_data).
3. [x,y]=size of train_data
4. [w,y]=size of test_data
5. for each document in training data and for each document in testing data
6. Calculate similarity matrix dist_matrix[i,j] for each testing instance to training instance for different metrics.
7. end first for
8. end second for
9. for each document in testing data
10. identify the k nearest neighbors of test_data in train_data
11. assign class labels to test_data using voting rule
12. end for

Algorithm for similarity measure SMTP

The similarity degrees between testing and training documents are computed using proposed similarity measure SMTP.

Input: train_data(d1,d2….dm) //training document vectors with n features

test_data(d1,d2…dk) // testing document vectors p features

Output: sim_matrix // Similarity matrix

Steps:

1. [m,n]= size of train_data
2. [k,p]= size of test_data
3. Take Lamda=1 and sigma(1:n) =2
4. Take N*D=0 and NuD=0
5. for each document in test data and for each document in training data
6. if test_data[r,j] . train_data[i,j]>0
7. then n1=0.5*(1+exp(-((train_data[r,j]-test_data[i,j])/sigma[j])^2))
8. if test_data[I,j] and train_data[r,j]=0
9. then n1=0
10. otherwise n1=-lamda
11. end if
12. N*D[i,r]=N*D[i,r]+n1[r,j]
13. end for
14. for each feature in train data
15. if train_data[r,j] and test[i,j]=0
16. then n2=0
17. otherwise n2=1
18. end if
19. NuD[i,r]=NuD[i,r]+n2[r,j]
20. end loop
21. F[i,r]=N*D[i,r]/NuD[i,r]
22. Sim_matrix[i,r]=(F[i,r]+lamda)/(1+lamda)
23. Return sim_matrix

VI. METRICS

Classification accuracy

The performance of proposed system can be evaluated by measuring the classification accuracy AC, which compares the predicted label of each document with that provided by the document corpus.

$$AC = \frac{\sum_{i=1}^{n} E\left(c_i, c_i'\right)}{n},$$

Where

n - the number of testing documents,
ci - target label
ci'- predicted label
E(ci, ci') = 1 if ci = ci', and E(ci, ci') = 0 otherwise.

Methods to be compared:

The proposed measure is compared with four other existing similarity measures. They are as follows:

1. Euclidean distance
2. Cosine similarity
3. Extended jaccard coefficient
4. Pairwise adaptive similarity

5.3.1 Euclidean distance

Euclidean distance is the well-known distance metric from Euclidean geometry. It is defined as the root of square differences between the respective coordinates of d1 and d2, i.e.

$$d_{Euc}(\mathbf{d}_1, \mathbf{d}_2) = [(\mathbf{d}_1 - \mathbf{d}_2)\cdot(\mathbf{d}_1 - \mathbf{d}_2)]^{1/2}$$

5.3.2 Cosine similarity

Cosine similarity measures cosine of the angle between two documents d1 and d2 as follows.

$$S_{Cos}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1\cdot\mathbf{d}_2}{(\mathbf{d}_1\cdot\mathbf{d}_1)^{1/2}(\mathbf{d}_2\cdot\mathbf{d}_2)^{1/2}}.$$

5.3.3 Extended jaccard coefficient

The jaccard coefficient measures the ratio of the number of commonly active features of d1 and d2 to the number of active in d1 or d2. This measure is often used in retail market based applications. For binary features, the jaccard coefficient measures the ratio of the intersection of the product sets to the union of the product sets. The extended jaccard coefficient is the extended version of the jaccard coefficient for data processing:

$$S_{EJ}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1\cdot\mathbf{d}_2}{\mathbf{d}_1\cdot\mathbf{d}_1 + \mathbf{d}_2\cdot\mathbf{d}_2 - \mathbf{d}_1\cdot\mathbf{d}_2}$$

5.3.4 Pairwise adaptive similarity

In order to decrease the degree of noise in the feature set, pairwise adaptive similarity measure is proposed based on a combination of the original cosine similarity measure and an observation specific feature selection. The measure is calculated similarly to the original similarity measure, but for each pair of observations the inner product is obtained within a customized subspace. Pairwise-adaptive similarity dynamically selects a number of features out of d1 and d2 and is defined to be

$$d_{Pair}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_{1,K} \cdot \mathbf{d}_{2,K}}{(\mathbf{d}_{1,K} \cdot \mathbf{d}_{1,K})^{1/2} (\mathbf{d}_{2,K} \cdot \mathbf{d}_{2,K})^{1/2}}$$

VII.Datasets

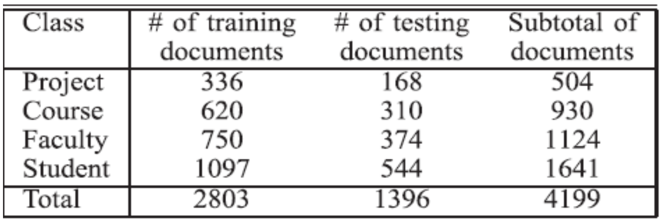
For the purpose of evaluating the performance and effectiveness of proposed system the following datasets are used.

- WebKB
- 20 newsgroups

WebKB

The documents in the WebKB data set are webpages collected by the World Wide Knowledge Base (Web→Kb) project of the CMU text learning group. The documents were manually classified into several different classes. The documents of this data set were not pre designated as training or testing patterns. The documents are randomly devided into training and testing subsets. Among the 4199 documents, 2803 are randomly selected for training and the rest, 1396, are for testing. The number of features involved is 7786.

Table 1: Distribution documents per class in webKB



Class	# of training documents	# of testing documents	Subtotal of documents
Project	336	168	504
Course	620	310	930
Faculty	750	374	1124
Student	1097	544	1641
Total	2803	1396	4199

20 newsgroups

This data set consists of 2000 messages taken from 20 newsgroups. One hundred Usenet articles were taken from each of the following 20 newsgroups.

alt.atheism, comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.guns, talk.politics.mideast, talk.politics.misc, talk.religion.misc.

Approximately 4% of the articles are crossposted. The articles are typical postings and thus have headers including subject lines, signature files, and quoted portions of other articles. Each newsgroup is stored in a subdirectory, with each article stored as a separate file.

VIII.Results (Matlab)

Results on webKB dataset

Number of documents =100
Number of features = 1343
Number of classes = 4

The documents are randomly divided into training and testing documents by considering 70% as training and rest as testing. The class labels are 1. Course 2.Faculty 3.Project 4.**Student**. The class labels for test documents are identified for different k values.

Iteration 1: Randomly divided documents

Iteration 1

```
Test labels
 1    4    1    4    4    1    1    4    4    3    1
4    1    1    4    4    1    1    2    3    3    2
1    4    3    3    3    1    1
Enter k nearest neighbors K=1
```

BY USING SMTP

```
Predicted labels
 Columns 1 through 17
 2   4   1   4   4   1   2   2   4   2   2   4   4
2   4   4   4
Columns 18 through 30
 2   1   2   2   2   3   1   2   3   2   3   1   1
Classification accuracy=0.5667
```

BY USING EUCLIDEAN DISTANCE

```
Predicted labels
Columns 1 through 17
2   4   4   4   4   2   2   2   4   2   2   4   4
2   4   2   4
Columns 18 through 30
2   2   2   4   4   4   1   2   3   2   3   2   2
Classification accuracy=0.1667
```

BY USING COSINE SIMILARITY

```
Predicted labels
Columns 1 through 17
2   3   1   1   1   1   4   1   3   3   1   1   1
1   1   1   1
Columns 18 through 30
 1   1   3   1   2   1   4   1   3   2   2   2   4
Classification accuracy=0.3000
```

BY USING EXTENDED JACCARD COEFFCIENT

```
Predicted labels
Columns 1 through 17
1   3   1   4   4   4   3   3   4   2   4   4   3
2   4   4   4
Columns 18 through 30
4   4   1   4   2   4   2   4   2   4   4   4   2
Classification accuracy=0.3000
```

BY USING PAIR WISE-ADAPTIVE SIMILARITY

Predicted labels
Columns 1 through 17
4 1 4 1 4 1 2 3 1 2 1 4 4
2 1 4 1
Columns 18 through 30
1 3 1 1 2 4 4 1 4 4 3 3 2
Classification accuracy=0.2333

Table 2: Accuracy for webKB dataset Iteration1

	SMTP	Euclidean distance	Cosine similarity	EJ coefficient	Pairwise adaptive similarity
K=1	0.5667	0.1667	0.3000	0.3000	0.2333
K=3	0.5333	0.3667	0.3000	0.3333	0.3000
K=5	0.5667	0.3333	0.1667	0.3333	0.2333
K=7	0.5667	0.3667	0.2667	0.2667	0.1667
K=9	0.6000	0.4000	0.3667	0.3667	0.2000

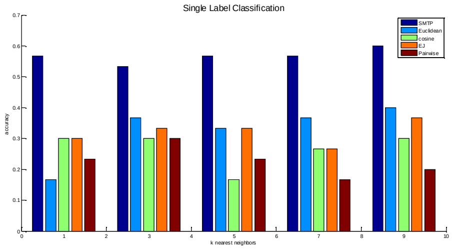
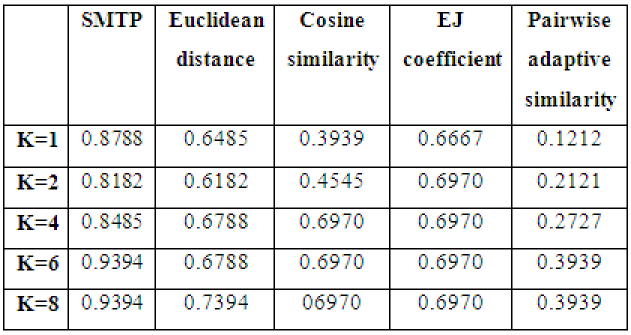


Fig: 3Accuracy comparison graph for webKB dataset for Iteration1

Iteration 2:

Randomly divided documents

Table 3: Accuracy for webKB dataset Iteration 2



	SMTP	Euclidean distance	Cosine similarity	EJ coefficient	Pairwise adaptive similarity
K=1	0.8788	0.6485	0.3939	0.6667	0.1212
K=2	0.8182	0.6182	0.4545	0.6970	0.2121
K=4	0.8485	0.6788	0.6970	0.6970	0.2727
K=6	0.9394	0.6788	0.6970	0.6970	0.3939
K=8	0.9394	0.7394	06970	0.6970	0.3939

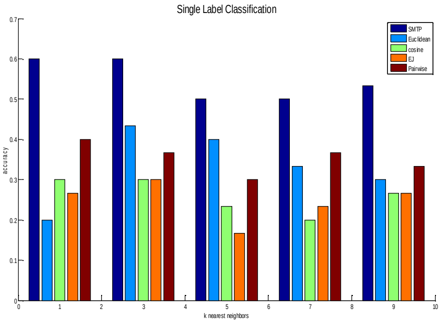


Fig 4: Accuracy comparison graph for webKB dataset for Iteration 2

Results on 20 newsgroups dataset

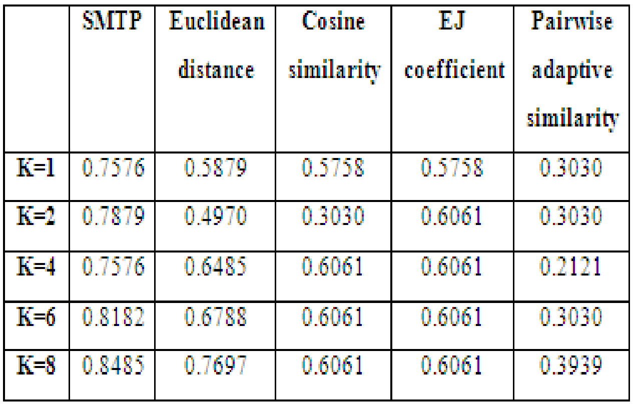
Number of documents =110
Number of features = 1056
Number of classes = 3

The documents are randomly divided into training and testing documents by considering 70% as training and rest as testing. The class labels are 1.atheism 2.comp.graphics 3.comp.os.ms-windows.misc. The class labels for test documents are identified for different k values.

Iteration 1

The accuracy of knn classifier using SMTP, Euclidean distance, cosine similarity, EJ coefficient and pairwise adaptive similarity for different k values is shown in the below table:

Table 4: Accuracy comparison graph for 20 newsgroups dataset for Iteration 1



	SMTP	Euclidean distance	Cosine similarity	EJ coefficient	Pairwise adaptive similarity
K=1	0.7576	0.5879	0.5758	0.5758	0.3030
K=2	0.7879	0.4970	0.3030	0.6061	0.3030
K=4	0.7576	0.6485	0.6061	0.6061	0.2121
K=6	0.8182	0.6788	0.6061	0.6061	0.3030
K=8	0.8485	0.7697	0.6061	0.6061	0.3939

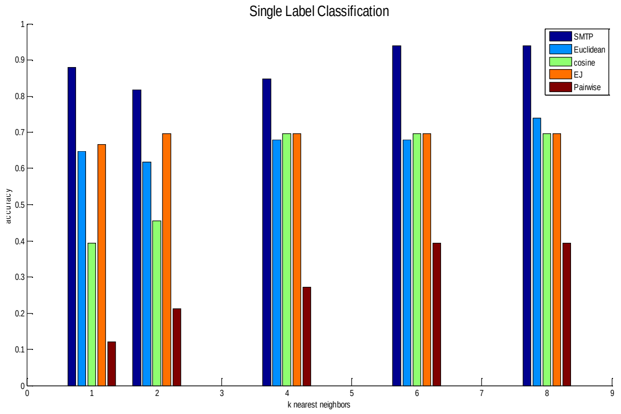


Fig 5: Accuracy comparison graph for 20 newsgroups dataset for Iteration 1

Iteration 2

The accuracy of knn classifier using SMTP, Euclidean distance, cosine similarity, EJ coefficient and pairwise adaptive similarity for different k values is shown in the below table:

Table 5: Accuracy for 20 newsgroups dataset Iteration2

	SMTP	Euclidean distance	Cosine similarity	EJ coefficient	Pairwise adaptive similarity
K=1	0.6000	0.2000	0.3000	0.2667	0.4000
K=3	0.6000	0.4333	0.3000	0.3000	0.3667
K=5	0.5000	0.4000	0.2333	0.1667	0.3000
K=7	0.5000	0.3333	0.2000	0.2333	0.3667
K=9	0.5333	0.3000	0.26667	0.2667	0.3333

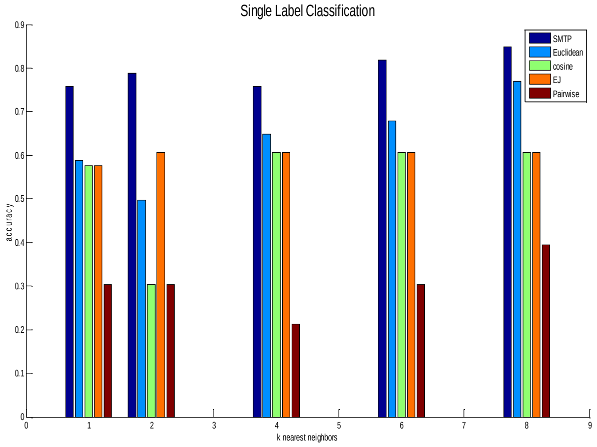


Fig 6: Accuracy comparison graph for 20 newsgroups dataset for Iteration 2

Analysis of results obtained

In this work the proposed similarity measure is compared with four other existing similarity measures. They are Euclidean distance, cosine similarity, extended jaccard coefficient and pairwise adaptive similarity. The results have shown that the accuracy of proposed measure is better than that of other existing similarity measures. The accuracy(%) of knn classifier on webkb dataset and 20 news groups dataset are shown in the following table. The proposed similarity measure achieved 93.9% accuracy on 20 newsgroup dataset while other existing measures achieved low(%) of accuracy. On webkb dataset the proposed measure have shown that 60-70% of accuracy but other existing measures have shown that 30-40% only. The results show that the performance obtained by the proposed measure is better than that achieved by other measures.

Table 6: Analysis of results on different datasets

Dataset	Number of documents	Number of features	Accuracy of knn classifier(%)				
			SMTP	Euclidean distance	Cosine similarity	EJ coefficient	Pairwise adaptive similarity
webkb	100	1343	60	40	38.6	36.6	35
20 news groups	110	1056	93.9	78.9	69.9	68	40

IX.CONCLUSION

A similarity measure is presented to compute the similarity between two documents. Several desirable properties are embedded in this measure. For example, the similarity measure is symmetric. The presence or absence of a feature is considered more essential than the difference between the values associated with a present feature. The similarity degree increases when the numbers of presence-absence feature pair decreases. Two documents are least similar to each other if none of the features have non-zero values in both documents. The effectiveness of proposed measure is investigated by applying it in *k*-NN based single-label classification on several real world datasets. The results have shown that the performance obtained by the proposed measure is better than that achieved by other measures.

References

[1] A. Strehl and J. Ghosh, “Value-based customer grouping from large retail data-sets,” in Proc. SPIE, vol. 4057. Orlando, FL, USA, Apr. 2000, pp. 33–42.

[2] C. G. González, W. Bonventi, Jr., and A. L. V. Rodrigues, “Density of closed balls in real-valued and autometrized boolean spaces for clustering applications,” in Proc. 19th Brazilian Symp. Artif. Intell., Savador, Brazil, 2008, pp. 8–22.

[3] F. Sebastiani, “Machine learning in automated text categorization,”ACM CSUR, vol. 34, no. 1, pp. 1–47, 2002.

[4] G. Yu, Y.J. Pei, Z.Y. Zhu and H.Y. Chen, Research of text similarity based on word similarity computing. Computer Engineering and Design, 27 (2), 2006.

[5] H. Chim and X. Deng, “Efficient phrase-based document similarity for clustering,” IEEE Trans. Knowl. Data Eng., vol. 20, no. 9, pp. 1217–1229, Sept. 2008.

[6] H. Kim, P. Howland, and H. Park, “Dimension reduction in text classification with support vector machines,” J. Mach. Learn. Res., vol. 6, pp. 37–53, Jan. 2005.

[7] I. S. Dhillon and D. S. Modha, “Concept decompositions for large sparse text data using clustering,” Mach. Learn., vol. 42, no. 1, pp. 143–175, 2001.

[8] I. S. Dhillon, S. Mallela, and R. Kumar, “A divisive information theoretic feature clustering algorithm for text classification,” J. Mach. Learn. Res., vol. 3, pp. 1265–1287, Mar. 2003.

[9] J. A. Aslam and M. Frost, “An information-theoretic measure for document similarity,” in Proc. 26th SIGIR, Toronto, ON, Canada, 2003, pp. 449–450.

[10] J. D’hondt, J. Vertommen, P.-A. Verhaegen, D. Cattrysse, and J. R. Duflou, “Pairwise-adaptive dissimilarity measure for document clustering,” Inf. Sci., vol. 180, no. 12, pp. 2341–2358,2010.

[11] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann; Boston, MA, USA: Elsevier, 2006.

[12] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell, “Learning to classify text from labeled and unlabeled documents,” in Proc. 15th Nat. Conf. Artif. Intell., Menlo Park, CA, USA,1998.

[13] Lam, W. and Ho, C.Y. (1998). Using a Generalized Instance Set for Automatic Text Categorization. SIGIR’98, pages 81-89.

[14] LOVINS, J.B. Development of a Stemming Algorithm. Mechanical Translation and computation Linguistics. 11 (1) March 1968 pp 23-31.

[15] L. Wang, X. Zhao, “Improved knn Classification Algortihm Research in Text Categorization”, In the Proceedings of the 2nd International Conference on Communications and Networks (CECNet), (2012), pp. 1848-1852.

[16] Martin F. Porter, An algorithm for suffix stripping, Program 14 (1980) 130–137.

[17] M. Craven et al., “Learning to extract symbolic knowledge form the world wide web,” in Proc. 15th Nat. Conf. Artif. Intell., Menlo Park, CA, USA, 1998.

[18] M. G. Michie, “Use of the bray-curtis similarity measure in cluster analysis of foraminiferal data,” Math. Geol., vol. 14, no. 6, pp. 661–667, 1982.

[19] Oznur Kirmemis and Gulen Toker “Text Categorization Using k-Nearest Neighbour Classification,” Survey Paper, , Middle East Technical University Computer Engineering Department.

[20] R. O. Duda, P. E. Hart, and D. J. Stork, Pattern Recognition. New York, NY, USA: Wiley, 2001.

[21] T. Joachims, a Probalilistic Analysis of the Tocchio Algorithm with TF-IDF for Text Categorization. Prof. of the 14th International Conference on Machine Learning, ICML97,1997.

[22] T. Joachims, (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features, In Proceedings of 10th European Conference on Machine Learning, Chemnitz, Germany, pages 137-142.

[23] T. Zhang, Y. Y. Tang, B. Fang, and Y. Xiang, “Document clustering in correlation similarity measure space,” IEEE Trans. Knowl. Data Eng., vol. 24, no. 6, pp. 1002–1013, Jun. 2012.

[24] T. W. Schoenharl and G. Madey, “Evaluation of measurement techniques for the validation of agent-based simulations against streaming data,” in Proc. ICCS, Kraków, Poland, 2008.

[25] X. Luo, D.L. Xia, P. Yan, Improved feature selection method and TF-IDF formula based on word frequency differentia. Computer Applications, 25(9), 2005.

[26] Y. Zhao and G. Karypis, “Comparison of agglomerative and partitional document clustering algorithms,” in Proc. Workshop Clustering High Dimensional Data Its Appl. 2nd SIAM ICDM, 2002, pp. 83–93.

[27] Yang .C and Jun Wen, “Text Categorization using cosine function”, School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 610054, P.R. China

[28] The datasets can be downloaded at http://web.ist.utl.pt/~acardoso/datasets and http://www.cs.technion.ac.il/~ronb/thesis.html

AUTHOR

M. Radha is pursuing her Ph.D. in Acharya Nagarjuna University, Andhra Pradesh under the domain of Text Mining. She is working as an Asst.Prof in CSE Department,QIS Institute of technology,ongole. She has a total of 9 Years Experience in Teaching. She has two International Journal Publications. She has participated in two National level conferences and attended many workshops. She attended for Two AICTE sponsored 2-week workshops on data mining. She is member in Institution of Engineers (India) IEI.She is also the member of ISTE.

M .Nagaraju received his M.Tech. Degree from Gokul Institute of Technology and Sciences ,Vijayanagaram in 2011. Presently, he is working as an Asst.Prof in the Dept.of CSE of QIS Institute of Technology, Ongole.He is having 12 years of teaching experience. His current research interest includes Data Mining,Artificial Intelligence.

Chilukuri Divya received her M.Tech. Degree from R.V.R&J.C College of Engineering, Guntur in 2015. Presently, She is working as an Asst.Prof in the Dept.of CSE of QIS Institute of Technology, Ongole.She is having 1.5 years of teaching experience. Her current research interest includes Data Mining.