International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

A Document Classification Framework for Efficient Retrieval

Aijazahamed Qazi, R.H. Goudar

Abstract*: Document classification has become an evolving field of exploration with the significant rise in the volume of computerized information. Weighting of a term is an elementary research issue in document classification. Several alternatives to the traditional techniques to weight a term like TF_IDF have been proposed by the researchers. This paper introduces a novel method to weight a term by calculating the semantic similarity between the category label and the term. Also the proposed term weighting technique includes the co-occurrence relation between the terms. Experiments were carried on the 20 Newsgroups and Reuters_21578 benchmark datasets. The results obtained infer that the proposed method outperforms the other weighting methods using various classifiers.

***Index Terms*: TF_IDF, Fuzzy kNN, Accuracy.**

I. INTRODUCTION

The growth of e-documents has made information retrieval more thought-provoking task to discover significant data. As metadata is available in multimedia document, classification of documents is becoming more challenging as presented by Qazi and Goudar [1]. The vector space model represents a document as a feature vector. The documents are assigned to categories using machine learning procedures. The effectiveness of an information retrieval model depends on the technique that is applied to weight a term in the corpus. Term Frequency and Inverse Document Frequency (TF_IDF) is broadly used statistical weighting method in the field of information retrieval. TF_IDF does not consider semantic information related to the term. Recent study by Mironczuk and Protasiewicz [2] provided a holistic view of the information retrieval techniques to obtain better classification accuracy. Evaluation of a user's query to retrieve information with imprecise knowledge is one of the search engine research issues. The following are the key contributions of this paper: Initially, a novel semantic term weighting approach (semTF_IDF) is proposed. Secondly, the proposed approach captures the semantic similarity between the categories and the terms. Further, comparative analysis with other weighting techniques is carried by conducting experiments on 20 Newsgroups and Reuters_21578 benchmark corpus. The paper is structured as follows: part 2 of the paper reviews the related work. Part 3 provides an overview of the preliminaries required. Part 4 elaborates the proposed term weighting approach. The experimental analysis is discussed in part 5. Part 6 draws the conclusion of the paper.

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
Aijazahamed Qazi*, Department of CSE, SDMCET, Dharwad, India.
R.H.Goudar, Department of CNE, Center for PG Studies Visvesvaraya Technological University, Belgaum, India.

II. RELATED WORK

Information Retrieval is fragment of Artificial Intelligence. It deals with efficient retrieval of information across the systems to a user's query. In the context of document classification, weighting of a term is carried to specify the importance of a word in the document. Supervised and unsupervised approaches are largely considered to compute the weight of a term in the document corpus. The objective of any information retrieval model is to increase the correctness of classification. Sebastiani [3] discussed several approaches for text categorization using Machine Learning. The issues pertaining to representation of documents and performance of various classifiers have also been discussed. Debole and Sebastiani [4] presented a supervised term weighting methodology by substituting inverse document frequency with category based term assessment function. Peng, Liu and Zuo [5] introduced an improved TF_IDF weighting approach by applying a voting classifier for better results on the dataset. Jiang et al. [6] proposed a correlation based weight assignment technique for features using sigmoid function. Feng et al. [7] proposed a technique to weight a term using a latent variable, conjugate prior and a scoring function. Al-Anzi and AbuZeina [8] examined the results on Arabic text classification using Latent Semantic Indexing. Figueiredo et al. [9] introduced an approach to create composite features based on the co-occurrence of terms in documents. Elhadad et al. [10] introduced a feature generation method based on the ontological structure of WordNet. The non-semantic words with no relation to any of the WordNet groups were removed and the feature vectors were concatenated. Experiments were carried with classifiers like Naïve-Bayes, SVM, kNN and better results were obtained. Qazi and Goudar [11] proposed a term weight computing method based on ontology to select the features of a specific domain. Luo et al. [12] presented a semantic procedure to weight a term based on the WordNet. Trstenjak et al. [13] introduced a framework for cataloging text documents using kNN classifier. An experimental evaluation was carried out on 20 Newsgroups and Reuters_21578 benchmark collection and better results were obtained. Tar and Khaing [14] introduced a technique of concept weighting for text clustering using the standards of ontology. Wang et al. [15] presented a weighting technique called Inverse Category Frequency (ICF) based on the terms existing in few categories, rather than few documents. This methodology combined relevance frequency and ICF for weighting the terms with a supervised approach.

III. PRELIMINARIES

A. *Term Frequency-Inverse Document Frequency*

Boolean, Term Frequency and TF_IDF are the traditional approaches applied to weight a term for retrieval of information. TF_IDF is a statistical quantity used to estimate the influence of a word in the collection as proposed by Jones [16]. The occurrence of a word in a document is represented by the term frequency. Inverse Document Frequency measures the term's importance in collection of documents. Variants of TF_IDF technique are applied by the search engines to assess the relevance of a user's query to the document.

$$w(t_k) = tf_k . log\left(\frac{N}{df_k}\right) \qquad (1)$$

where, the count of the term is represented by t_k in the document, df_k represents the total document count having term t_k and N is the total document count in the corpus. TF_IDF does not include the information pertaining to the category of the terms to be weighted.

B. *Lin's Semantic Similarity*

The interpretation of textual information involves the evaluation of semantic likeness amongst the terms as presented by Lin et al. [17]. Ontology provides description of concepts and the relations amongst concepts. The concept of Information Content refers to the probability of occurrence of each concept in the corpus.

$$IC(a) = -log\ P(a) \qquad (2)$$

Semantic similarity can be determined by the Least Common Subsumer (LCS) shared amongst the two terms in the ontology.

$$sim_{lin}(a,b) = \frac{2xIC\big(LCS(a,b)\big)}{\big(IC(a) + IC(b)\big)} \qquad (3)$$

C. *kNN Algorithm*

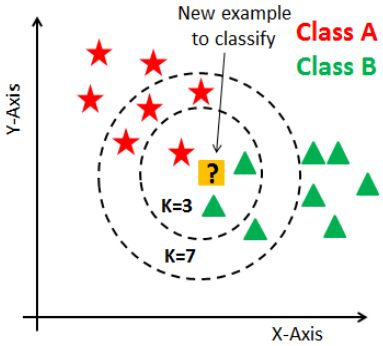

Fig. 1. kNN classifier

The k-Nearest Neighbor (kNN) is a lazy learner algorithm. The kNN is a machine learning algorithm without any input parameter. kNN classifier has been largely applied in the field of pattern recognition as discussed by Aggarwal and Zhai [18]. kNN classifier is based on the learning by correlating a test and training data. The training data is represented in n-dimensional space as a feature vector. For a test feature vector to be classified, kNN algorithm examines the k trained vectors neighboring to the test feature vector. Then, the test vector is given the class label of the k nearest neighbors. The Euclidean distance calculates the proximity between two points. The Euclidean distance amongst the two points, M = (m_1, m_2, … m_i) and N = (n_1, n_2, … n_i) is

$$dist(M,N) = \sqrt{\sum_{i=1}^{n}(m_i - n_i)^2} \qquad (4)$$

D. *Support Vector Machines*

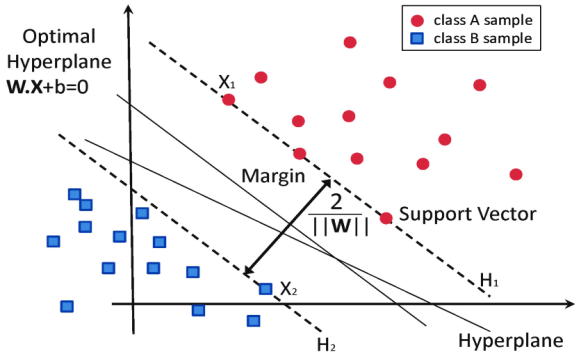

Fig. 2. SVM classifier

Support vector machine (SVM) is a machine learning algorithm presented by Vapnik. SVM works on the principle of minimization from theory of computational learning as discussed by Aggarwal and Zhai [18]. It introduces a hyperplane to separate training features called support vectors into a set of classes. The hyperplane separates the training features by the broadest possible margin. This property of SVM makes it popular amongst the other classifiers for the purpose of data classification. SVM is classified as linear SVM and non-linear SVM. The type of the kernel function decides the category of SVM. Some of the kernel functions are:

Linear Kernel: $K_l(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y}$ (5)

Gaussian Kernel: $K_g(\mathbf{x},\mathbf{y}) = \exp(\frac{\|\mathbf{x}-\mathbf{y}\|^2}{A})$ (6),

where, A is a constant.

Polynomial Kernel: $K_p(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T\mathbf{y}+1)^B$ (7),

where, B is a constant.

E. *Fuzzy kNN*

Fuzzy k Nearest Neighbor (FkNN) algorithm groups the vectors V = {v_1, v_2, ... v_n} into z [1 < z < n] fuzzy subsets. For i = 1,2, ...z and j = 1, 2,...n the matrix of fuzzy membership is given by A, where A_{ij} is the fuzzy membership degree of v_j in class i. The matrix A has two limitations as follows:

$$\sum_{i=1}^{z} A_{ij} = 1 \ \ A_{ij} \in [0,1] \qquad (8)$$

$$0 < \sum_{j=1}^{m} A_{ij} < n, A_{ij} \in [0,1] \qquad (9)$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Eqn. 8 confirms that all the entities across the classes have membership degrees and the summation of all the membership degrees equals to one. Eqn. 9 indicates the fuzzy membership degree for an entity lies in the interval of [0,1]. FkNN algorithm assigns fuzzy membership degree to each of the sample vector as,

$$Ai(V) = \sum_{j=1}^{k} \frac{\frac{A_{ij}}{\left\|v - v_j\right\|^{\left(\frac{2}{m-1}\right)}}}{\sum_{j=1}^{k} \frac{1}{\left\|v - v_j\right\|^{\left(\frac{2}{m-1}\right)}}} \qquad (10)$$

where k refers to the count of nearest neighbors and m is a constant parameter. Eqn. 10 computes the membership degree of an entity to the i_{th} class.

IV. PROPOSED TERM WEIGHTING APPROACH

The following is the algorithm for the proposed term weighting approach,

Algorithm 1: semTF_IDF to compute Document Term Weight Matrix	
Input: Document corpus D	
Output: W ← W_{ij}, Document Term Weight Matrix of size |D| x |T|	
Procedure:	
1.	for each $d_i \in D$ do
2.	Remove stopwords, lemmatize and construct set T of tokens in d_i
3.	for each term $t_i \in T$ do
4.	Compute term frequency tf_i
5.	end for
6.	end for
7.	Construct the co-occurrence matrix C_T of size |T| x |T| with ijth element $C_T(i,j)$ being the number of joint occurrences of ith and jth terms in a document of D
8.	for each ith row of C_T do
9.	Compute rowsum $\alpha_i \leftarrow \sum_{j=1}^{\|N\|} C_T(i,j) / \|T\|$
10.	end for
11.	Let CL={$l_1,l_2,\ldots,l_8$} be the set of labels of 8 categories of web documents in D
12.	for each label $l_k \in CL$ do
13.	Extract the first synonym S_{l_k} from the WordNet
14.	end for
15.	Construct the set of synonym labels, $S_{C_L} = \{S_{L_1}, S_{L_2}, \ldots, S_{L_8}\}$
16.	for each term ti ∈ do
17.	Compute βi ← $\max_{s_{l_k} \in S_{CL}} (SIM_{LIN}(t_i, S_{l_k}))$
18.	where, $SIM_{LIN}(x,y \leftarrow 2\log(LCS(x,y))/(\log(P(x)) + \log(P(y)))$
19.	with x,y ← terms, LCS(x,y) ← least common subsumer of terms
20.	end for
21.	Compute modified term frequency mtf_i for each $t_i \in$ T,
22.	$mtf_i \leftarrow tf_i + \alpha_i + \beta_i$
23.	for each $d_j \in D$ do
24.	for each $t_i \in T$ of dj do
25.	Compute, $W_{ij} \leftarrow mtf_i \times \log(\|D\|/df_i)$
26.	where |D|← total count of documents
27.	df_i ← document count containing t_i
28.	end for
29.	end for
30.	Normalize each row vector of W

Algorithm 1 discusses to the proposed method to calculate term weight in a document corpus. Initially preprocessing is carried and token set is constructed for all the documents. Term frequency is calculated for each of the term in the token set. Then co-occurrence relation between the terms is computed. Lin's similarity is calculated for the WordNet synonym of the category and the term. Modified term frequency is determined. Finally, semTF_IDF is calculated and normalized.

V. EXPERIMENTS

The experimentation section comprises of details about benchmark datasets, preprocessing, evaluation parameters and analysis of results obtained using various weighting approaches with different classifiers. Also, this section includes comparative analysis of the proposed term weighting technique against other techniques.

A. *Dataset*

Reuters_21578 and 20 Newsgroups benchmark datasets are used for the purpose of experimentation for document classification. The above-mentioned datasets vary in their features. The term weighting approaches are evaluated for performance analysis with the classifiers. In the process of experimentation, 8 classes of the Reuters-ModApte split are considered. Reuters-ModApte split is used in many research works of document classification. Reuters_21578 has varying count of documents in each of the class. Reuters_21578 dataset is fragmented into training set and test set for the purpose of experimentation. Table 1 provides the details of 8 classes of Reuters_21578 dataset.

Table 1. Reuters_21578 dataset

Sl No	Class Name	Training Set	Test Set
1.	ship	95	31
2.	interest	154	60
3.	trade	197	98
4.	crude	215	104
5.	grain	39	8
6.	money-fx	201	82
7.	acq	700	290
8.	earn	1224	562

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

20 Newsgroups dataset is widely chosen document collection in English. It comprises 19,997 documents representing 20 classes. The documents of selected classes are fragmented into train and test set for the purpose of experimentation. Table 2 provides the information of 20 Newsgroups dataset.

Table 2. 20 Newsgroups dataset

Sl No	Class Name	Training Set	Test Set
1.	alt.atheism	50	20
2.	comp.graphics	50	20
3.	comp.windows.x	50	20
4.	rec.autos	50	20
5.	rec.sport.hockey	50	20
6.	sci.electronics	50	20
7.	soc.religion.christian	50	20
8.	talk.politics.misc	50	20

B. *Experimental phases and techniques*

The preprocessing of the above datasets involves tokenization, conversion of tokens to lower case, elimination of stop words and stemming. Feature weighting of documents is carried out after the above preprocessing steps are completed. The term weighting approaches are tested with top {500, 1400, 3500, 5000} terms selected for the Reuters_21578 and 20 Newsgroups dataset. Four weighting approaches, i.e. TF_IDF [16], OnCW [14], ICF based [15] and the proposed term weighting approach, semTF_IDF are applied in the experiments with the classifiers. kNN, SVM, Decision Tree and Fuzzy kNN are the classifiers used in the experiments for the purpose of classification of documents. These classifiers are discussed in the preliminaries section of the paper. The performance of the term weighting techniques is assessed using Micro_F1 and Macro_F1 metrics. The above mentioned metrics are derived using two popular performance measures, i.e. precision and recall. Micro_F1=2*A*B/(A+B), where A represents the precision of the obtained classified results and B represents the recall for the set to be classified. Micro_F1=sum(F_{1j})/m, with j=1,2,…,m and F_{1j}=2*A_j*B_j/(A_j+B_j), where F1 represents the measure of the j^{th} class, A_j and B_j represent the precision and recall of j^{th} class respectively and m indicates the label count.

C. *Result and Analysis*

Performance analysis on Reuters_21578 dataset

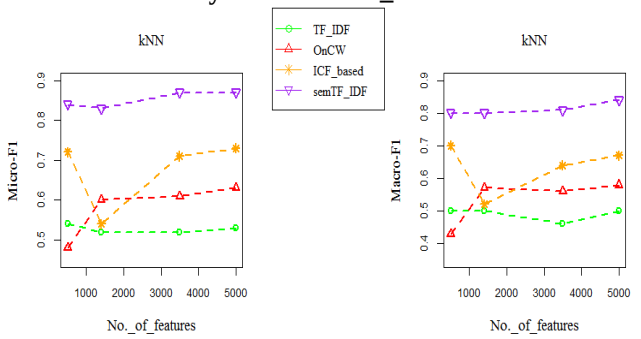

Fig.3.Micro_F1 and Macro_F1 values obtained for multiclass document classification with kNN (k=13) and four weighting techniques on the Reuters_21578 dataset with varying feature size.

For kNN classifier, the Micro_F1 and Macro_F1 values of semTF_IDF are high for most of the features as shown in Fig 3. TF_IDF demonstrates the minimum performance with lower Micro_F1 and Macro_F1 values. The Micro_F1 and Macro_F1 values of ICF-based decrease for an intermediate feature size and then gradually increase. The Micro_F1 and Macro_F1 values of OnCW gradually increase with rise in the quantity of features.

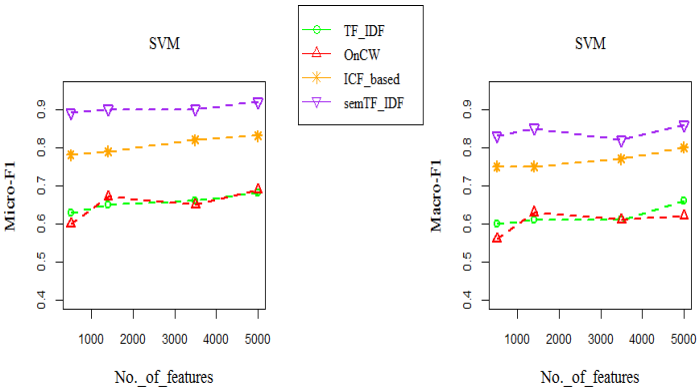

Fig.4.Micro_F1 and Macro_F1 values obtained for multiclass document classification with SVM and four weighting techniques on the Reuters_21578 dataset with varying feature size.

For SVM classifier, the Micro_F1 and Macro_F1 values of semTF_IDF are high for most of the features as shown in Fig 4. TF_IDF and OnCW demonstrate the minimum performance with lower Micro_F1 and Macro_F1 values. The Micro_F1 and Macro_F1 values of ICF-based vary with increase in the number of features. For Fuzzy kNN classifier, the Micro_F1 and Macro_F1 values can be represented in descending (semTF_IDF > ICF-based > OnCW > TF_IDF) as shown in below Fig 5.

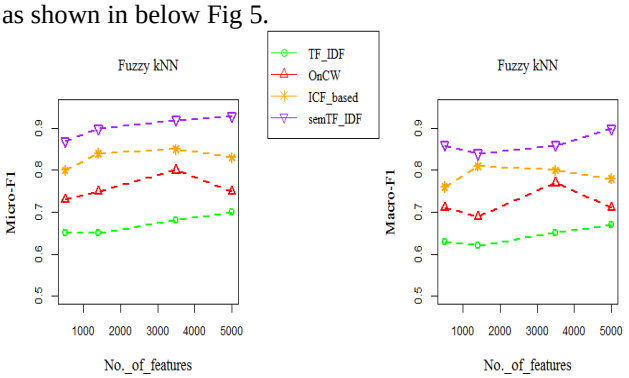

Fig.5.Micro_F1 and Macro_F1 values obtained for multiclass document classification with Fuzzy kNN and four weighting techniques on the Reuters_21578 dataset with varying feature size.

Performance analysis on the 20 Newsgroups dataset

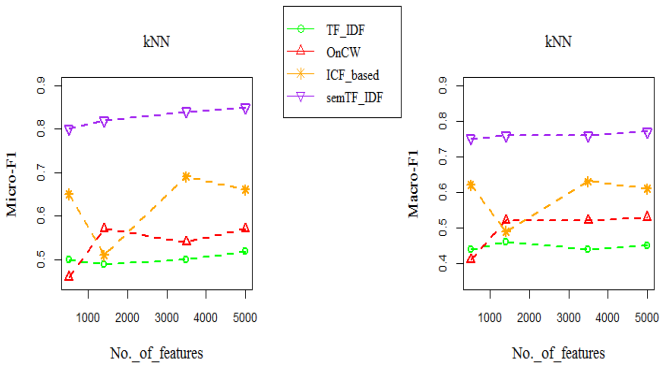

Fig.6.Micro_F1 and Macro_F1 values obtained for multiclass document classification with kNN (k=13) and four weighting techniques on the 20 Newsgroups dataset with varying feature size.

For kNN classifier, the classification accuracy curve (Micro_F1 and Macro_F1) of semTF_IDF increases with rise in the number of features as shown in Fig 6. TF_IDF demonstrates the minimum performance with lower Micro_F1 and Macro_F1 values. The Micro_F1 and Macro_F1 values of ICF-based decrease for an intermediate feature size and then gradually increase. The Micro_F1 and Macro_F1 values of OnCW gradually increase with surge in the number of features.

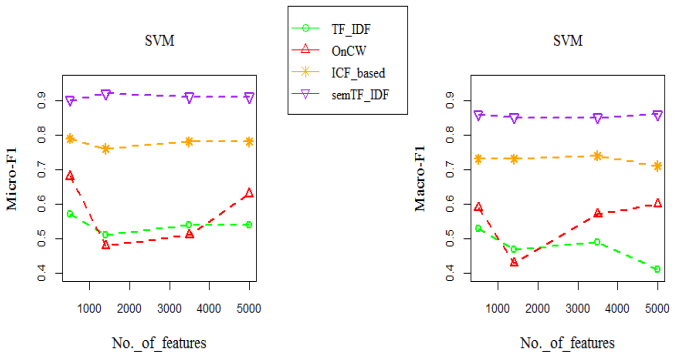

Fig.7.Micro_F1 and Macro_F1 values obtained for multiclass document classification with SVM and four weighting techniques on the 20 Newsgroups dataset with varying feature size.

For SVM classifier, the Micro_F1 and Macro_F1 values of semTF_IDF are high for most of the features as shown in Fig 7. There is a rise and fall in the accuracy curve of ICF-based with increase in the feature size. TF_IDF and OnCW indicate minimum performance.

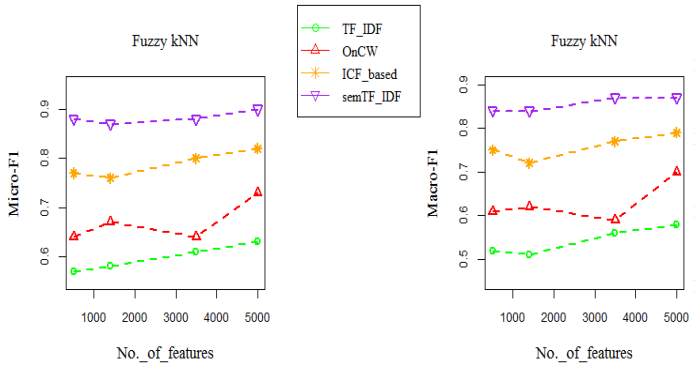

Fig.8.Micro_F1 and Macro_F1 values obtained for multiclass document classification with Fuzzy kNN and four weighting techniques on the 20 Newsgroups dataset with varying feature size.

For Fuzzy kNN classifier, the Micro_F1 and Macro_F1 values can be represented in descending (semTF_IDF > ICF-based>OnCW > TF_IDF) as shown in Fig 8.

The performance of proposed term weighting technique, semTF_IDF is significantly better than OnCW, TF_IDF and ICF-based techniques on both the datasets.

VI. CONCLUSION

The paper provides a novel methodology to weight terms for document classification. semTF_IDF considers the co-occurrence relation between the terms. Also, a semantic similarity is calculated between the category label and the term. This paper offers an experimental assessment and comparative analysis between term weighting methods and infers semTF_IDF yields better classification accuracy. As a part of future scope, we propose to investigate and apply the technique for multi-label document classification.

REFERENCES

1. A. Qazi and R. H. Goudar, "Emerging Trends in Reducing Semantic Gap towards Multimedia Access: A Comprehensive Survey," *Indian Journal of Science and Technology*, vol. 9, no. 30, Aug. 2016.
2. M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, vol. 106, pp. 36–54, Sep. 2018.
3. F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002.
4. F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Proceedings of the 2003 ACM symposium on Applied computing - SAC '03*, Melbourne, Florida, 2003, p. 784.
5. T. Peng, L. Liu, and W. Zuo, "PU text classification enhanced by term frequency-inverse document frequency-improved weighting" *Concurrency and Computation: Practice and Experience*, vol. 26, no. 3, pp. 728–741, Mar. 2014.
6. L. Jiang, L. Zhang, C. Li, and J. Wu, "A Correlation-Based Feature Weighting Filter for Naive Bayes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 201–213, Feb. 2019.
7. G. Feng, S. Li, T. Sun, and B. Zhang, "A probabilistic model derived term weighting scheme for text classification,"*Pattern Recognition Letters*, vol. 110, pp. 23–29, Jul. 2018.
8. F. S. Al-Anzi and D. AbuZeina, "Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 2, pp. 189–195, Apr. 2017.
9. F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira Jr., "Word co-occurrence features for text classification," Information Systems, vol. 36, no. 5, pp. 843–858, Jul. 2011.
10. M. K. Elhadad, K. M. Badran, and G. I. Salama, "A Novel Approach for Ontology-Based Feature Vector Generation for Web Text Document Classification:," International Journal of Software Innovation, vol. 6, no. 1, pp. 1–10, Jan. 2018.
11. A. Qazi and R. H. Goudar, "An Ontology-based Term Weighting Technique for Web Document Categorization," *Procedia Computer Science*, vol. 133, pp. 75–81, 2018.
12. Q. Luo, E. Chen, and H. Xiong, "A semantic term weighting scheme for text categorization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12708–12716, Sep. 2011.
13. B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF_IDF based Framework for Text Categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.
14. H. H. Tar and M. M. Khaing, "Text Document Clustering with Ontology Applying Modify Concept Weighting," in *Genetic and Evolutionary Computing*, vol. 388.
15. Wang, Deqing and Hui Zhang. "Inverse Category Frequency based supervised term weighting scheme for text categorization." J. Inf. Sci. Eng. 29 (2013): 209-225.

16. K. Sparck Jones, “A Statistical Interpretation of Term Specificity and its Application in Retrieval,” *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, Jan. 1972.
17. Y.-S. Lin, J.-Y. Jiang, and S.-J. Lee, “A Similarity Measure for Text Classification and Clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1575–1590, Jul. 2014.
18. C. C. Aggarwal and C. Zhai, “A Survey of Text Classification Algorithms,” in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 163–222.
19. J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy K-nearest neighbor algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 4, pp. 580–585, Jul. 1985.

AUTHORS PROFILE

Aijazahamed Qazi, currently working as an Assistant Professor, Dept. of CSE, SDMCET, Dharwad. He has published papers in International Journals and Conferences. His areas of interest include Semantic Web and Information Retrieval.

Dr.R.H.Goudar, currently working as an Associate Professor, Dept. of CNE, Visvesvaraya Technological University, Belagavi. He has 14 years of teaching experience at Professional Institutes across India. He worked as a faculty at International Institute of Information Technology, Pune for 4 years and at Indian National Satellite Master Control Facility, Hassan, India. He has published over 130 papers in International Journals, Book Chapters and Conferences of high repute. His subjects of interest include Semantic Web, Network Security and Wireless Sensor Networks.