Journal of Integrated
SCIENCE & TECHNOLOGY

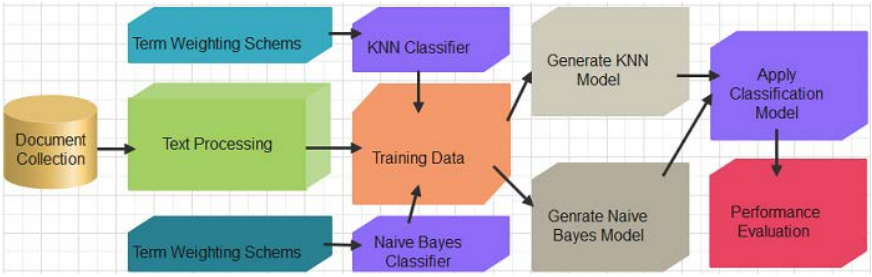
Analysis of Term Weighting schemes in Vector Space model for text classification

Shitanshu Jain,[1] Santosh Vishwakarma,[2] S.C.Jain[1*]

[1]*Amity University, Gwalior, Madhya Pradesh 474005, India.* [2]*Manipal University Jaipur, Rajasthan – 303007, India*

Received on: 12-Oct-2022, Accepted and Published on: 16-Nov-2022

ABSTRACT



The term weighting system (TWS) is an important component for the text matching system whenever the vector space model is employed for information retrieval from text sources. This work reports an innovative way of term-weighting approach to enhance the performance of classification. With respect to text categorization approaches, the term weighting system that we present in this study has the highest accuracy. We examined several weighting - schemes, weight information - gain, SVM, and the current technique's performance parameters with K-NN and Naïve- Bayes technique. A variety of term-weighting techniques (TWM) in conjunction with Information-Gain, SVM, K-NN, and Naïve-Bayes techniques have been used for analysis on Amazon data collections.

Keywords: Text Classification, Text Minig, K-NN, Naïve-Bayes, Term-Weightng, SVM

INTRODUCTION

Large datasets are regularly mined using data-mining techniques to obtain precise information.[1] Structured and unstructured data are dealt with through data mining and text mining (Figure 1). Any text-based document, email, etc. is an example of a semi-structured dataset. Finding unknown knowledge from various sources is the objective of text mining.[2]

It is possible to produce (convert) text documents into pre-defined classes using a variety of text-classification methods.[3]

Obtaining information from a group of documents is called information retrieval. User submits a search request to the information-retrieval system (Figure2), which evaluates the the document's ranking and indexes the collection of documents before displaying the results that correspond most closely to the user's search.[4]

A text file is represent as a vector in the Vector-Space model, where K is the the quantity of terms in the set, and Dj = (W1j,...,Wkj).. The value of WKj reveals how much the phrase tk enhances the semantics of text Dj, it varies from (0, 1).[5]

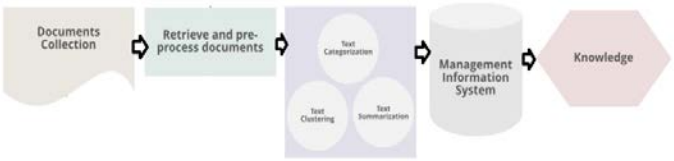


Figure 1. Text Mining Framework

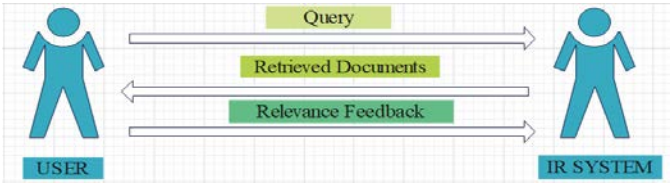


Figure 2. Information-Retrieval System

TERM-WEIGHTING SCHEME

During the text indexing process, a mechanism known as term-weighting is employed to recognise its significance of each word to the text. In order to increase the effectiveness of retrieval, terms are given numerical values that reflect their relevance in a document. In essence, it takes into account the respective weights of various information retrieval system words, considering that not every term in a given document collection is equally significant, this can increase system efficacy. The information retrieval system can

*Maj Gen (Dr) SC Jain, VSM (Retd) Director, Amity School of Engineering & Technology, Director. Amity Directorate of Distance & Online Education, Amity University Madhya Pradesh, Gwalior 474005, India
Tel: 9836682118
Email: scjain@gwa.amity.edu
Cite as: *J. Integr. Sci. Technol., 2023, 11(2), 469.*
URN:NBN:sciencein.jist.2023.v11.469

 ISSN: 2321-4635
http://pubs.thesciencein.org/jist

assess the meaning of a particular term in a particular document or by weighing the terms in a query. It is significant element of any system for retrieving information and one that has established significant assure for enhancing retrieval efficiency.[6]

The Vector Space Model(Figure3) uses a vector of index terms to represent each text document, where every term is assigned a weight indicating how discriminate or formative it is.[7]

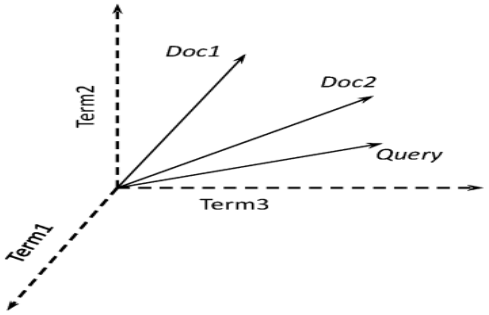


Figure 3. Vector Space IR Model

Term-Weighting Scheme is the method of giving each term a weight. The fundamental concept behind a vector-space model is to signify a text in a way that a computer can comprehend. Vectors can be used to represent any written material in the vector-space model. The vector must represent each spatial dimension as a separate feature, and weight is determined by a number of weighting methods. The document is denoted by the letter $D = (t1, W1;\ t2, W2 \ldots . tn, Wn)$, where ti refers to a term and Wi referes to term's weight in the text. To demonstrate a term's significance in a text file, use word weighting.[8]

In a corpus, constructed 2 brief documents, combined them, and used Rapid-Miner to execute the task of all the term weighting techniques.

Document:1 - $Amity\ university\ is\ very\ big\ university.$
Document: 2 - $I\ am\ student\ in\ Amity\ university.$

A. Term - Occurrence

This is the fundamental term-weighting technique for word token vectorization. The term occurrence word vector N_{ij} we created for this technique counts the occurrences of word tokens Ti in document Dj.[5]

$$Nij = Ti\ (Dj) \qquad (1)$$

Term occurrence (Figure 4) is use to point how frequently the term T_i appears in $document\ Dj\ (Nij)$.

Row No.	amity	big	i	student	university
1	1	1	0	0	2
2	1	0	1	1	1

Figure 4. Two documents' term occurrences

B. Binary Term Frequency

The only distinction between this approach and others is that it just converts word occurrences into 0 and 1 as shown in figure 2, where the term "university" appears twice in document1 but is quickly changed from 2 to 1, It demonstrates(Figure5) that the university word only appeared once.[5]

Row No.	amity	big	i	student	university
1	1	1	0	0	1
2	1	0	1	1	1

Figure 5. Two documents' Binary terms frequency

C. Term-Frequency

The Term-Frequency operates in a extremely simple and direct manner, Term-Frequency measures the ratio of a word's frequency relative to overall words tokens in a text file. We take the operator for the token number from the process documents to calculate the overall word-token count in each text file.[5]

Normalizing term frequency word-lists vectors is the same as that of unit vector normalisation, the Euclidean norm, size, or length of a vector. Using the Pythagorean Theorem, we can identify the norm. Then, the document1 Term-Frequency vector's norm was 0.408, while the document2 Term-Frequency Vector's norm was 0.510 .

$$TFij = Nij / \sum_K Nij \qquad (2)$$

$TFij$ stands for the frequency of term i in document j.

To analysis the file as a whole, similarly we require that the length of every file vector be the same. We therefore dividing each term-frequency vector in a text by the related standard to obtain term-frequency for documents, To accomplish this, split term frequency vector for each document by its norms as illustrated in figure 6, and we will obtain each file's normalised term-frequency vectors.

Row No.	amity	big	i	student	university
1	0.408	0.408	0	0	0.816
2	0.500	0	0.500	0.500	0.500

Figure 6. Term-frequency in 2 documents

D. Term Frequency - Inverse Document frequency

TF-IDF method's procedure is utilised in retrieval of information and text mining. It operates on datasets or collections of documents. The TF-IDF method is utilised to ascertain the word count in a document, how many documents have that word in them, and the proportion of documents that contain that word to all other documents. TF-IDF eliminates low frequency, high frequency and stop-words,. Text classification and summarization can be done using TF-IDF algorithms.[9]

$$IDF(W) = log\ (N/dft) \qquad (3)$$
$$TF.IDF = TFij * IDF(W) \qquad (4)$$

Consequently, figure7 provides TF-IDF of the documents.

Row No.	amity	big	i	student	university
1	0	1	0	0	0
2	0	0	0.707	0.707	0

Figure 7. TF-IDF in 2 documents

METHODOLOGY

Rapid Miner 9.1 is open source software for data mining the studies in this research. It is java-based and offers a user interface with a variety of operators for constructing the strategy. It employs a sophisticated idea in which numerous operators are employed to develop the solution to a particular problem.[10]

The Text Classification Techniques using TF-IDF approach discusses implementation of the different classification algorithms and their results and comparisons of our research work with different algorithms.[11]

Firstly it represents implementation of various methods and selects the best one in our work. This work has made a comparison with different classification methods for this different term weighting methods and similarity measures are used. We have utilised a number of measures, including correlation, recall, precision, accuracy, and error as K-NN (Figure8) & Naive-Bayes (Figure9) classification, term-weighted K-NN & Naive-Bayes classification modification using SVM, term-weighted K-NN & Naive-Bayes classification modification using Information Gain with appropriate similarity measure.[12]

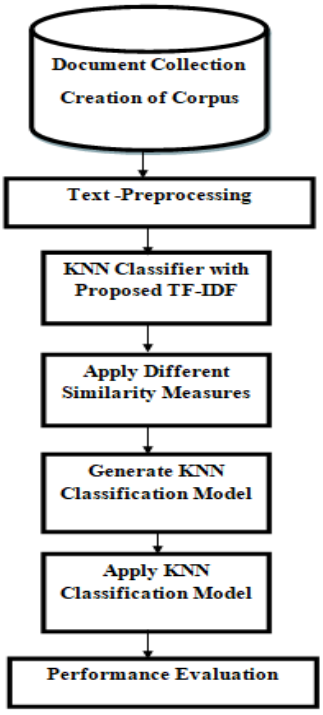


Figure 8. Text Classification Model using K-NN

The datset is processed using the categorization methods Naive-Bayes and K-Nearest Neighbor (KNN). To train the dataset and create a performance model, both methods are applied, We analyse the dataset using this performance model and assess its accuracy, and evaluate how well it performs in the testing dataset.[13,14]

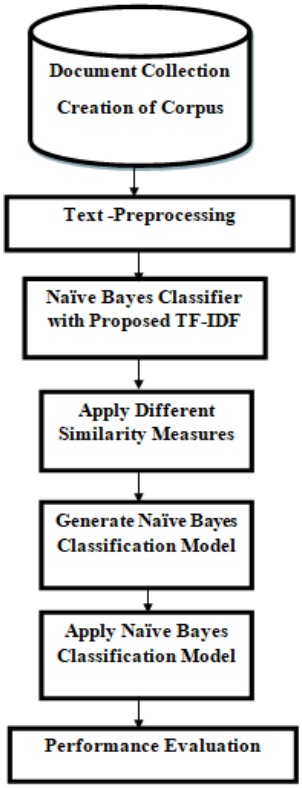


Figure 9. Text Classification Model using Naive- Bayes

The probabilistic-classifier is utilised by the Naive Bayes method (Figure10) to categorise a collection of text documents. The Navie-Bayes Model responds relatively quickly to the selection of an attribute that only manages small dimensions. This approach is very simple to use.[15]

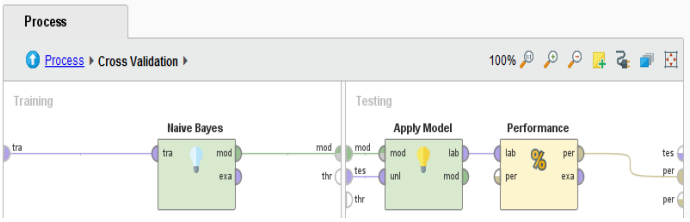


Figure 10.Implementation of Naïve Bayes Model

Below is a formula for Naive Bayes:

$$P(C\_i/X) = P(X/C\_i).P(C)/P(X) \quad (5)$$

$P(C\_i|X)$: Given input X, posterior probability is determined for the Class 'C'..
$P(C)$: Class 'C' prior probability.
$P(X|C\_i)$: Probability that feature 'X' will exist given class
$P(X)$: The feature's prior- probability.

The target class of the object is displayed as having the highest posterior probability, according to a prediction made by the Naïve-

Bayes Classifier. Therefore, the Target Class is one whose posterior-probability is the highest.

For text classification, the K-NN technique (Figure11) is crucial and simple to implement and utilise in data mining. K's value is determined by the user and is arbitrary. An unknown item will be assigned to a certain class utilizing tokens obtained from the neighbours using an algorithm that predicts a target class feature and is based on neighbours.

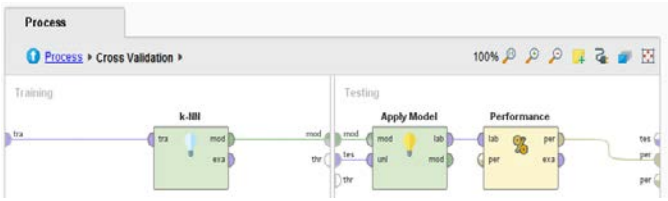


Figure 11. Implementation of K-NN Model

The closest neighbour of the object, also known as the Target-Class for the object, will be its group of association. Distance functions are used to calculate the neighbours' proximity (i.e., depending on how far or how near they are) such is the Hamming, Manhattan, and Euclidean. The formula for K-NN distance formulas can be expressed as:

$$\text{Euclidean Measure} = \sqrt{\sum_{i=1}^{k} (x_i - y_i)} \qquad (6)$$

$$\text{Manhattan Measure} = \sqrt{\sum_{i=1}^{k} |x_i - y_i|} \qquad (7)$$

$$\text{Hamming Measure} = \sum_{i=1}^{k} |x_i - y_i| \qquad (8)$$

where the two target class labels are x and y. The distance is determined for all neighbours, and the anticipated target class for unidentified objects is the one with the shortest distance from the item.

RESULT AND DISCUSSION

K-NN and Naive Bayes classifiersare used in this work, we implemented both the classifier in this paper. We can use a variety of classification algorithms to implement this job, thus it is not just restricted to the data-mining classifier indicated above. For dataset classification and output analysis, numerous other well-liked classification methods are employed. This research compares a variety of categorization methods and their various performance metrics. We evaluate each classifier's output and do research on how well they work and how accurately they generate predictions.

Performance table in Table 1, which is shown below, compares numerous categorization techniques using a variety of weighting schemes and we have conducted additional analysis utilising graphics. These graphs display the study of numerous performance metrics that we have applied to many different classifiers, including K-NN and Naive Bayes.

Table-1: classification of texts using a different term weighting system

Term-Weight Methods	Classification Techniques	Accuracy - Parameter
Term – Occurrence	K-NN	97.50
Term – Occurrence	Naïve – Bayes	92.70
Term - Binary Occurrence	K-NN	97.50
Term - Binary Occurrence	Naïve – Bayes	92.70
Term - Frequency	K-NN	97.50
Term - Frequency	Naïve – Bayes	92.80
TF- IDF	K-NN	97.50
TF- IDF	Naïve – Bayes	92.80

Table1 shows various term weighting schemes for the K-NN and Naïve-Bayes techniques. It exhibits (Figure12) the best performance when both Binary Term Occurrence and Term Occurrence are taken into account. Binary-term occurrence weighting and term-occurrence weighting in K-NN both have accuracy of 97.50%.

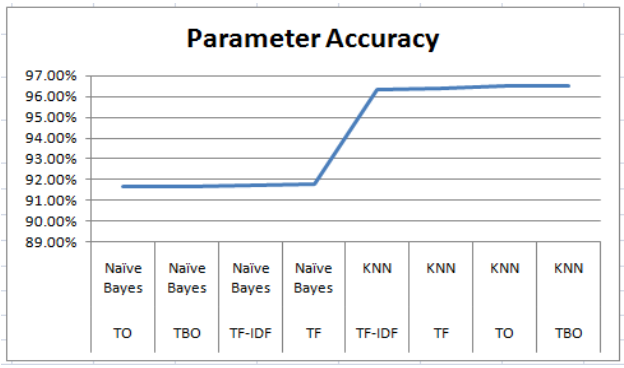


Figure 12. Text categorization accuracy using various term-weights

Table-2: Text categorization using information gain and different term weights

Term–Weight and Information-Gain	Classification - Techniques	Accuracy - Parameter
Term – Occurrence	K-NN	64.60
Term – Occurrence	Naive-Bayes	38.40
Term - Binary Occurrence	K-NN	38.40
Term - Binary Occurrence	Naive-Bayes	64.60
Term - Frequency	K-NN	64.60
Term - Frequency	Naive-Bayes	38.40
TF – IDF	K-NN	38.40
TF – IDF	Naive-Bayes	64.60

The Naive-Bayes and K-NN categorization techniques are shown in Table-2 along with various term weighting schemes and information gain that demonstrate (Figure13) the best performance when both Term Occurrence and Binary Term Occurrence are taken into account. KNN's accuracy with term-occurrence is 64.60%, and its accuracy with binary-term occurrence is also 64.60%. outcome from this implementation are fairly similar.

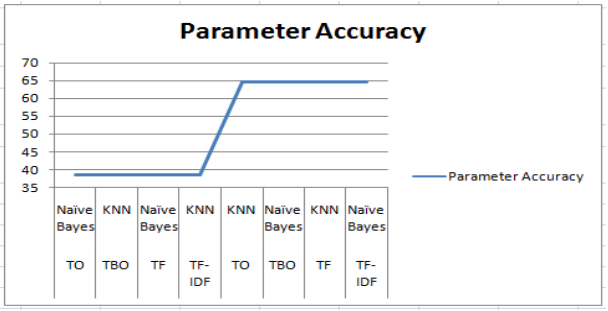


Figure 13. Accuracy of information gain-based text categorization with different term weights

Table-3: SVM-based text categorization with different term weights

Term-Weight and SVM	Classification -Method	Accuracy - Parameter
Term - Occurrence	K-NN	87.20
Term - Occurrence	Naïve – Bayes	54.80
Term - Binary Occurrence	K-NN	87.20
Term - Binary Occurrence	Naïve – Bayes	57.50
Term - Frequency (TF)	K-NN	86.90
Term - Frequency (TF)	Naïve – Bayes	60.60
TF-IDF	K-NN	92.20
TF-IDF	Naïve – Bayes	58.50

In table-3, the K-NN and Naive-Bayes SVM displays the best performance among the classification techniques when both binary term occurrence and term occurrence are taken into account, along with other term weighting schemes. K-NN with term occurrence weighting has an accuracy of 87.20%, while K-NN with binary term occurrence weighting has an accuracy of 87.20% (Figure 14).

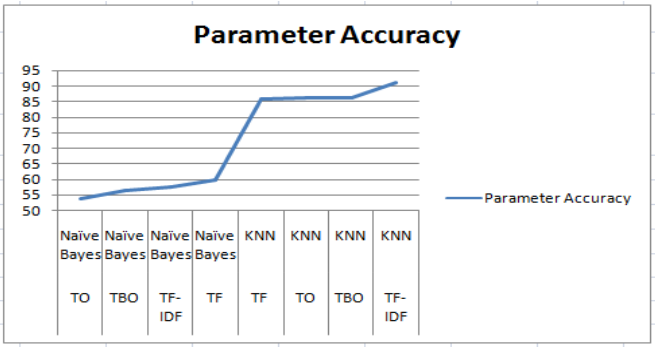


Figure 14. Text classifications' accuracy using SVM with various term weights

In the performance table stated earlier, K-NN and Naive-Bayes performance are compared including metrics like accuracy, recall, precision, classification-error rate, and kappa.

We create classifiers using K-NN and the Naïve-Bayes approach, and three tables also display comparisons of the performance of various classifier techniques. All tables show that K-NN and Naïve-Bayes were used as the starting point for our analysis, with K-NN achieving the maximum accuracy of 96.50% and Naïve-Bayes classifier achieving the highest accuracy of 91.70%.

The classification of Amazon reviews, which have few elements but a strong emphasis on feedback, may not be relevant for this task. K-NN has outperformed other classifiers in terms of overall performance.

CONCLUSION

The goal of this research work is to provide an overview of several term weighting schemes that have been given for text-classification. Compared to standard text categorization techniques, the suggested method offers the most accuracy. Because of this, it has the most actual positives, which raises accuracy. In our suggested strategy, the accuracy parameter has been improved. The proposed method outperforms most measures when compared using a variety of term weighting schemes, information gain, and SVM.

ACKNOWLEDGMENTS

Authors express their gratitude towards AUMP Gwalior for infrastructural support.

CONFLICT OF INTEREST

Authors declare no conflict of interest is there.

REFERENCES AND NOTES

1. J. Han, M. Kamber, J. Pei. Data Mining: Concepts and Techniques. *Data Min. Concepts Tech.* **2012**.
2. R. Talib, M. Kashif, S. Ayesha, F. Fatima. Text Mining: Techniques, Applications and Issues. *Int. J. Adv. Comput. Sci. Appl.* **2016**, 7 (11).
3. C. ShrihariR, A. Desai. A Review on Knowledge Discovery using Text Classification Techniques in Text Mining. *Int. J. Comput. Appl.* **2015**, 111 (6), 12–15.
4. M. Balamurugan, E. Iyswarya. A Trend Analysis of Information Retrieval Models. *Int. J. Adv. Res. Comput. Sci.* **2017**, 8 (5), 531–534.
5. G. Salton, C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, 24 (5), 513–523.
6. G. Salton, M.J. McGill. Introduction to Modern Information Retrieval; International student edition; McGraw-Hill, **1983**.
7. E.E. Ogheneovo, R.B. Japheth. Application of Vector Space Model to Query Ranking and Information Retrieval. Int. J. Adv. Res. Comp. Sci. Software Engin. **2016**, 6(5).
8. S. Brindha, K. Prabha, S. Sukumaran. The Comparison Of Term Based Methods Using Text Mining. *Int. J. Comput. Sci. Mob. Comput.* **2016**, 5 (9), 112–116.
9. W. Zhang, T. Yoshida, X. Tang. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Syst. Appl.* **2011**, 38 (3), 2758–2765.
10. F. Jungermann. Information Extraction with RapidMiner. *GSCL Symp. Sprachtechnologie und eHumanities 2009* **2009**, 2009 (09/28), 1–16.
11. Nidhi, V. Gupta. Recent Trends in Text Classification Techniques. *Int. J. Comput. Appl.* **2011**, 35 (6), 45–51.
12. Y.S. Lin, J.Y. Jiang, S.J. Lee. A similarity measure for text classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2014**, 26 (7), 1575–1590.
13. Y. Wang, Z.O. Wang. A fast KNN algorithm for text categorization. *Proc. Sixth Int. Conf. Mach. Learn. Cybern. ICMLC 2007* **2007**, 6, 3436–3441.
14. A. Lamba, D. Kumar. Survey on KNN and Its Variants. *Int. J. Adv. Res. Comput. Commun. Eng.* **2016**, 5 (5), 430–435.
15. Y. Wang, J. Hodges, B. Tang. Classification of Web Documents Using a Naive Bayes Method. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*; ICTAI '03; IEEE Computer Society, USA, **2003**; p 560.