Asian Journal of Information Management

ISSN 1819-334X

Academic Journals Inc.

www.academicjournals.com

Asian Journal of Information Management 2 (1): 14-22, 2008
ISSN 1819-334X

A Comprehensive Comparative Study Using Vector Space Model with K-Nearest Neighbor on Text Categorization Data

[1]Wa'el Musa Hadi, [2]Fadi Thabtah, [1]Salahideen Mousa, [1]Samer Al Hawari,
[1]Ghassan Kanaan and [1]Jafar Ababnih
[1]Department of Computer Information Systems,
Arab Academy for Banking and Financial Sciences, Amman, Jordan
[2]Department of MIS, Philadelphia University, Amman, Jordan

Abstract: On 20 text categorization data sets, the research investigated different variations of VSM using KNN algorithm and different term weighting approaches compared in term of F1 measure. The experimental results provide evidence that Dice and Jaccard Coefficient outperformed the Cosine Coefficient approach with regards to F1 results and the Dice-based TF. IDF achieved the highest average scores.

Key words: Data mining, text categorization, term weighting, vector space model

INTRODUCTION

Text Categorization (TC) is one of the important problems in Information Retrieval (IR) and data mining communities. This is because of the significance of natural language text, the huge amount of text stored on the internet and the available information libraries and document corpus. Further, TC importance rises up since it concerns with natural language text processing and classification using different techniques, in which it makes the retrieval and other text manipulation processes easy to execute.

They are many TC techniques exists such as: decision trees (Quinlan, 1993), Support Vector Machine (SVM) (Joachims, 1998), rule induction (Moulinier *et al*., 1996), Neural Network (Wiener *et al*., 1995) and k-nearest neighbor (KNN) (Yang, 1999).

In this study, we focus on a text similarity strategy, known as VSM in order to compute the similarity between incoming text (new test cases) and the pre-categorized text in the training data set. We use KNN algorithm to classify incoming text to one of categories. Generally, TC based on text similarity goes through two steps: Similarity measurement and classification assignment.

Term weighting is one of the known concepts in TC, which can be defined as a factor given to a term in order to reflect the importance of that term. There are many term weighting approaches, including, Inverse Document Frequency (IDF) and Weighted Inverse Document Frequency (WIDF) (Tokunaga and Iwayama, 1994). IDF and WIDF focus on terms occurrences inside a text corpus. WIDF distinguishes between two terms that have different occurrences, whereas, IDF treats both terms equally.

Since TC stands at the cross junction to modern IR and ML, Several research papers have focused on it but each of which has concentrated on one or more issues related to such task. There are some research works (Deng *et al*., 2004; Debole and Sebastiani, 2003), which have focused on the different term weighting approaches related to TC such as Term Frequency (TF), WIDF, IDF, Chi-square (Deng *et al*., 2004) and ITF (Leopold and Kindermann, 2002). For example, the researchers of Tokunaga and Iwayama (1994) have achieved good improvement with reference to the retrieval

Corresponding Author: Wa'el Musa Hadi, Department of Computer Information Systems,
Arab Academy for Banking and Financial Sciences, Amman, Jordan

accuracy using WIDF on Japanese language if compared to TF. IDF approach using KNN (Yang, 1999) and Bayesian Model (Tzeras and Hartman, 1993). Specifically, the KNN. WIDF implementation achieved 7.4% higher than that of the TF.IDF.

Yang and Liu (1999) have tested five categorization algorithms SVM (Joachims, 1998), KNN (Yang, 1999), NNet (Wiener *et al*., 1995), LLSF (Yang and Chute, 1994) and NB based on Network (Tzeras and Hartman, 1993) on the Reuters-21578 TC data set. The results showed that SVM, KNN and LLSF outperformed NNet and NB-network when the number of positive training instances per category is less than ten. Further, all the methods performed well when the categories are well distributed in the training data set.

The researchers of Lan *et al*. (2005) proposed a term weighting method called tf*rf and compared their method using the traditional SVM, with other term weighting methods, i.e., (tf.×2, tf.ig, tf.or), on two widely used data sets from (20News, 1999). The experimental results showed that methods based on information theory, i.e., (tf.×2, tf.ig, tf.or), perform poorly if compared with their proposed term-weighted method in terms of accuracy. In Lan *et al*. (2006) a comprehensive comparative study conducted on different term weighting methods using SVM showed that the term weighting method developed in Lan *et al*. (2005) achieved better accuracy than other term weighting methods such as tf.ig and tf.or.

Finally, Hadi *et al*. (2007) conducted a comparative study on IDF and WIDF term weighting with KNN, Experimental results against eight different 20 newsgroup data sets provide evidence that the Cosine Coefficient outperformed Jaccard and Dice Coefficient approaches with regards to F1 measure results and the Cosine-based IDF achieved the highest average scores.

In this study, we compare different variations of VSM (Dice, Jaccard, Cosine) with KNN (Yang, 1999) algorithm using IDF and WIDF. The base of our comparison between the different implementations of KNN is the F1 measure (Van Rijsbergan, 1979). In other words, we want to determine the best VSM, which if merged with KNN produces good results with reference to F1 measure results.

To the best of the authors knowledge, there are no comparisons which have been conducted against English language data collections using different variations of VSM with KNN.

TEXT CATEGORIZATION PROBLEM

TC, also known as text classification, is the task of automatically sorting a set of documents into categories (or classes, or topics) from a predefined set. Such task is related to IR and ML communities. Automated text classification tools are attractive since they free organizations from the need of manual categorization of document, which can be too expensive, or simply not feasible given the constraints of the application or the number of documents involved (Sebastiani, 2005).

TC involves many applications such as automated indexing of scientific articles according to predefined thesauri of technical terms, filing patents into patent directories, selective dissemination of information to information consumers, automated population of hierarchical catalogues of web resources, spam filtering, identification of document genre, authorship attribution, survey coding and even automated essay grading.

After preprocessing, indexing and transforming documents into the vector model representation, we will have a data set $D = \{d_1,\ldots,d_n\}$ represented as a set of vectors for n documents, also we have a set of category $C = \{C_1,\ldots,C_m\}$.

TC problem can be defined according to Sebastiani (1999) as follows: The documents divided in two datasets, for training and testing. Let training data set = $\{d_1,d_2,\ldots,d_g\}$, where, g documents are used

Table 1: Representation of text categorization problem

	Training data set			Test data set		
Category	d_1	…	d_j	d_{j+1}	…	d_n
C_1	C_{11}	…	C_{1j}	$C_{1(j+1)}$		C_{1n}
…	…	…	…	…	…	…
C_m	C_{m1}	…	C_{mg}	$C_{m(j+1)}$	…	C_{mn}

as examples for the classifier and must contain sufficient number of positive examples for all the categories involved. The testing data set $\{d_{g+1},d_{g+2},\ldots,d_n\}$ used to test the classifier effectiveness. The following matrix represents data splitting into training and testing parts, A document d_k is considered a positive example to C_y if C_{ky} =1 and a negative example if c_{ky} = 0.

Term Weighting

Term weighting is one of the important issues in TC, which has been widely investigated in IR (Salton and McGill, 1983; Salton, 1988). Term weighting corresponds to a value given to a term in order to reflect the importance of that term in a document.

Although there are different weighting approaches for text indexing, they all share the following two observations:

- The more the number of times a term occurs in documents that belong to some category, the more it is relative to that category.
- The more the term appears in different documents representing different categories, the less the term is useful for discriminating between documents as belonging to different categories (Table 1).

Term Frequency (TF)

One of the simplest term weighting methods that used to measure the importance of each term in a given document is TF (Tokunaga and Iwayama, 1994). Using this method, each term is assumed to have a value proportional to the number of times it occurs in a text. Generally, for a document d and a term t, the weight of t in d is given as:

$$W(d, t) = TF(d, t) \quad (1)$$

TF can help in improving an IR and TC evaluation measure named recall (Van Rijsbergan, 1979) since frequent terms tend to appear in many documents, such terms have little discriminative power. Recall is the fraction of the relevant documents which has been retrieved and is represented in Eq. 11 according to Table 4. To some extend, we can say that TF follows the normal distribution curve with regards to the importance of terms to the retrieval process, which means too much frequency or less frequency does not improve the retrieval process.

Figure 1 shows that the term frequencies in the interval [0, 5] and the interval [15, 20] are too low, so we remove them from the term list. We also remove the stop words, which often has high frequencies. We keep the interval [5, 8.5] and the interval [11.5, 15] since the term frequencies are ideal for the retrieval process.

Inverse Document Frequency (IDF)

TF reflects the importance of the term in a single document, however, what if we are interested in the frequency of a term in the set of documents. This is called the Inverse Document Frequency

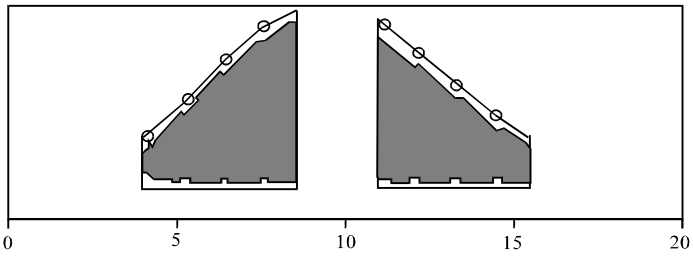

Fig. 1: Term frequency

Table 2: The relation between n documents and the importance of terms they contain

N: Total No. of documents	n: No. of documents contain the term	IDF = Log(N/n)	Importance of the term
1000	10	2.000	Maximum
1000	20	1.699	
1000	40	1.399	
1000	80	1.097	
1000	160	0.795	
1000	320	0.494	
1000	640	0.190	Minimum

(IDF), meaning the importance of each term inversely proportional to the number of documents that contain that term (Sparck, 1972). Table 2 shows that for a given corpus of documents, when a given term frequency increases within a document, the importance of that term decreases according to the IDF. In other words, when the term occurs in a small number of documents, this signifies it (when n equal ten). Whereas, when the term occurs frequently within a large number of documents, then it has insignificant importance according to IDF.

For a given N documents, if n documents contain the term t, IDF is given as follows:

$$IDF(t) = \log(N/n) \tag{2}$$

Sometimes n is replaced by the document frequency (the number of documents that contain t) i.e., df(t). This approach follows Slaton's definition (Salton, 1988), which combined TF and IDF to weight the terms and he showed that his approach gives better performance with reference to accuracy than IDF and TF. The product of TF and IDF is given in Eq. 3 as:

$$W(d, t) = TF(t).IDF(t) \tag{3}$$

Weighted Inverse Document Frequency (WIDF)

One of the IDF drawbacks is that all documents containing a certain term are treated equally due to the binary counting. In other words, if a term sea occurred in 4 documents with different frequencies in each of these documents, the IDF does not consider the number of times in which sea has occurred in these 4 documents, rather it mainly considers the fact that sea has occurred. WIDF of a term t in document d is given by:

$$WIDF(d,t) = \frac{TF(d,t)}{\sum_{i \in D} TF(i,t)} \tag{4}$$

where, TF(d, t) is the occurrence of t in d and i ranges over the documents in the collection D. WIDF corresponds to the normalized term frequency over the collection. The weight of a term with reference to WIDF is given as:

$$W(d, t) = WIDF(d, t) \tag{5}$$

Similarity Measurements

There are several well-known similarity techniques, such as: VSM and Probabilistic Model (PM) (Tokunaga and Iwayama, 1994). In this study, we focus on VSM by adapting Cosine as shown in Eq. 6, Jaccard as shown in Eq. 7 and Dice as shown in Eq. 8.

$$Sim(Vi,Vj) = \frac{\sum_{k=1}^{m}(W_{ik} \times W_{jk})}{\sqrt{\sum_{k=1}^{m} W^2_{ik} \times \sum_{k=1}^{m} W^2_{jk}}} \tag{6}$$

$$Sim(Vi,Vj) = \frac{\sum_{k=1}^{m}(W_{ik} \times W_{jk})}{\sum_{k=1}^{m} W^2_{ik} + \sum_{k=1}^{m} W^2_{jk} - \sum_{k=1}^{m}(W_{ik} \times W_{jk})} \tag{7}$$

$$Sim(Vi,Vj) = \frac{2\sum_{k=1}^{m}(W_{ik} \times W_{jk})}{\sum_{k=1}^{m} W^2_{ik} + \sum_{k=1}^{m} W^2_{jk}} \tag{8}$$

where, W_{ik} corresponds to the weight of the k-th element of the term vector V_i, i.e., pre-categorized documents and W_{jk} is the weight of K-th element of the term vector V_j i.e., incoming text. The greater the value of Sim (V_i, V_j), the more similar these two texts are.

KNN Algorithm

There are many approaches to assign category to incoming text such as (Quinlan, 1993; Thabtah *et al*., 2004; Tzeras and Hartman, 1993). In present study, we implemented Text-to-Text Comparison (TTC), which is also known as the k-nearest neighbour (k-NN) (Yang, 1999). KNN is a statistical classification approach, which has been intensively studied in pattern recognition over four decades. KNN has been successfully applied to TC problem, i.e., (Yang and Liu, 1999; Yang, 1999) and showed promising results if compared with other statistical approaches such as Baysian based Network (Tzeras and Hartman, 1993).

The KNN algorithm is quite simple: Given a test document to be classified, the algorithm searches for the k nearest neighbors among the pre-classified training documents based on some similarity measure and ranks those k-neighbors based on their similarity scores, the categories of the k-nearest neighbors are used to predict the category of the test document by using the ranked scores of each as the weight of the candidate categories, if more than one neighbor belong to the same category then the sum of their scores is used as the weight of that category, the category with the highest score is assigned to the test document provided that it exceeds a predefined threshold, more than one category can be assigned to the test document. One draw back in KNN is the difficulty to determine the value of k, a series of experiments with different k values should be conducted to determine the best value of k, another disadvantage of KNN is the complexity of computation time needed to traverse all the training documents.

EXPERIMENT RESULTS

Experiments on the 20NewsGroups data sets (20NG) (20News, 1999) using three TC techniques based on vector model similarity (Cosine, Jaccard, Dice) have been conducted. We used F1 evaluation measure as the base of our comparison, where F1 is computed based on the following equation:

$$F1 = \frac{2 * Precision * Recall}{Recall + Precision} \tag{9}$$

Precision and recall are widely used evaluation measures in IR and ML, where according to Table 3.

$$Precision = \frac{X}{(X + Y)} \tag{10}$$

$$Recall = \frac{X}{(X + Z)} \tag{11}$$

To explain precision and recall, let's say someone has 5 blue and 7 red tickets in a set and he submitted a query to retrieve the blue ones. If he retrieves 6 tickets where 4 of them are blue and 2 that are red, it means that he got 4 out of 5 blue (1 false negative) and 2 red (2 false positives). Based on these results, precision = 4/6 (4 blue out of 6 retrieved tickets) and recall = 4/5 (4 blue out of 5 in the initial set).

Three TC techniques based on vector model similarity (Cosine, Jaccard and Dice) have been compared in term of F1 measure. These methods use same strategy to classify incoming text i.e., KNN. We have several options to construct a text classification method; we compared techniques using different term weighting IDF and WIDF. KNN was implemented using VB.NET on 2.8 Pentium IV machine with 256 RAM.

The dataset is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g., comp.sys.ibm.pc.hardware/ comp.sys. mac.hardware), while others are highly unrelated (e.g., misc.forsale/soc.religion.christian). The dataset is sorted by date into training (60%) and test (40%) sets, does not include cross-posts (duplicates) and does not include newsgroup-identifying headers (Xref, Newsgroups, Path, Followup-To, Date) (20News, 1999).

Table 4 and 5 shows the F1 results of the text categorizers generated against the 20 data sets, K parameter in the KNN algorithm is varied from 3 to 11 by 2.

After analyzing Table 4 and 5, we found out that we discovered that there is consistency between Dice based on TF.IDF and Jaccard based on TF.IDF algorithm in which both of them outperformed Cosine based TF.IDF, Cosine based WIDF, Dice based WIDF and Jaccard based WIDF. Particularly, Dice based TF.IDF outperformed Dice based WIDF, Jaccard based WIDF, Cosine based TF.IDF and Cosine based WIDF on 10, 10, 19 and 12 data sets, respectively.

There are similarities between (1) Dice based TF.IDF and Jaccard based TF.IDF and (2) Dice based WIDF and Jaccard based WIDF with respect to the average results of F1 measure.

Finally, larger values of k reduce the effect of noise on the classification; this result is entirely consistent with that in (Tokunaga and Iwayama, 1994).

Table 3: Documents possible sets based on a query in IR

Iteration	Relevant	Irrelevant
Documents retrieved	X	Y
Documents not retrieved	Z	W

Table 4: F1 results of the Cosine implementations with KNN

	Cosine									
	K = 3		K = 5		K = 7		K = 9		K = 11	
Category	TF.IDF	WIDF	TF.IDF	WIDF	TF.IDF	WIDF	TF.IDF	WIDF	TF.IDF	WIDF
Atheism	49.28	46.88	54.17	54.36	56.47	53.02	60.09	56.47	59.02	57.50
Graphics	38.37	35.93	45.26	39.75	46.15	44.87	47.01	44.76	48.35	47.89
Windows.misc	30.11	38.11	33.57	42.68	38.00	45.57	41.41	47.88	42.73	50.53
Pc.hardware	39.68	37.59	43.67	41.98	44.61	42.58	45.96	45.07	47.71	46.87
Mac.hardware	38.92	38.47	44.61	45.29	47.47	51.29	48.49	54.44	49.44	56.51
Windows.x	46.51	45.64	50.18	53.65	51.14	54.92	51.74	55.49	54.89	57.46
Forsale	28.54	37.76	29.61	41.21	32.06	45.16	29.52	47.97	31.98	48.53
Autos	37.16	46.21	43.19	52.63	46.71	56.87	49.35	60.68	53.59	61.61
Motorcycles	57.93	64.95	63.60	68.28	67.33	70.41	67.33	72.81	67.24	73.38
Baseball	46.15	50.35	47.06	59.73	49.43	64.55	49.38	68.99	51.06	71.62
Hockey	49.47	54.37	57.58	59.44	63.49	62.33	66.35	66.29	68.56	67.10
Crypt	59.20	55.77	65.63	62.91	67.68	68.80	69.29	69.48	71.47	69.54
Electronics	32.84	36.68	35.82	40.78	39.58	43.81	40.22	46.56	40.73	48.61
Med	48.48	35.18	51.79	44.23	54.97	48.42	57.76	50.72	56.39	54.43
Space	53.12	44.38	54.95	50.78	58.51	55.65	60.47	60.66	62.53	62.47
Christian	43.27	52.55	47.96	58.74	54.71	61.18	56.81	63.25	59.18	66.40
Guns	47.59	52.28	49.57	58.24	53.14	59.55	55.82	64.17	55.41	64.01
Mideast	61.49	51.38	63.92	57.44	65.21	64.54	66.15	67.14	67.67	69.11
Politics.misc	33.50	31.54	36.05	39.19	40.07	43.74	42.36	46.15	43.64	47.85
Religion.misc	31.22	34.54	32.27	43.68	32.74	45.19	30.02	48.31	30.11	47.25
Average (%)	43.64	44.53	47.52	50.75	50.47	54.12	51.78	56.87	53.08	58.43

Table 5: F1 results of the Jaccard and Dice implementations with KNN

	Cosine									
	K = 3		K = 5		K = 7		K = 9		K = 11	
Category	TF.IDF	WIDF	TF.IDF	WIDF	TF.IDF	WIDF	TF.IDF	WIDF	TF.IDF	WIDF
Atheism	62.93	45.99	62.60	53.76	63.44	55.41	62.92	56.77	63.72	56.61
Graphics	40.84	34.93	45.72	41.98	46.41	44.90	46.44	46.92	48.03	48.52
Windows.misc	39.17	36.45	44.10	42.67	49.63	44.47	53.96	47.53	57.85	49.07
Pc.hardware	40.88	37.93	48.45	41.40	49.26	42.67	49.56	43.95	50.91	44.64
Mac.hardware	40.90	38.77	45.98	46.64	50.30	53.35	49.58	55.10	50.98	55.89
Windows.x	50.84	45.38	54.83	52.67	57.70	53.86	57.64	56.65	58.68	56.81
Forsale	38.77	39.81	42.69	44.36	42.52	50.27	47.35	50.67	47.48	51.52
Autos	48.76	45.74	51.26	52.59	53.69	56.73	55.08	59.54	58.43	62.07
Motorcycles	66.50	64.88	69.51	69.25	71.79	70.85	71.68	72.64	73.43	73.62
Baseball	49.34	49.17	57.21	59.98	60.80	64.71	58.92	67.56	61.12	71.82
Hockey	58.95	54.50	68.97	60.77	72.45	64.52	74.65	66.89	75.26	68.67
Crypt	68.87	53.82	73.45	62.55	74.28	65.54	75.46	65.79	75.20	67.06
Electronics	42.20	36.39	46.76	41.43	48.52	44.02	49.24	46.92	48.88	49.50
Med	48.02	35.55	52.98	46.06	56.29	51.01	58.70	52.45	58.64	55.42
Space	60.47	42.75	66.14	50.35	65.23	57.10	67.09	60.89	68.97	63.59
Christian	59.84	51.13	63.67	63.16	65.91	62.70	67.50	64.20	69.61	67.37
Guns	55.86	54.11	59.40	57.14	60.47	60.24	62.44	63.84	62.23	63.41
Mideast	64.37	50.22	65.22	57.06	66.20	62.80	67.31	66.10	67.85	69.10
Politics.misc	52.08	34.15	50.96	39.64	52.84	44.32	54.91	47.58	54.51	48.42
Religion.misc	42.40	34.25	39.43	42.06	40.40	45.74	39.11	46.98	41.55	46.90
Average (%)	51.60	44.30	55.47	51.28	57.41	54.76	58.48	56.95	59.67	58.50

CONCLUSIONS

In this study, we investigated different variations of VSM using KNN algorithm, these variations are: Cosine coefficient, Jacaard coefficient and Dice coefficient, using IDF and WIDF term weighting measures. The base of our comparisons is the F1 evaluation measure. The average F1 results obtained against 20 data sets indicated that Dice based TF.IDF and Jaccard based TF.IDF outperformed Cosine

based TF.IDF, Cosine based WIDF, Dice based WIDF and Jaccard based WIDF. We plan in near future to experiment other TC data collections especially Arabic data sets. Also we plan to propose a new TC technique based on association rule mining.

REFERENCES

Debole, F. and F. Sebastiani, 2003. Supervised term weighting for automated text categorization. Proceedings of the 2003 ACM Symposium on Applied Computing. ACM Press, pp: 784-788.

Deng, Z.H., S.W. Tang, D.Q. Yang, M. Zhang, L.Y. Li and K.Q. Xie, 2004. A comparative study on feature weight in text categorization. Lecture Notes in Computer, pp: 588-597.

Hadi, W., F. Thabtah and H. Abdeljaber, 2007. A comparative study using vector space model with K-nearest neighbor on text categorization data. Proceedings of the 2007 International Conference of Data Mining and Knowledge Engineering. London, UK., pp: 296-300.

Joachims, T., 1998. Text Categorization with support vector machines: Learning with many relevant features. Proceedings of the European Conference on Machine Learning (ECML), Berlin, Springer, pp: 173-142.

Lan, M., S.Y. Sung, H.B. Low and C.L. Tan, 2005. A comparative study on term weighting schemes for text categorization. Proceedings of the International Joint Conference on Neural Networks, pp: 1032-1033.

Lan, M., C.L. Tan and H.B. Low, 2006. Proposing a new term weighting scheme for text categorization. Proceedings of the 21st National Conference on Artificial Intelligence, pp: 763-768.

Leopold, E. and J. Kindermann, 2002. Text categorization with support vector machines. How to represent texts in input space? Mach. Learn., 46 (1-3): 423-444.

Moulinier, I., G. Raskinis and J. Ganascia, 1996. Text categorization: A symbolic approach. Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval, pp: 87-99.

Quinlan, J., 1993. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

Salton, G. and M.J. McGill, 1983. Introduction to Modern Information Retrieval. McGill-Hill.

Salton, G., 1988. Automatic Text Processing: The Transformation, Analysis Retrieval of Information by Computer. Addison-Wesley.

Sebastiani, F., 1999. A tutorial on automated text categorisation. Proceedings of the ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, pp: 7-35.

Sebastiani, F., 2005. Text Categorization. In: Text Mining and its Applications, Alessandro Zanasi (Ed.). WIT Press, Southampton, UK., pp: 109-129.

Sparck, K.J., 1972. A statistical interpretation of term specificity and its application in retrieval. J. Documentation, 28 (1): 11-21.

Thabtah, F., P. Cowling and Y. Peng, 2004. MMAC: A new multi-class, multi-label associative classification approach. Proceedings of the 4th IEEE International Conference on Data Mining (ICDM '04), Brighton, UK., pp: 217-224.

Tokunaga, T. and M. Iwayama, 1994. Text categorization based on weighted inverse document frequency. Technical Report 94 TR0001, Department of Computer Science, Tokyo Institute of Technology: Tokyo, Japan.

Tzeras, K. and S. Hartman, 1993. Automatic indexing based on bayesian inference networks. Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93), pp: 22-34.

Van Rijsbergan, C., 1979. Information Retrieval. 2nd Edn. Buttersmiths, London.

Wiener, E., J.O. Pedersen and A.S. Weigend, 1995. A neural network approach to topic spotting. Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95), Las Vegas, Nevada, pp: 317-332.

Yang, Y. and C.G. Chute, 1994. An example-based mapping method for text categorization and retrieval. ACM Trans. Inform. Syst. (TOIS), 12 (3): 252-277.

Yang, Y., 1999. An evaluation of statistical approaches to text categorization. J. Inform. Retreiv., 1 (1/2): 67-88.

Yang, Y. and X. Liu, 1999. A re-examination of text categorization methods. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp: 42-49.

20News, 1999. Groups: http://people.csail.mit.edu/jrennie/ 20Newsgroups/.