TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams

Joel W. Reed[1], Yu Jiao[1], Thomas E. Potok[1], Brian A. Klump[1], Mark T. Elmore[1], and Ali R. Hurson[2]

[1]Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

[2]Computer Science and Engineering Department
The Pennsylvania State University
University Park, PA 16801

{reedjw, jiaoy, potokte, klumpba, elmoremt} @ornl.gov, hurson@cse.psu.edu

ABSTRACT

In this paper, we propose a new term weighting scheme called Term Frequency – Inverse Corpus Frequency (TF-ICF). It does not require term frequency information from other documents within the document collection and thus, it enables us to generate the document vectors of *N* streaming documents in linear time. In the context of a machine learning application, unsupervised document clustering, we evaluated the effectiveness of the proposed approach in comparison to five widely used term weighting schemes through extensive experimentation. Our results show that TF-ICF can produce document clusters that are of comparable quality as those generated by the widely recognized term weighting schemes and it is significantly faster than those methods.

1. INTRODUCTION

Document clustering is an enabling technique for many other machine learning applications, such as information classification, filtering, routing, topic tracking, and new event detection [2]. Today, dynamic data stream clustering poses significant challenges to traditional methods.

Typically, clustering algorithms use the *Vector Space Mode*l (VSM) [17] to encode documents. The VSM relates terms to documents, and since different terms have different importance in a given document, a term weight is associated with every term [18]. These term weights are often derived from the frequency of a term within a document or set of documents. Many term weighting schemes have been proposed [5,9,18]. Most of these existing methods work under the assumption that the whole data set is available and static. For instance, in order to use the popular Term Frequency - Inverse Document Frequency (TF-IDF) approach and its variants, one needs to know the number of documents in which a term occurred at least once (document frequency). This requires a priori knowledge of the data, and that the data set does not change during the calculation of term weights.

The need for knowledge of the entire data set significantly limits the use of these schemes in applications where continuous data streams must be analyzed in real-time. For each new document, this limitation leads to the update of the document frequency of many terms and therefore, all previously generated term weights needs recalibration. For *N* documents in a data stream, the computational complexity is $O(N^2)$, assuming that the term space *M* per document is much less than the number of documents. Otherwise, the computational complexity is $O(N^2MlogM)$, where $O(MlogM)$ computations are needed to update a document.

In order to address the problem of finding and organizing information from dynamic document streams, we proposed a new term weighting scheme called Term Frequency – Inverse Corpus Frequency (TF-ICF). It does not require term frequency information from other documents within the set and thus, it can process document streams in linear time.

To assess the effectiveness of this proposed approach in document clustering tasks, we compared our scheme with five widely used term weighting schemes through extensive experimentation. Our results show that TF-ICF can produce clusters that are of comparable quality to those generated by the widely recognized term weighting schemes, such as TF-IDF [18], Okapi [9] and LTU [5], and it is significantly faster than those methods.

The major contributions of our work can be summarized as follows:

- We removed the technical bottleneck in dynamic document clustering by proposing a new term weighting scheme, TF-ICF, in which the inverse document frequency of TF-IDF is replaced by inverse corpus frequency (ICF) values, reducing the computational complexity of generating representations for *N* dynamic documents from $O(N^2)$ to $O(N)$.
- The effectiveness of TF-ICF is shown to be comparable to standard methods through extensive experimentation.
- Since calculating the TF-ICF term weights for a set of documents is not dependent on any global features of the set, it is easily parallelizable and can be used to supply downstream parallel algorithms with document vectors.

The remainder of this paper is organized as follows: Section 2 introduces the preliminaries; Section 3 details the TF-ICF approach; Section 4 presents the performance evaluation; and finally, Section 5 concludes our work.

2. PRELIMINARIES

In this section, we present an overview of the concept of the vector space model and different term weighting schemes pertinent to document representation. Throughout this paper, we will use the symbols *N*, *M*, and *K* to denote the number of documents, the number of terms in a document, and the number of classes, respectively.

The majority of the clustering algorithms proposed to date use the vector space model (VSM) [17] to represent a document. In this model, each document is represented by a vector in the term-space.

In order to quantitatively judge the similarity between a pair of documents, a method is needed to determine the significance of each term in differentiating one document against other documents. Various weighting schemes have been proposed to help define the significance of terms [5,9,18].

TF-IDF and its variants are commonly used term weighting schemes. TF-IDF uses the inverse of a term's document frequency within the data collection to balance the phenomenon observed by Zipf [23]. The basic form of TF-IDF can be described by equation (1).

$$w_{ij} = \log(f_{ij}) \times \log(N / n_j) \quad (1)$$

Here, w_{ij} is the weight of the term j in document i; f_{ij} is the number of occurrences of term j in document i (TF); N is the total number of documents; and n_j is the number of documents in which term j occurs at least once. (n_j/N) is often referred to as the document frequency (DF) of term j and naturally, (N/n_j) is called the inverse document frequency (IDF) of term j.

Other widely used term weighting methods include: mutual information (MI) from information theory [13], ATC, Okapi [8], and LTU [5]. ATC, Okapi, and LTU are TF-IDF variations that take additional parameters into consideration. ATC uses the maximum term frequency, *max_f*; Okapi and LTU utilize the document length (*dl*) and the average document length (*avg_dl*) in their term weighting equations. Okapi and LTU often produce the best clustering results when compared with other weighting schemes [8]. Table 1 lists the weighting function used by each of the schemes.

Table 1. Term Weighting Schemes

Name	Term Weighting Scheme
TF-IDF	$w_{ij} = \log(f_{ij}) \times \log(N / n_j)$
MI	$w_{ij} = \log \dfrac{\frac{f_{ij}}{N}}{\frac{\sum_{i=1}^{N} f_{ij}}{N} \times \frac{\sum_{j=1}^{M} f_{ij}}{N}}$
ATC	$w_{ij} = \dfrac{\left(0.5 + 0.5 \times \frac{f_{ij}}{max_f}\right) \log\left(\frac{N}{n_j}\right)}{\sqrt{\sum_{i=1}^{N} \left[\left(0.5 + 0.5 \times \frac{f_{ij}}{max_f}\right) \log\left(\frac{N}{n_j}\right)\right]^2}}$
Okapi	$w_{ij} = \left(\dfrac{f_{ij}}{0.5 + 1.5 \times \frac{dl}{avg_dl} + f_{ij}}\right) \log\left(\dfrac{N - n_j + 0.5}{f_{ij} + 0.5}\right)$
LTU	$w_{ij} = \dfrac{(\log(f_{ij}) + 1.0) \log\left(\frac{N}{n_j}\right)}{0.8 + 0.2 \times \frac{dl}{avg_dl}}$

One common characteristic of these term weighting schemes is that they all require knowledge of the entire document collection. In other words, if a TF-IDF based method is used to generate document representation, a newly arriving document requires the weights of existing document vectors to be recalculated. Consequently, any applications that rely on the document vectors will also be affected. This fact significantly hinders their use in applications where dynamic data streams need to be processed in real-time.

In this paper, we propose a new term weighting scheme, namely TF-ICF, which generates document representations independently without the knowledge the document stream being examined. Its computational complexity is $O(N)$.

3. TF-ICF – A NEW TERM WEIGHTING SCHEME

In trying to formulate a new term weighting method that is better suited for a fluid collection of documents, we came upon the fundamental question to this problem: Assume that the document frequencies of terms in a specific domain of English usage follow some distribution. Can we preserve such a distribution through sampling techniques? More specifically, we need to address the following three questions:

I. Can we use the document frequency distribution of a smaller data set to approximate that of a larger data set?

II. Can we use the document frequency distribution of one corpus to approximate another?

III. Can we compose a corpus such that it covers nearly all the words commonly found in writing?

If the answers to these three questions were 'yes,' the global elements, the IDF, in the TF-IDF scheme could be replaced with information gathered from a known static corpus. As a result, the term weights of a document in a dynamic data stream become independent of other documents, and the previously created document vectors do no need to be updated when a new document arrives.

Unfortunately, it would be extremely difficult to obtain answers to these questions through rigorous mathematical and statistical analysis, if such methods do exist. Therefore, we decided to gain insights through observations over large data sets.

3.1 Test Data

The test data are from the Text Retrieval Conference's Text Research Collection Volume 5 (TREC-5) [21] and the novel "War and Peace" by Leo Tolstoy [15]. The TREC-5 collection contains 130,471 news feed documents from the Foreign Broadcast Information Service (FBIS) and 127,742 news feed documents from the Los Angeles Times (LATIMES). These documents have been generally collected from the same domain – news and events. Considering the usage of the English language, the novel "War and Peace" is from a distinctively different domain than the TREC-5 data collection, and we used it in the experiments attempting to elucidate question II.

3.2 Observation I

The intuition behind this set of experiments is that high frequency words like 'the' or 'of' will occur in approximately the same percentage of documents no matter whether the document set is small or large and similarly, low frequency words like 'dressage' will occur very rarely across small and large datasets.

If we assume that each smaller set of documents is a sub-set of the next larger set studied, then the comparable term set is limited by the smallest experimental document collection. The fact that a larger set of documents will likely introduce new terms will be explored in Section 3.4.

To test our intuition, we designed a set of experiments where we took sets of documents of various sizes (1K -- 100K) from the LATIMES collection and calculated the document frequencies of all terms in each data set. Figure 1 shows the document frequency distribution of all terms. On the x-axis, the terms from the 100K size set are sorted in increasing order of their document frequencies. From this figure, we only observe small variations in the document frequencies calculated based on document sets of different sizes.

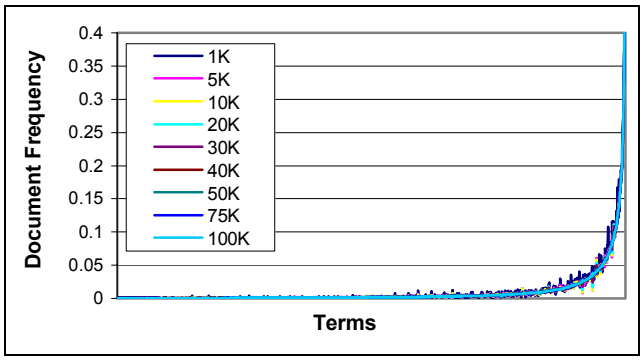

Figure 1. LATIMES data collection document frequency distribution.

The same set of experiments was also conducted on the FBIS data collection and Figure 2 plots the results. This figure shows slightly greater variations in the document frequency distribution than that seen in the LATIMES. This behavior is probably explained by FBIS being transcripts of foreign broadcasts and therefore, the word usage may exhibit some degree of inconsistency.

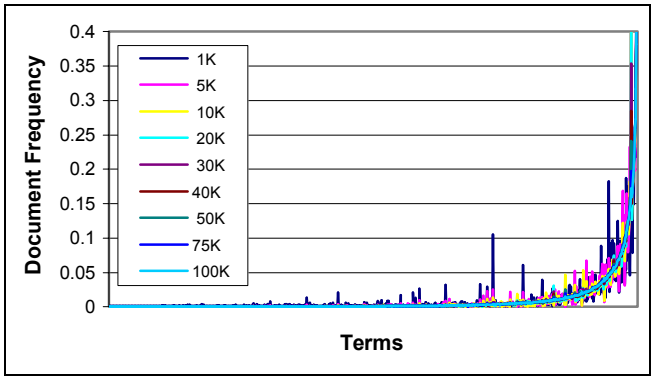

Figure 2. FIBS data collection document frequency distribution.

Observations made over these two figures lead us to believe that the document frequency of a term in a corpus of unknown size can be estimated by using a set of known documents. Note that in order to cover a reasonably large vocabulary, the document collection used for estimation should be sufficiently large.

3.3 Observation II

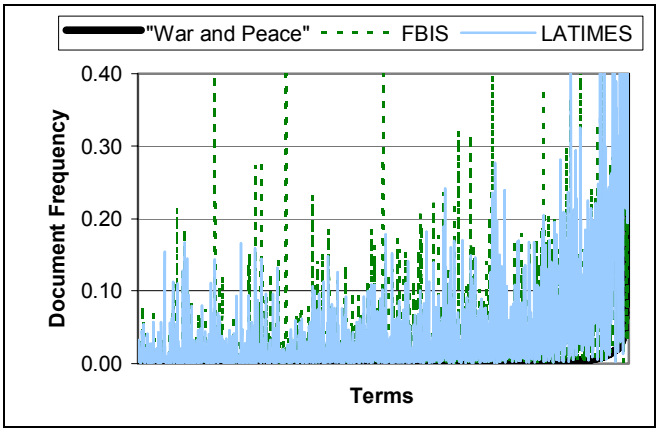

Figure 3. The document frequency distributions of "War and Peace," FBIS, and LATIMES.

Intuitively, documents from the same domain should demonstrate similar document frequencies of terms, and documents from different domains may bear different characteristics. Thus, we expect documents from LATIMES and FBIS to demonstrate similar document frequency distributions because these two document collections belong to the same domain of news and events. However, the novel "War and Peace" should exhibit a different distribution than that of LATIMES and FBIS because it is from a distinctively different domain of English usage.

To test this intuition, 120,000 articles were randomly taken from each of the LATIMES collection and the FBIS collection. There are 17,546 unique terms in "War and peace" and we sorted the results in increasing order their document frequencies. Figure 3 plots the document frequency distributions of these terms in LATIMES, FBIS, and "War and Peace."

The results agree with our expectation: While LATIMES and FBIS demonstrate similar document frequency distributions, but "War and Peace" exhibits a very different distribution. These results imply a positive answer to question II: Document frequency data obtained from one source can effectively be used to estimate a different, but similar, source. However, it is not effective in approximating that of documents obtained from a very different domain.

3.4 Observation III

Although many sources suggest that somewhere between 750,000 and 1,000,000 unique English words exist [12], only a subset of them appear in writings commonly seen. Our intuition is that if the sample document set size is large enough, any additional documents will introduce very few new words. We conducted an experiment where the LATIMES and FBIS collections were combined and the number of unique terms (after stemming) was calculated for several different sized sub-collections.

Results in Figure 4 show that when the document set size is small, the unique term count continues to climb up as the number of documents increases. However, this growth of the unique term count is reduced sharply as the number of documents becomes very large. This observation indicates that if the document collection is sufficiently large, we can expect to see very few new words by adding more documents.

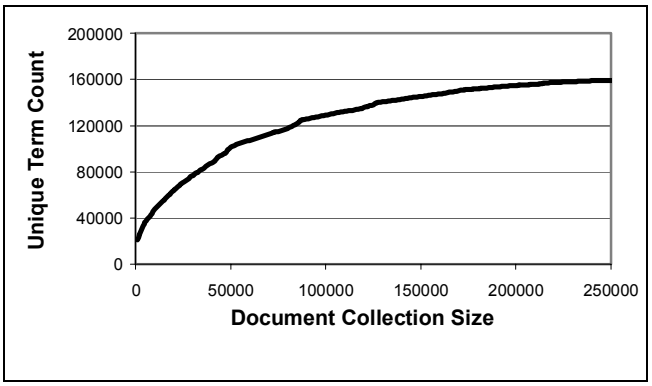

Figure 4. Unique term count.

3.5 TF-ICF

Since we have observed positive evidences for questions I, II, and III through empirical studies, we believe that it is reasonable to express the IDF component of TF-IDF using term occurrence rates from a sufficiently large and diverse static corpus. We propose a new term weighting scheme called Term Frequency – Inverse Corpus Frequency (TF-ICF) in which the weight of each term is calculated as follows:

$$w_{ij} = \log(1 + f_{ij}) \times \log\left(\frac{N+1}{n_j+1}\right) \tag{2}$$

We composed a corpus of 258,231 documents from the LATIMES and FBIS data collections. In the ICF table, we store N, which is the total number of documents in the corpus. Also, for each unique term j, after removing the stopwords and applying Porter's Stemming Algorithm [14], we store n_j, which is the number documents in the corpus where term j occurred one or more times. As a result, the task of generating a weighted document vector for a document in a dynamic data stream is as simple as one table lookup. The computational complexity of processing N documents is therefore, $O(N)$. In contrast, in order to achieve the same effect, the computational complexity of TF-IDF and its variants is $O(N^2)$. It is important to note that it is possible for a document in the data stream to contain a term that is not represented in the corpus. In the event that this happens, n_j is defined as 0.

Note that the idea of TF-ICF is similar to the use of training data in classification tasks. However, to the best of our knowledge, in the domain of unsupervised document clustering, no previous research has provided concrete evidences that the document frequency distribution derived from training data set can be successfully used to approximate the document frequency distribution of an unknown data stream. In this work, we present such evidence through extensive experimentations.

4. EFFECTIVENESS EVALUATION

In this section, we evaluate the performance of TF-ICF in the context of unsupervised document clustering. We first discuss the evaluation methodology and performance metrics, then the test data, and finally, we present the experimental results and performance analysis.

4.1 Evaluation Methodology

We evaluate the quality of the document representation generated by TF-ICF in comparison to other commonly used term weighting schemes (listed in Table 1) by examining the quality of the clustering results. We expect TF-ICF to result in document clusters of comparable quality as other popular TF-IDF variations, yet perform significantly faster than the other methods.

Unsupervised document clustering algorithms can be divided into two major categories [13]: partitional algorithms and hierarchical algorithms. The K-Means algorithm and its variations are a family of partitional clustering algorithms. Single link [20], complete link [10], and UPGMA [7] are hierarchical agglomerative algorithms. Although there are many variations on these basic clustering algorithms, in our experiments, we chose the basic K-Means, single link, complete link, and UPGMA as the representative clustering algorithms of their class.

Two commonly used similarity measures include Euclidean distance and cosine similarity and they are defined as follows:

$$EuclideanDistance(X,Y) = \sqrt{\sum (x_i - y_i)^2} \tag{3}$$

$$CosineSimilarity(X,Y) = \frac{X \bullet Y}{\|X\| \times \|Y\|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \times \sqrt{\sum y_i^2}} \tag{4}$$

In order to isolate the impact of different similarity measures on the final clustering result, we ran each clustering algorithm with both the cosine similarity and the Euclidean distance measures.

Entropy is often used to evaluate clusters created by partitional clustering algorithms [3]. Let the correct classification of documents be $C_1 \ldots C_k$, where each C_i is a class. This is typically produced by expert opinion. Next, let $\widetilde{C}_1 \ldots \widetilde{C}_l$ denotes the clusters. In a flat document cluster set generated by a partitional algorithm, each $\widetilde{C}_j$ represents a cluster. The entropy of a cluster $\widetilde{C}_j$ is defined as:

$$Entropy(\widetilde{C}_j) = \sum_{i=1}^{k} -\left(\frac{|C_i \cap \widetilde{C}_j|}{|\widetilde{C}_j|}\right) \log\left(\frac{|C_i \cap \widetilde{C}_j|}{|\widetilde{C}_j|}\right) \tag{5}$$

The entropy of K-clustering $\widetilde{C}_1 \ldots \widetilde{C}_k$ is the weighted sum of the entropies of the clusters [6].

$$Entropy = \sum_{j=1}^{k} \left(Entropy(\widetilde{C}_j) \times \frac{|\widetilde{C}_j|}{N} \right). \tag{6}$$

F-Measure is commonly used to evaluate the quality of the clustering result of a hierarchical clustering algorithm [22]. In a document hierarchy generated by a hierarchical algorithm, each $\widetilde{C}_j$ represents a node of the hierarchy which includes the subset of nodes in the tree below it. The F-Measure of each class C_i is defined as:

$$F(C_i) = \max_{i=1}^{l} \frac{2P_j R_j}{P_j + R_j} \tag{7}$$

Where P is precision and R is recall:

$$P_j = \frac{|C_i \cap \widetilde{C}_j|}{|\widetilde{C}_j|}, R_j = \frac{|C_i \cap \widetilde{C}_j|}{|C_i|}$$

The F-Measure of the clustering is defined as [6]:

$$F - Measure = \sum_{i=1}^{k} F(C_i) \times \frac{|C_i|}{N} \tag{8}$$

4.2 Test Data Sets

To conduct our experiments, we used three different data sets that are commonly used in document clustering research (Table 2): Reuters-21578 [16], SMART [19], and 20 Newsgroups [1].

Reuters-21578 consists of 21578 articles from the Reuters news service. In order to reduce the document set to a manageable size, we chose a subset of the articles according to two criteria: (i) the document belongs to one and only one category and (ii) the value of the attribute "LEWISSPLIT" is "TEST." There are 2349 such documents. We used all the documents from the SMART data set. The 20 Newsgroups data set is a collection of approximately 20,000 documents from 20 different news groups. To reduce the size of this data set to a more manageable one, in alphabetical order, we selected all the documents in the first 12 of the 20 groups. This results in a subset of 4650 documents.

Table 2. Test Data Sets

Data Set	# of Docs	# of Classes	Largest Class	Smallest Class
Reuters	2349	58	1041	1
SMART	3891	3	1460	1033
20 News	4650	12	399	385

In order to conduct a fair comparison, the ICF table used in our experiments was generated from a corpus that only includes LATIMES and FBIS documents. None of the documents from the three test data sets were included in building the ICF table.

4.3 Experimental Results and Analysis

We evaluated the effectiveness of TF-ICF and the five term weighting approaches listed in Table 1: TF-IDF, MI, ATC, Okapi, and LTU. Thus, each test data set has six different document vector representations in accordance to the six term weighting schemes we investigate.

We set up the experiments such that the impact of the term weighting scheme in a document clustering task can be isolated from other factors and therefore, the quality of the clustering result can be used to reflect the effectiveness of the weighting scheme. More specifically, we applied the same clustering algorithm and similarity measure to document vector representations generated by different term weighting schemes. The representation that leads to better quality clusters is considered to be more effective.

4.3.1 K-Means Experiments

This set of the experiments executes the basic K-Means algorithm over the test data. The maximum number of iterations is 100. In the legend, "K" means "K-Means algorithm," "E" means "Euclidean Similarity," "Co" means "Cosine Similarity," "DR" means "Reuters Dataset," "DS" means "Smart Dataset," and "D20" means "20 Newsgroups Dataset."

For each term weighting scheme, we first used the Euclidean distance as the similarity measure. Then, we ran the same experiment with the cosine similarity measure. Because the basic K-Means algorithm is sensitive to the centroids randomly initially selected, we repeated the experiments 10 times with different seed values for the random number generator. Figures 5 and 6 plot the average entropy of each experiment.

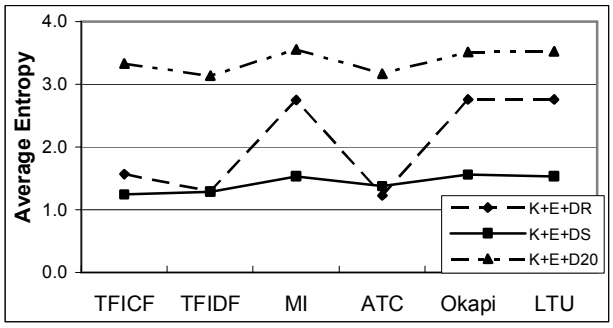

Figure 5. K-Means algorithm using Euclidean Distance.

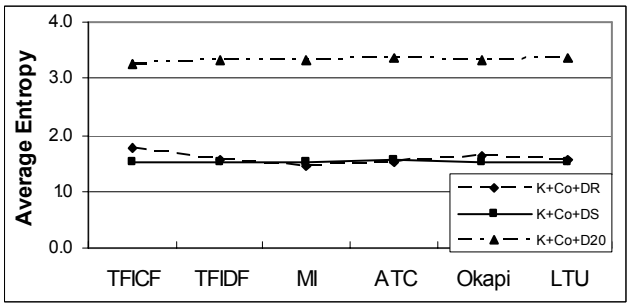

Figure 6. K-Means using Cosine Similarity.

Figure 5 shows that when the Euclidean distance was used as the similarity measure, TF-ICF achieved above average performance in all three test data sets. In the best case, with the Reuters data set, TF-ICF achieved an entropy that is 24% lower than the average. When the cosine similarity was used (Figure 6), TF-ICF performed above average with the SMART and 20 News data sets, but its entropy is 11% higher than the average on the Reuters data.

Two general observations are (*i*) when the same algorithm is executed, the cosine similarity measure leads to more similar clustering results irrespective of the term weighting schemes, and (*ii*) the results of TF-ICF, TF-IDF, and ATC methods are consistent regardless of the similarity measure used in the experiment. In contrast, MI, TF, Okapi, and LTU produce noticeably better results when the cosine similarity measure is used.

4.3.2 Single Link, Complete Link, and UPGMA Experiments

These experiments examine the impact of different term weighting schemes on the clustering quality of hierarchical algorithms. The test data are the same as those used in the K-Means experiments. We first ran each algorithm using the Euclidean distance as the similarity measure. Then, we repeated all experiments with the cosine similarity measure.

Table 3 lists the numerical results. The underlined bold values indicate that TF-ICF achieved above average performance in those experiments. The last row of the table shows the difference between the F-Measure score of TF-ICF and the average F-Measure score. In all six algorithm-similarity measure combinations examined, TF-ICF scored above average in 3 cases. In the worst case, the F-Measure score of TF-ICF is 10% lower than average.

Table 3. Hierarchical Algorithm F-Measure Scores.

(**S**=Single Link, **C**=Complete Link, **U**=UPGMA)

	Euclidean Distance			Cosine Similarity		
	S	**C**	**U**	**S**	**C**	**U**
Min	0.47	0.43	0.47	0.56	0.42	0.63
Max	0.59	0.50	0.69	0.60	0.51	0.71
Avg	0.53	0.48	0.58	0.59	0.46	0.68
TFICF	**0.59**	0.43	**0.63**	**0.59**	0.42	0.63
TFICF vs. Avg	+11.3%	-10%	+8.6%	+0%	-8.6%	-7.4%

5. CONCLUSION

In this paper, we presented a new term weighting scheme, TF-ICF, as a solution to the real-time unsupervised document clustering problem. The principal idea is to use a well-conceived static corpus to approximate information about an unknown document set. Thus, it overcomes the limitations of the existing TF-IDF based approaches in which the document set to be clustered must be known in advance or otherwise, high computational complexity is inevitable.

We examined the effectiveness of TF-ICF in comparison to five other commonly used term weighting methods in the context of document clustering tasks. Experimental results show that TF-ICF achieved above average performance in most cases. In the worst-case scenario, it performed 11% below average, and one of the reasons is attributed to the limited diversity of the corpus from which the ICF table used in the experiments was created. When used for general purposes, we expect TF-ICF to perform even better if the ICF table is generated from a sufficiently large and diverse corpus. When used for a specific application domain, historical data can certainly improve the performance of TF-ICF.

Acknowledgements

Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

6. REFERENCES

[1] 20 Newsgroups. *http://people.csail.mit.edu/jrennie/20Newsgroups*.

[2] Aslam, J., Pelekhov, K., and Rus, D. Static and dynamic information organization with start clusters. In *Proc. of the 7th International Conference on Information and Knowledge Management* (*CIKM'98*), 208-217, 1998.

[3] Beil, F., Ester, M., and Xu, X. Frequent term-based text clustering. In *Proc. of the 8th ACM International Conference on Knowledge Discovery and Data Mining* (*SIGKDD'02*), 436-442, 2002.

[4] Brants, T., Chen, F., and Farahat, A. A system for new event detection. In *Proc. of the 26th ACM International Conference on Research and Development in Information Retrieval* (*SIGIR'03*), 330-337, 2003.

[5] Buckley, C., Singhal, A., and Mitra, M. New retrieval approaches using SMART. In *Proc. of the 4th Text Retrieval conference (TREC-4)*, Gaithersburg, 1996.

[6] Cheng, D., Kannan, R., Vempala, S., and Wang, G. A divide-and-merge methodology for clustering. In *Proc. of the 24th ACM International Conference on Principles of Database Systems* (*SIGMOD/PODS'05*), 196-204, 2005.

[7] Han, J. and Kamber, M. *Data Mining –Concepts and Techniques*. Morgan Kaufmann, 2001.

[8] Jin, R., Falusos, C., and Hauptmann, A.G. Meta-scoring: automatically evaluating term weighting schems in IR without precision-recall. In *Proc. of the 24th ACM International Conference on Research and Development in Information Retrieval* (*SIGIR'01*), 83-89, 2001.

[9] Jones, K.S. and Willett, P. *Readings in Information Retrieval*, Chap. 3. Morgan Kaufmann Publishers, San Francisco, CA, 305-312, 1997.

[10] King, B. Step-wise clustering procedures. *Journal of the American Statistical Association*, *69*, 86-101, 1967.

[11] Luhn, H.P. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Develop, I, 4*, 1957.

[12] Oxford. *http://www.askoxford.com/asktheexperts/faq/aboutenglish/numberwords*.

[13] Pantel, P. and Lin D. Document clustering with committees. In *Proc. of the 25th ACM International Conference on Research and Development in Information Retrieval* (*SIGIR'02*), 199-206.

[14] Porter, M.F. An algorithm for suffix stripping. *Program*, *14*(3), 130-137, 1980.

[15] Project Gutenberg. *http://www.gutenberg.org/etext/2600*.

[16] Reuters-21578 Text Categorization Test Collection v1.0, *http://kdd.ics.uci.edu/databases/reuters21578*.

[17] Salton., G. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989.

[18] Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Journal of Information Processing and management*, *24*(5): 513-523, 1988.

[19]SMART. *ftp://ftp.cs.cornell.edu/pub/smart*.

[20] Sneath, P.H.A. and Sokal, R.R. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. Freeman. London, UK.

[21] TREC-5. *http://trec.nist.gov*, 1999.

[22] Zhao, Y. and Karypis, G. Evaluation of hierarchical clustering algorithms for document datasets. In *Proc. of the 11th International Conference on Information and Knowledge Management* (*CIKM'02*), 515-524, 2002.

[23] Zipf, G.K. *Selective studies and the principle of relative frequency in language*. Harvard University Press, Cambridge, Massachusetts, 1932.

.