Cluster Based Term Weighting Model for Web Document Clustering

B. R. Prakash, M. Hanumanthappa and M. Mamatha

Abstract The term weight is based on the frequency with which the term appears in that document. The term weighting scheme measures the importance of a term with respect to a document and a collection. A term with higher weight is more important than a term with lower weight. A document ranking model uses these term weights to find the rank of a document in a collection. We propose a cluster-based term weighting models based on the TF-IDF model. This term weighting model update the inter-cluster and intra-cluster frequency components uses the generated clusters as a reference in improving the retrieved relevant documents. These inter cluster and intra-cluster frequency components are used for weighting the importance of a term in addition to the term and document frequency components.

Keywords Term weighting scheme · Document clustering · Information retrieval · Data mining

1 Introduction

A document clustering algorithm helps to find groups in documents that share a common pattern [1–5]. It is an unsupervised technique and is used to automatically find clusters in a collection without any user supervision. The main goal of the clustering is to find the meaningful groups so that the analysis of all the documents within clusters is much easier compared to viewing it as a whole collection.

The Vector Space Model (VSM) represents a document using a vector of T unique terms in a collection (T-dimension). Each term in a vector is associated

B. R. Prakash (✉) · M. Hanumanthappa
Department of Computer Science and Applications, Bangalore University, Bangalore, India
e-mail: India.brp.tmk@gmail.com

M. Mamatha
Department of Computer Science, Sri Siddaganga College for Women, Tumkur, India

M. Pant et al. (eds.), *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 259,
DOI: 10.1007/978-81-322-1768-8_70, © Springer India 2014

with a weight (term weight) [6, 7]. The term weight is based on the frequency with which the term appears in that document. The term weighting scheme measures the importance of a term with respect to a document and a collection. A term with higher weight is more important than a term with lower weight. Each document can be located in T-dimensional space, where T is the number of unique terms in a collection (Euclidean space). With a document represented by a location in Euclidean space, we can compare any two documents by measuring the actual distance between them. In the same way, a user-supplied query can be represented as a vector and mapped in Euclidean space. In order to find a set of documents relevant to a query, we can find documents that are closer to this query in Euclidean space. A document ranking model finds the similarities between these documents and a query. If a document is more relevant to a query, it will get a higher ranking. VSM and term weighting schemes are widely used in many research areas such as document clustering, classification, information retrieval, document ranking, etc.

2 Cluster-Based Retrieval

Cluster-based retrieval uses the cluster hypothesis to retrieve a set of documents relevant to a query [8]. Cluster hypothesis Documents in the same cluster behave similarly with respect to relevance to information needs [9]. If a document in a cluster is relevant to a query, then the rest of the documents in that cluster are potentially relevant to the same query. There are two approaches in cluster-based retrieval. The first approach retrieves one or more clusters relevant to a query instead of retrieving documents relevant to a query. In other words, this approach retrieves and ranks the relevant clusters instead of the relevant documents. Based on the cluster hypothesis, the documents from the highly ranked clusters are more relevant to a query than the documents from the clusters with lower ranking. The main motive of this approach is to achieve higher efficiency and faster search.

The second approach uses the generated clusters as a reference in improving the retrieved relevant documents. In this approach, the given document collection is clustered (static clustering) beforehand. When a set of documents is retrieved for a query, the generated clusters (static clusters) of the collection are used as a reference to update the retrieved relevant document list. The main goal of this approach is to improve the precision-oriented searches.

3 Term Weighting Scheme

The Boolean retrieval model assigns 1 or 0 based on the presence or absence of a term in a document. This model performs undesirably in querying for a document. Later, VSM was introduced for ranked retrieval [10]. It is widely used in querying documents, clustering, classification and other information retrieval applications because it is simple and easy to understand. It uses a bag of word approach. Each document d_i in the collection ζ is represented as a vector of terms with weight [11, 12].

One of the most commonly used term weighting schemes, TF-IDF, assigns weights to each term using the term frequency (tf) and inverse document frequency (idf). The term frequency of the $term_t$ is the number of times the given $term_t$ occurs in a document. The inverse document frequency is the total number of documents in a collection containing the $term_t$ with respect to the total number of documents (N) in a collection. Then, the document vector d_i, represented as:

$$d_i = \left\{ term_{1,} tf_{i1} \cdot \log \frac{N}{N_1}; term_{2,} tf_{i2} \cdot \log \frac{N}{N_2}; \ldots \; term_{T,} tf_{it} \cdot \; \log \frac{N}{N_t}; \right\}$$

The term weight w_{it} determines whether the $term_t$ will be included in the further steps. Only certain terms extracted from a document can be used for identifying and scoring a document within the collection. The term weighting schemes are used to assign weight to each term in a document. These term weights represent the importance of a term with respect to a collection. Document clustering uses these term weights to compare two documents for their similarity.

Table 1 shows representation of some of the term weighting schemes commonly used. Here, TF is the Term Frequency, IDF is the Inverse Document Frequency and ICF is the Inverse Cluster Frequency.

w_{itj} is the weight of the term $term_t$ in the document d_i of the cluster C_j.
tf_{if} = fit is the term frequency of the term $term_t$ in the document d_i.
$idf_t = \log \frac{N}{N_t}$ is the inverse document frequency for the term $term_t$ in the collection

where N is the total number of documents in the collection and d_t is the number of documents that contain the term $term_t$.

$icf_t = \log \frac{k}{k_t}$ is the inverse cluster frequency of the term $term_t$ in the collection ζ, where K is the total number of clusters in the collection and Kt is the number of clusters that contains the term $term_t$.

We present a new term weighting method based on the traditional TF-IDF term weighting scheme. Our motivation is based on the idea that the terms of the documents in a same cluster have similar importance compared to the terms of the documents in a different cluster. We concentrated on the terms that are important within a cluster and considered the other terms as irrelevant and redundant. We presented this idea by giving more weight to the terms that are common within a cluster but uncommon in other clusters.

In our new term weighting scheme, we used unsupervised partitional (K-means) clustering algorithms [2, 13, 14]. First, we ran the K-means algorithm with the four term weighting schemes, to show that the CBT term weighting scheme improves the quality of the clusters generated by a partitional clustering algorithm.

Table 1 Term weighting schemes

$\mathrm{Norm} - \mathrm{TFw}_{it} = \frac{f_{it}}{\sqrt{\sum_T f_{it}^2}}$
$\mathrm{TF} - \mathrm{IDE} \quad w_{it} = f_{it} \log \frac{N}{N_t}$
$\mathrm{TF} - \mathrm{IDE} - \mathrm{ICF} \quad w_{itj} = f_{it} \log \frac{N}{N_t} \log \frac{K}{K_t}$

Table 2 List of components in CBT term weighting scheme

tf_{if}	Term frequency component. High when term t occurs often in a document i
idf_t	Collection frequency component. High when term t occurs less of ten in the entire collection
df_{tj}	Intra-cluster frequency component. High when term t occurs more often in a cluster j
icf_t	Inter-cluster frequency component. High when term t occurs less often in clusters other than cluster j

From our experiment, we found that the new term weighting scheme based on the clusters gives better results than the other well-known term weight schemes traditionally used for both partitional and hierarchical clustering algorithms.

3.1 The Proposed Term Weighting Scheme

We introduce our new term weighting scheme. For the term $term_t$, document d_i and cluster C_j, CBT is given as:

$$\begin{aligned} w_{itj} &= tf_{it} \cdot idf_t \cdot df_{tj} \cdot icf_t \\ &= f_{it} \cdot \log \frac{N}{N_t} \cdot \frac{df_j}{|c_j|} \cdot \log \frac{K}{K_t} \end{aligned}$$

Here, $df_{tj} = df_j$

jC_jj is the document frequency of the term $term_t$ within the cluster C_j, where df_j is the number of documents in the cluster C_j that contain the term $term_t$, and jC_jj is the total number of documents in the cluster C_j.

Our new term weighting scheme has four components. The first two components are based on the term weighting components discussed in [15]. The last two components are the cluster components as shown in Table 2. In other words, CBT assigns a weight to a term which is

- Highest when the term occurs more frequently in the documents of a cluster and uncommon in other clusters.
- Higher when the term occurs less frequently in the documents of a cluster and uncommon in other clusters.
- Lower when the term occurs often in a few clusters.

Lowest when the term occurs in most of the documents in a collection.

4 K-Means Algorithm with CBT

Initially, the K-means algorithm doesn't have any information about the cluster components, so we start the algorithm by setting df_{tj} and icf_t to 1 and update the inter- and intra-cluster components on each iteration. If a document has a set of terms that doesn't belong to a cluster, then its term weight will be reduced so that it will move to other clusters. It will be repeated until it finds a suitable cluster of its type.

```
Require: An integer $K \geq 1$, Document Collection $\zeta$
if K = 1 then
  return $\zeta$
else
  Initialize l = 0
  $\{C_1^{(0)}, \cdots\cdots, C_k^{(0)} \leftarrow \text{RANDOM} \quad \text{CLUSTERS}(\zeta, K)$
  repeat
   for all $d_i \in \zeta, i : 1\ldots\ldots N$ do
   $m = \arg\min_j |c_j - d_i|$
   $C_m^{(l+1)} \leftarrow C_m^{(i+1)} \cup d_i$
   end for
   $l \leftarrow l+1$ + 1
    $w_{itj} tf_{it} \cdot idf_t \cdot df_{tj} \cdot icf_t;$
for each term $term_t$ in a document $d_i$ for a cluster
$C_j^{(t)}, t : 1\ldots T, i : 1\ldots N, j : 1\ldots K$
  for j = 1 to K do
 $C_j \leftarrow \frac{1}{|C_j^{(l)}|} \sum_{d_i \in C_j^{(l)}} d_i$
   end for
  until No change in K centroids
  return $\left\{C_1^{(l)}, \cdots\cdots, C_k^{(0l)}\right\}$
end if
```

4.1 Data Collections for CBT

We used the TREC [16], 20 Newsgroup and Reuters-21578 [17] data collections for our experiment. TR11, TR12, TR23, TR31, and TR45 collections are taken from TREC-5, TREC-6 and TREC-7. 20 NG S1–S5 are the five randomly chosen subsets of 20 Newsgroup documents [18]. RE S1 and RE S2 data sets are from Reuters-21578 collection. We got 4645 documents that have only one category. In addition to that, we used the Reuters transcribed subset (RE S2) [19]. For all the data sets shown in Table 3, we removed the stop words and stemmed using the Porter stemming algorithm [20].

Table 3 Data sets

Data set	Collection	No of documents	No of class
TR11	TREC	414	9
TR12	TREC	313	8
TR23	TREC	204	6
TR31	TREC	927	7
TR45	TREC	690	10
20 NG S1	20 newsgroup	2,000	20
20 NG S2	20 newsgroup	2,000	20
20 NG S3	20 newsgroup	2,000	20
20 NG S4	20 newsgroup	2,000	20
20 NG S5	20 newsgroup	2,000	20
RE S1	Reuters-21578	4,645	59
RE S2	Reuters-21578	200	10

Table 4 K-means clustering algorithm—avg. Entropy measured for norm TF, CBT, TF IDF ICF and TF IDF term weighting schemes

Data sets	Term weighting schemes			
	Norm TF	TF-IDF	CBT	TF-IDE-ICF
TR11	0.8413	0.7905	0.8535	0.7749
TR12	0.9009	0.6834	0.6139	0.6261
TR23	1.0424	0.9246	0.839	0.8501
TR31	0.9379	1.2781	0.9657	1.0822
TR45	1.0787	1.3469	1.0443	1.1485
20 NG S1	2.4824	0.4037	0.2995	0.4475
20 NG S2	2.4954	0.5791	0.4164	0.801
20 NG S3	2.4727	0.9366	0.5082	0.7886
20 NG S4	2.4923	0.3138	0.2845	0.521
20 NG S5	2.7464	0.2983	0.3274	0.5665

Table 5 Bisecting K-means clustering algorithm—avg. Entropy, avg. F-measure and avg. Purity measured for the TF-IDF and CBT term weighting schemes

Data source	CBT			TF-IDF		
	Avg entropy	Avg F-measure	Avg purity	Avg entropy	Avg F-measure	Avg purity
TR11	1.3435	0.2067	0.5039	1.4102	0.2478	0.485
TR12	1.548	0.2256	0.3936	1.7344	0.1946	0.3514
TR23	1.3337	0.1803	0.4853	1.3351	0.1719	0.4853
TR31	1.2539	0.1473	0.515	1.4105	0.1407	0.4344
TR45	1.507	0.3003	0.4404	1.5922	0.2627	0.421
RE S1	2.0039	0.0504	0.414	2.0061	0.0519	0.4137
RE S2	1.9169	0.2663	0.2764	1.9981	0.2444	0.2518
20 NG	2.8548	0.09863	0.1079	2.2575	0.1894	0.2141

5 Conclusion

Importance of using inter- and intra- cluster components in the term weights using the average entropy measure. Since the K-means clustering algorithm is unstable and sensitive to initial centroids, we ran the algorithm 10 times with different random seed for the initial centroids on each run. We repeated this experiment for the four term weighting schemes on the data collections listed in Table 3.

We calculated the entropy for the term weighting schemes, as given in Eq. (2.13), for each run after the algorithm converged. Then, we computed the average of the entropies obtained in each run. Similarly, we computed the average F-measure and average purity measures. Table 4 shows the average entropy calculated for each data set. Table 5 shows the average entropy, average F-Measure and average purity measured for the TF-IDF and CBT term weighting schemes for the K-means clustering algorithm. Both experiments show that the results obtained from the K-means clustering algorithms with the CBT term weighting scheme have shown better results compared to the other term weighting schemes on each data set. According to the cluster-based term weighting scheme, a term is considered important to a cluster if it is unique to that cluster and occurs frequently within the documents of that cluster. The inter- and intra- cluster components try to identify these important terms by analyzing the term frequency distribution at three levels: document, cluster and collection. And our experimental results have shown that adding these cluster components in the term weighting scheme significantly improves the results on each data set. We believe that some of the deviations in the results are due to the clustering algorithms' lack of handling the noise in the data collection.

References

1. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, vol. 8, 1st edn. Addison-Wesley, Boston (2005)
2. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining, vol. 400, pp. 525–526. Department of Computer Science and Engineering University of Minnesota, Citeseer (2000)
3. Guha, Sudipto, Rastogi, Rajeev, Shim, Kyuseok: CURE: an efficient clustering algorithm for large databases. ACM SIGMOD Rec. **27**(2), 73–84 (1998)
4. Zhao, Ying, Karypis, George, Fayyad, Usama: Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Disc. **10**(2), 141–168 (2005)
5. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: a cluster-based approach to browsing large document collections. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92), pp. 318–329. ACM Press, New York (1992)
6. Chisholm, E., Kolda, T.G.: New term weighting formulas for the vector space method in information retrieval. Technical report (1999)
7. Singhal, Amit, Buckley, Chris, Mitra, Mandar, Mitra, Ar: Pivoted document length normalization, pp. 21–29. ACM Press, New York (1996)

8. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04), pp. 186–193. ACM, New York (2004)
9. Voorhees, E.M.; The cluster hypothesis revisited. In: Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '85), pp. 188–196. ACM, NewYork (1985)
10. Salton, Gerard: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Boston (1989)
11. Cummins, Ronan, O'Riordan, Colm: Evolving general term weighting schemes for information retrieval: tests on larger collections. Artif. Intell. Rev. **24**, 277–299 (2005)
12. Jung, Y., Park, H., Du, D.Z.: An effective term weighting scheme for information retrieval (2000)
13. Manning, C.D., Raghavan, P., Schutze, H.: Introduction to information retrieval, Chapter 16, p. 496. Cambridge University Press, Cambridge (2008)
14. David MacKay, J.C.: Information Theory Inference and Learning Algorithms. Cambridge University Press, Cambridge (2002)
15. Salton, Gerard, Buckley, Christopher: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)
16. Text REtrieval Conference (TREC) (1999)
17. Lewis, D.D.: The reuters-21578 text categorization test collection (1999)
18. Zhou, X., Zhang, X., Hu, X.: Dragon toolkit: incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), pp. 197–201. IEEE (2007)
19. Hettich, S., Bay, S.D.: Reuters transcribed subset (1999)
20. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
21. Murugesan, K.: Cluster-based term weighting and document ranking models. kentucky (2011)