IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4922

Improve the Query Hit List Precision by Documents Clustering Technique

Ciya Liao
Oracle Corporation, david.liao@oracle.com

Shamim Alpha
Oracle Corporation, shamim.alpha@oracle.com

Paul Dixon
Oracle Corporation, paul.dixon@oracle.com

ABSTRACT

We propose a new approach to improve query hit list precision in document information retrieval. We use the k-mean clustering technique to group returned hit list documents. The relevancy of each cluster is evaluated according to document relevancy scores in the clusters. The final relevancy score of each document is a combination of the relevancy score of cluster and individual document. To form clusters with features more related to the query, we use pseudo-feedback documents to construct a latent semantic index (LSI), which transforms all the documents in the hit list into LSI feature vectors. Feature vectors constructed with relevant features are input to the clustering algorithm. We show that LSI based on relevant documents can improve the hit list cluster coherence significantly, in the sense that clusters group query relevant and irrelevant documents separately. We also show that the improved cluster quality, which results to better separation between relevant and irrelevant documents, can be used to improve the precision of a query hit list significantly.

INTRODUCTION

Conventional search engines suffer from the problem of low precision and long hit list. Conventional search engines return long lists of documents ranked by the probabilities of relevance to the query. One reason for low precision is that relevance of a document is based only on the occurrence of the query or expanded query terms inside the document without using any contextual information surrounding the query terms. For example, polysemy of the query term “bank” causes returning documents about bank of rivers when searching for bank services. This paper‘s technique will recalculate the relevance of each document in the query hit list based on context information beyond query terms. In particular, we employ a clustering technique to group documents in the hit list. The criteria used to group documents are features related to but beyond query terms. The relevance re-rank of each document in the hit list takes into account two types of information: the conventional information retrieval relevance score and the grouping characteristics of documents.

Document clustering technique has been used to find connections between documents. Given a collection of documents, document clustering forms groups of documents such that documents within one group are more similar with each other than documents across different groups. Apropos of information retrieval, a Cluster Hypothesis was introduced by van Rijsbergen (van Rijsbergen, 1979), stating that closely associated documents tend to be relevant to the same request. According to this hypothesis, the probability that a document is relevant is increased by knowledge that there is a relevant document within the same cluster, but decreased by an irrelevant document’s sharing the same cluster. Based on this hypothesis, early works have been focused on improving information retrieval by clustering the basic collection (Jarding, 1971, Voorhees, 1985).

Recently, document clustering has been applied during query processing (Cutting, 1992; Zamir, 1998; Leuski, 2001). Scatter/Gather provides a browse method to narrow down candidate documents from a collection interactively (Cutting, 1992). For each step, users are allowed to choose interesting clusters based on their query need; the resultant sub-collection is then clustered into more refined sub-clusters. Scatter/Gather clustering has been applied to query retrieval results (Hearst, 1996). This study found that most relevant documents tend to fall into one “best” cluster and choosing this best cluster for a ranked documents list yield better precision than then original ranked list at the same cutoff point. In another paper (Zamir, 1998), the above conclusion of the existence of a best cluster was also reached by using different clustering techniques for web page hit list clustering. Leuski clustered the documents in a hit list and re-ranked the documents based on clustering associations and manual relevance feedback of previously returned documents (Leuski, 2001).

Because of the heterogeneity of the documents in the hit list, a cluster is formed sometimes based on the content totally irrelevant to the query. This mixes the relevant and irrelevant documents in one cluster and affects the quality of clusters. We solve this problem by using a pseudo-feedback technique that has been shown to improve precision in TREC ad hoc tracks (Robertson, 1999). We know in practice that the documents at the top of a hit list will have high probability of being relevant. We then use these documents to build a latent semantic index (LSI) model. The LSI model is used to transfer the features in the original space into those in the space that only is related to the relevant documents hence the query itself. We believe that clustering based on these transferred features can improve clusters’ quality hence hit list ranking precision.

LATENT SEMANTIC INDEXING

LSI is the application of Singular Value Decomposition (SVD) to document-term matrices in information retrieval (Manning, 2000). The SVD decomposes the document-term matrix into the product of three matrices. Let **A** denotes the document-term matrix with m rows and n columns, element a_{ij} is the weight of the i-th term in the j-th document. Each column $\mathbf{a}_j$ is the feature vector of j-th document. The SVD of matrix **A** is:

$$A_{mn} = U_{mk}L_{nk}V_{nk} \qquad k=n \qquad \text{eq.(1)}$$

Where the matrices **U**, **V** are orthogonal in the sense that their columns are orthonormal, that is to say $\mathbf{U}^T\mathbf{U}=\mathbf{V}^T\mathbf{V}=\mathbf{I}$, the matrix **L** is diagonal. Multiplying the transpose matrix of **U** to the both side of eq.(1), one can get: $\mathbf{U}^T\mathbf{A}=\mathbf{LV}$. $\mathbf{U}^T\mathbf{A}$ is the transformed matrix of original matrix **A** by transforming matrix **U**. The transformed matrix has size k x n, each column is the transformed feature vector of each document.

This conference paper appears in the book, Innovations Through Information Technology, edited by Mehdi Khosrow-Pour.

It is easy to prove from eq. (1) and orthonormality of matrix **U** that **$(U^TA)^T(U^TA)=A^TA$**, which says that the transformation maintains the same vector length of feature vectors and the same dot product between any pair feature vectors or similarity if they have been normalized to unity length.

The dimensionality reduction of LSI is accomplished by approximating the eq. (1) with k<n.

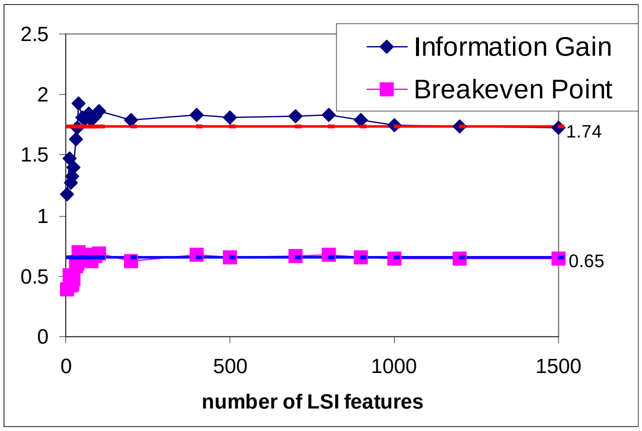
$$A_{mn} » U_{mk}L_{nk}V_{nk} \qquad k<n \qquad \text{eq. (2)}$$

Where we rank the diagonal elements in the diagonal matrix as descending order, and only pick k largest values. In this case, the transformed matrix $\mathbf{(U^TA)}_{kn}$ is composed of n feature vectors of size k (k<n). The approximation of eq.(2) is in the sense that the 2-norm distance between the matrix $\mathbf{A_{mn}}$ and $\mathbf{U_{mk}L_{nk}V_{nk}}$ is minimized.

LSI has been applied to information retrieval by transforming both the original documents and the queries to discover co-occurrence of terms (Deerwester, 1990). LSI provides a mechanism to wrap and transform original features represented by terms into new features with reduced dimensionality, which has been used in text categorization (Schutze, 1995).

In Fig. 1, we show the k-mean clustering result based on LSI features. Here, we only use the top 25 most populated predefined classes in of Reuters-21578 collection (Yang,99), and documents assigned to multiple classes have been deleted. The number of documents in the collection to be clustered is about 9200. We report two types of measures for cluster quality evaluation. One is the information gain (Bradley, 1998), which estimates the amount of information gained by clustering the collection as measured by reduction in class impurity within clusters. Before clustering, the original collection entropy is calculated as the entropy of class distribution. For a collection with L known classes, let C^l be the number of documents in class l where l=1,…,L. Let N be the total number of documents in the collection. The total entropy of the collection before clustering is: $\text{OriginalEntropy}_0 = -S_l\, C^l/N \log(C^l/N)$. After clustering, the collection is partitioned into sub-groups (clusters) with each group having more purity of class membership than the original whole collection. Let CS_k be the number of documents in cluster k, C_k^l be the number of documents in class l within cluster k. We can calculate the entropy for each cluster as: $\text{Entropy}(k) = -S_l\, C_k^l/CS_k \log(C_k^l/CS_k)$. The whole entropy of the collection after clustering is then the weighted averaged entropy of each cluster: $\text{Entropy} = S_k\, CS_k/N\, \text{Entropy}(k)$. The information gain is the entropy decrease of the collection by clustering: $\text{Information Gain} = \text{OriginalEntropy}_0 - \text{Entropy}$. The other measure of cluster quality is the breakeven points. We artificially assign all documents in one cluster to the class that has the largest number of documents in this cluster. The document assignment rules construct a classifier, and the breakeven points, where the precision and recall is equal, convey the information of the class impurity within clusters. Fig 1. shows consistent variation of these two measurements.

Figure 1: The information gains and breakeven points for clustering reuters-21578 collection vs a different number of LSI features. The information gain and breakeven points of clustering using original features are shown as horizontal lines.



In the original feature space, we represent each document with a feature vector. The feature vector dimensionality is 3000. We first perform kmean clustering based on the original feature space and find the information gain is 1.74 and the breakeven point is 0.65. We then employ eq. 2 to decompose the original document-feature matrix with different k values (number of LSI features). For each k value, we perform kmean clustering for the transformed feature vector of size k, and get information gain and breakeven point. In Fig. 1, we see the information gains and breakeven points will approach to the values of the original feature space. This is because as k increases and approaches the original dimensionality n, eq. 2 becomes more accurate. When eq. 2 becomes eq.1, the kmean clustering which calculates the dot product as document similarity should give exactly the same result in the original or transformed feature space.

From Fig.1, we also find that as k is very small (about 10 to 20), the cluster quality based on LSI features is better than that based on the original space. This finding confirms that reduction of dimensionality by LSI does not worsen the cluster quality, but improve it.

GROUPS OF DOCUMENTS IN THE HIT LIST

According to the Cluster Hypothesis (van Rijsbergen, 1979), documents associated with the same cluster tend to be relevant for the same query. If we consider a hit list of documents as a collection with two classes (relevant and irrelevant), then clustering partitions on this collection will have positive information gain. We clustered the hit list of TREC-8 adhoc queries. Each query has relevancy judgments supplied by NIST (Voorhees, 1999). The hit lists are formed by converting the TREC topics 401-450 to queries which are executed against the TREC-8 collection. Our information retrieval system returns the top 1000 documents for each query. We delete documents in the hit list whose relevancy has not been judged. We consider two types of queries: short and long query. The short query is constructed from only the query title, while the long query is constructed from the query title, description, narration and their stemming terms. For the short query, kmean clustering improves the relevance purity by 14.4% information gain compared to original entropy averaged for 50 topics.

By carefully examining the formed clusters, we found that some clusters are formed because the documents in the cluster share some completely unrelated terms with the original query. This phenomenon that relevant and irrelevant documents are grouped to the same cluster because of irrelevant common terms tends to make the Cluster Hypothesis shaky. One way to improve cluster quality and hence enlarge the separation of relevant and irrelevant documents is to only use relevant terms as features in clustering. To distinguish relevant or irrelevant terms, we need document relevance feedback. Once we have relevant documents, we can assume that the terms or the major representing terms in relevant documents are relevant. We use the LSI equation to wrap those relevant terms. In specific, we only apply LSI eq. 2 to the document-term matrix that only includes the relevant documents. The derived transfer matrix **U** can then be used to transfer each document's feature vector in the hit list to LSI feature space. This process can be viewed as projecting each original feature vector to a sub-space that is spanned only by relevant features. In Fig. 2 we report an experiment result by clustering short query hit lists using LSI features derived by using relevant feedback. The averaged information gain across 50 topics is almost doubled compared to that of using original feature vectors. Of 50 topics, more than 40 topics' hit lists have improved cluster quality or relevant-irrelevant document separation, and only about 5 topics become worse. A surprise finding is that the cluster quality reaches a plateau when the number of LSI features is larger than three, which may mean that the relevant subspace is of very low dimensionality.

Fig. 2 demonstrates that the effectiveness of the idea of applying relevant document based LSI to derive relevant features for clustering hit lists. In reality, relevance judgments in hit list are not available. We then use the pseudo-feedback technique to derive the relevant features by LSI just based on the documents at the top of the hit list. The pseudo-

Figure 2: The quality of clustering the hit list documents of short queries of the TREC-8 adhoc collection by using LSI features based on relevance feedback. The LSI transfer matrix is calculated based on relevant documents. The LSI features of each document are derived by applying the transfer matrix to original feature vectors. The horizontal line is the averaged information gain of clustering with original features. The information gain is reported as a percentage.

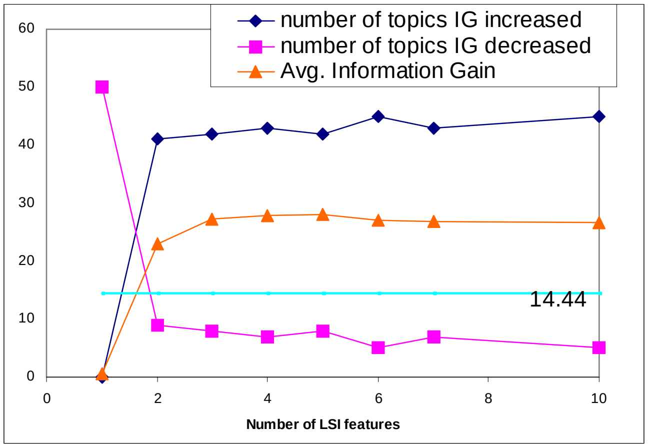


Table 1: The clustering quality of clustering hit list documents of the TREC-8 adhoc collection by using LSI features based on pseudo-feedback. The LSI transfer matrix is calculated based on the top 20% of documents in the hit list. The LSI features of each document are derived by applying the transfer matrix to original feature vectors. The number of LSI features for both cases is fixed at 10.

queries	Number of topics with IG increased	Number of topics with IG decreased	Avg. information gain (%)	Avg. information gain in original space (%)
Short queries	36	14	18.04	14.44
Long queries	32	17	20.86	15.16

feedback technique has been used to improve hit list precision in adhoc queries (Buckley, 1996; Kwok, 1998), where the top documents in the hit list are used to suggest terms or term weights to modify the original queries. In theory and practice, the top documents in the hit list have high probability of being relevant. In the absence of relevance judgments, those top documents can be treated as relevant. In Table 1, we report the cluster quality based on pseudo-feedback. The documents used in building the LSI transfer matrix are the top 20% of documents in the hit list. The LSI feature number is fixed at 10. We see that in short queries, the improved information gain is not as much as shown in Fig. 2. That is expected as the pseudo-feedback documents are not all relevant. However, the information gains are still improved by 25 % for short queries and 38% for long queries.

RE-RANKING THE HIT LIST

We believe that clustering of the hit list, which separates relevant and irrelevant documents, can be used to improve the precision of the hit list. From query execution we have document relevance score returned by information retrieval system, we call them IR scores. For the hit list clusters derived by LSI features from pseudo-feedback documents, we define a cluster relevance score as a combination of overall IR scores of documents inside the cluster and IR scores of documents near the centroid of the cluster: clusterRelevantScore = ((average doc IR scores in cluster) + A*(average doc IR score near the centroids))/(1+A), where A is a predefined weight.

The cluster relevance score conveys two types of information: the IR relevance of documents inside the cluster and connections between documents inside the cluster. The cluster relevance score is the common characteristics related to the original query shared by the documents inside the same cluster. Since the separation of the relevant and irrelevant documents could be accomplished by clustering the hit list based on LSI features, a cluster having many relevant documents will have high cluster relevance score and a cluster having few relevant documents will have a much lower cluster relevance score.

Table 2: The improvement of hit list of TREC-8 adhoc queries through kmean clustering based on LSI features. The LSI features are derived from the top documents in the original hit list.

Parameter A	Parameter C	Avg.precision@ 10 improvement (%)for short query	Avg. precision @10 improvement (%)for long query
1	0.7	0	40.9
0.5	0.3	4.8	8.9
0.5	0.2	7.1	12.3
0.5	0.1	4.5	11.1

The special consideration of the documents near cluster centroids is to take care of the case where some clusters have only a few irrelevant documents but these documents have high IR scores. In this case, the clusters will have much higher cluster relevance score than clusters that have relevant and irrelevant documents mixed due to averaging effect. Increasing the weight parameter A will make the cluster relevance score calculation biased to the clusters in which there are relevant documents of high IR score near the cluster centroids, hence increase the relevance score of these clusters.

The hit list returned by the conventional information retrieval system is sorted according to document IR scores. After the hit list is clustered, we associate each cluster with the cluster relevance score. The documents in the hit list then could be re-ranked by a combination score of IR scores and cluster relevance score. We use a linear combination of those two types of score: $\text{RelevantScore}_d = (\text{IRScore}_d + C*\text{ClusterScore}_d)/(1+C)$, where C is a predefined weight, ClusterScore_d is the cluster relevance score of the cluster to which the document d is assigned.

We report the result of hit list re-ranking of short and long queries of the TREC-8 adhoc collection in table 2. We only report the average precision at 10 as the evaluation of the hit list. The average precision @10 improvement in table 2 is the percentage of the improvement compared with the average precision @10 of the original hit list. The average precision reported in table 2 for each type of query is the average value for 50 topics.

Table 2 shows significant improvement both for short and long queries. In our IR system, the average precisions of the original hit lists for both short and long query are similar. However, the magnitudes of the improvement for short and long queries are different. The long query is improved from 8.9% to 40.9%, while the short query is improved from 0% to 7.1%. We also see big variations of the hit list quality according to variations of parameters A and C.

The difference in improvement between short and long queries may be due to the fact that the documents in the hit list for long queries share more terms than those for short queries. The clustering technique employs connections among documents to improve the precision of the hit list. Thus, if there are more common terms or connections among documents, the clustering technique will produce more benefits. That difference between short and long queries can also be seen from table 1, where clustering improves the information gain by 25 % for short queries and 38 % for long queries.

CONCLUSIONS

We have shown that clustering technique can be used to separate the relevant and irrelevant documents in the hit list returned by conventional information retrieval search engines. We have also shown that latent semantic indexing can be used to form more coherent clusters with features more related to the query, when the latent semantic index is built by the documents with large probabilities of being relevant to the query. We defined the relevance of clusters according to documents relevant scores in the clusters and cluster centroids. We then re-ranked the original hit list based on the combination of the document's IR score and cluster relevance score. The re-ranked hit list shows an improvement in precision compared to original hit list.

REFERENCES

Van Rijsbergen, C. J. (1979), Information Retrieval, Butterworths, London, 1979, Second Edition.

Jardine, N. and van Rijsbergen, C.J.,(1971) The Use of Hierarchy Clustering in Information Retrieval. *Inform. Stor. & Retr.,* **7**, 217-240.

Voorhees, Ellen M.,(1985), The Cluster Hypothesis Revisited. *Proceedings of the Eight ACM SIGIR,*1985 188-196.

Cutting, D.R. Karger, D.R., Pederson, J.O. and Tukey J.W., (1992), Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. 15th Annual Int'l ACM SIGIR.*

Zamir, O. and Etzioni, O.,(1998), Web Document Clustering: A Feasibility Demonstration, in *Proc. 21st Annual Int'l ACM SIGIR Conferenc*e.

Leuski, A., (2001), Evaluating Document Clustering for Interactive Information Retrieval, in *Proc. Of the 10th International Conference on Information and Knowledge Management, 2001*.

Hearst, M.A. and Pederson, J.O.,(1996), Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *Proceedings of ACM SIGIR'96*.

Robertson, S.E. and Walker S.(1999), Okapi/Keenbow at TREC-8, In *Proceeding of TREC-8, 1999*.

Voorhees, E.M., Harman, D., (1999), Overview of the Eighth Text Retrieval Conference (TREC-8). In *Proceedings of TREC-8*

Manning, C.D. and Schutze H., (2000), Foundations of Statistical Natural Language Processing, the MIT Press, 2000.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, t.k., and Harshman, R.,(1990) Indexing by latent semantic analysis, *Journal of the American Society for Information Science,* 41(6):391-407.

Schutze, H., Hull, D.A, and Pederson, J.O.,(1995), A Comparison of Classifiers and Document Representations for the Routing Problem, *Proc. of SIGIR'95,* 1995**.**

Yang Y.(1999),An evaluation of statistical approaches to text categorization.*Journal of Information Retrieval*, Vol.1, No.1/2, 1999.

Bradley, P.S. AND Fayyad, U.M., (1998), Refining Initial Points for K-Means Clustering, in *Proc. 15th International Machine Learning*, 1998, Morgan Kaufmann, San Fransisco, CA.

Buckley, C, Singhal, A, Mitra, M. & Salton, G., (1996). New retrieval approaches using SMART. In *Proceedings of TREC-4*.

Kwok, K.L. and Chan, M., (1998), Improving Two-Stage Ad-hoc Retrieval for Short Queries." In the Proceedings of the 21th ACM SIGIR,1998.