Indian Journal of Science and Technology, *Vol 8(15), 46754, July 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Improving Weak Queries using Local Cluster Analysis as a Preliminary Framework

Amir H. Jadidinejad[1*] and Hossein Sadr[2]

[1]Computer and IT Engineering Faculty, Islamic Azad University, Qazvin Branch, Qazvin, Iran;
amir.jadidi@qiau.ac.ir
[2]Young Researchers and Elite Club, Lahijan Branch, Islamic Azad University, Lahijan, Iran;
Sadr@raiau.ac.ir

Abstract

In a web retrieval task, the query is usually short and the users expect to find the relevant documents in the first several result pages. To address this issue, the possibilities of using Local Cluster Analysis as a preliminary framework with the intention of improving the effectiveness of weak queries by clustering search results and creating high-precision retrieval is explored in this paper. Moreover, employing this notion makes our approach an apt choice to be embedded in other applications such as Pseudo Relevance Feedback that requires high-precision results and cannot be applied on weak queries currently.

The clustering method is notably an important part in our approach. Therefore, the problem of creating effective and meaningful clusters is addressed in this paper and different well-known and state-of-the-art clustering methods are evaluated in order to achieve superior efficiency and effectiveness in the proposed approach. Consequently, various experiments are conducted to evaluate the impact of the proposed architecture and different clustering variants in large Persian text collection created based on TREC specifications. Furthermore, extensive experiments results present promising improvements over existing measures that emphasize on weak queries.

Keywords: Local Cluster Analysis, Persian Text Retrieval, Query-Specific Clustering, Search Result Clustering, Weak Queries

1. Introduction and Motivation

In web retrieval tasks, the number of terms in a query is usually small (two to three on average)[40]. According to[8], if the terms do not provide enough information of the user's need, the retrieval result may be poor. These are known as weak queries[24]. Moreover, the relevant documents are likely to be scattered along the retrieval list.

In this case, users of retrieval systems are often forced to spend a lot of time to sift through a diversity of the results. As it is clear, users are confronted with various complexities to find the specific information that they are looking for.

Although information retrieval research has been concerned with improving the effectiveness of retrieval in some applications, such as Pseudo Relevance Feedback, a more specific requirement exists for high-precision retrieval[39,25]. Pseudo Relevance Feedback is a well-known method for improving retrieval effectiveness. Whereas, it is based on the assumption that top retrieved documents are relevant, it may actually harm the performance when the initial retrieval's top ranked documents are irrelevant.

Some previous works have been done to address the issue of weak queries in information retrieval[1,14,24,25,39,47,48]. In this context, a simple high-precision information retrieval system is introduced by clustering and re-ranking search results with the intention of eliminating these shortcomings without using any external evidence, complicated algorithms and etc. The proposed architecture has some key features:

- Simple and high performance. Our experimental results (Section 4) present that the proposed method performs better than the best known standard Persian retrieval systems[3,4,16].

*Author for correspondence

- Independent of initial system architecture. It can be embedded in any fabric information retrieval system. Moreover, it causes the proposed architecture to be able to envisage the web search engines[2].
- High-Precision. Relevant documents are exhibited at top of the result list. Therefore, the proposed method can be used in applications that need high-precision such as Pseudo Relevance Feedback.

The rest of the paper is organized as follows. In Section 2, related works in close domains are presented. The architecture of the proposed method and experimental details are outlined in Section 3 before introducing used dataset and evaluating the algorithm's performance in Section 4. Conclusion is given in Section 5.

2. Related Works

Although using some kind of documents clustering technique to help improving retrieval results is not new field of research, the proposed architecture to the best of our knowledge is the first method that explicitly presents and deals with the low-precision and weak queries problems in terms of clustering search results and re-ranking.

Document clustering can be performed on the collection as a whole (static clustering)[1,10,26,32,37], but post-retrieval document clustering (dynamic clustering) has shown that can produce superior results[15,31,42]. Tombros et al.[42] conducted a number of experiments using five document collections and four hierarchical clustering methods to show that if hierarchic clustering is applied to search results (query-specific clustering), then it has the potential to increase the retrieval effectiveness compared to both of static clustering and conventional inverted file search. The actual effectiveness of hierarchic clustering can be gauged by Cluster-based retrieval strategies, which perform a ranking of clusters instead of individual documents in response to each query[22,42]. Furthermore the generation of precision-recall graphs is not possible in such systems, and in order to derive an evaluation function for clustering systems some effectiveness function was proposed by[22]. The formula for the measure is given by 1 - where P and R correspond to the standard definitions for precision and recall (over the set of documents of a specific cluster), and is a parameter that reflects the ratio of importance attached to precision and recall[22,42,]. In this paper, Firstly, simple architecture is proposed which uses query-specific clustering to improve the effectiveness of retrieval and utilizes traditional precision-recall evaluation and ranks list presentation instead of cluster-based retrieval. Secondly, this paper is devoted to high-precision retrieval to improve weak queries tremendously. Thirdly, larger Persian standard test collection is employed which is created based on TREC specifications that validate[42] findings in a wider context. Lastly, Tombros et al.[42] believe that partitioning methods have some limitation to handle query-specific clustering and prefer to use hierarchical methods. On the other hand, our experimental results (Section 3) revealed that simple partitioning methods such as k-means has a great potential in query-specific clustering especially that partitioning methods need low computational requirement than hierarchical methods. Whereas in our architecture clustering step is overhead, using simple methods instead of complicated ones is preferred.

Query expansion is another approach that can improve the effectiveness of information retrieval. These techniques can be categorized as either global or local. While global techniques rely on analysis of a whole collection to discover word relationships, local techniques emphasize on analysis of the top-ranked retrieved documents for a query[29,47.] Furthermore, local techniques have shown to be more effective than global techniques in general[4,47–49]. Consequently, the proposed architecture is called Local Cluster Analysis (LCA)[18,19,29,49] in contrast to Local Context Analysis because clusters are analyzed instead of contexts.

Many research efforts such as[13,46] have been made on how to solve the keyword barrier which exists because there is no perfect correlation between matching words and intended meaning[13] presents TermRank, a variation of the PageRank algorithm based on a relational graph representation of the content of web document collections. Search Result Clustering has successfully served this purpose in both commercial and scientific systems[1,9,15,27,32,38,41,43,50,54]. The proposed methods focus on separating search results into meaningful groups and user can browse and view retrieval results. One of the first approaches for searching results clustering called Suffix Tree Clustering would group documents according to the common phrases[50,51]. STC has two key features: the use of phrases and a simple cluster definition. This is very important when attempting to describe the contents of a cluster. Proposes a new approach for web search result clustering to improve the performance of methods that use the previous STC algorithms[21]. Search Results Clustering has a few interesting characteristics and one of them is the fact that it is only based on document snippets. Certainly, document snippets returned by search engines are usually very short and noisy. Another shortage of these systems is the cluster's name. Cluster's name must describe the contents of the cluster accurately and concisely, so that the user can decide quickly if the cluster is interesting or not. This aspect of these systems is difficult and sometimes neglected[41,54]. In this context our tendency is to provide very simple high-precision system based on cluster hypothesis[44] without any user feedback.

Web assistance and data fusion methods[2,24] have been employed to address the issue of weak queries in IR. These approaches probe a web search engine to form new queries, and then combine the corresponding retrieval lists. These approaches are efficient but need external corpus or auxiliary search engines.

Lee et al.[26] published a "re-ranking model based on document clusters". Their goal is the same as our first motivation (improve the effectiveness of retrieval) and they also use a hierarchical clustering method to identify and classify results. There is significant difference between our approaches. Lee et al.[26] applies static hierarchical agglomerative clustering to the set of whole documents and view clusters dynamically depending on retrieval results in the initial ranking. On the other hand, they used static clustering and dynamic view. This approach has two disadvantages: First, since the data sets are mostly dynamic, pre-computed clusters would have to be constantly updated, and most clustering algorithms cannot perform incrementally. This would require a huge amount of resources and it has been shown that such an approach results in clusters of lower quality[42]. Second, after the costly partitioning step, the results of cluster partitioning can contain documents which are not in the result of the first step. These have a negative effect on cluster centroid for a query. They need to adjust the value of cluster centroid to minimize the negative effects. Moreover most experiments in[26] are about evaluating basic retrieval methods such as inverted file and different weighting schemes; our focus is primarily to improve cluster analysis and precision without any heavy solution.

3. System Architecture and Experimental Details

Retrieval systems generally look at each document as a unique one in assigning a page rank. If the document is viewed as a combination of other related documents in the query area (local context), better results can be obtained. The conjecture that relevant documents tend to cluster was made by[44].

Irrelevant documents share many terms with relevant documents in the local context but about two completely different topics, these may demonstrate some patterns. On the other hand, an irrelevant cluster can be viewed as the retrieval result for different queries that share many terms with the original query. This implication is more important in the weak queries.

Xu et al.[46,48] believe that document clustering can make mistake and when this happens, it adds more noise to the query expansion process. Moreover, as it is discussed in Section 3.4, the proposed architecture exploits document clustering with a conservative fusion method for high-precision information retrieval systems and weak queries.

In this context, this architecture (Figure 1.) and experimental details are proposed to cluster search results and re-rank them based on cluster analysis. Although Persian

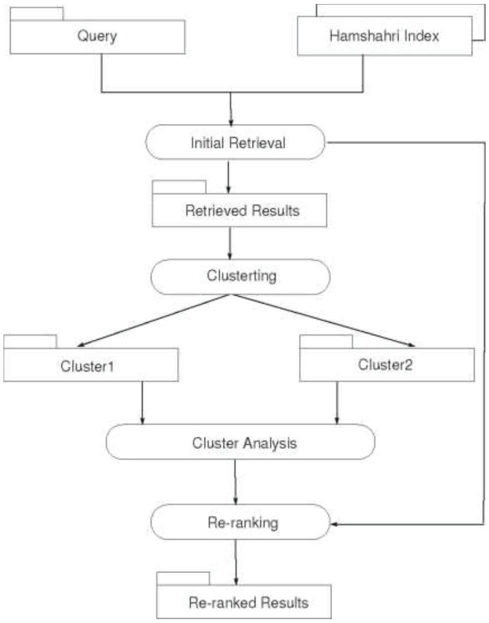

Figure 1. System architecture.

language benchmark dataset is used, it is almost clear that the same results must be exhibited in other benchmarks because it seems that the proposed architecture is independent of the language.

3.1 Document Collection and Initial Retrieval

At the initial retrieval step, documents based on the query-document similarity are retrieved. As matter of fact, he focus at this retrieval step is on each document. The initial retrieval step ranks the retrieved documents in decreasing order of query-document similarities[42] suggest that there is not a statistically significant variation in query-specific cluster effectiveness for different values of top-ranked documents so the top-100 documents for each query can be used.

First initial results retrieve per query based on standard method (Section 3.1), and then clustering is applied on initial results and separates it into two clusters (Section 3.2). After the clustering step, we have to choose relevant cluster (Section 3.3) and then re-ranked results based on it (Section 3.4).

The Persian language is one of the dominant languages in Middle-East, so there are significant amount of Persian documents available on the Web. Some experimental results[3,4,5,16,33] show that 4-gram and term based vector space model with Lnu.ltu weighting scheme has acceptable performance for Persian text retrieval so far. However some previous research such as[4] uses some additional query that was not created according to TREC specifications so that they are not included in this paper and all of the results are based on TREC specification.

This methods is supposed in the initial retrieval step and empirical experiments will present that the proposed method achieves significant improvement over all of the best existing methods in this field[3,4,16,17].

In this paper, a standard Persian text collection, named[2] *Hamshahri*[3,11] is used, which is built from a large number of newspaper articles according to TREC specifications. Hamshahri is the largest Persian text collection.

Hamshahri[1] is one of the first online Persian newspapers in Iran. It has presented its archive to the public through its website since 1996. Darrudi et al.[11] employed a crawler to download available online news from the website. The collection contains 166,774 articles covering the following subject categories: politics, city news, economics, reports, editorials, literature, sciences, society, foreign news, sports, etc. Table 1 presents the complete attributes of this collection. It contains 65 natural language queries and relevant information of entry lists related to each query according to TREC specifications[3,11]. The proposed architecture is evaluated based on this corpus.

Table 1. Attributes of Hamshahri collection

Attributes	Value
Collection size	564 MB
Collection Length	63,513,827
Documents Format	Terms Text
No. of documents	166
No. of unique terms	774 417
Average length of documents	339
No. of categories	380 Terms
No. of Topics	82 65

3.2 Construction of Clusters

This step is overhead in the architecture. In other word, in this step retrieved documents must be clustered with a fast clustering algorithm and produce two different clusters (Relevant and Irrelevant).

The data clustering, as a class of data mining techniques, aims to partition a given dataset into separate clusters where each cluster is composed of the data objects with similar characteristics. Most existing clustering methods can be broadly classified into two categories: partitioning methods and hierarchical methods. Partitioning algorithms, such as k-means, k-medoid and EM attempt to partition a dataset into k clusters where a previously given evaluation function can be optimized. The basic idea of hierarchical clustering methods is to first construct a hierarchy by decomposing the given dataset, and then use agglomerative or divisive operations to form clusters. In general, an agglomeration-based hierarchical method starts with a disjoint set of clusters, placing each data object into an individual cluster and then merges pairs of clusters until the number of clusters is reduced to a given number k. On the other hand, the division-based hierarchical method treats the whole data set as

[1] http://www.hamshahrionline.ir/

one cluster at the beginning, and divides it iteratively until the number of clusters is increased to k. See[20,23] for more information.

Although[9,30,32,34,37,38,41,51,54] have developed some special algorithms for clustering search results, employing traditional simple and fast methods is preferred in the experiments of this paper. In fact, algorithms that are assumed as the vector space representation for documents and modeled as feature-object matrices (especially term-document matrix) are considered. Empirical results will present that the proposed method with basic clustering algorithms such as k-means[20,23,36] and Principal Direction Divisive Partitioning[6,7,28,7] and some state-of-the-art variants[53] achieves significant improvement over the methods based on similarity search ranking alone.

K-means[20,23] is probably the most celebrated and widely used clustering technique; hence it is the best representative of the class of iterative centroid-based divisive algorithms. On the other hand, PDDP[6,7,28] is representative of the non-iterative techniques based upon the Singular Value Decomposition (SVD) of a matrix built from the data set. PDDP can be quite efficient in comparison to other agglomerative hierarchical algorithms[1,7,36] presented a comparative analysis on the bisecting k-means and PDDP clustering algorithms. Two well-known disadvantages of the k-means algorithm are that the generated clusters depend on the specific selection of initial centroid, so the algorithm can be trapped at local minima of the objective function[53]. Therefore, one run of k-means can easily lead to clusters that are not satisfactory and users are forced to initialize and run the algorithm multiple times.

Existing approaches for document clustering are generally based on either probabilistic methods, or distance and similarity measures. Although there are many well defined distance measures in information retrieval and especially for clustering in high dimensional situations, initial retrieval still depends on similarity measures. Therefore, non-similarity-based methods such as PDDP are employed for clustering search results. Principal Direction Divisive Partitioning proposed by[6] is capable of partitioning a set of documents based on using a high dimensional Euclidean space[6,7,28]. The basic idea is to split the dataset recursively into sub-clusters based on principal direction vectors. PDDP has many key features such as unsupervised, deterministic, good scalability, high quality and identifies the distinct features of the individual clusters. Furthermore, the splits are not based on any distance or similarity measure[6,36], so they seem suitable for our approach[52] created a flexible implementation of this method.

Boley et al.[6,36] believe that using K-means with PDDP clusters as initial configuration can achieve higher quality (Hybrid approach), so it was examined to clustering search results but any improvement was not gained[18]. Regarding the PDDP, despite the convenient deterministic nature, it is easy to construct examples where PDDP produces inferior partitioning than k-means[53]. Although PDDP is known to be an effective clustering method for text mining, term-document matrix are very large and extremely sparse[7,28].

In LCA approach deterministic clustering algorithm with high quality semantic is required, so we turn to some state-of-the-art researches[53] that have been studying the characteristics of PDDP and considering ways to improve its performance[53] shows how to leverage the power of k-means and some interesting recent theory in order to steer the partitioning decision efficiently at each iteration of PDDP.

Six hierarchical methods were employed in the experiments: PDDP[6,7,28], Euclidean K-means[20], Spherical K-means[12], PDDP-2MEANS[53], PDDP-OPT- 2MEANS[53], PDDP-OPTCUT-PD[53]. The main reason behind the choice of these six methods is the fact that they have been extensively used and examined in the context of IR.

3.3 Cluster Analysis

Cluster Analysis is a technique that allows the identification of groups (clusters) of similar objects in multi-dimensional space. After clustering step, two groups of documents are obtained (Figure 1). In cluster analysis step, clusters content has to be analyzed and the relevant and irrelevant cluster must be choose that is an important selection. We conjecture that the context of a document can be considered in the retrieved results by the combination of information search (IDF evidence) and cluster analysis (cluster evidence).

An irrelevant cluster can be viewed as the retrieval result for a different query that shares many terms with the original query. On the other hand, for a cluster of irrelevant documents, there are some terms' query that do not occur in any of the documents in that cluster[48].

Each cluster has a cluster centroid in the form of a vector which is useful as a representative of that cluster. We conjecture that relevant cluster centroid must be near than irrelevant cluster centroid to the query. Consequently, unlike previous work[18] that re-ranked results based on both clusters and after that choose better one manually, in this paper near clusters are choose automatically. On the other hand, cluster centroids and query vector are compared using cosine similarity measure[20] and the relevant cluster is chosen.

Experimental results shows (Section 4) that aforesaid approach is not effective at all and can only choose some of the relevant clusters (Table 2 and Figure 2). Therefore, it would be the purpose of future research to find methods that will 'guide' users towards finding the relevant clusters in the LCA. As you see in Section 4, despite the weak and simple cluster analysis, the proposed method obtained dominant improvement over the best previous methods.

3.4 Documents Re-Ranking

The architecture of document re-ranking model is shown in Figure 3. This model is combining the initial retrieved documents (IDF evidence) and the cluster analysis results (cluster evidence).

Table 2. Interpolated Recall-Precision with manual[18] and automatic cluster analysis (Section 3.3) for PDDP clustering algorithm. Manual cluster analysis result shows with '+' and automatic cluster analysis result shows with '*'

Recall	Precision		
	Initial	LCA (PDDP) +	LCA (PDDP)*
0.0	0.5037	0.9075	0.8345
0.1	0.4103	0.8204	0.6603
0.2	0.3817	0.7299	0.5663
0.3	0.3653	0.6500	0.5230
0.4	0.3489	0.5805	0.4681
0.5	0.3314	0.5140	0.4154
0.6	0.3139	0.4600	0.3703
0.7	0.2886	0.3918	0.3241
0.8	0.2533	0.3227	0.2801
0.9	0.1612	0.1990	0.1799
1.0	0.0342	0.0522	0.0417
11pt avg	0.2766	0.4960	0.3957

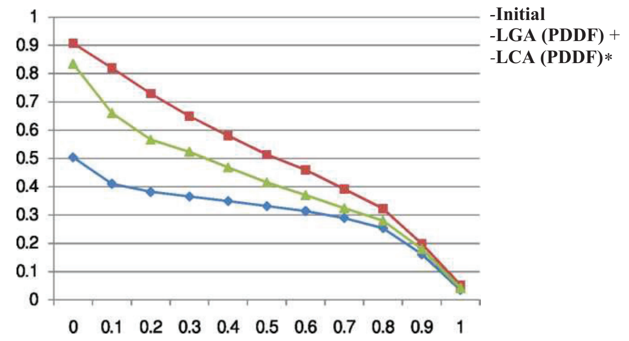

Figure 2. Interpolated Recall-Precision for initial retrieval, manual[18] and automatic cluster analysis (Section 3.3) for PDDP clustering algorithm. Manual cluster analysis result shows with "+" and automatic cluster analysis result shows with,

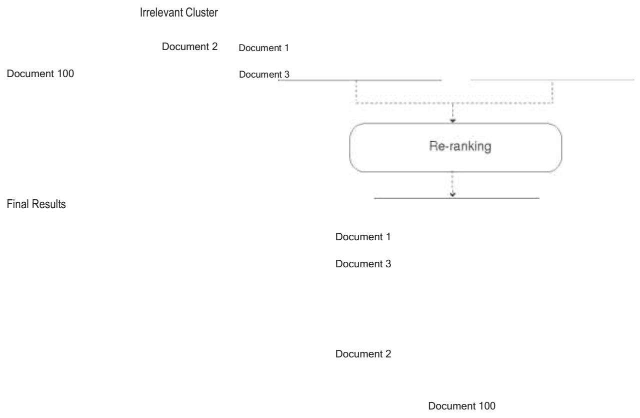

Figure 3. Search result re-ranking architecture. Document 2 does not exist in relevant cluster so sit at the down mid of output list.

In other word, it has been focused on initial retrieved documents and they were combined with clusters evidence (Figure 3). Re-ranked list consist of two sections. Relevant section contains documents in the relevant cluster and the irrelevant section contains documents in the irrelevant cluster based on initial retrieved documents.

Average R-P for Initial, PDDP and Hybrid results is 0.3384, 0.6034 and 0.6010. See Table 2 for more details.

The re-ranking method is very simple. If current document exist in the relevant cluster, go to the re-ranked output otherwise sit in the down mid. As it is clear, using this simple fusion method, it is emphasized on the IDF evidence instead of cluster evidence because the probability of mistakes in clustering step is high[14,46,48] especially with our simple and weak cluster analysis (Section 3.3). Here is pseudo code for re-ranking and synthesize of evidences:

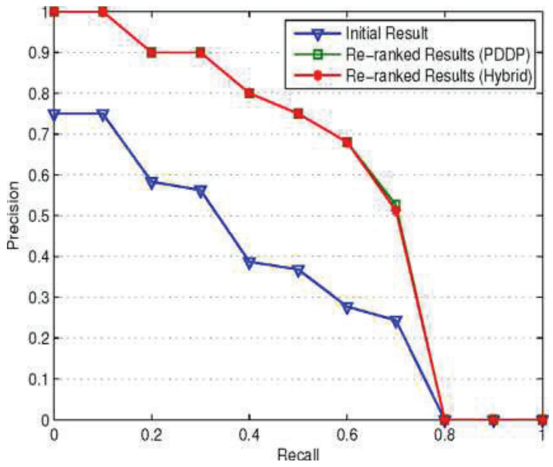

Figure 4. Interpolated Recall-Precision at the best query.

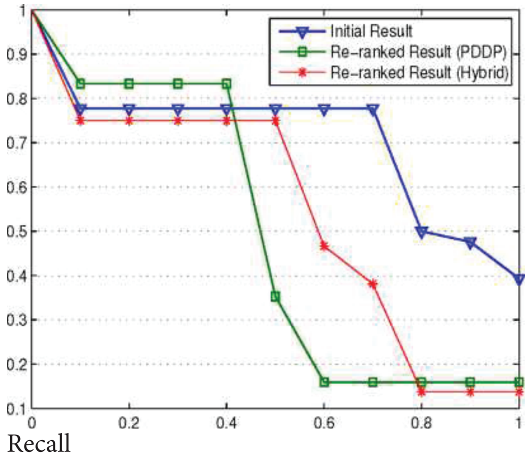

Figure 5. Interpolated Recall-Precision at the worst query.

Table 3. Interpolated Recall-Precision with the best initial retrieval method and different clustering variants

Recall	Precision						
	Initial	PDDP	E.K-means*	S.K-means*	2M*	O-2M	PD
0.0	0.5037	0.8345	0.833500	0.820945	0.845005	0.8284	0.8413
0.1	0.4103	0.6603	0.725994	0.684740	0.718321	0.6992	0.6654
0.2	0.3817	0.5663	0.661786	0.607776	0.657949	0.5774	0.5776
0.3	0.3653	0.5230	0.600378	0.547230	0.603402	0.5096	0.5060
0.4	0.3489	0.4681	0.539994	0.493411	0.549842	0.4586	0.4687
0.5	0.3314	0.4154	0.478150	0.445955	0.497103	0.4185	0.4209
0.6	0.3139	0.3703	0.420919	0.396584	0.445274	0.3732	0.3686
0.7	0.2886	0.3241	0.365966	0.348759	0.389229	0.3330	0.3250
0.8	0.2533	0.2801	0.303962	0.289828	0.321746	0.2806	0.2746
0.9	0.1612	0.1799	0.189509	0.183478	0.200872	0.1740	0.1735
1.0	0.0342	0.0417	0.043988	0.040782	0.045484	0.0362	0.0363
11pt avg	0.2766	0.3957	0.451731	0.416990	0.459822	0.3982	0.3949

*Average over 100 runs. Best initial result, PDDP[6,7,28], Euclidean K-means[20], Spherical K-means[12], PDDP-2MEANS[53], PDDP-OPT-2MEANS[53], PDDP-OPTCUT-PD[53]

```
    FOR each document in the retrieved result list IF document exist in relevant cluster THEN CALL Append To Relevant Section (document)
    ELSE
        CALL Append To Irrelevant Section (document)
    END IF END FOR
```

4. Discussions of the Results

The TREC Robust Track[45] was started in 2003 to focus on poor performing queries. Several new measures were introduced to evaluate the effectiveness on weak queries. Since 2004, another new measure Geometric MAP (GMAP)[35] was introduced as an alternative to the Mean Average Precision (MAP). GMAP takes the geometric mean of average precisions of all the queries instead of their arithmetic mean, in order to emphasize scores close to 0. Table 4 shows a comparison between best initial results[3,4,5,16,33] and LCA approach using different variants on Hamshahri corpus.

The main problem with local cluster analysis[18,19] is its inconsistency. A query-by-query TREC specific analysis on Hamshahri collection shows that it can improve some queries seriously and hurt others in contrast (Table 5). On

Table 4. Other evaluation measures with the best initial retrieval method and different clustering variants.

Criterion	Values						
	Initial	PDDP	E.K-means*	S.K-Means*	2M*	O-2M	PD
MAP	0.2766	0.3957	0.451731	0.416990	0.459822	0.3982	0.3949
GMAP	0.2186	0.3060	0.360739	0.332635	0.369182	0.3155	0.3112
R-Prec	0.2898	0.3822	0.454330	0.414220	0.465161	0.3903	0.3866
P5	0.2646	0.6000	0.636797	0.583229	0.618098	0.5908	0.6000
P10	0.2800	0.5185	0.577830	0.522268	0.560636	0.5077	0.5077
P15	0.2974	0.4800	0.534447	0.487479	0.523282	0.4749	0.4708
P20	0.3023	0.4469	0.505373	0.459854	0.503223	0.4385	0.4369

*Average over 100 runs. Best initial result, PDDP[2,6,7,8], Euclidean K-means[20], Spherical K-means[12], PDDP-2MEANS[53], PDDP-OPT-2MEANS[53], PDDP-OPTCUT-PD[53].

Table 5. Average precision by per query and cluster. Note that this table for manual cluster analysis

Query NO.	Average precision (non-interpolated) over all rel docs					Improvement		Description
	Best init-retrieval	Re-ranked (PDDP)		Re-ranked (Hybrid)		PDDP	Hybrid	
		Cluster 1	Cluster 2	Cluster 1	Cluster 2			
1	0.7318	0.7383	0.4084	0.7472	0.6218	0.0065	0.0154	
2	0.7638	0.7452	0.7120	0.7615	0.7110	-0.0186	-0.0023	
3	0.5404	0.5142	0.5562	0.5173	0.5562	0.0158	0.0158	
4	0.5420	0.3875	0.5433	0.3875	0.5433	0.0013	0.0013	
5	0.3430	0.0915	0.5461	0.0985	0.4148	0.2031	0.0718	
6	0.6552	0.7062	0.2862	0.6538	0.2927	0.0510	-0.0014	
7	0.4322	0.3951	0.5218	0.3951	0.5218	0.0896	0.0896	
8	0.5262	0.7113	0.3078	0.6990	0.3676	0.1851	0.1728	
9	0.1531	0.0342	0.2447	0.0582	0.2683	0.0916	0.1152	
10	0.8336	0.8244	0.7991	0.8244	0.7991	-0.0092	-0.0092	
11	0.7504	0.6941	0.3348	0.7187	0.2469	-0.0563	-0.0317	
12	0.4685	0.4840	0.4306	0.4833	0.4371	0.0155	0.0148	
13	0.4190	0.1280	0.4565	0.1549	0.4257	0.0375	0.0067	
14	0.2423	0.1504	0.3504	0.1550	0.3627	0.1081	0.1204	
15	0.3901	0.3230	0.3495	0.3438	0.3852	-0.0406	-0.0049	
16	0.6284	0.6247	0.4434	0.5646	0.4888	-0.0037	-0.0638	
17	0.7620	0.7232	0.6931	0.7169	0.6989	-0.0388	-0.0451	
18	0.1587	0.1361	0.1010	0.1596	0.0944	-0.0226	0.0009	
19	0.1771	0.0843	0.2375	0.0843	0.2375	0.0604	0.0604	
20	0.6937	0.1314	0.8338	0.2234	0.8077	0.1401	0.1140	
21	0.0766	0.0617	0.1195	0.0648	0.1689	0.0429	0.0923	
22	0.7130	0.5777	0.7337	0.6193	0.7377	0.0207	0.0247	
23	0.7671	0.7343	0.7563	0.7467	0.7847	-0.0108	0.0176	
24	0.2257	0.1360	0.1860	0.1360	0.1860	-0.0397	-0.0397	
25	0.5261	0.3944	0.2936	0.4572	0.2777	-0.1317	-0.0689	
26	0.5350	0.4262	0.6526	0.5055	0.5679	0.1176	0.0329	

(*continued*)

Table 5. (*continued*)

Query NO.	Average precision (non-interpolated) over all rel docs					Improvement		Description
	Best init-retrieval	Re-ranked (PDDP)		Re-ranked (Hybrid)		PDDP	Hybrid	
		Cluster 1	Cluster 2	Cluster 1	Cluster 2			
27	0.0770	0.0419	0.2194	0.0421	0.2100	0.1424	0.1330	
28	0.2928	0.2994	0.2224	0.3004	0.1983	0.0066	0.0076	
29	0.4402	0.1452	0.5525	0.1787	0.4690	0.1123	0.0288	
30	0.3001	0.1837	0.3919	0.2110	0.3866	0.0918	0.0865	
31	0.6390	0.5638	0.6239	0.5543	0.6248	-0.0151	-0.0142	
32	0.6352	0.6457	0.2490	0.6457	0.2490	0.0105	0.0105	
33	0.4081	0.4554	0.2306	0.4114	0.3159	0.0473	0.0033	
34	0.7402	0.7573	0.5485	0.7573	0.5485	0.0171	0.0171	
35	0.7230	0.6437	0.6165	0.6187	0.6348	-0.0793	-0.0882	
36	0.3955	0.3719	0.2041	0.3719	0.2041	-0.0236	-0.0236	
37	0.4675	0.3193	0.4817	0.3193	0.4817	0.0142	0.0142	
38	0.6604	0.4459	0.6493	0.4571	0.6428	-0.0111	-0.0176	
39	0.7776	0.4803	0.6314	0.4803	0.6314	-0.1462	-0.1462	
40	0.1191	0.2715	0.0697	0.1730	0.0892	0.1524	0.0539	
41	0.6139	0.6759	0.5198	0.6759	0.5198	0.0620	0.0620	
42	0.4237	0.3374	0.2032	0.4275	0.1852	-0.0863	0.0038	
43	0.6106	0.2790	0.5716	0.4108	0.5708	-0.0390	-0.0398	
44	0.8041	0.5005	0.8069	0.5005	0.8069	0.0028	0.0028	
45	0.4445	0.2390	0.3703	0.4475	0.1886	-0.0742	0.0030	
46	0.5481	0.3092	0.5801	0.4872	0.4719	0.0320	-0.0609	
47	0.3735	0.2958	0.2758	0.2676	0.2932	-0.0777	-0.0803	
48	0.8477	0.8241	0.4480	0.8277	0.4814	-0.0236	-0.0200	
49	0.5265	0.5191	0.4563	0.5191	0.4563	-0.0074	-0.0074	
50	0.3384	0.1152	0.6034	0.1163	0.6010	0.2650	0.2626	Best Query
51	0.6396	0.3795	0.7213	0.3780	0.7312	0.0817	0.0916	
52	0.4369	0.4299	0.3889	0.4299	0.3889	-0.0070	-0.0070	
53	0.0517	0.0605	0.0539	0.0605	0.0539	0.0088	0.0088	
54	0.3650	0.3919	0.2374	0.3420	0.2962	0.0269	-0.023	
55	0.2810	0.3006	0.1878	0.3006	0.1878	0.0196	0.0196	
56	0.6709	0.6171	0.4837	0.6413	0.4906	-0.0538	-0.0296	
57	0.0033	0.0049	0.0037	0.0133	0.0027	0.0016	0.01000	
58	0.3373	0.3568	0.1641	0.3513	0.1517	0.0195	0.01400	
59	0.3019	0.3836	0.1364	0.4046	0.1456	0.0817	0.1027	
60	0.2890	0.2853	0.1211	0.4855	0.1503	-0.0037	0.1965	
61	0.2129	0.0907	0.3779	0.0915	0.4182	0.1650	0.2053	
62	0.6611	0.3768	0.4629	0.3973	0.5119	-0.1982	-0.1492	Worst Query
63	0.5849	0.3867	0.5978	0.4189	0.5490	0.0129	-0.0359	
64	0.6124	0.4550	0.4791	0.4965	0.5033	-0.1333	-0.1091	
65	0.3231	0.2321	0.3231	0.2696	0.2707	0.0000	-0.0524	

the other hand, our experiments express that total average precision on all queries can improve search results effectiveness (Table 3 and Figure 6).

Average R-P for initial, PDDP and Hybrid results is 0.6611, 0.4629 and 0.5119. See Table 2 for more details.

In this paper two different variants of K-means are evaluated. Spherical k-means[12] and Euclidean K-means[20]. As it is illustrated in Table 3 and Table 4, Euclidean K-means gives the better results than Spherical K-means among all measures.

Although Euclidean k-means appears to give the better results between all variants and all measures (except PDDP-2MEANS[53]) especially PDDP, it must be noted that these plots report mean values attained by k-means and related variants. In practice, a single run of k-means may lead to poor results. As a result, a 'good' partitioning may require several executions of the algorithm.

Unlike[53] results, compared to the basic algorithm, PDDP-2MEANS appear to give the best results between all variants and all measures even better than k-means. See[53] for more details about PDDP-2MEANS.

The precision curve of local cluster analysis in Figure 4 predicts the best query improvement by using the proposed architecture on search results (Table 2). In this specific query, using the architecture improve 0.265 average precision quantity (Table 5, QueryNO = 50).

The precision curve in Figure 5 in contrast to Figure 4 predicts the worst query by applying the proposed architecture which is very instructive. As it is clear, even in the worst query while Recall <= 0.4 our system remains high-precision (Figure 5.) and it curves upper the best initial retrieval, although average precision 0.1982 decrease (Table 5, QueryNO = 62). Note that the best methods for initial retrieval are employed in this paper[3,4,16].

5. Conclusion and Future Works

In this paper, a novel model for retrieval systems is proposed which is based on a simple document re-ranking method using Local Cluster Analysis (LCA). Experimental results on a Hamshahri collection present that this method performs more efficiently than other existing techniques[3–5,16].

Whereas in our approach the context of a document is considered in the retrieved results by the combination of information search (IDF evidence) and local cluster analysis (cluster evidence), it causes some consequences. First, relevant cluster are tailored to the user information need and can improve the search results efficiently. Second, it constructs high-precision system that contains more relevant documents at top of the result list that makes

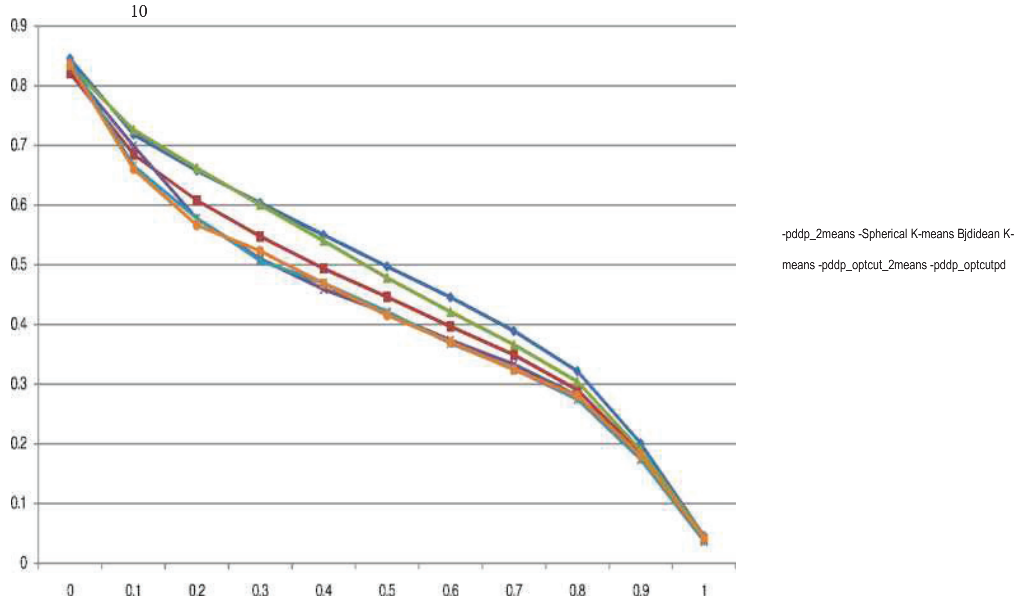

Figure 6. Interpolated Recall-Precision with the best initial retrieval method and different clustering variants.

it good framework for weak queries in the following. Experiments results on Hamshahri corpus stipulate it. As it was shown, even in the worst query that average precision is decreased to 0.1982 percent, our system is still remained in high-precision.

Consequently, this work can be persuaded in several directions as follows. Firstly, as mentioned before in Section 3.3, using cluster centroids as a representative vector for each cluster and comparing them with query weak vector is not efficient and causes to lose a lot of accuracy (Table 2 and Figure 2). Therefore, it can be the motivation of future research to find methods that choose relevant cluster efficiently and if it works well then the results using proposed method can be improved.

Secondly, using dimensional reduction methods in LCA approach is beloved. Apart from efficiency, effectiveness will also put forward for the use of LCA in IR systems.

Finally, the proposed architecture is evaluated in ad-hoc retrieval. As mentioned before, this approach is independent of initial system architecture, so it can be embedded on any fabric search engine. One of the high-precision needful systems is Web search engines. Indisputable evaluations of this approach on Web search engines can also be a prominent future work.

6. References

1. Abdulla HD, Abdelrahman AS, Snasel V, Aldosari H. Using singular value decomposition as a solution for search result clustering. Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2014); Springer; 2014. p. 53–61.
2. Alam M, Sadaf K. A review on clustering of web search result. Advances in Computing and Information Technology. 2013; 153–9.
3. Aleahmad A, Amiri H, Darrudi E, Rahgozar M, Oroumchian F. Hamshahri: A standard Persian text collection. Knowl Base Syst. 2009; 22(5):382–7.
4. Aleahmad A, Hakimian P, Mahdikhani F, Oroumchian F. N-gram and local context analysis for Persian text retrieval. International Symposium on Signal Processing and its Applications; 2007.
5. Amiri H, Aleahmad A, Oroumchian F, Lucas C, Rahgozar M. Using owa fuzzy operator to merge retrieval system results. The Second Workshop on Computational Approaches to Arabic Script-based Languages (LSA 2007); USA: Linguistic Institute, Stanford University; 2007.
6. Boley D. Principal direction divisive partitioning. Data Min Knowl Discov. 1998; 2(4):325–44.
7. Boley D. A scalable hierarchical algorithm for unsupervised clustering. Data Mining for Scientific and Engineering Applications; 2001.
8. Buckley C. Why current IR engines fail. Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'04); New York, NY, USA. 2004. p. 584–5.
9. Calli C, Ucoluk G, Sehitoglu T. Improving search result clustering by integrating semantic information from Wikipedia [PhD thesis, MS Thesis]. Middle East Technical University, Department of Computer Engineering; 2010.
10. Cutting DR, Karger DR, Pedersen JO, Tukey JW. Scatter/gather: A cluster-based approach to browsing large document collections. Proceedings of the 15th annual international ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'92); New York, NY, USA. 1992. p. 318–29.
11. Darrudi E, Hejazi MR, Oroumchian F. Assessment of a modern farsi corpus. Proceedings of the 2nd Workshop on Information Technology and its Disciplines (WITID); 2004.
12. Dhillon IS, Modha DS. Concept decompositions for large sparse text data using clustering. Mach Learn. 2001; 42(1–2):143–75.
13. Gelgi F, Davulcu H, Vadrevu S. Term ranking for clustering web search results. WebDB; 2007.
14. He J, Meij E, de Rijke M. Result diversification based on query-specific cluster ranking. J Am Soc Inform Sci Tech. 2011; 62(3):550–71.
15. Hearst MA, Pedersen JO. Reexamining the cluster hypothesis: scatter/gather on retrieval results. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'96); New York, NY, USA. 1996. p. 76–84.
16. Jadidinejad A, Mahmoudi F. Cross-language information retrieval using metalanguage index construction and structural queries. In: Peters C, Nunzio G, Kurimo M, Mandl T, Mostefa D, Peas A, Roda G, editors. Multilingual Information Access Evaluation I. Text Retrieval Experiments. Lecture Notes in Computer Science. Berlin Heidelberg: Springer; 2010. p. 70–7.
17. Jadidinejad A, Mahmoudi F, Dehdari J. Evaluation of perstem: A simple and efficient stemming algorithm for persian. In: Peters C, Nunzio G, Kurimo M, Mandl T, Mostefa D, Peas A, Roda G, editors. Multilingual Information Access Evaluation I. Text Retrieval Experiments. Lecture Notes in Computer Science, Springer Berlin Heidelberg; 2010. p. 98–101.
18. Jadidinejad AH, Amiri H. Local cluster analysis as a basis for high-precision information retrieval. Proceedings of the International Conference on Informatics and Systems (INFOS'08); Cairo, Egypt. 2008. p. 93–100.
19. Jadidinejad AH, Toroghi Haghighat A. Local cluster analysis: A new approach for evaluating different document clustering algorithms by huge corpora. Proceedings of the

International Conference on Asian Language Processing (IALP '08); Chiang Mai, Thailand. 2008. p. 155–60.
20. Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surv. 1999; 31(3):264–323.
21. Janruang J, Kreesuradej W. A new web search result clustering based on true common phrase label discovery. Proceedings of the International Conference on Computational Intelligence for Modeling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06); Washington, DC, USA: IEEE Computer Society; 2006. p. 242.
22. Jardine C, van Rijsbergen N. The use of hierarchic clustering in information retrieval. Information Storage Retrieval. 1971; 7:217–40.
23. Kogan J. Introduction to clustering large and high-dimensional data. New York, NY, USA: Cambridge University Press; 2007.
24. Kui-Lam Kwok LG, Deng P. Improving weak ad-hoc retrieval by web assistance and data fusion. Lecture Notes in Computer Science. 2005; 3689:17–30.
25. Kwok KL. Higher precision for two-word queries. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'02); New York, NY, USA. 2002. p. 395–6.
26. Lee K-S, Park Y-C, Choi K-S. Re-ranking model based on document clusters. Inf Process Manage. 2001; 37(1):1–14.
27. Leuski A. Evaluating document clustering for interactive information retrieval. Proceedings of the Tenth International Conference on Information and Knowledge Management (ACM CIKM'01); New York, NY, USA. 2001. p. 33–40.
28. Littau D, Boley D. Clustering very large datasets with PDDP. Grouping Multidimensional Data: Recent Advances in Clustering; 2006. p. 99–126.
29. Liu Z, Natarajan S, Chen Y. Query expansion based on clustered results. Proceedings of the VLDB Endowment. 2011; 4(6):350–61.
30. Mecca G, Raunicha S, Pappalardoa A. A new algorithm for clustering search results. Data and Knowledge Engineering. 2007 Sep; 62(3):504–22.
31. Moreno JG, Dias G, Cleuziou G. Post-retrieval clustering using third-order similarity measures. ACL; 2013; (2):153–8.
32. Nguyen SH, Swieboda W, Jaskiewicz G. Extended document representation for search result clustering. Intelligent Tools for Building a Scientific Information Platform. Springer; 2012. p. 77–95.
33. Oroumchian F, Darrudi E, Taghiyareh F, Angoshtari N. Experiments with persian text compression for web. 13th International World Wide Web Conference; New York, NY, USA. 2004.
34. Osinski S, Weiss D. A concept-driven algorithm for clustering search results. IEEE Intelligent Systems. 2005; 20(3):48–54.
35. Robertson S. On GMAP: and other transformations. Proceedings of the 15th ACM International Conference on Information and Knowledge Management (ACM CIKM'06); New York, NY, USA. 2006. p. 78–83.
36. Savaresi SM, Boley DL. A comparative analysis on the bisecting k-means and the PDDP clustering algorithms. Intell Data Anal. 2004; 8(4):345–62.
37. Scaiella U, Ferragina P, Marino A, Ciaramita M. Topical clustering of search results. Proceedings of the Fifth ACM International Conference on Web Search and Data Mining; 2012. p. 223–32.
38. Schuhmacher M, Ponzetto SP. Exploiting dbpedia for web search results clustering. Proceedings of the 2013 Workshop on Automated Knowledge base Construction (ACM); 2013. p. 91–6.
39. Shah C, Croft WB. Evaluating high accuracy retrieval techniques. Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'04); New York, NY, USA. 2004. p. 2–9.
40. Spink A, Jansen BJ, Wolfram D, Saracevic T. From e-sex to e-commerce: Web search changes. Computer. 2002; 35(3):107–9.
41. Stefanowski J, Weiss D. Carrot and language properties in web search results clustering. AWIC. 2003; 240–9.
42. Tombros A, Villa R, Rijsbergen CJV. The effectiveness of query-specific hierarchic clustering in information retrieval. Inf Process Manage. 2002; 38(4):559–82.
43. Tsai CW, Liang TW, Ho JH, Yang CS, Chiang MC. A document clustering approach for search engines. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (ICSMC'06); Taipei, Taiwan. 2006. p. 1050–5.
44. Van Rijsbergen CJ. Information Retrieval. 2nd ed. Department of Computer Science, University of Glasgow; 1979.
45. Voorhees EM. Overview of trec 2003. TREC; 2003. p. 1–13.
46. Xu J. Solving the word mismatch problem through automatic text analysis [PhD thesis]. Amherst: MA, USA; 1997.
47. Xu J, Croft WB. Query expansion using local and global document analysis. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR '96); New York, NY, USA. 1996. p. 4–11.
48. Xu J, Croft WB. Improving the effectiveness of information retrieval with local context analysis. ACM Trans Inf Syst. 2000; 18(1):79–112.
49. Yin J, Xu WR. Query expansion associated with clustering. Appl Mech Mater. 2014; 441:647–50.
50. Zamir O, Etzioni O. Grouper: A dynamic clustering interface to web search results. Proceeding of the Eighth International Conference on World Wide Web (WWW'99); New York, NY, USA: Elsevier North-Holland, Inc; 1999. p. 1361–74.

51. Zamir OE. Clustering web documents: A phrase-based method for grouping search engine results [PhD thesis]. 1999.
52. Zeimpekis D, Gallopoulos E. Tmg: A matlab toolbox for generating term-document matrices from text collections. Grouping Multidimensional Data: Recent Advances in Clustering. Springer; 2006. p. 187–210.
53. Zeimpekis D, Gallopoulos E. Principal direction divisive partitioning with kernels and k-means steering. Survey of Text Mining II: Clustering, Classification and Retrieval. Springer; 2008. p. 46–64.
54. Zeng HJ, He QC, Chen Z, Ma WY, Ma J. Learning to cluster web search results. Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'04); New York, NY, USA. 2004. p. 210–17.