CMIC@TREC-2009: Relevance Feedback Track

Kareem Darwish, Ahmed El-Deeb
Cairo Microsoft Innovation Center (CMIC)
Bldg. B115, Smart Village,
Km. 28 Cairo-Alexandria Desert Rd.,
Abou Rawash, Egypt
{kareemd, v-ahelde}@microsoft.com

Abstract

This paper describes CMIC's submissions to the TREC'09 relevance feedback track. In the phase 1 runs we submitted, we experimented with two different techniques to produce 5 documents to be judged by the user in the initial feedback step, namely using knowledge bases and clustering. Both techniques attempt to topically diversify these 5 documents as much as possible in an effort to maximize the probability that they contain at least 1 relevant document. The basic premise is that if a query has n diverse interpretations, then diversifying results and picking the top 5 most likely interpretations would maximize the probability that a user would be interested in at least one interpretation. In phase 2 runs, which involved the use of the feedback attained from phase 1 judgments, we attempted to use positive and negative judgments in weighing the terms to be used for subsequent feedback. .

1. Introduction

Phase 1 of the runs involved nominating 5 documents to a user for which the user would provide relevance judgments. In the second phase, these judgments are used for relevance feedback. In nominating the 5 documents, it is essential to present users with some relevant documents exemplars, where having at least one relevant document is better than having none. We opted to eliminate the worst case scenario, where none of the documents that a user is judging is relevant. To this end, we attempted to topically diversify the documents to be judged by the user to increase the probability that at least 1 of the 5 document is relevant, at the possible expense of decreasing the number of relevant documents in these 5 documents. The basic premise is that if a query has n diverse interpretations $I_n = \{i_1, i_2, \ldots i_n\}$ (ex. Jaguar: cat, car, OS, etc.) with P_j = Prob(interest_to_user|i_j), where sum(P_j | j = 1 .. n) = 1, then picking one example of each of the top 5 most likely interpretations would maximize the probability that a user would be interested in one interpretation. To achieve this kind of diversity we tried two different techniques for diversification: one relied on a knowledge base, namely Wikipedia, and the other on cluster analysis.

For phase 2 submissions involving feedback, we employed a fairly simple equation to expand the queries based on the probability of the existence of a certain term in a relevant document versus the probability of its existence in an irrelevant document.

The rest of the paper is organized as follows: section 2 surveys issues relating to results diversification; section 3 describes experimental setup; section 4 reports on submissions results; and section 5 concludes the paper.

2. Prior Work

Diversification:

Though the work on results diversification is relatively scant, a few methods have been suggested to diversify search results. Carbonell and Goldstein [2] suggested the so-called Maximal Marginal Relevance (MMR) which attempts to reduce redundancy while maintaining relevance. MMR combines query relevance with information-novelty as follows [2]:

$$MMR \stackrel{\text{def}}{=} Arg \max_{Di \in R \setminus S} \left[\lambda(Sim1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim2(D_i, D_j))\right] \quad (1)$$

Where Q is the query, D_i is the i^{th} document in the ad-hoc retrieval ranked list, $Sim_1(D_i, Q)$ is the similarity between Q and D_i, max $Sim_2(D_i, D_j)$ is the maximal similarity between D_i and all documents D_j where j ranges between 1 and i – 1, and λ is a weighting factor that is less than 1 and gives varying weights to Sim_1 and max Sim_2 to favor similarity to query or dissimilarity to previously seen documents. MMR favors documents that are most similar to the query while penalizing documents that contain redundant information. The newly computed MMR for each document is used to re-rank search results, hopefully selecting non-redundant relevant documents.

Chen and Karger [3] argued against the optimality of the Probability Ranking Principal, stating that "in a probabilistic context, one should directly optimize for the expected value of the metric of interest". Most web search engines optimize for metrics such as DCG and NDCG, which often hurt diversity. To achieve diversity, they used a greedy algorithm to optimize for a specific objective, namely finding at least one relevant document, integrating diversity into their ranking formula. What is noteworthy in their work is their treatise on the applicability of different evaluation metrics such as search length, MRR, %no, which measures one document sufficiency, and k-call, which is k document sufficiency.

Radlinski and Dumais [20] explored re-ranking on the client side, to efficiently incorporate personalization with diversification, and they achieved diversification (or disambiguation) by augmenting a query with its most common reformulations, which were acquired from web search engine query logs.

Agrawal et al. [1] assumed the existence of a taxonomy of information to achieve diversification. They mapped both queries and documents to one or more entries in the taxonomy and query results were diversified to cover different entries in the taxonomy. They also proposed some interesting generalizations to standard IR metrics, like MAP, MRR, and NDCG to explicitly account for diversification.

Zhai et al. [25] proposed a framework for evaluating algorithms for subtopic retrieval in an effort to account for the intrinsic difficulty of a query, as well as the coverage of subtopics. They also did some work on generalizing evaluation metrics and introduced so-called S-recall and S-precision. In another work, Zhai and Lafferty [26] proposed a risk minimization framework that attempts to minimize a certain loss function that represents the user's dissatisfaction.

Clustering:

Cluster analysis is an unsupervised machine learning technique that attempts to find clusters of related n-dimensional objects within a data collection using different objectives or criteria. Partitioning algorithms such as k-means [10] attempts to optimize an objective function to form clusters around centers/means.

K-medoids [7], PAM [13] and CLARANS [19] are related techniques that substitute cluster means with medoids or representative data objects. Hierarchical algorithms such as single, average, and complete linkage produce dendrograms which provide clustering at several possible numbers of clusters. Other approaches include Grid-based algorithms such as DenClue [9] and STING [23], Density-based algorithms such as DBSCAN [5], Graph-based algorithms such as Chameleon [12], and distance-relatedness-based algorithms such as Mitosis [24]. Partitional algorithms that are k-means like and average or complete linkage are only able to detect clusters of globular/hyper-spherical shapes. However, single linkage algorithms are known to find elongated shaped clusters, but are greatly affected by outliers. Density based algorithms as DBSCAN, tend to find clusters of arbitrary shapes and identify outliers, and more recent algorithms such as Chameleon and Mitosis find clusters of arbitrary shapes and arbitrary densities [24]. Some of the popular types of clustering include partitional k-means, hierarchical single-link, and density-based DBSCAN clustering, which are O(nkd), $O(n^2)$, and O($n$log$n$) respectively, where n is the number of documents and for k-means k is the number of clusters and d is the number of iterations [18][22]. K-means proceeds through the following steps: k documents are picked randomly to form centroids of clusters, each document is assigned to the closest centroid, the center of each cluster is chosen as the new centroid, and the process iterates until the algorithm converges. K-means requires k to be specified a priori, the resulting clusters are globular in shape, and the choice of initial centroids may change the assignment of documents to clusters. Bisecting k-means is a variant of the popular k-means clustering algorithm in which a document set is split into two clusters using the generic k-means algorithm and then some (or all) of the resulting clusters of elements are iteratively split into two until the desired k clusters are formed. Although bisecting k-means is slower than k-means clustering, bisecting k-means is insensitive to the choice of initial centroids.

Hierarchical clustering organizes documents in a tree like structure called a dendrogram, where each document is assumed to be a singleton cluster, and then clusters are merged successively in descending similarity until all documents are merged into a large cluster at the root of the dendrogram. The merge process can be applied successively until a desired number of clusters is reached.

DBSCAN [5] is a density based clustering technique in which an initial set of "core" elements, which are elements that have a minimum number of M neighbors that fall within ε distance away, are used to form the seeds of clusters, and then these seed clusters are allowed take-in more points or clusters within ε distance of any of their member elements (conflating clusters if need be). Some of the advantages of DBSCAN include: clusters can be arbitrary shaped, unlike k-means clustering which produces globular clusters, "core" elements in a cluster are found automatically, and unlike partitional or hierarchical techniques not all elements belong to clusters, because elements that are further than ε away from other elements are deemed as outliers. The major disadvantage of DBSCAN is that the values of M and ε need to be determined a priori. A distance metric or a similarity measure is used to measure proximity between objects in a clustering algorithm. Some popular similarity measures include cosine similarity and TF-IDF weighing.

As for the use of clustering in IR, subsequent to Van Rijsbergen cluster hypothesis [11] and Salton's suggestion to use clustering in IR [21], much work has been done on applying clustering to IR [18]. Van Rijsbergen's cluster hypothesis states that "closely associated documents tend to be relevant to the same request" [11]. Attempts were made to exploit this hypothesis in various ways. As examples, post hoc clustering of retrieval results has been used to improve retrieval effectiveness [8][17], to improve

presentation [16], blind relevance feedback [14], and non-blind relevance feedback [15]. This list is by no means comprehensive, but gives samples of the four main directions in which the cluster analysis was used in IR. In Lee et al. [14], single link clustering was successfully used to identify core topics of a query, which are identified as dense clusters, for which "dominant" documents were used for blind relevance feedback. Leuski and Allen [16] used hierarchical clustering as part of an interactive retrieval system in which documents that cluster together would appear together and clusters are clearly demarked.

3. Experimental Setup

Phase 1

For the first stage, we experimented with two different techniques to produce 5 documents to be judged by the user. Both techniques attempt to topically diversify the 5 documents to be presented to the user as much as possible in an effort to maximize the probability that these documents contain at least 1 relevant document.

The first diversification technique **(employed in run CMIC.1)** utilized Wikipedia with the assumption that Wikipedia articles are naturally diverse, i.e. no two articles cover exactly the same topic. We issued all the queries against Bing, a web search engine, while restricting results to Wikipedia (ex. "jaguar site:en.wikipedia.org"). The RF track collection includes a 2008 snapshot of Wikipedia. We nominated the top 5 results to show to the user, automatically excluding Wikipedia articles that point to non-article content such as images and discussion pages.

The second diversification technique **(employed in run CMIC.2)** relied on DBSCAN [5] — a density based clustering technique — to cluster top 100 results for each query. The IR group at Microsoft Research, Cambridge, kindly provided us with 2,500 search results for each query [4]. These results were obtained by searching Category B of the ClueWeb09 collection using the OKAPI-BM25 weighting formula. One of the main challenges in this track was to effectively search the 50 million documents in the collection. This was done using a distributed grep-like function, implemented on Microsoft's Dryad framework, to select all documents containing the query terms and then to compute appropriate weights for ranking.

In our application of DBSCAN, all the terms in documents were tokenized, stemmed using Porter stemmer, and stopwords were removed. Distance between documents was computed as (1 - cosine similarity). Clustering was performed using parameter values M equal to 4 and ε equal to 0.65. We picked these specific parameters using extensive side experiments on the TREC 2001 and TREC 2002 filtering collection, which includes a set of approximately 880,000 documents from Reuters, 184 topics, and associated relevance judgments. The highest ranked document in each cluster, excluding outliers (singleton clusters), was picked to represent the cluster. Subsequently, the highest ranked 5 documents representing clusters were nominated to be shown to the users. If the number of clusters was less than 5, the remaining documents were picked from the highest ranked outliers. Aside from being easy to implement and having an agreeable time complexity, DBSCAN has many relevant advantages including its capacity to form arbitrarily shaped clusters and to automatically detect outliers. We did some previous experiments that suggested that the use of DBSCAN for this purpose is more effective than that of k-means and bisecting k-means. These experiments also showed the favorable effect of detecting outliers.

Phase 2

For phase 2, we attempted to re-rank the top 2,500 results from searching using the original queries. The re-ranking was done by indexing these top 2,500 documents using Indri and searching the index using expansion terms from judged documents (original query terms were excluded because they were used to produce the initial results and all documents generally contained the original query terms). For expansion terms, we attempted to make use of both positive and negative judgments in weighing the terms to be used in expanding queries. To do so, we used a variant of Acc2 feature selection metric referred to by Forman [6]. Forman originally used Acc2 — among other metrics— as a metric for feature selection in text classification tasks. In his paper, he compared several such metrics for precision and F-measure with Acc2 being one of the best feature selection metrics and one of the easiest to implement. We use Acc2 in a different sense however: to determine the weights to assign to each term while expanding the query. The weight W for a certain term t is calculated as follows:

$$W(t) = P(t|pos) - P(t|neg)$$

$$= \frac{\#\ of\ relevant\ \ docs\ containg\ \ t}{\#\ of\ relevant\ \ docs} - \frac{\#\ \text{of non}-\text{relevant}\ \ \text{docs containg t}}{\#\ \text{of non}-\text{relevant}\ \ \text{docs}}$$

In original formula used by Forman, Acc2 was the absolute value of W(t). Only terms with positive weights were used to re-rank the initial set of results of 2,500 documents by searching them using Indri. The weights W(t) were used in weighting query terms in Indri (using Indri's #wsum operator).

CMIC was assigned 8 different phase 1 runs to use for expansion, namely: CMIC.1, CMIC.2, ilps.1, MSRC.1, udel.1, udel.2, ugTr.2, and UMas.2.

4. Results

To ascertain the effectiveness of our phase 1 results, we used mean reciprocal rank (MRR) and precision at 5 (P@5) as the measures of quality. The rationale for picking MRR is based on the assumption that including at least 1 relevant document in the feedback set is better than having none. Thus, MRR would hopefully correlate with the probability of having at least one such document, as the higher the rank of relevant documents the greater the probability and having no relevant documents would result in an MRR of 0. The rationale for using P@5 stems from the assumption that having more relevant documents in the feedback set would yield better feedback results. Figure 1 shows the results of CMIC (marked) compared to the results of the rest of the groups for phase 1 sorted by descending MRR values.

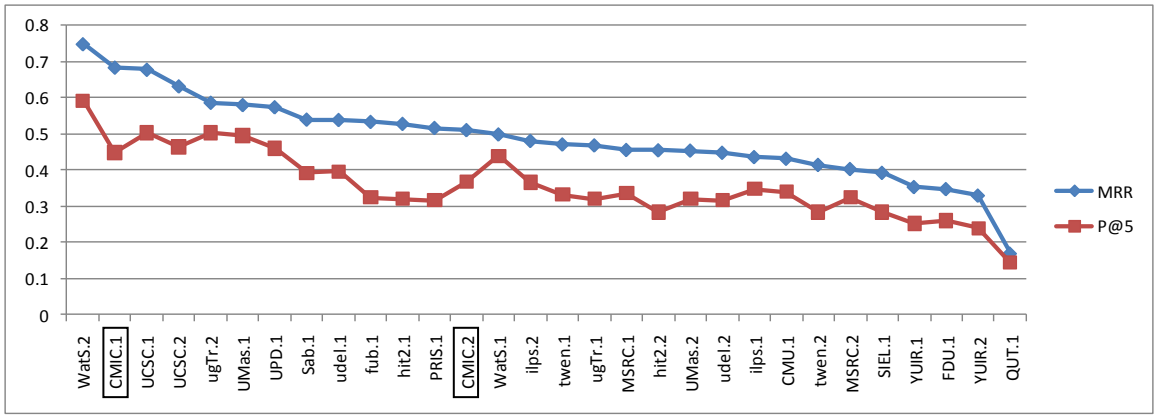

Figure 1. Phase 1 results using MRR and P@5

Using MRR as the metric of effectiveness, CMIC.1 and CMIC.2 appeared in positions 2 and 13 respectively compared to the other submissions. P@5 did not correlate perfectly with MRR. Using P@5, CMIC.1 and CMIC.2 appeared in positions 7 and 11 respectively compared to the other submissions.

It is worth noting at this point that many [1][2][3] consider using such traditional metrics such as MRR and P@5 insufficient to judge diversification results as it is not the goal of diversification to optimize for them. Thus, some efforts went into devising modified metrics that account for diversity [1][3].

Table 1 reports the official results phase 1 runs, where score is the ratio of runs that are better to the number of runs that are better and worse accumulated over all measures and groups using the overall average in the feedback step. For each the metrics for each of the runs, the 1st and 2nd numbers indicate the number of runs where the run did worse or better than respectively.

Table 1. Official phase 1 results

Topic	Emap	Map	P@10	stAP	Score
CMIC.1	6 27	0 0	0 0	9 24	0.7727
CMIC.2	8 5	5 9	4 10	9 4	0.5185

Table 2. Official phase 2 results

run ID	emap	stAP
Base	-	0.1582
CMIC.CMIC.1	0.0340	0.1511
CMIC.ugTr.2	0.0318	0.1520
CMIC.ilps.1	0.0314	0.1600
CMIC.UMas.2	0.0312	0.1409
CMIC.udel.1	0.0299	0.1363
CMIC.udel.2	0.0285	0.1216
CMIC.CMIC.2	0.0284	0.1293
CMIC.MSRC.1	0.0284	0.1331

As for the official phase 2 results, Table 2 reports expected mean average precision (emap) and statistical average precision (stAP) scores for all our phase 2 submissions sorted by emap. Unfortunately, we did not have access to the scores for submissions of other groups, which does not allow us to compare to other groups. The only information that was provided to us indicate that emap scores across all groups ranged between 0.0168 and 0.0536, and stAP scores ranged between 0.0434 and 0.2638.

Using all the above metrics, CMIC.1 which involved restricting phase 1 results to Wikipedia only did better than CMIC.2, which relied on density-based clustering.

5. Conclusion

We conclude that the use of diversification can benefit many retrieval scenarios especially when we would like to minimize the probability of a user not finding any relevant documents, hence it can be used for feedback tasks. The results suggest that relying on knowledge bases – namely Wikipedia in this work – can be more effective than unsupervised approaches such as cluster analysis. Although diversification has the potential of decreasing the number of relevant documents in the diversified results (5 in the case of phase 1 results), it also has the effect of increasing the probability of finding at least one relevant document, which would improve relevance feedback. We tried to make use of non-relevant documents in our query expansion scheme. Due to the fact that we don't have the full relevance judgments, we cannot ascertain the effect of accounting for relevant as well as non-relevant documents in relevance feedback, however, we hope that this can be more effective than using only relevant documents for feedback.

References

1. Agrawal, R., Gollapudi, S. Halverson, A. and Ieong, S. Diversifying search results. Proceedings of the Second ACM International Conference on Web Search and Data Mining, (Barcelona, Spain: ACM, 2009), pp. 5-14.
2. Carbonell, J. and Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. SIGIR-1998 (1998), pp. 335-336.
3. Chen, H. and Karger, D. R. Less is more: probabilistic models for retrieving fewer relevant documents. SIGIR-2006, (Seattle, Washington, USA: ACM, 2006), pp. 429-436.
4. Craswell, N., D. Fetterly, M. Najork, S. Robertson, E. Yilmaz. Microsoft Research at TREC 2009: Web and Relevance Feedback Track. TREC 2009 (2009).
5. Ester, M., Kriegel, H. P., Sander, J., and Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. KDD-96 (1996): 226-231.
6. G. Forman, "An extensive empirical study of feature selection metrics for text classification," J. Mach. Learn. Res., vol. 3, 2003, pp. 1289-1305.
7. Hastie, T., Tibshirani, R., and Friedman, J. The Elements of Statistical Learning. Data Mining, Inference and Prediction, Springer, New York, (2001).
8. Hearst, M. A. and Pedersen, J.O. Reexamining the cluster hypothesis: scatter/gather on retrieval results. SIGIR-1996 (Zurich, Switzerland: ACM, 1996), pp. 76-84.
9. Hinneburg, A. and Keim, D. An efficient approach to clustering in large multimedia databases with noise. Knowledge Discovery and Data Mining, 1998.
10. Jain, A., Murty, M., and Flyn, P. Data clustering: a review. ACM Computer Surveys 31 (3) (1999).
11. Jardine, N. and Van Rijsbergen, C. J. The use of hierarchic clustering in information retrieval. Information Storage and Retrieval, vol. 7, Dec. (1971), pp. 217-240.

12. Karypis, G., Han, E., and Kumar, V. CHAMELEON: a hierarchical clustering algorithm using dynamic modeling. Computer 32 (8) (1999) pp. 68–75.
13. Kaufman, L. and Rousseeuw, P. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, (1990).
14. Lee, K. S., Croft, W. B., and Allan, J. A cluster-based resampling method for pseudo-relevance feedback. SIGIR-2008 (2008).
15. Leuski, A. and Allan, J. Improving Interactive Retrieval by Combining Ranked Lists and Clustering. Proceedings of RIAO'2000, (2000), pp. 665--681.
16. Leuski, A. and Allan, J. Interactive Information Retrieval Using Clustering and Spatial Proximity. In User Modeling and User-Adapted Interaction 14: 259-288 (2004).
17. Liu, X. and Croft, W.B. Cluster-based retrieval using language models. SIGIR-2004, (Sheffield, United Kingdom: ACM, 2004), pp. 186-193.
18. Manning, C. D., Raghavan, P. and Schütze, H. Introduction to Information Retrieval. Cambridge University Press, (2008).
19. Ng, R., and Han, J. CLARANS: a method for clustering objects for spatial data mining, IEEE Trans. Knowl. Data Eng. 14 (5) (2002) 1003–1016.
20. Radlinski, F. and Dumais, S. Improving personalized web search using result diversification. SIGIR-2006 (Seattle, Washington, USA: ACM, 2006), pp. 691-692.
21. Salton, G. Automatic Information Organization and Retrieval., McGraw Hill Text, (1968).
22. Steinbach, M., Karypis, G., and Kumar, V. A comparison of document clustering techniques. In: Proc. of the KDD'2000 Workshop on Text Mining, (2000).
23. Wang, W., Yang, J., and Muntz, M. STING: a statistical information grid approach to spatial data mining. International Conference on Very Large Data Bases VLDB'97 (1997), pp. 186–195.
24. Yousri, N. A., Kamel, M. S., and Ismail, M. A. A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities. Pattern Recognition, vol. 42, (2009), pp. 1193-1209.
25. Zhai, C. and Lafferty, J. A risk minimization framework for information retrieval. Information Processing Management, vol. 42, (2006), pp. 31-55.
26. Zhai, C. X., Cohen, W. W., and Lafferty, J. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. SIGIR-2003, (2003), pp. 10-17.