When Just One is Enough

Chumki Basu *   Paul B. Kantor †

November 20, 1995

Abstract

Information retrieval systems today scan large data spaces in response to the user's information need, returning lists of retrievables ordered by relevance [11]. In the general case, the tendency is to return as many relevant items as the system can locate. However, in situations where the retrievables are highly similar to one another, one just may be enough. In this extended abstract, we describe an agent-based model of information retrieval and presentation wherein the user can settle for a single retrievable knowing that it satisfies the information need as well as any other similar retrievable.

1 Introduction

Retrieving information from networked-computing environments [13, 16] adds new dimensions to traditional document-based information retrieval[11]. Vast amounts of heterogeneous information are distributed across local and remote locations[3]. The agent paradigm provides a metaphorical bridge that permits transparency of access to this information[17, 5, 1, 2, 14]. Both the distributed nature of the data and agent-based access also broaden the notion of relevance[15]. Relevance can acquire new dimensions, e.g.,of timeliness, comprehensiveness, and authoritativeness of the retrievables. However, all of these increase the information "load" which must be processed by the agent and eventually, the end-user, upon examining the results. In this extended abstract, we present an abstract model of information retrieval and presentation which decomposes the feasible set of retrievables along multiple dimensions into a collection of smaller classes with the property that any representative of a class is as good as any of its members in satisfying the user's information need.

*Department of Computer Science, Rutgers University, cbasu@paul.rutgers.edu

†School of Communication, Library and Information Studies, Rutgers University, kantorp@cs.rutgers.edu

2 Motivation

For the purposes of the discussion, let's assume that a user client has access to a collection of documents which are distributed over a number of servers. We can describe a "waiter-agent" approach where one agent is customized to the needs of the client. The agent takes the client's query request and proceeds to search over all the servers to find the optimal match to the query[20].

Alternatively, we suggest the task of organizing a "buffet" of candidate documents for the user. In the real world, when implementing a buffet, the waiter-agent is replaced with multiple agents, working behind the scenes. The latter can be distinguished from the former in that their task is simplified: each of these "buffet" agents is responsible for choosing a buffet item from a predetermined set of highly similar items.

When a buffet is designed, an entire menu of possibilities is divided along a number of dimensions, inducing the formation of *classes* of individual items. An agent then presents a representative from each of these classes for consumption.

A buffet model can be described in terms of the methods for presentation and the methods for consumption. For example, a restaurateur must set criteria for how a buffet is organized, usually grouping similar foods together and choosing one from each group for inclusion in the buffet. Once this is done, it is the responsibility of the consumer to select items from this buffet which conform to her idea of a balanced meal. In this paper, we focus on the first issue of presentation or organization of of a similar food groups retrieval buffet.

3 Constructing a Buffet

First, we formally define what we mean by a buffet. In general terms, given a set of N documents, $D = \{d_1, d_2, ..., d_n\}$, a buffet, B, is a collection of subsets, $S^{(1)}$,...,$S^{(k)}$, containing elements of D such that $S^{(i)} \bigcap S^{(j)} = \emptyset, (i \neq j)$.

We have identified the need to decompose our universe of retrievables into smaller classes, along a fixed number of dimensions. One way to proceed is to relate the notion of dimension with the multi-facted nature of relevance. Relevance is no longer seen in static terms defined by just one criterion for comparison–instead we identify a number of similarity metrics, sim_i, which can be used to gauge relevance of a query, q, to a document, d_i, or the similarity of documents, d_i and d_j, to one another.

Each similarity metric is based on a particular scale of measurement. As previously stated, we can measure and compare documents on the basis of timeliness, authoritativeness, and comprehensiveness. These units of measure define dimensions. Therefore, every document can be measured according to k variables and plotted in k-dimensional space.

There seems to be a natural correspondence between what we have been calling a *class* and a cluster. A cluster is a collection of documents sharing certain properties. An agent can be responsible for all the documents belonging to a cluster. A particular buffet item is just a representative chosen from one of these clusters. We have considered clustering algorithms [6, 7, 10, 18], in particular, partitional clustering, which can be distinguished from hierarchical clustering techniques in that the latter do not permit reallocation of entities which may have been misclassified in early stages [7]. A number of partitional clustering techniques have been described in [6].

If we are given one similarity measure, we can perform ordinary clustering and identify, from the resultant clusters, the one most relevant to our needs. However, in our buffet, we assume each item originated from a distinct cluster. Hence, in the construction of our buffet we must perform an additional partitioning of the initial relevant cluster of documents, on the basis of our dimensions, to give us multiple relevant clusters.

Consider a typical retrieval scenario in which the user has requested the retrieval of 5 documents. She has not provided any additional information about what is considered relevant to her. Given this incomplete information, we can approximate the construction of a buffet of clusters wherein the items in each cluster are similar along a particular dimension. We do this by choosing one dimension and one similarity metric (for gauging similarity along that dimension) and grouping retrievables into clusters such that the items in any one cluster are highly similar to each other and the items in different clusters are highly dissimilar to each other. We consider different measures of intra-cluster *cohesiveness* and inter-cluster *distinctiveness* and are interested in exploring the interplay between these measures.

Cohesiveness

In this discussion, we refer to individual items in a cluster as cases. In the restaurant world, a case corresponds to a food item; in on-line retrieval, a case can correspond to a text document. Once we have decided what the cases are, we can define the measurement to be taken per case and the distance between two cases, in terms of this measurement. We shall rely on word frequency as our unit of measurement, i.e., the number of occurrences of each word in the document [6]. The distance measures we consider are Euclidean distance[10], cosine similarity [4], and group average similarity [6]. We use these distance measures to define similarity of one case to another.

For each of the distance measures, we optimize the sum of the distances between cases within the same cluster. In the first situation, we minimize the Euclidean distance between every case and the mean or average case of the cluster. For cosine similarity, we maximize the cosine similarity between every case and the mean. Finally, for group average similarity, we maximize the average cosine similarity between every pair of cases in the cluster. Each of the above distance measures can be considered as a *cohesiveness* measure for any particular cluster. As the "glue" which holds a cluster together, this measure serves as our primary objective function.

Distinctiveness

In addition to looking at intra-cluster distance, we must also examine inter-cluster distances. In other words, although we want our cluster diameters to be as small as possible, we would like the "spread" between clusters to be as large as possible [12]. This spread can be considered the *distinctiveness* of a cluster; we compute it to be the total inter-group distance. In the future, we also would like to consider entropy measurements as part of the definition of the distinctiveness of a cluster. This distinctiveness measure can serve as a secondary objective function.

4 Experiments

In the following section, we describe a series of experiments we conducted on a corpus of data taken from the TREC4 routing task. The purpose of our experiments is to examine the role of the cohesiveness and the distinctiveness measures in the formation of related document clusters.

For each of the experiments, we apply the *k-means* clustering algorithm as follows[10]:

- Start with an initial clustering. Compute cluster means and and the initial error. This error will be the primary objective function to optimize.
- For every case, in turn, compute the decrease (increase) in the objective function resulting from moving the case from the current cluster to every other cluster.
- For every case, move it to the cluster for which the objective function is decreased (increased) the most.
- Continue to examine cases, one at a time, until there is one complete pass through the cases which does not result in movement from one cluster to another.

As a stepwise-optimal algorithm, *k-means* starts with an initial partition and proceeds to search neighboring partitions in search of improving the objective function [10, 18].

We assembled the set of all documents that had been judged to be relevant to each of the so-called "routing topics" for the TREC4 text retrieval conference (NIST). The relevance scores and documents are available via ftp, but access is restricted to participants in the TREC workshops. We then selected groups of documents which were relevant to each of 5 distinct topics or problems. Our working assumption is that a "good" clustering algorithm will find that there are exactly 5 clusters in this set of documents, and will assign documents to clusters in a way that corresponds to their original relation to the topics or problems to which they have been judged relevant.

Each document is broken up into constituent words and the frequency of distinct words is noted. A document vector whose components are the frequencies of the distinct words is formed. Using a stop-list provided in [8], we narrow the total number of distinct words in our corpus to 3473.

Now, we are ready to form our initial partition. To test the extent to which the topic-based clustering is preserved under the previously stated cohesiveness measures, we performed two tests – one in which the initial partition is the final partition we expect and another in which, in the initial partition, three cases are moved away from the the expected final partition. For this experiment, we applied the *k-means* algorithm to the sum of squared Euclidean distances (ESS) and to the cosine similarity cohesiveness measures. We also used the n-gram center of gravity approach [4] in addition to stop-words for cosine similarity.

We score our results on the basis of the number of incorrect pairings of documents within the final clusters. Within the original groups of relevant documents, we can count those pairs of documents which fall within a particular group and those pairs which do not. This is assumed to be the correct grouping and hence these pairs are deemed correct. This calculation is done once at the beginning. Then for every final clustering we get, we examine each cluster and recalculate those pairs which fall in the same group and those which do not. We can then compute the total number of incorrect pairings compared to the original groups and divide this number by the total number of possible pairs. Let us call these scores, the "badness" values; the lower the scores, the better our final cluster conforms to the expected clustering.

We also measured the average "distinctiveness" scores for each of the final clusterings. For any two clusters, this measure is equal to the distance between their means [12]. We then compute the average over all cluster pairs.

Our results for the first test (using 5 clusters) are:

Method	Badness Score	Average Distinctiveness
ESS	.34	54.98
Cosine	.03	.11
Cosine (center of gravity)	.23	.052

Our results for the second test (using 5 clusters) are:

Method	Badness Score	Average Distinctiveness
ESS	.21	52.78
Cosine	.03	.11
Cosine (center of gravity)	.21	.071

5 Discussion

The above results bring out some important points. First, various measures of similarity have more or less ability to separate documents into relevant topic clusters (as determined in TREC4). The cosine measure performs the best of the three reported, although what it considers the best clustering still has one case appearing in a different relevant topic cluster.

Early evidence suggests that using stop-words plus the center of gravity approach may perform better at achieving distinctiveness, but does not do as well at achieving correct classifications.

6 Related Work

There are a number of excellent references on clustering algorithms [7, 10, 12, 18, 9]. Hartigan classifies clustering algorithms in terms of their "modes of search." Based on our description, the search for new buffet items falls best under the general category of "adding" (objects to existing clusters).

In exploring the interplay between the ideas of the cohesiveness and the distinctiveness of clusters, our work is conceptually, closely related to the *Scatter/Gather* method [6] and to some of the ideas used in the *Acquaintance* technique by Damashek[4]. Cutting, et. al [6] make a case for the use of document clustering techniques in the context of browsing. We hope that document clustering may prove viable for goal-directed pursuits such as search, if we relax the requirement for retrieving the optimal matching document.

7 Future Work

In this extended abstract, we have presented our preliminary ideas for the buffet model. At this time, we plan further experimentation and analysis to test our claims and determine the effectiveness of a buffet.

References

[1] M. Balabanovic and Y. Shoham. Learning Information Retrieval Agents: Experiments with Automated Web Browsing, *AAAI-95 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments,* 1995.

[2] W. P. Birmingham, E. H. Durfee, T. Mullen, and M. P. Wellman. The Distributed Agent Architecture of the University of Michigan Digital Library. University of Michigan Department of Computer Science Technical Memo. (1995)

[3] C. M. Bowman, P. B. Danzig, U. Manber, and M. F. Schwartz. Scalable Internet Resource Discovery: Research Problems and Approaches. In *Communications of the ACM,* pp. 98-114, Vol. 37, No. 8, August 1994.

[4] M. Damashek. Gauging Similarity via N-Grams: Language-Independent Sorting, Categorization, and Retrieval of Text. In *Science,*267, Feb. 1995.

[5] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proceedings of IPSJ Conference*, Tokyo, Japan, October, 1994.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, J. W. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the 15th annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 318-329, June 1992.

[7] B. Everitt. *Cluster Analysis.* Halstead Press, 1974.

[8] W. B. Frakes and R. Baeza-Yates (eds). *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, 1992.

[9] H. P. Friedman and J. Rubin. On some invariant criteria for grouping data. In *Journal of the American Statistical Association.*62, pp. 1159-1178.

[10] J. A. Hartigan. *Clustering Algorithms.* John Wiley and Sons, 1975.

[11] P. B. Kantor. Information Retrieval Techniques. In M. E. Williams, editor, *Annual Review of Information Science and Technology,*pp. 53-90, Vol. 29, ASIS, 1994.

[12] M. Kendall. *Multivariate Analysis*, Charles Griffin, and Co., 1980.

[13] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Frystyk Nielsen, A. Secret. The World-Wide Web. In *Communications of the ACM,*pp. 76-82, Vol. 37, No. 8, August 1994.

[14] A. Reinhardt. The Network with Smarts. *BYTE,* October 1994.

[15] L. Schamber. Relevance and Information Behavior. In M. E. Williams, editor, *Annual Review of Information Science and Technology,*pp. 3-48, Vol. 29, ASIS, 1994.

[16] B. Schatz and J. B. Hardin. NCSA Mosaic and the World Wide Web: Global Hypermedia Protocols for the Internet. In *Science,* pp. 895-901, Vol. 265, August 12, 1994.

[17] B. Sheth and P. Maes. Evolving Agents for Personalized Information Filtering. In *Proceedings of the 9th IEEE Conference on AI for Applications,* 1993.

[18] H. Spath. *Cluster Analysis Algorithms for data reduction and classification of objects*, Halstead Press, 1980.

[19] UMDL: The University of Michigan Digital Library Project. *http://http2.sils.umich.edu/UMDL/HomePage.html.* (1995)

[20] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning Collection Fusion Strategies. In *Proceedings of the 18th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 172-179, July, 1995.