Evaluating Hierarchical Clustering of Search Results

Juan M. Cigarran, Anselmo Peñas, Julio Gonzalo, and Felisa Verdejo

Dept. Lenguajes y Sistemas Informáticos, E.T.S.I. Informática UNED**
{juanci, anselmo, julio, felisa}@lsi.uned.es

Abstract. We propose a goal-oriented evaluation measure, *Hierarchy Quality*, for hierarchical clustering algorithms applied to the task of organizing search results -such as the clusters generated by *Vivisimo* search engine-. Our metric considers the content of the clusters, their hierarchical arrangement, and the effort required to find relevant information by traversing the hierarchy starting from the top node. It compares the effort required to browse documents in a baseline ranked list with the minimum effort required to find the same amount of relevant information by browsing the hierarchy (which involves examining both documents and node descriptors).

1 Motivation

Clustering search results is an increasingly popular feature of web search and meta-search engines; examples include many industrial systems like Vivisimo, Kartoo, Mooter, Copernic, IBoogie, Groxis, Dogpile and Clusty and research prototypes such as Credo [1] or SnakeT [4].

For navigational search goals (finding a specific web site), clustering is probably useless. But for complex information needs, where there is a need to compile information from various sources, hierarchical clustering has clear advantages over plain ranked lists. According to [9], around 60% of web searches are informational, suggesting that techniques to organize and visualize search results can play an increasingly important role in search engines.

Although evidences show that this kind of systems seems to work fine and may be helpful, there is not a consensus about what kind of metrics are suitable to evaluate and to compare them in a quantitative way.

The research described in this paper attempts to design task-oriented evaluation metrics for hierarchical clusters that organize search results, according to their ability of minimizing the amount of irrelevant documents that has to be browsed by the user to access all the relevant information. Our metrics compare generated clusters with the original ranked list considering the navigational properties of the cluster, counting all clusters with relevant information and also

** This work has been partially supported by the Spanish Ministry of Science and Technology within the project(TIC-2003-07158-C04) Answer Retrieval From Digital Documents (R2D2).

M. Consens and G. Navarro (Eds.): SPIRE 2005, LNCS 3772, pp. 49–54, 2005.

how they are interconnected. Although our metrics compare the original ranked list with the clustering created, we have to remark that we work in an scenario where recall is maximum.

Regarding task-oriented clustering, there has been some attempts to evaluate the quality of clusters for Information Retrieval tasks. Scatter/Gather system [5], for instance, generates a hierarchical clustering but the evaluation issues are only focused on the highest scoring cluster to be compared with the original ranked list of documents and without considering the effort required to reach it. [8] also assume that a good clustering algorithm should put relevant and nonrelevant documents into separate classes but, again, in a flat approach that ignores the hierarchical and navigational properties of the cluster. More recently, [7] and [6] consider the hierarchical properties of the clusters in a task-oriented evaluation methodology and propose metrics to measure the time spent to find all the relevant information. [7] scores hierarchies estimating the time it takes to find all relevant documents by calculating the total number of nodes that must be traversed and the number of documents that must be read. In this case, the measure is used to compare the structure of hierarchies built using different approaches. The algorithm calculates the optimal path to each relevant document and then averages the results. No differences between the cognitive cost of reading a document and a cluster description are made in this approach. [6] present a similar strategy that compares the retrieval improvements of different clustering strategies versus the original ranked list. They consider differences between the cognitive cost of reading a document and a cluster description but, again, the algorithm operates over each relevant document separately and then averages the results. Our metrics do not calculate an optimal path to each relevant document averaging the results. Instead, a Minimal Browsing Area (MBA) (i.e. an optimal area within the hierarchy containing all the relevant documents) is calculated and then the measures (i.e. Distillation Factor and Hierarchical Quality) are applied. MBA is a way to reflect the power of the clustering to isolate relevant information allowing to work with it as a whole (i.e. we suppose the user is going to access all the relevant documents but taking the advantages of the set of nodes previously traversed). This approach relies on the idea of working at maximum recall.

The paper is organized as follows. First we present some preliminaries about how should be a good clustering organization and the basic assumptions in which our metrics are based. Then we explain the metrics, Distillation Factor and Hierarchical Quality and finally we present the conclusions and the future work.

2 Basic Assumptions

At least, four features of a hierarchical clustering have to be considered: a) *the content of the clusters*. A clustering that effectively groups relevant information is better than a clustering that mixes relevant and non-relevant documents; b) *the hierarchical arrangement of the clusters*. For instance, if the user has to browse a chain of clusters with irrelevant information before reaching a cluster with several relevant documents, the clustering is non-optimal for the task;

c) *the number of clusters*. To be effective, the number of clusters that have to be considered to find all relevant information should be substantially lower than the actual number of documents being clustered. Otherwise, the cognitive cost of examining a plain ranked list would be smaller than examining the cluster hierarchy, and; d) *how clusters are described*. A good cluster description should help the user predicting the relevance of the information that it contains. Optimal cluster descriptions will discourage users from exploring branches that only contain irrelevant information. We think that this last feature cannot be measured objectively without user studies. But before experimenting with users - which is costly and does not produce reusable test beds - we can optimize hierarchical clustering algorithms according to the first three features: contents of the clusters, number of clusters and hierarchical arrangement. Once the clustering is optimal according to these features, we can compare cluster descriptions by performing user studies. In this paper we will not discuss the quality of the cluster descriptions, focusing on those aspects that can be evaluated objectively and automatically (given a suitable test bed with relevance judgments).

In order to make our metrics to work, we propose the following two main assumptions: a) the first assumption considers that a hierarchical clustering should build each cluster only with those documents fully described by its descriptors. For instance, a cluster about *physics* with sub-clusters such as *nuclear physics* *astrophysics* should only contain those generic documents about physics, but not those about nuclear physics or astrophysics. Moreover, specific documents about any of the physics subtopics should be place in lower clusters (i.e. otherwise, we do not have a hierarchical clustering but only a hierarchical description of clusters). From the evaluation point of view, it has no sense to have a top high level cluster containing all the documents retrieved because it will will force the user to read the whole list at the very first time. This approach is the same as used in web directories and is considered the natural way of browsing hierarchical descriptions; b) the second assumption considers that clustering is made with an 'open-world' view. This means that, if a document is about very different topics, it should be placed (i.e. repeated) in its corresponding topic clusters and, as a consequence, it should appear in different parts of the hierarchy. For instance, if we have a document about *physics* which also includes some *jokes about physics* and a clustering hierarchy with clusters about *physics* and *jokes* without any connection between them, clustering should place the document anywhere in the physics hierarchy and it should repeat it in the jokes hierarchy. This 'open-world' view is more realistic than the classical 'closed-world' view, applied by some hierarchical clustering algorithms, and where a document only belongs to one part of the hierarchy. As a drawback of this assumption, it is very difficult to deal with evaluation issues when repeated documents appear in different parts of the hierarchy. As a possible solution, we propose the use of lattices instead of hierarchies as the data models used to represent the clustering. From the modeling and browsing point of view there are no differences between lattices and hierarchies and it is always possible to unfold a lattice into a hierarchy and

viceversa. [2], [3] and [1] show how to deal with concept lattices in a document clustering framework.

3 Evaluation Measures

Let us start with a ranked list obtained as a result of a search which is going to be clustered using a lattice. Let N be the set of nodes of the lattice L, where each node is described by a pair $(DOCS, DESC)$, with $DOCS$ the set of documents associated to the node (but not to its subnodes) and $DESC$ the description that characterizes the cluster (if any).

Our proposal is to measure the quality of a lattice, for the purpose of browsing search results, as the potential reduction of the cognitive load required to find all the relevant information as compared to the original ranked list. This can be expressed as a gain factor:

$$\text{Quality(lattice)} \equiv \frac{\text{cognitive load(ranked list)}}{\text{cognitive load}(lattice)}$$

The effort required to browse a ranked list is roughly proportional to the number of documents in the list. Of course, the non-trivial issue is how to estimate the effort required to browse the lattice. In the remainder of this section, we will discuss two approaches to this problem: the first one is an initial approach that only considers the cost of examining documents. The second approach, in addition, also considers the cost of taking decisions (which nodes to explore, which nodes to discard) when traversing the lattice.

3.1 Distillation Factor

Let us assume that the user begins browsing the lattice at the top node; Let us also assume that, in our evaluation testbed, we have manual assessments indicating whether each document is relevant or not for the query that produced the ranked list of documents.

We can then define the *Minimal Browsing Area* (MBA) as the smallest part of the lattice that has to be explored by the user to find all relevant information contained in the lattice (i.e. a complete description of how to build MBA can be found in [2]). If we compute the cognitive cost of exploring the lattice as the cost of examining the documents contained in the MBA, then the quality of a lattice L would be given by the ratio between the cost of examining the list of documents and the cost of examining the documents in the MBA.

Then, if k_d is the cognitive cost of examining a document, $D_{RankedList}$ is the number of documents in the ranked list, and D_{MBA} is the number of documents in the minimal browsing area, then the quality of the lattice, according to the definition above, would be:

$$\text{DF}(L) = \frac{k_d * D_{RankedList}}{k_d * D_{MBA}} = \frac{D_{RankedList}}{D_{MBA}}$$

We call this measure *Distillation Factor* (DF), because it describes the ability of the lattice to 'distill' or filter relevant documents in a way that minimizes user effort. Its minimum value is 1 (when all nodes in the lattice have to be examined to retrieve all relevant documents), which means that there is no improvement over the ranked list.

Notice that DF can also be seen as the factor between the *precision* of the set of documents in the MBA and the *precision* of the original ranked list:

$$DF(L) = \frac{D_{Relevant}/D_{MBA}}{D_{Relevant}/D_{RankedList}} = \frac{D_{RankedList}}{D_{MBA}}$$

3.2 Hierarchy Quality

The DF measure is only concerned with the cost of reading documents, but browsing a conceptual structure has the additional cost (i.e. as compared to a ranked list of documents) of examining node descriptors and deciding whether each node is worth exploring. For instance, a lattice may lead us to n relevant documents and save us from reading another m irrelevant ones, ... but force us to traverse a thousand nodes to find the relevant information! Obviously, the number of nodes has to be considered when computing the cost of using the lattice to find relevant information.

To compute the cost of browsing the lattice we need to count all node descriptions that have to be considered to explore all relevant nodes. Let us call this set of nodes N_{view}. It can be computed starting with the nodes in the MBA and adding the lower neighbors of every node in the MBA.

Then, if the cognitive cost of examining a node description is k_n, the quality of the lattice can be defined as:

$$\mathrm{HQ}(L) = \frac{k_d * D_{RankedList}}{k_d * D_{MBA} + k_n * |N_{view}|} = \frac{D_{RankedList}}{D_{MBA} + \frac{k_n}{k_d}|N_{view}|}$$

We call this measure *Hierarchy Quality* (HQ). This is the improved measure that we propose to evaluate the quality of a lattice for the task of organizing and visualizing search results. Note that it depends on a parameter $k \equiv \frac{k_n}{k_d}$, which estimates the ratio between the effort needed to examine a document (i.e. its title, its snippet or another kind of description) and the effort required to examine a node description. This value has to be settled according to the retrieval scenario and the type of node descriptions being considered.

Unlike the Distillation Factor, the HQ measure can have values below 1, if the number of nodes to be considered is too large. In this case, the HQ measure would indicate that the lattice is worse than the original ranked list for browsing purposes.

Of course, this formula implies to fix (i.e. for each retrieval scenario) a value for k. This value has a strong influence on the final HQ values and should be estimated conducting user studies.

4 Conclusions

We have introduced task-oriented evaluation metrics for hierarchical clustering algorithms that organize search results. These metrics consider the features of a good clustering for browsing search results. Our main evaluation measure, HQ (*Hierarchy Quality*), compares the cognitive effort required to browse the lattice, with the effort required to browse the original ranked list of results. Our measure is computed using the concept of *Minimal Browsing Area* and the related concept N_{view} of minimal set of node descriptions which have to be considered in order to traverse the minimal browsing area. Our future work aims at trying to estimate the value of k over different kinds of documents and cluster descriptions using different clustering strategies.

References

1. C. Carpineto and G. Romano. *Concept Data Analysis. Data and Applications.* Wiley, 2004.
2. J. Cigarran, J. Gonzalo, A. Peñas, and F. Verdejo. Browsing search results via formal concept analysis: Automatic selection of attributes. In *Concept Lattices.* Second International Conference on Formal Concept Analysis, ICFCA 2004, Springer.
3. J. Cigarran, A. Peñas, J. Gonzalo, and F. Verdejo. Automatic selection of noun phrases as document descriptors in an fca-based information retrieval system. In *Formal Concept Analysis.* Third International Conference, ICFCA 2005, Springer, 2005.
4. P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 801–810, New York, NY, USA, 2005. ACM Press.
5. M. Hearst and J. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 76–84, Zurich, CH, 1996.
6. K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 658–665, New York, NY, USA, 2004. ACM Press.
7. D. Lawrie and W. Croft. Discovering and comparing topic hierarchies. In *Proceedings of RIAO 2000.*, 2000.
8. A. Leouski and W. Croft. An evaluation of techniques for clustering search results, 1996.
9. D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM Press, 2004.