Active Information Retrieval

Tommi Jaakkola  
MIT AI Lab  
Cambridge, MA  
tommi@ai.mit.edu

Hava Siegelmann  
MIT LIDS  
Cambridge, MA  
hava@mit.edu

Abstract

In classical large information retrieval systems, the system responds to a user initiated query with a list of results ranked by relevance. The users may further refine their query as needed. This process may result in a lengthy correspondence without conclusion. We propose an alternative active learning approach, where the system responds to the initial user’s query by successively probing the user for distinctions at multiple levels of abstraction. The system’s initiated queries are optimized for speedy recovery and the user is permitted to respond with multiple selections or may reject the query. The information is in each case unambiguously incorporated by the system and the subsequent queries are adjusted to minimize the need for further exchange. The system’s initiated queries are subject to resource constraints pertaining to the amount of information that can be presented to the user per iteration.

1 Introduction

An IR system consists of a collection of documents and an engine that retrieves documents described by users queries. In large systems, such as the Web, queries are typically too vague, and hence, an iterative process in which the users refine their queries gradually has to take place. Since much dissatisfaction of IR users stems from long, tedious repetitive search sessions, our research is targeted at shortening the search session. We propose a new search paradigm of active information retrieval in which the user initiates only one query, and the subsequent iterative process is led by the engine/system. The active process exploits optimum experiment design to permit minimal effort on the part of the user.

Our approach is related but not identical to the interactive search processes called relevance feedback. The primary differences pertain to the way in which the feedback is incorporated and queried from the user. In relevance feedback, the system has to deduce a set of “features” (words, phrases, etc.) that characterize the set of selected relevant documents, and use these features in formulating a new query (e.g., [5, 6]). In contrast, we cast the problem as a problem of estimation and the goal is to recover the unknown document weights or relevance assessments.
Our system also relates to the Scatter/Gather algorithm of browsing information systems [2], where the system initially scatters the document collection into a fixed number $k$ of clusters whose summaries are presented to the user. The user selects clusters from a new sub-collection, to be scattered again into $k$ clusters, and so forth, until enumerating single documents. In our approach, documents are not discarded but rather their weighting is updated appropriately. Like many other clustering methods, the scatter/gather is based on hierarchical orderings. Overlapping clusters were recently proposed to better match real-life grouping and allow natural summarizing and viewing [4].

This short paper focuses on the underlying methodology of the active learning approach.

2 Active search

Let $\mathcal{X}$ be the set of documents (elements) in the database and $\mathcal{C} = \{C_1, \ldots, C_m\}$ a set of available clusters of documents for which appropriate summaries can be generated. The set of clusters typically includes individual documents and may come from a flat, hierarchical, or overlapping clustering method. The clustering need not be static, however, and could be easily defined dynamically in the search process.

Given the set of available clusters, we may choose a query set, a limited set of clusters that are presented to the user for selection at each iteration of the search process. The user is expected to choose the best matching cluster in this set or, alternatively, annotate the clusters with relevant/irrelevant labels (select the relevant ones). We will address both modes of operation.

The active retrieval algorithm proceeds as follows: (1) it finds a small subset $\mathcal{S}$ of clusters to present, along with their summaries, to the user; (2) waits until the user selects one, or more, of the presented clusters; (3) uses the evidence from the user’s selections to update the distribution over documents or relevance assessments; (4) outputs the top documents so far, ranked by their weights, and the iteration continues until terminated by the user or the system (based on any remaining uncertainty about the relevant documents or the implied ranking).

The following sections address three primary issues: the user model, how to incorporate the information from user selections, and how to optimize the query set presented to the user. All the algorithms should scale linearly with the database size (and the size of the query set).

3 Contrastive selection model

We start with a contrastive selection model where the user is expected to choose only the best matching cluster in the query set. In case of multiple selections, we will interpret the marked clusters as a redefined cluster of the query set. While this interpretation will result in sub-optimal choices for the query set assuming the user consistently selects multiple clusters, the interpretation nevertheless obviates the need for modeling user’s selection biases in this regard. An empty selection, on the other hand, suggests that the clusters outside the query set are deemed more likely.