Towards Semantic Search

Ricardo Baeza-Yates, Massimiliano Ciaramita,
Peter Mika, and Hugo Zaragoza

Yahoo! Research, Barcelona, Spain

Abstract. Semantic search seems to be an elusive and fuzzy target to many researchers. One of the reasons is that the task lies in between several areas of specialization. In this extended abstract we review some of the ideas we have been investigating while approaching this problem. First, we present how we understand semantic search, the Web and the current challenges. Second, how to use shallow semantics to improve Web search. Third, how the usage of search engines can capture the implicit semantics encoded in the queries and actions of people. To conclude, we discuss how these ideas can create virtuous feedback circuit for machine learning and, ultimately, better search.

1 Introduction

From the early days of Information Retrieval (IR), researchers have tried to take into account the richness of natural language when interpreting search queries. Early work on natural language processing (NLP) concentrated on tokenisation and normalisation of terms (detection of phrases, stemming, lemmatisation, etc) and was quite successful [11]. Sense disambiguation (needed to differentiate between the different meanings of the same token) and synonym expansion (needed to take into account the different tokens that express the same meaning) seemed the obvious next frontier to be tackled by researchers in IR. There were a number of tools available that made the task seem easy. These tools came in many forms, from statistical methods to analyse distributional patterns, to expert ontologies such as WordNet. However, despite furious interest in this topic, few advances were made for many years, and slowly the IR field moved away: concepts like synonymy were no longer discussed and topics such as term disambiguation for search were mostly abandoned [5]. In lack of solid failure analysis data we can only hypothesize why this happened and we try to do so later.

Semantic search is difficult because language, its structure and its relation to the world and human activities, is complex and only partially understood. Embedding a *semantic model*, the implementation of some more or less principled model of both linguistic content and background knowledge, in massive applications is further complicated by the dynamic nature of the process, involving millions of people and transactions. Deep approaches to model meaning have failed repeatedly, and clearly the scale of the Web does not make things easier. In [6] we argue that semantic search has not occurred for three main reasons. First, this integration is an extremely hard scientific problem. Second, the Web

E. Kapetanios, V. Sugumaran, M. Spiliopoulou (Eds.): NLDB 2008, LNCS 5039, pp. 4–11, 2008.

imposes hard scalability and performance restrictions. Third, there is a cultural divide between the Semantic Web (SW) and IR disciplines. Our research aims at addressing these three issues.

Arguably, part of the reason the Web and search technologies have been so successful is because the underlying language model is extremely simple, and people expected relatively little from it. Although search engines are impressive pieces of engineering they address a basic task: given a query return a ranked list of documents from an existing collection. In retrospect, the challenges that search engines have overcome, mostly have to do with scalability and speed of service, while the ranking model which has supported the explosion of search technology in the past decade is quite straightforward and has not produced major breakthroughs since the formulation of the classic retrieval models [4] and the discovery of features based on links and usage. Interestingly, however, the Web has created an ecosystem where both content and queries have adapted. For example, people have generated structured encyclopedic knowledge (*e.g.*, Wikipedia) and sites dedicated to multimedia content (*e.g.*, Flickr and YouTube). At the same time users started developing novel strategies for accessing this information; such as appropriate query formulation techniques ("mammals Wikipedia" instead of just "mammals"), or invented "tags" and annotated multimedia content otherwise almost inaccessible (videos, pictures, etc.) in the classic retrieval framework because of the sparsity of the associated textual information.

Clearly the current state of affairs is not optimal. One of our lines of research, semantic search, addresses these problems. To present our vision and our early findings, we first detail the complexity of semantic search and its current context. Second, we survey some of our initial results on this problem. Finally, we mention how we can use Web mining to create a virtuous feedback circle to help our quest.

2 Problem Complexity and Its Context

Search engines are hindered by their limited understanding of user queries and the content of the Web, and therefore limited in their ways of matching the two. While search engines do a generally good job on large classes of queries (*e.g.* navigational queries), there are a number of important query types that are undeserved by keyword-based approach. Ambiguous queries are the most often cited examples. In face of ambiguity, search engines manage to mask their confusion, by (1) explicitly providing diversity (in other words, letting the user choose) and (2) relying on some notion of popularity (*e.g.* PageRank), hoping that the user is interested in the most common interpretation of the query. As an example of where this fails, consider searching for George Bush, the beer brewer. The capabilities of computational advertising, which is largely also an information retrieval problem (*i.e.* the retrieval of the matching advertisements from a fixed inventory), are clearly impacted due to the relative sparsity of the search space. Without understanding the object of the query, search engines are also unable to perform queries on descriptions of objects, where no key exists. A typical, and important example of this category is product search. For example, search engines are unable to look for "music players with at least 4GB of RAM"