Searching the Wikipedia with Contextual Information

Antti Ukkonen[†][*]
antti.ukkonen@tkk.fi

Carlos Castillo[‡]
chato@yahoo-inc.com

Debora Donato[‡]
debora@yahoo-inc.com

Aristides Gionis[‡]
gionis@yahoo-inc.com

[†]Helsinki University of Technology
Helsinki, Finland

[‡]Yahoo! Research Labs
Barcelona, Spain

ABSTRACT

We propose a framework for searching the Wikipedia with contextual information. Our framework extends the typical keyword search, by considering queries of the type $\langle q, p \rangle$, where q is a set of terms (as in classical Web search), and p is a *source* Wikipedia document. The query terms q represent the information that the user is interested in finding, and the document p provides the *context* of the query. The task is to rank other documents in Wikipedia with respect to their relevance to the query terms q given the context document p. By associating a context to the query terms, the search results of a search initiated in a particular page can be made more relevant.

We suggest a number of features that extend the classical query-search model so that the context document p is considered. We then use RankSVM (Joachims 2002) to learn weights for the individual features given suitably constructed training data. Documents are ranked at query time using the inner product of the feature and the weight vectors. The experiments indicate that the proposed method considerably improves results obtained by a more traditional approach that does not take the context into account.

Categories and Subject Descriptors: H.3.3 Information Search and Retrieval: Retrieval models

General Terms: Algorithms, Experimentation

1. INTRODUCTION

We consider the problem of incorporating *contextual information* while performing a keyword-search in a hyperlinked document collections, such as Wikipedia, or the Web. We model contextual information by considering one document of the collection to be the *context page*, which, for instance can be the page that the user is currently browsing when submitting a query. By attaching a context to the query terms, the search results of a search initiated in a particular page can be made more relevant, for example, the context page can be used for disambiguation. For instance, a relevant answer for the query "*apple*", given the query page `en.wikipedia.org/wiki/Fruit` is the page `en.wikipedia.org/wiki/Apple`, while a relevant answer to the query "*apple*", given the query page `en.wikipedia.org/wiki/Computer` is the page `en.wikipedia.org/wiki/Apple_Inc`.

[*]Part of this work was done while the author was visiting Yahoo! Research Labs, Barcelona

2. OVERVIEW OF THE APPROACH

We model Wikipedia as a directed graph $G = (V, E)$, where a document d_v is associated to each node $v \in V$. The problem of context-sensitive search in the graph G is defined as follows. A query consists of a pair $\langle q, p \rangle$, where q is a set of terms as usual in Web search, and p is the query node of the graph G that provides the context for q. The task is to rank the nodes of G in decreasing order of their relevance to the query $\langle q, p \rangle$.

The basic approach for solving the problem is to compute a score for each node $v \in V$, so that the score measures the relevance of the node v with respect to the query $\langle q, p \rangle$. We construct the scoring function as a linear combination of a number of features, the weights of which are learned by using RankSVM [5]. Let d_v denote the page whose score we are computing. The features we use are BM25, the Jaccard coefficient between the context document d_p and d_v, the Jaccard coefficient between the successors of p and v in G, the Jaccard coefficient between the predecessors of p and v in G, the spectral distance between p and v, and a context-sensitive PageRank [2, 4] of v given p. The spectral distance is obtained by embedding the vertices of G to $\mathbb{R}^n$ [3, 6, 7], and computing their euclidean distance in this space.

The ideal context-sensitive PageRanks are based on adjusting the teleport vector so that the random walk always jumps back to the context vertex p. As we can not compute the PageRanks for every possible p, and computing PageRank at query time is in practice very hard, we consider two heuristics to obtain context-sensitive PageRanks. The first is based on random landmarks, where we compute the adjusted PageRanks for fixed number of vertices of G, and at query time use the PageRanks that correspond to a landmark that is closest to p in G. The second approach is based on clustering the graph G, and computing the context-sensitive PageRanks separately for each cluster. Here the teleport vector is modified so that the random walk jumps back to any vertex that belongs to the cluster with the same probability. At query time we use the PageRanks that correspond to the cluster p belongs to.

In order to combine all these features in a single scoring function we use a machine-learning approach: first we compute the proposed features for a given set of (ambiguous) queries, which we complement with ground-truth information about which nodes should be ranked preferably than others for each query. From this training set and using the RankSVM algorithm we learn the weights of the scoring function.

The set of relevant nodes to which the scoring function is applied is formed by first finding all documents that are at a distance of at most k links from p. These are found simply by running a BFS at query time starting from p. Of these documents only those are retained that contain all terms in q. The remaining documents are ranked according to the learned scoring function. The motivation for using BFS comes from our initial experiments in which it was shown that most of the "correct" (underlying truth) documents are within short a distance from p. We note that the use of BFS was proven to be an effective strategy for the search problem in Wikipedia. The situation might be different in other applications.

3. EXPERIMENTS

We implemented a prototype system using the ideas discussed above for context-sensitive search in the English Wikipedia. The system is built over a snapshot from April 2007 with roughly 5 million pages, of which about 1.7 million are real articles, with the others being mostly redirect pages or images. We use the Terrier Information Retrieval Platform [8] for indexing and the Webgraph library [1] for storing the links between articles.

3.1 Evaluation methodology

Wikipedia resolves the problem of ambiguous entity names by the so called disambiguation pages. For example, when issuing the query "bach" in the regular Wikipedia search, the result is the page about J. S. Bach, on top of which is a link to a page called "Bach (disambiguation)". This page is a list of links to other persons (or things) called "bach", usually with short descriptions.

We make use of the disambiguation pages for evaluating our framework. To create a test query $\langle q, p\rangle$ we proceed by first picking one disambiguation page at random. The title of this disambiguation page defines our query string q. Then we pick a random page that has a link from the disambiguation page. We call this page the *target page*, as it is one of the relevant pages for the query string q in some context. The query vertex p that defines this context is found by picking uniformly at random one of the pages that have a link to the target page. This procedure is repeated for a number of times for the same disambiguation page to obtain a few instances of a query with the same query string, but with different contexts and target pages. Note that the same approach is used when creating training data for RankSVM.

We conducted a number of experiments with the system using the features discussed above and different versions of the context-sensitive PageRank. These are compared to a simple BM25 based ranker. The set of queries we use for evaluation contains roughly 25 different query strings, each of which is used in roughly four different contexts. (A few queries appear in less than four contexts, but the total number of queries is 100.) The same set of queries is used in all experiments, but this test set is different than the one used to train the RankSVM. BFS is computed to depth 3. The number of clusters as well as the number of landmarks for computing approximate versions of the context-sensitive PageRanks was set to 100. This number was chosen as it is large enough to capture a number of different conceptual clusters (such as the Wikipedia categories) but is still small enough to prevent storage requirements for the PageRank vectors to become an issue.

3.2 Results

To compare the methods we measure the *success rate* defined as the number of times the correct target (as defined by the disambiguation page) appears within the top-k results (with k equal to 1, 5 and 10). This is shown in Table 1 for different variations of the context-sensitive PageRank, with and without BFS pruning. Also results for the baseline algorithm (BM25) are shown. Computing the "true" context-sensitive PageRanks at query time is very inefficient, but we include it here for comparison.

We observe that in every case including the context-sensitive features to the scoring function considerably improves the retrieval performance of ambiguous queries when compared with simple

	success rate @ k				
	k=1	k=5	k=10	mean	median
BFS_BM25	9	49	61	15.6	5
BFS_TPR	44	80	83	2.1	1
BFS_CPR	23	56	71	11.6	3
BFS_LPR	20	55	67	14.3	4
BFS_NPR	25	59	66	16.7	3
NOBFS_BM25	0	12	25	285	38
NOBFS_TPR	2	92	99	3.1	2
NOBFS_CPR	0	55	72	49.2	5
NOBFS_LPR	0	50	63	66.1	6
NOBFS_NPR	0	59	67	81.7	5

Table 1: Success rates of BM25 and our RankSVM based system with (top) and without (bottom) using BFS pruning. TPR, CPR, LPR and NPR stand for "true", "cluster", "landmark" and "no" Pagerank, respectively.

BM25. Also, using BFS for pruning the set of relevant pages leads to a dramatic increase in the success rate. This comes with the price that in some cases the correct target page is not within the BFS tree, and can hence not be found by our system. Of the approximate context-sensitive PageRank measures the one based on graph clustering outperforms the one using randomly selected landmarks.

We also evaluated the system qualitatively by performing a number of ambiguous queries in different contexts and looking at the results. For instance, for the query ⟨ "watson", "Behaviorism" ⟩, the first result is "John B. Watson" (psychologist who established the school of behaviorism), for ⟨ "watson", "DNA" ⟩ it is "James D. Watson" (the co-discoverer of the structure of DNA), and for ⟨ "watson", "Mystery"⟩ we get "Doctor Watson" (the sidekick of Sherlock Holmes) as the first hit. Also most of the other pages appearing in the topmost results are highly relevant given the context and the query string.

4. CONCLUSIONS

We have described a system for context-sensitive search based on the hyperlinks between documents. The novelty in our approach is to localize the query at a node in the graph and use this node as the context of the query. The experiments indicate that we can improve upon a baseline system that only uses BM25 for ranking and does not take context into account. Especially in case of ambiguous queries the proposed system is able to return different but relevant results depending on the context.

5. REFERENCES

[1] P. Boldi and S. Vigna. The webgraph framework I: compression techniques. In *WWW Conference*, 2004.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 1998.

[3] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[4] T. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th WWW Conference*, 2002.

[5] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[6] Y. Koren. On spectral graph drawing. In *COCOON*, 2003.

[7] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[8] I. Ounis, G. Amati, P. V., B. He, C. Macdonald, and Johnson. Terrier Information Retrieval Platform. In *ECIR*. Springer, 2005.