Information Systems 37 (2012) 372–390

Contents lists available at SciVerse ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

PEST: Fast approximate keyword search in semantic data using eigenvector-based term propagation

Klara Weiand [a,*], Fabian Kneißl [a], Wojciech Łobacz [a], Tim Furche [b,a], François Bry [a]

[a] Institute for Informatics, Ludwig-Maximilians-Universität, 80538 Munich, Germany
[b] Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom

ARTICLE INFO

Available online 10 November 2011

Keywords:
Keyword search
Indexing methods
Approximate matching
Eigenvector
Semantic data
Wikis

ABSTRACT

We present PEST, a novel approach to the approximate querying of graph-structured data such as RDF that exploits the data's structure to propagate term weights between related data items. We focus on data where meaningful answers are given through the application semantics, e.g., pages in wikis, persons in social networks, or papers in a research network such as Mendeley. The PEST matrix generalizes the Google Matrix used in PageRank with a term-weight dependent leap and accommodates different levels of (semantic) closeness for different relations in the data, e.g., friend vs. co-worker in a social network. Its eigenvectors represent the distribution of a term after propagation. The eigenvectors for all terms together form a (vector space) index that takes the structure of the data into account and can be used with standard document retrieval techniques. In extensive experiments including a user study on a real life wiki, we show how PEST improves the quality of the ranking over a range of existing ranking approaches, yet achieves a query performance comparable to a plain vector space index.

1. Introduction

Mary wants to get an overview of software projects in her company that are written in Java and that make use of the Lucene library for full-text search. According to the conventions of her company's wiki, a brief introduction to each software project is provided by a wiki page tagged with "*introduction*".

Thus, Mary enters the query for wiki pages containing "*java*" and "*lucene*" that are also tagged with "*introduction*". In the (semantic) wiki KiWi, this can be achieved by the KWQL [1] query `ci (java lucene tag(introduction))`, where `ci` indicates wiki pages, see Section 3.2.

However, the results fall short of Mary's expectations for two reasons that are also illustrated in the sample wiki of Fig. 1:

(1) Some projects may not follow the wiki's conventions (or the convention might have changed over time) to use the tag "*introduction*" for identifying project briefs. This may be the case for Document 5 in Fig. 1. Mary could loosen her query to retrieve all pages containing "*introduction*" (but that are not tagged with it). However, in this case pages that follow the convention are not necessarily ranked higher than other matching pages.

(2) Some projects use the rich annotation and structuring mechanisms of a wiki to split a wiki page into subsections, as in the case of the description of KiWi in Documents 1 and 2 from Fig. 1, and to link to related projects or technologies (rather than discuss them inline), as in the case of Document 4 and 5 in Fig. 1. Such projects are not included in the results of the original query at all. Again, Mary could try to change her query to allow keywords to occur in sub-sections or in linked documents, but such queries quickly become rather complex (even in a flexible query language such as KWQL) or impossible with the limited search facilities most wikis provide.

[*] Corresponding author.
E-mail address: oucl@furche.net (T. Furche).

0306-4379/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.is.2011.10.004

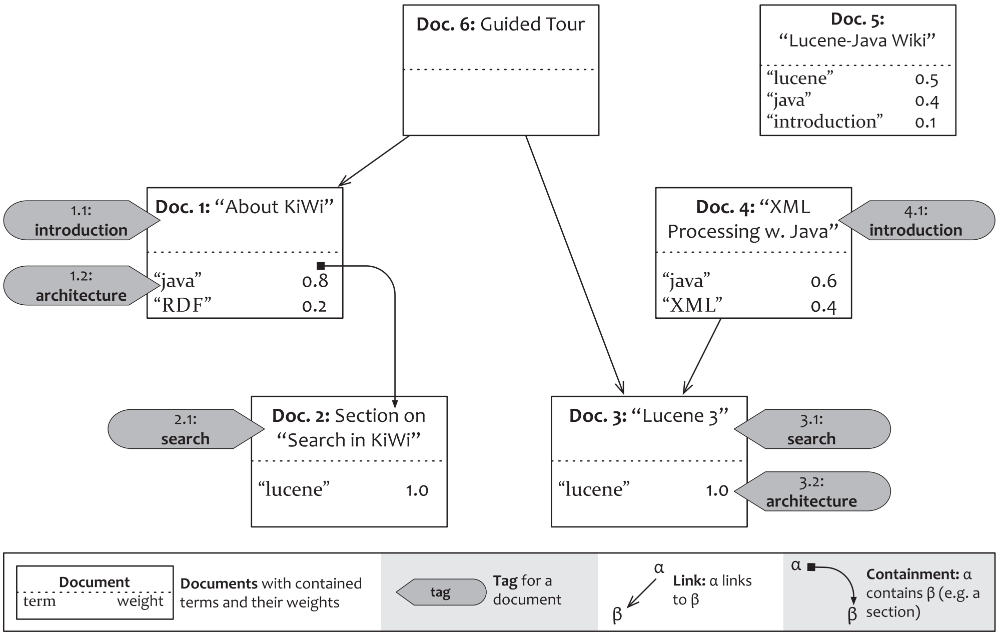

Fig. 1. Link and containment graph for a sample wiki.

Furthermore, this solution suffers from the same problem as addressed above: Documents following the wiki's conventions are not necessarily ranked higher than those only matched due to the relaxation of the query.

Though we choose a wiki to introduce these challenges, they appear in a wide range of applications involving (keyword) search on structured data, e.g., in social networks, in ontologies, or in a richly structured publication repository. The common characteristic of these applications is that relevant answers (e.g., a wiki page or a person in a social network) are not fully self-contained documents as in the case of standard web search, but obtain a big part of their relevance by virtue of their structural relations to other pages, persons, etc. At the same time, they are sufficiently self-contained to serve as reasonable answers to a search query, in contrast to, e.g., elements of an XML document.

Since data items such as wiki pages tend to be less self-contained than common web documents, PageRank and similar approaches that use the structure of the data merely for ranking of a set of answers do not suffice: As Fig. 1 illustrates, even pages that do not contain the relevant keyword can be highly relevant answers due to their relations with, e.g., tags that prominently feature the keyword.

To address this challenge, not only the ranking, but also the selection of answers needs to be influenced by the structure of the data. As a first step in this direction, PageRank propagates the anchor text of links to a page as if they are content of that page.

In this article, we generalize the idea of term propagation over structured data: PEST, short for term-*p*ropagation using *e*igenvector computation over *st*ructural data, is a novel approach to *approximate matching over structured data*. PEST is based on a unique technique for propagating term weights (as obtained from a standard vector-space representation of the documents) over the structure of the data using eigenvector computation. It generalizes the principles of Google's PageRank [2] to data where the content of a data item is not sufficient to establish the relevant terms for that item, but where rich structural relations are present that allow us to use the content of related data items to improve the set of keywords describing a data item.

It is worth noting, that PEST is tailored to search in structured data such as semantic wikis or social networks where nodes often contain significantly less data than documents in standard web search. Finding nodes only based on the content of a node is infeasible in this cases. Furthermore, in this setting global relevance rankings such as basic PageRank tend to severely bias towards popular documents over documents relevant to the query. Even if the nodes are full documents (e.g., in a wiki about composers). E.g., if one asks where classical organ music composers spend most of their career, pages about cities such as Leipzig or Weimar are certainly relevant, but a document about Sachsen or the courts of German dukes of the 18th century would be just as relevant, whether they specifically mention any of the composers or not.

In contrast to many other fuzzy matching approaches (see Section 2), PEST relies solely on modifying term weights in the document index and requires no runtime query expansion, but can use existing document retrieval technologies such as Lucene. Furthermore, the modified term weights represent how well a data item is connected to others in the structured data and therefore one can omit a separate adjustment of the answer ranking as in PageRank. Our experimental evaluation shows that, indeed, query time is comparable to Lucene, despite incorporating both term propagation and structural centrality into the vector space index.