11-10-2009

Intelligent Search from Multiple Resources of Purdue Library

Dzung Hong
Purdue University - Main Campus

Luo Si
Purdue University, lsi@cs.purdue.edu

Paul Bracke
Purdue University - Main Campus, pbracke@purdue.edu

Michael Witt
Purdue University, mwitt@purdue.edu

Timothy C. Juchcinski
tjuchcin@purdue.edu

PURDUE UNIVERSITY LIBRARIES

Access. Knowledge. Success.

Intelligent Search from Multiple Resources of Purdue Library

Dzung Hong[1], Luo Si[1], Paul Bracke[2], Michael Witt[2] and Tim Juchcinski[1]

[1] Department of Computer Sciences, Purdue University

[2] Purdue University Libraries

- The Purdue Library has access to more than 400 databases.
- Databases span across different categories such as: news, engineering & technology, arts & humanities, business & economics, etc.
- Most of them contain documents that are not reachable by traditional web crawlers due to security, technical limitations or copyright agreement.
- Each database uses different methods of indexing and searching for documents.
- The aim of this project is to enhance search efficiency by automatically suggesting and searching in the most appropriate databases, depending on users' queries.

The three steps

1. Collect information about each database
2. Based on information about each database and user's query, propose the most relevant ones
3. Forward user's query to those databases, retrieve the results, merge them and present to user

1 Query-based sampling

(*Callan, J. & Connell, M. Query-based sampling of text databases. ACM Transactions on Information Systems (TOIS), ACM New York, NY, USA, 2001, 19, 97-13*)

Repeat these steps for each database:

1. Select a keyword
2. Send that keyword as a query to the database
3. Retrieve the top *n* documents from the results
4. Add those documents to the sample database
5. Choose another keyword from the sample database and repeat step 2

After all, we will get a set of documents representing the database

2 Choosing the best databases

♦ Independent model

- Training on a set of features
  - Big document feature
  - Relevant document distribution estimation (ReDDE)
  - Geometric average
- Ranking databases using logistic regression model

$$P(db_i \mid \vec{f}(db_i)) = \frac{1}{1+\exp(\vec{f}(db_i)\cdot\vec{w})}$$

where $\vec{f}(db_i)$: feature vector of the i-th database

$\vec{w}$: the weight corresponding to each feature

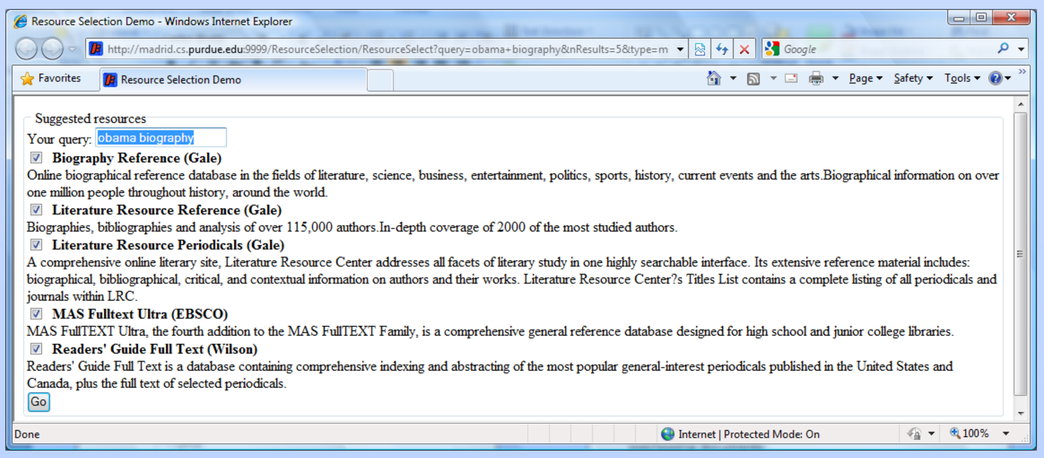
♦ Joint prediction model

- Ranking databases based on how they are similar with another good, relevant databases

$$P(\vec{v} \mid db) = \frac{1}{Z}\exp\Big(\sum_i^n (1-v_i)(\vec{f}(db_i)\cdot\vec{w}) - \log(1+\exp(\vec{f}(db_i)\cdot\vec{w}) + \frac{\alpha}{|\vec{v}|}\sum_{i,j(i<j)} sim(db_i, db_j)v_i v_j\Big)$$

where $\vec{v}$: relevant vector, $v_i = 1$ if i-th database is relevant, $v_i = 0$ otherwise

$sim(db_i, db_j)$: similarity score between db_i and db_j



Demo: Resource Selection with Query "Obama biography"

3 Merging the results

(Si, L. & Callan, J. A semisupervised learning method to merge search engine results. *ACM Transactions on Information Systems (TOIS), ACM,* 2003*, 21*, 491)

1. Sending query to the selected databases from step 2; meanwhile sending query to the central database built from step 1
2. Inspecting the scores of overlapping documents from both central database and remote database
3. Using those scores to interpolate scores of another non-overlapping documents
4. Merging documents and present the rank based on the interpolating scores

Future works

- Building a new merging model that performs well even when there is no overlapping document
- Exploring the use of users' log and personalizing searching
- Enhancing database selection with another probabilistic model
- Implementing the whole system, providing user-friendly interface

PURDUE UNIVERSITY