DWRank: Learning concept ranking for ontology search

In Semantic Web Journal, vol. 7, no. 4, Pages 447-461, May 2016

Anila Sahar Butt, Armin Haller, and Lexing Xie
The Australian National University
CONCEPT SEARCH AND RANKING
Concept Search and Ranking

- Concept Search
  - Matching a search term with a more expressive class description

- Matching terms are defined with differing
  - Perspectives
  - Levels of detail
  - Reuse and Extensions

How to rank similar concepts with different levels of modelling details?
Relationship based Concept Retrieval Framework

- The framework retrieves and rank concepts for keyword query
  - DWRank Ranking Model
  - Top-k Filters
DWRank – Dual Walk Ranking Model

• Query independent scores for each concept of ontologies based on their importance
  – HubScore: Centrality of the concept
  – AuthScore: Authoritativeness of the Ontology

• Relevance score of a concept to a query:
  – DWRank Function: Linear model combines
    • Text relevancy of the concept label to a query
    • HubScore and AuthScore
HubScore – Centrality of a Concept

• **Connectivity:**
  – Relations starting from the concept

• **Neighbourhood:**
  – Relations starting from the concept to another central concept
AuthScore – Authoritativeness of an Ontology

- **Reuse**: Relations ending at the ontology
- **Neighbourhood**: Relations starting from another authoritative ontology to the ontology

**PageRank**

- :Location: 0.10
- :Restaurant: 0.145
- :People: 0.471
DWRank Function

- The ranking model is function of
  - Concept Relevancy: \( F_v(v, Q) = \sum_{q \in Q} f(q, \varphi(q)) \)
  - Hub Score: \( h(v, Q) \)
  - Auth Score: \( a(O) \)

- \( R(v, Q) = Fv(v, Q) \times [\alpha h(v, Q) + (1 - \alpha) a(O)] \)
  \[= 1 \times [0.5 \times 0.46 + 0.5 \times 0.471]\]
  \[= 0.466\]
DWRank vs. Linked-based Ranking Models

1. Direction of the walk varies based on the link type
   - Intra-ontology links: Reverse PageRank
   - Inter-ontology links: PageRank

2. Linked Analysis:
   - HubScore – Concept
     • Independently on each ontology
   - AuthScore – Ontology
     • Ontology Corpus
Top-K Filter

- **Intended Type Filter**
  - Intended Type vs. Context Resource
    - Name of the Person
      - Intended Type: Name
      - Context Resource: Person

- **Distinct Resource Filter**
  - Select Resources that are less overlapping
Evaluation

• Effectiveness of the approach
  – DWRank
  – DWRank + Filter

• CBRBench
  – Queries and Gold standard
  – Baseline Ranking models
DWRank and Filter Effectiveness

![Graph showing comparison between DWRank and MaxPerformance in AP@10 and NDCG@10 across different attributes: Person, Name, Event, Title, Location, Address, Music, Organization, Author, and Time.]

![Graph showing filter effectiveness across different attributes: Person, Name, Event, Title, Location, Address, Music, Organization, Author, and Time, comparing DWRank and DWRank + Filter.]
LEARNING CONCEPT RANKING
Need for LTR

• None of the commonly used evaluation algorithm performs adequately for all type of queries.

• So most of the proposed approaches used more than one ranking/evaluation metric to optimise the effectiveness of ranking models.
  – However for optimal performance of such algorithms the metrics’ weights need to be reset for each user query.
    • Manually setting metrics’ weights for each and every query is impractical
    • Solution: Learning to Rank
Learning Concept Ranking for DWRank

• Feature Set
  – HubScore, Max HubScore, Min HubScore, AuthScore, Text Relevancy
  – Target Feature: Relevance Score

• Training Data
  – CBRBench
    • <target> qid:<qid> <feature1>:<value> <feature2>:<value> ... <featuren>:<value> #<info>
  – Data Set
    • Training Set,
    • Validation Set,
    • Test Set
Learning Concept Ranking for DWRank (1/2)

• Metrics
  – P@10, AP@10, DCG@10, NDCG@10

• Learning to Rank Algorithm
  – LambdaMART
    • RankLib: [http://sourceforge.net/p/lemur/wiki/RankLib/](http://sourceforge.net/p/lemur/wiki/RankLib/)
  – Leave-one-out Cross Validation (LOOCV)
  – Optimised NDCG and Tested DCG, AP, P
Framework Overview

Index Construction and Learning Phase

- Ontology Collection
- CBRBench Gold standard
- Generate Training dataset
- Indexes
  - ConHub
  - OntAuth
- Ranking Model
- Learn Ranking Model
- Generate Training dataset

Query Processing Phase

- Top-K result
- HubScore, MaxHubScore, MinHubScore, AuthScore features value extraction for Candidate Resultset
- Candidate Resultset Selection
- Generate Ranking
- Text Relevancy

Candidate Resultset Selection

HubScore, MaxHubScore, MinHubScore, AuthScore features value extraction for Candidate Resultset

Generate Ranking

Top-K result

Generate Ranking

Text Relevancy

Candidate Resultset Selection

HubScore, MaxHubScore, MinHubScore, AuthScore features value extraction for Candidate Resultset
Learning to Rank Model Effectiveness

- P@10
- AP@10
- DCG@10
- NDCG@10

DWRank Fixed Weight Model
DWRank Learning to Rank Model