Exploiting Paths for Entity Search in RDF Graphs

Minsuk Kahng and Sang-goo Lee

Department of Computer Science and Engineering
Seoul National University, South Korea

Abstract

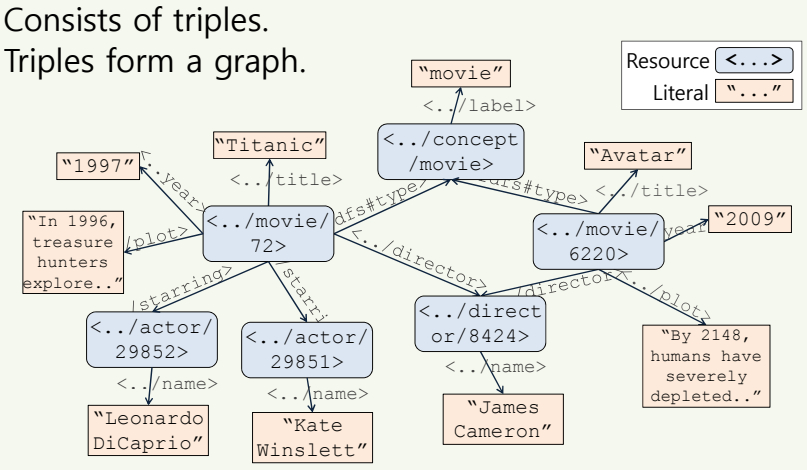
- Propose an entity retrieval model for RDF data.
- Aim to capture indirect relationships between nodes in the RDF graph by considering paths as fields.

Problem

- Retrieve entities using (semi-)structured RDF data.
  - Called 'Semantic search' or 'Ad-hoc object retrieval'.

Data: What is RDF?

- Semantic Web (standard) data model
- Consists of triples.
- Triples form a graph.



Subject	Predicate	Object
<http://a.bc/movie/72>	<../title>	"Titanic"
<http://a.bc/movie/72>	<../director>	<../director/8424>
<http://a.bc/director/8424>	<../name>	"James Cameron"
<http://a.bc/movie/6220>	<../title>	"Avatar"

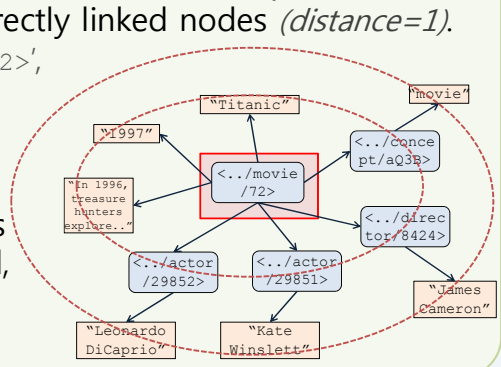
Problem

- Given a keyword query Q,
  rank RDF resources E, which represent entities.
  - Ex. For a query 'movie james cameron', resources like '<../movie/72>' & '<../movie/6220>' are retrieved.

Motivation

Observation

- Most existing models assume the descriptions of an entity exist only at directly linked nodes *(distance=1)*.
  - Ex. For '<../movie/72>', nodes like '"James Cameron"' are NOT considered.
- But even if two nodes are not directly linked, they are *somewhat related* to each other.



Direction

- We aim to capture indirect relationships btw nodes.
- We assume *the descriptions of an entity* exist at *any (literal) node* that is reachable from the resource node.
- Each *path* from E to L_j is considered as a field.

Proposed Model

- Simulates the generation process of query Q by following paths from a resource node E to several related literal nodes L_j.

$$P(E|Q) \overset{rank}{=} P(Q|E)P(E) \qquad \textit{Bayes's Rule}$$

$$= \prod_{i=1}^{|Q|} P(q_i|E) \qquad \textit{Term independence}$$

$$= \sum_{j=1}^{m} P(q_i|L_j)P(L_j|E) \qquad \textit{Through Literals } L_j \textit{ related to } E$$

Finally,

$$P(E|Q) = P(E) \prod_{i=1}^{|Q|} \sum_{j=1}^{m} P(L_j|E) P(q_i|L_j)$$

Similar to mixture of field LMs (MFLM)

Resource Prior

- Like doc prior
- Use '# of literals'

Path Importance

- Path: seq. of pred.
- How to determine?

Path	Weight
E —<..title>→ L	0.40
E —<..plot>→ L	0.25
E —<..director>→ E' —<..name>→ L	0.15

The weights are determined by defining the importance of predicates, then aggregated (above table). Otherwise, it can be learned.

Literal LM

- Lang Model for each literal
- Generation prob. of query term given L_j.

$$P(q_i|L_j) = \frac{tf(q_i, L_j) + \mu \frac{c_{q_i}}{|C|}}{|L_j| + \mu}$$

Evaluation

Setup

- Followed the standard evaluation framework used in *SemSearch Challenge 2010*.
  - Data : BTC collection (886M triples, 175M resources, 296M literals)
  - Query set : 92 entity queries from Yahoo query logs
  - Relevance Judgments : obtained from SemSearch10
  - Baselines approaches :
    - Pseudo-document (plain text)
    - Attributes (directly linked literals) as fields (uni. & diff. weights)

Results

Approach	Ret.Model	MAP	P@10	NDCG	
Plain text	BM25	0.2366 ▽	0.4293 ▽	0.4288 ▽	
	LM	0.2426 ▽	0.4272 ▽	0.4438 ▽	
Attr. (uniform w)	BM25F	0.2014 ▽	0.4380	0.3886 ▽	*LMs > BM25s*
	MFLM	0.2711 *†▽	0.4783	0.4765 *†▽	*MFLM > LM*
Attr. (different w)	BM25f	0.2523 ‡▽	0.4826 †‡	0.4484 ‡▽	
	MFLM	0.2889 *†‡▽	0.5076 †	0.4913 *†‡▽	
Path	**PathLM**	0.3268 †‡	0.5033 †	0.5245 †‡	*Path is best.*

Discussion

- Only few long paths improve performance (e.g. `sameAs`).
- More research needed to develop ways to learn path weights.
- We added extra judgments due to insufficient relevance judgments.
- We look forward to further research in our path-based approach.
  - See our paper "Ranking Objects by Following Paths in E-R Graphs" in Ph.D workshop at CIKM 2011.

SEOUL NATIONAL UNIVERSITY

SIGIR Special Interest Group on Information Retrieval