Entity Retrieval via Query Graph Inference

**Xinshi Lin**, Kwun Ping Lai, Zihao Wang and Wai Lam
The Chinese University of Hong Kong

**Github Project**: https://github.com/linxinshi/EntityRetrievalQGR
Background – some recent work

• FSDM (Zhiltsov et.al., 2015): term-based representation

• FSDM+ELR (Hasibi et.al., 2017): term-based representation + semantic annotations

• MLM+type (Garigliotti et.al., 2017): term-based representation + query type information

• MLM/FSDM+PAS(Lin et.al.,2018): term-based representation + hierarchical entity type information + hierarchical entity descriptions
Background – some recent work

• Advantages of FSDM or Markov random field based models
  • Solid theoretical foundation
  • Stable performance
  • Term-based

• Disadvantages (my opinion)
  • Ignore structure information in a knowledge graph
  • Only consider one single candidate entity at one time
    • Ignore its related entities
Our proposal

• Propose a path-centric entity retrieval framework
  • consider candidates entities and its related entities in a unified way
• Improve entity retrieval by incorporating
  • entity type information
  • semantic annotations in a query
  • entity descriptions
Model Description - path-centric framework

- Given a query $Q$ and a candidate entity $E$
- Enumerate entity type $c$ of $E$
- A knowledge graph $G$
  - nodes denote entities and edges denote relations
- Enumerate path $p$ starting from an identified query entity $e_q$ (inference chain)
  - $p = \{e_1 = e_q, e_2, ..., e_n\}$

The overall scoring function

$$f(Q, E, G) = \max_{c \in \text{types}(E)} \max_{p \in G} \tilde{f}(Q, E, p, c)$$

$\tilde{f}(Q, E, p, c)$: scoring function for inference chain
Model Description

Scoring function for each inference chain:

\[
\tilde{f}(Q, E, p, c) = \sum_{t \in \{text, rel, path, elr\}} w_t s_t(Q, E, p, c)
\]

• Consider each evidence(feature) \( t \)
  • text: term-based representation
  • rel: relations
  • path: query graph connectivity
  • elr: contextual probability of \( E \)
Model Description - Incorporate entity type information

Markov random field based framework
Example: query='barack Obama parents'
unigrams={‘barack’,‘Obama’,‘parents’}
bigrams={‘barack Obama’, ‘Obama parents’}

\[ D = \{D_f\}: \text{multi-fielded pseudo documents of } E. \]

\[
\begin{align*}
    s_{text}(Q,E,p,c) &= \lambda_T \sum_{q_i \in Q} g_T(q_i, D, H_c) + \\
    & \quad \lambda_O \sum_{q_i, q_{i+1} \in Q} g_O(q_i, q_{i+1}, D, H_c) + \\
    & \quad \lambda_U \sum_{q_i, q_{i+1} \in Q} g_U(q_i, q_{i+1}, D, H_c)
\end{align*}
\]

feature function for unigrams
feature function for ordered bigram occurrences
feature function for unordered bigrams occurrences
Model Description - Incorporate entity type information

• Dirichlet prior smoothed feature function

\[
g_{\{T,O,U\}}(W, D, H_c) = \log \sum_f w_f^T \frac{t_{fW,D_f,c} + \mu_f \frac{c_{fW,f}}{|C_f|} + N_t}{|D_f| + \mu_f + D_t}
\]

\(t_{fW,D_f}\): term frequency of \(W\) in the pseudo document of \(f\)

\(W\) generalizes to unigrams and bigrams

\(N_t, D_t\): type-aware smoothing components

\(H_c\): a tree rooted at \(c\) in the type taxonomy
Model Description - Incorporate entity type information

- Specifically, we add a **type-aware smoothing component** in the feature function

\[
g_{\{T,Q,U\}}(W,D,H_c) = \log \sum_f w_f^T \frac{tf_{W,D,f,c} + \mu_f \frac{cf_{W,f}}{|C_f|} + N_t}{|D_f| + \mu_f + D_t}
\]

\[
N_t = \beta \cdot \mu_{f,H_c} \frac{cf_{W,f,H_c}}{|C_{f,H_c}|}
\]

\[
D_t = \beta \cdot \mu_{f,H_c}
\]

- Can derive many existing models by taking particular parameter combinations (LM, MLM, SDM, PRMS...)

\[
\mu_f. \text{ Dirichlet priors}
\]

\[
\beta. \text{ model parameters}
\]

\[
c_{f_{W,f,v_j}} \text{ collection frequency of term } W \text{ in field } f \text{ under each type in } H_c
\]
Model Description

- Semantic similarity between inference chain and query

\[ s_{rel} = \prod_{r \in \rho} \text{sim}(r, Q) \]

Consider each edge (relation) in the path.
Split relation label into a sequence of terms. (e.g. ‘birthYear’ to [birth’,’year’])
Use a classic recurrent neural network to compute the semantic similarity.

- Contextual probability of \( E \)

\[ s_{elr} = \sum_{f} w_f \left[ (1 - \alpha) t_f(0,1)(E,D_f) + \alpha \frac{d_f(E,f)}{d_f} \right] \]

- Query graph connectivity

\( s_{path} \) is set to 1 if \( e_n \) in the path is the candidate entity \( E \), and 0 otherwise.
Experiment

• Knowledge graph: DBpedia 2015-10
• Knowledge source: Wikipedia 2015-10
• Type taxonomy: Wikipedia Category System

• Test collection: DBpedia-entity v2
  • Four benchmark datasets with 467 queries in total
    • INEX-LD: keyword based queries
    • SemSearch_ES: named entity targeted queries
    • ListSearch: queries that seek a particular list of entities
    • QALD2: natural language questions

• “DBpedia-Entity v2: A Test Collection for Entity Search”, SIGIR 2017
## Experiment

### Entity Search - Shared Task @ EYRE '18: NDCG

<table>
<thead>
<tr>
<th>Participants</th>
<th>SemSearch ES</th>
<th>INEX-LD</th>
<th>ListSearch</th>
<th>QALD-2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25-CA + Rerank (Ma et al.)</td>
<td>0.5984</td>
<td>0.6958</td>
<td>0.4159</td>
<td>0.5077</td>
<td>0.4290</td>
</tr>
<tr>
<td>BM25F-CA + Rerank</td>
<td>0.6317</td>
<td>0.7218</td>
<td><strong>0.4454</strong></td>
<td><strong>0.5324</strong></td>
<td>0.4235</td>
</tr>
<tr>
<td>FSDM+ELR + Rerank</td>
<td>0.6954</td>
<td>0.7301</td>
<td>0.4394</td>
<td>0.5171</td>
<td>0.4219</td>
</tr>
<tr>
<td>FSDM + Entity Semantics (Nasiri et al.)</td>
<td>0.6586</td>
<td>0.7281</td>
<td>0.4335</td>
<td>0.5119</td>
<td>0.4247</td>
</tr>
<tr>
<td>FSDM + Term Semantics</td>
<td>0.6500</td>
<td>0.7173</td>
<td>0.4224</td>
<td>0.5015</td>
<td>0.4272</td>
</tr>
<tr>
<td>FSDM + Entity &amp; Term Semantics</td>
<td>0.6583</td>
<td>0.7273</td>
<td>0.4291</td>
<td>0.5047</td>
<td>0.4242</td>
</tr>
<tr>
<td>EntityRetrievalQGR (Lin et al.)</td>
<td><strong>0.6686</strong></td>
<td><strong>0.7139</strong></td>
<td>0.4140</td>
<td>0.4647</td>
<td>0.4064</td>
</tr>
<tr>
<td>MLM+QGR</td>
<td><strong>0.6685</strong></td>
<td><strong>0.7314</strong></td>
<td>0.4330</td>
<td>0.4962</td>
<td>0.3043</td>
</tr>
<tr>
<td>FSDM+QGR/type</td>
<td>0.6355</td>
<td>0.7241</td>
<td>0.4235</td>
<td>0.4730</td>
<td>0.4235</td>
</tr>
<tr>
<td>BM25-CA</td>
<td>0.5958</td>
<td>0.6883</td>
<td>0.4120</td>
<td>0.5050</td>
<td>0.4220</td>
</tr>
<tr>
<td>BM25F-CA</td>
<td>0.6281</td>
<td>0.7200</td>
<td>0.4394</td>
<td>0.5296</td>
<td>0.4252</td>
</tr>
<tr>
<td>FSDM+ELR</td>
<td>0.6563</td>
<td>0.7257</td>
<td>0.4354</td>
<td>0.5134</td>
<td>0.4220</td>
</tr>
</tbody>
</table>

* These results were not reported by authors but were calculated by the organizers.
Summary

• Propose a Markov random field based framework incorporating both hierarchical entity type information and entity descriptions
  • Add a path aware smoothing component in the feature functions

• Future work includes investigation of incorporating more knowledge graph structures and query characteristics/intents
Thank you