Effective and Efficient Structured Retrieval

Le Zhao and Jamie Callan (lezhao, callan}@cs.cmu.edu)
Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Boolean Filtering

Why
- Required fields (date, etc.) have low IDF as index terms
- Important names may be missing from answer sentence
- Because of fields being smoothed with document model
  E.g., document on Wilt Chamberlain: sentence: “The big dipper" scored 100 points in a single game,” non-relevant, but scores high, because of document level smoothing
- Makes scoring fast

How
- Separate out key query concepts
- Each argument field of semantic role parse as a concept
- Target verbs ignored
- Conjunctive Normal Form (CNF) query
- OR within concept, AND between concepts
- E.g., Which did Wilt Chamberlain score 100 points?
  #band(#syn(#any date)) #with(Wilt Chamberlain) #from(100 points)
  arg0:arg1

Field Specific Smoothing

Why
- Sizes of fields vary
- Larger fields need more smoothing
- Optimal smoothing parameter depends on average field length
- Mismatched fields are often aligned to the outside sentence field

How
- Two level Dirichlet smoothing
- Field specific tuning of document & collection level smoothing parameters
- DocumentMu proportional to average length of that field type
- CollectionMu proportional to average aggregated length of that field type
- Within a document
- Only these two parameters/ration need be tuned on training data

Matching Alternative Answer Structures

Why
- Mismatch between query & answer structures
- Q. When did Wilt Chamberlain score 100 points?
- A. When he scored 100 points in a single game, Wilt Chamberlain lived in New York.
  #arg0 mismatch
- #syn(#Wilt Chamberlain) comes from noisy parses
- Also, how sensitive the parser is to syntactic variations in natural language.

How
- Training data: Align answer structure to question structure, to learn about the mismatch
- Given question structure, find maximally scored answer structure, and align fields to it
- Mismatched fields are assigned aligned to the outside sentence field
- Baseline: model field translations independently
- Cooccurrence (Cooc) model: jointly model all argument translations in the same query
  e.g., can model:arg0#arg1 from which when Q-A target verbs are antonyms: "buy" vs "sell."
  - Short question, low complexity

Structured query formulation

#combine(sentence)
#combine[|target|]
#combine[|arg0|]
#combine[|arg1|]

Efficiency Results

Time in minutes spent on running 50 queries

<table>
<thead>
<tr>
<th>Structure Baseline</th>
<th>Filt</th>
<th>Smoothing</th>
<th>Cooc Model</th>
<th>Mn w/ Keyword</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.07</td>
<td>1.37</td>
<td>2.29</td>
<td>3.05</td>
<td>9.9</td>
<td>10.6</td>
</tr>
</tbody>
</table>

Retrieval Results

Sentence Retrieval Performance in Mean Average Precision (MAP), Significance levels measured using sign-rank test

<table>
<thead>
<tr>
<th>Structure Baseline</th>
<th>Filt</th>
<th>Smoothing</th>
<th>Cooc Model</th>
<th>Mn w/ Keyword</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.155</td>
<td>0.279</td>
<td>0.225</td>
<td>0.140</td>
<td>0.717</td>
<td>0.475</td>
</tr>
</tbody>
</table>

Findings
- Automatically formulated Boolean filters significantly improve both retrieval accuracy and efficiency
- Field-specific smoothing improves performance consistently
- Modeling structural mismatch helps performance, but not very consistent
- Overall, structured retrieval outperforms keyword retrieval, and is efficient

Supported by: DARPA; Found grant IIS-0749601 and IIS-0543495