

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 verb ignored
 - Conjunctive Normal Form (CNF) query
 - OR within concept, AND between concepts
 - E.g. When did Wilt Chamberlain score 100 points?
- ```
#band(#syn(#any:date) #syn(Wilt Chamberlain) #syn(100 points))
 argm-tmp arg0 arg1
```

## Field Specific Smoothing

### Why

- Sizes of fields vary
- Larger fields need more smoothing
- Optimal smoothing parameter depends on average field lengths

### 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/ratios need to 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
- Structural mismatch comes from noisy parses
- Also, how sensitive the parser is to syntactic variations in natural language.

| Percentage of matching query & answer fields |       |       |       |       |          |        |
|----------------------------------------------|-------|-------|-------|-------|----------|--------|
| QA                                           | arg0  | arg1  | arg2  | tmp   | sentence | target |
| arg0                                         | .1000 | .0917 | .0150 | .0017 | 0.3167   | 0      |
| arg1                                         | .0583 | .2917 | .0483 | .0550 | 0.4683   | 0      |
| arg2                                         | .0117 | .0450 | .0150 | .0250 | 0.2017   | 0      |
| tmp                                          | .0133 | .0417 | .0067 | .1117 | 0.2033   | 0      |
| target                                       | 0     | 0     | 0     | 0     | 0.8600   | 0.1400 |

### 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 assumed aligned to the outside sentence field
- Baseline: model field translations independently
- Cooccurrence (Cooc) model: jointly model all argument translations in the same query
  - can model arg0 ⇔ arg1 switches when Q-A target verbs are antonyms: "buy" ⇔ "sell"
  - short question, low complexity

## Structured query formulation

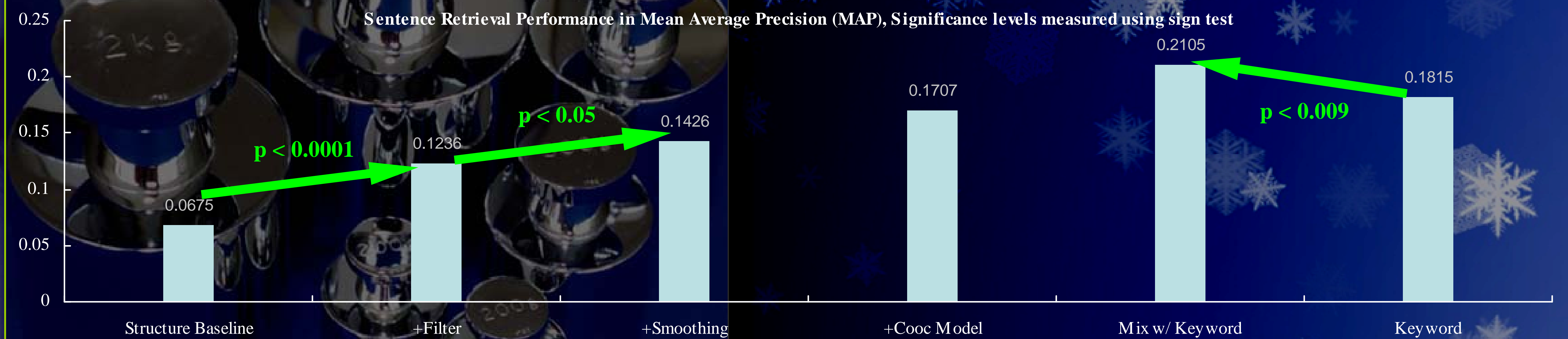
```
#combine[sentence](
#filreq(
#band(#syn(#any:date)
#syn(Wilt Chamberlain)
#syn(100 points))
#weight(0.9 #combine(Wilt Chamberlain score ...)
0.1 #max(
#combine[sentence](
#combine[target](score
#combine[./arg0](Wilt Chamberlain)
#combine[./arg1](100 points)
#combine[./argm-tmp](#any:date)))
#combine[sentence](
Wilt Chamberlain
#combine[target](score
#combine[./arg1](100 points)
#combine[./argm-tmp](#any:date)))
)
))
```

**Boolean Filter** (circled in green)

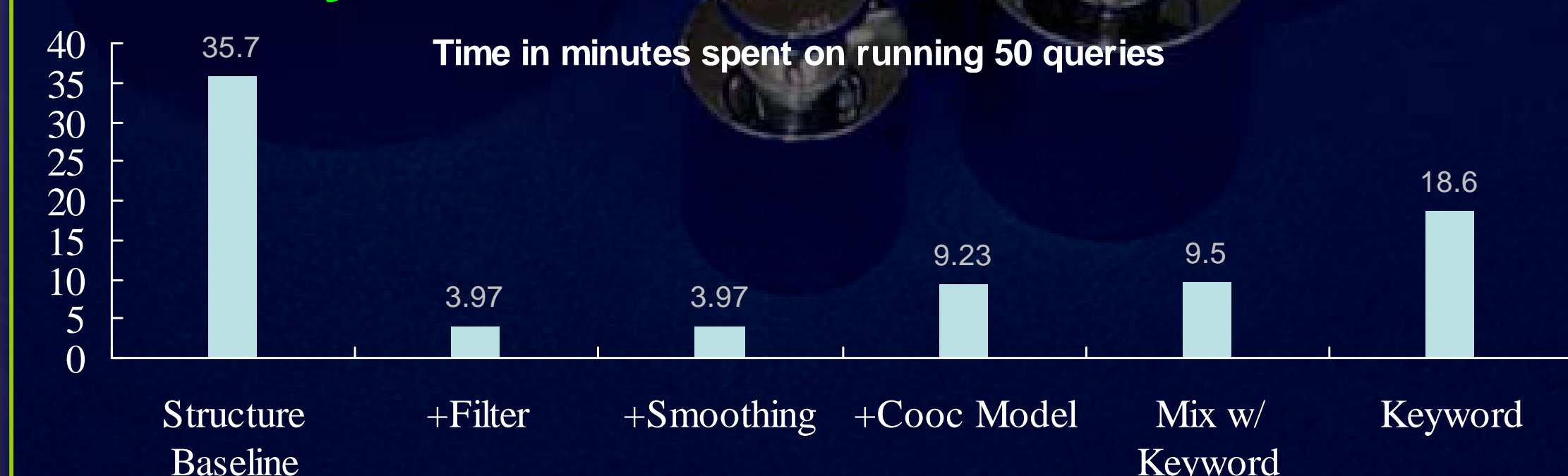
**Mixing w/ Keyword** (circled in green)

**Alternative Answer Structures** (circled in green)

## Retrieval Results



## Efficiency Results



## 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