Evaluating the Effectiveness of Axiomatic Approaches in Web Track

TREC 2013 Web Track

Peilin Yang and Hui Fang

University of Delaware

Semantic term matching is important.

Q : car

D1: driver

D2: fish

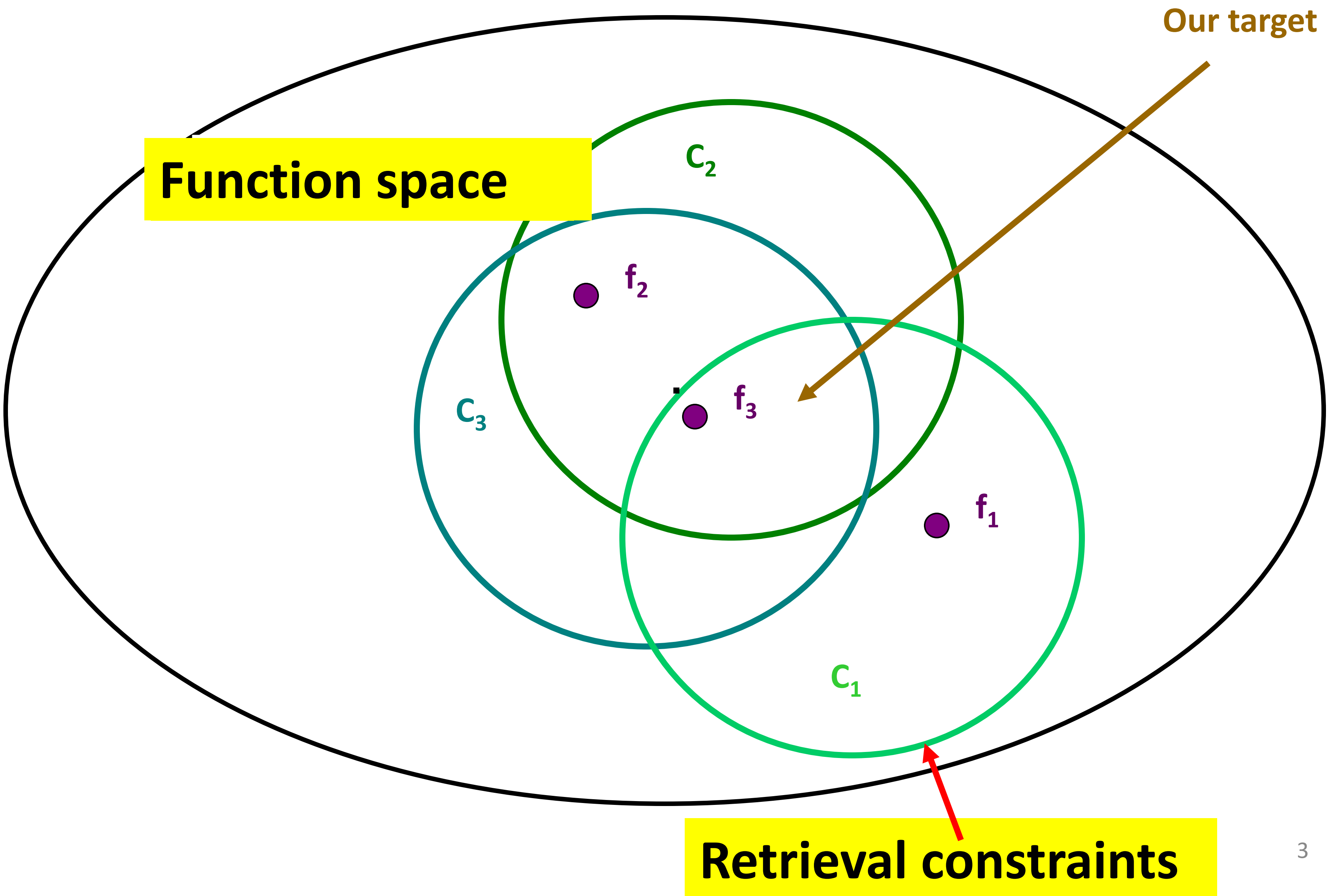
$$S(D_1, Q) > S(D_2, Q)$$

$S(D_1, Q) = S(D_2, Q)$ in most of existing keyword matching based functions

How to integrate term semantic relationship?

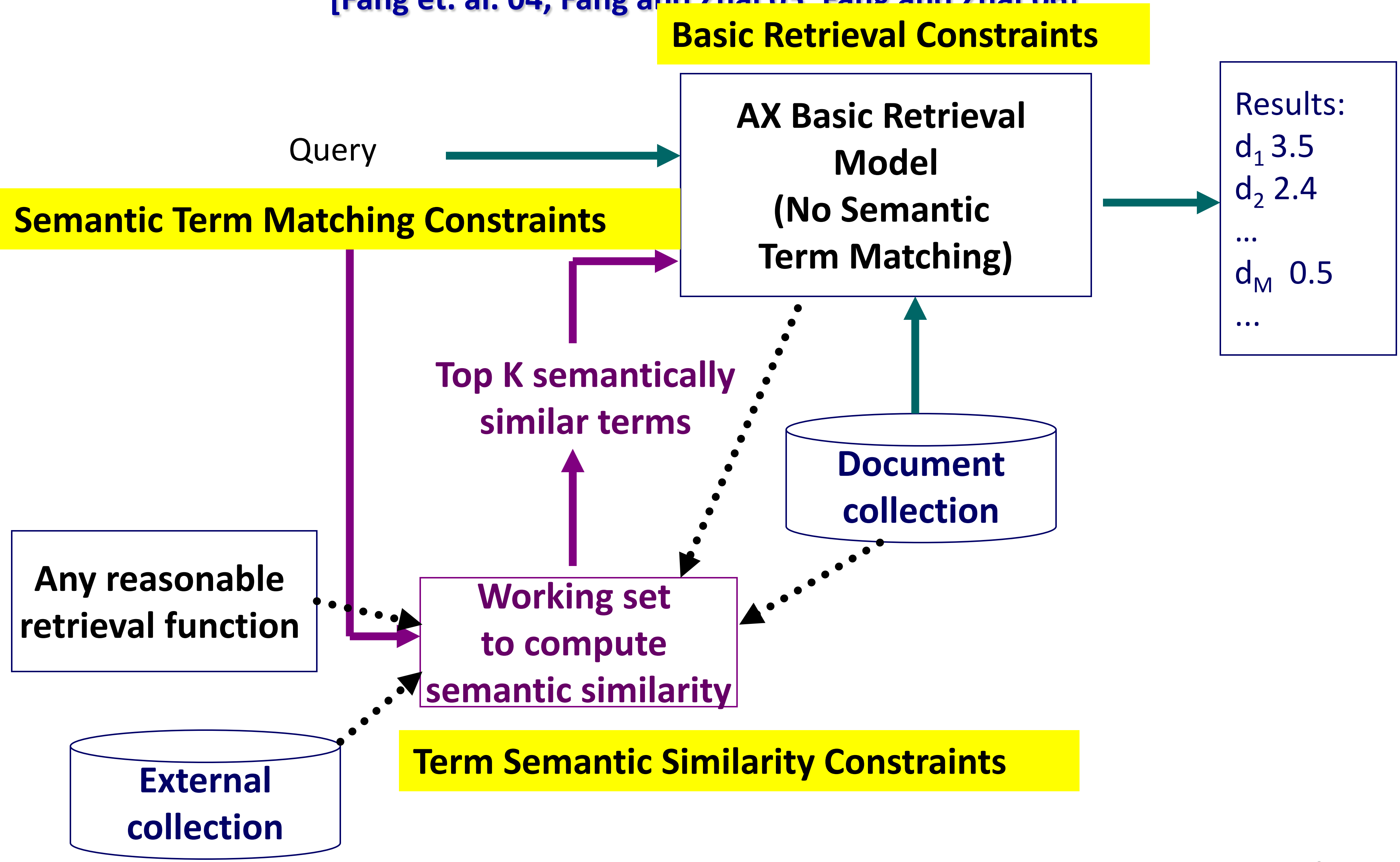
Basic Idea of Axiomatic Approach

[Fang et. al. 04, Fang and Zhai 05, Fang and Zhai,06]

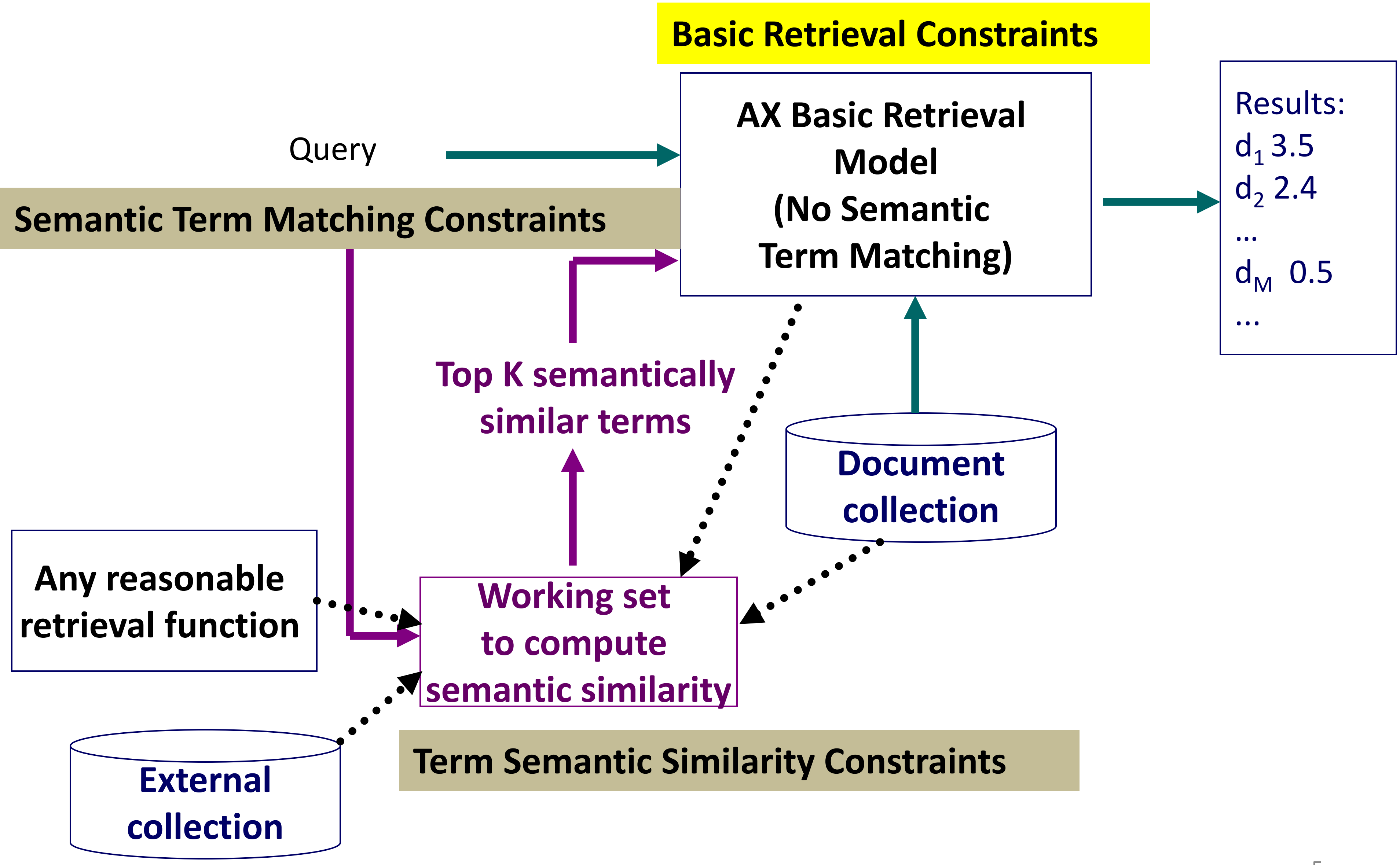


Semantic Term Matching in AX

[Fang et. al. 04, Fang and Zhai 05, Fang and Zhai 06]



Semantic Term Matching in AX

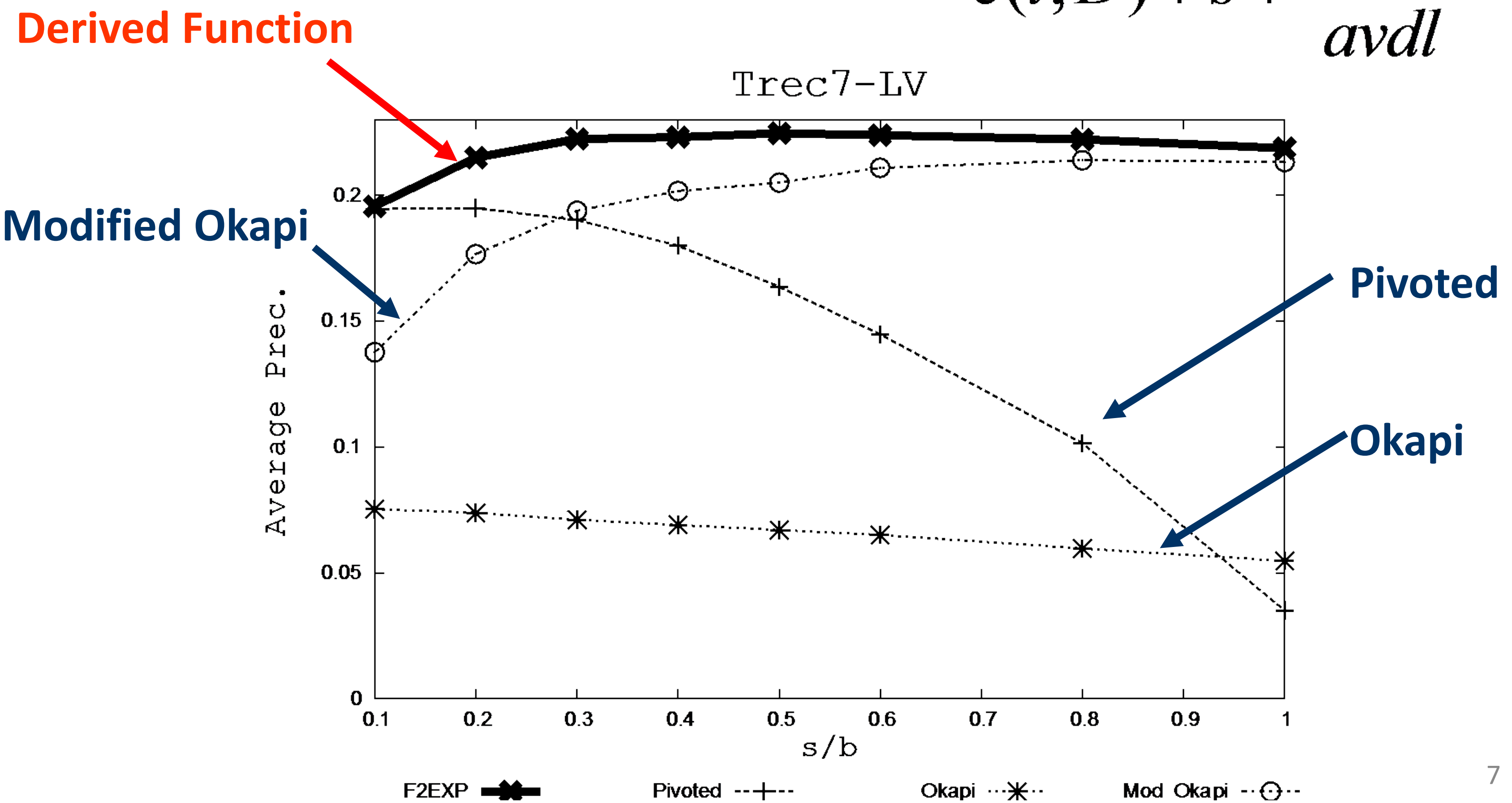


Basic Retrieval Constraints

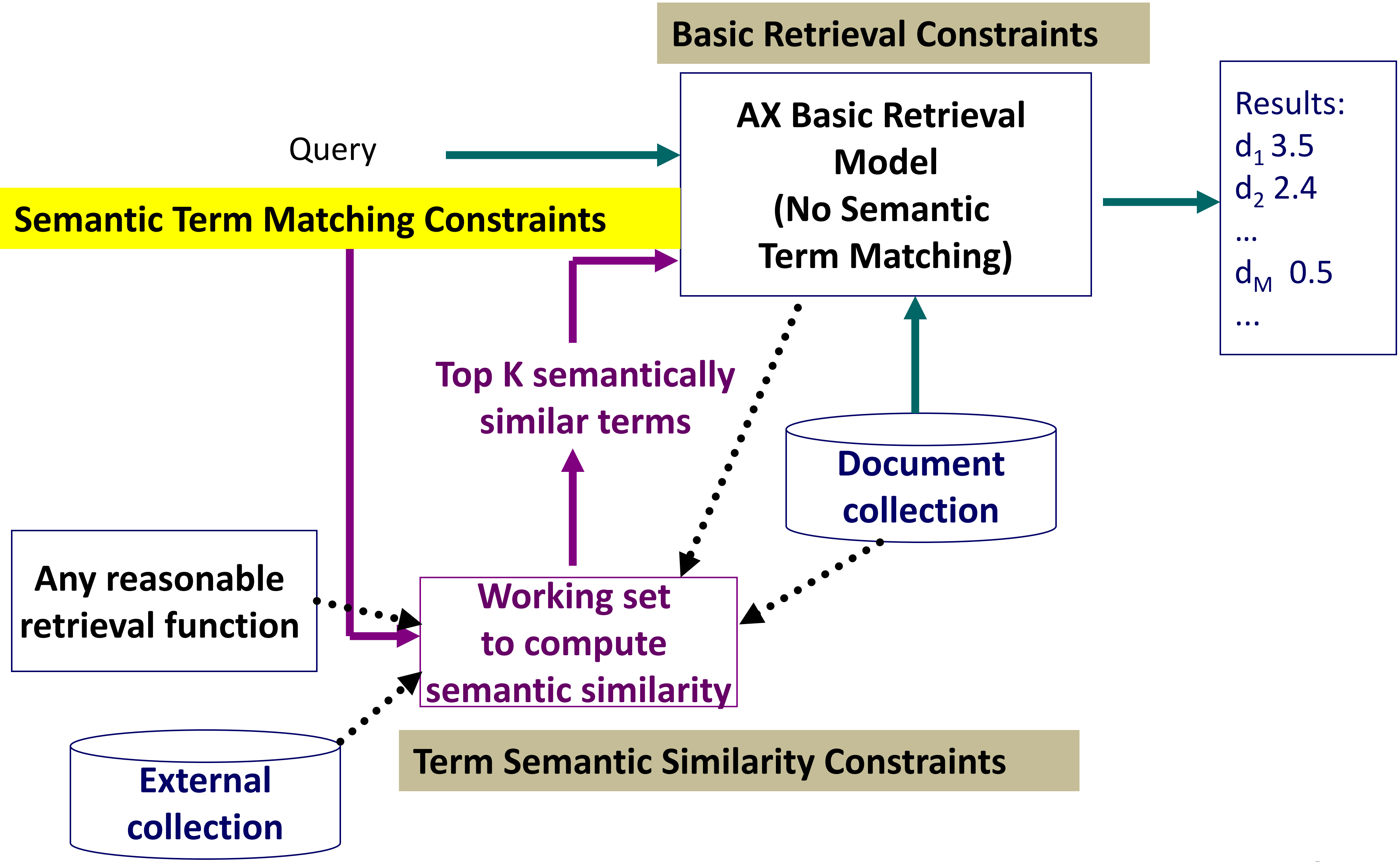
Constraints	Intuitions
TFC1	To favor a document with more occurrences of a query term
TFC2	To ensure that the amount of increase in score due to adding a query term repeatedly must decrease as more terms are added
TFC3	To favor a document matching more distinct query terms
TDC	To penalize the words popular in the collection and assign higher weights to discriminative terms
LNC1	To penalize a long document (assuming equal TF)
LNC2, TF-LNC	To avoid over-penalizing a long document
TF-LNC	To regulate the interaction of TF and document length

Derived Basic AX Retrieval Function

$$S(Q,D)=\sum_{t\in Q\cap D} c(t,Q)\cdot\left(\frac{N}{df(t)}\right)^k\cdot\frac{c(t,D)}{c(t,D)+s+\frac{s\cdot|D|}{avdl}}$$



Semantic Term Matching in AX



Semantic Term Matching Constraints (STMC1)

Semantic Term Matching Heuristic I:

Give a higher score to a document with a term that is more semantically related to a query term

- ***STMC1***

Let $Q=\{q\}$ be a query.

Let $D_1=\{d_1\}$, $D_2=\{d_2\}$ be two documents.

If $\boxed{s(q,d_1) > s(q,d_2)}$

then $S(Q,D_1) > S(Q,D_2)$

Q: "car"
D1: "driver"
D2: "fish"

$$S(Q,D2) < S(Q,D1)$$

s(q,d) is any given semantic similarity function between two terms q and d.

Moreover, s(t,t)>s(t,u)

Semantic Term Matching Constraints

(STMC2 & STMC3)

Semantic Term Matching heuristic II:

Favor semantically similar terms(SMTC3);

Avoid over-favoring semantically similar terms (SMTC2) .

- ***SMTC3***

Let $Q=\{q_1,q_2\}$ be a query and d be non-query term such that $s(d,q_2)>0$. $S(\{q_1\},\{q_1\})=S(\{q_2\},\{q_2\})$

If $|D_1|=|D_2|>1$, $c(q_1,D_1)=|D_1|$ and $c(d_1,D_2)=|D_2|-1$

then $S(Q,D_1) \le S(Q,D_2)$

Q: “safety car”
D1: “safety safety”
D2: “safety driver”

$$S(Q,D_1) \le S(Q,D_2)$$

- ***SMTC2***

Let $Q=\{q\}$ be a query and d be non-query term such that $s(q,d)>0$.
If $|D_1|=1$, $c(q,D_1)=1$, $|D_2|=k$, and $c(d,D_2)=k$

then $S(Q,D_1) \ge S(Q,D_2)$

Q: “car”
D1: “car”
D2: “driver driver”

$$S(Q,D_1) \ge S(Q,D_2)$$

Incorporate Semantic Term Matching

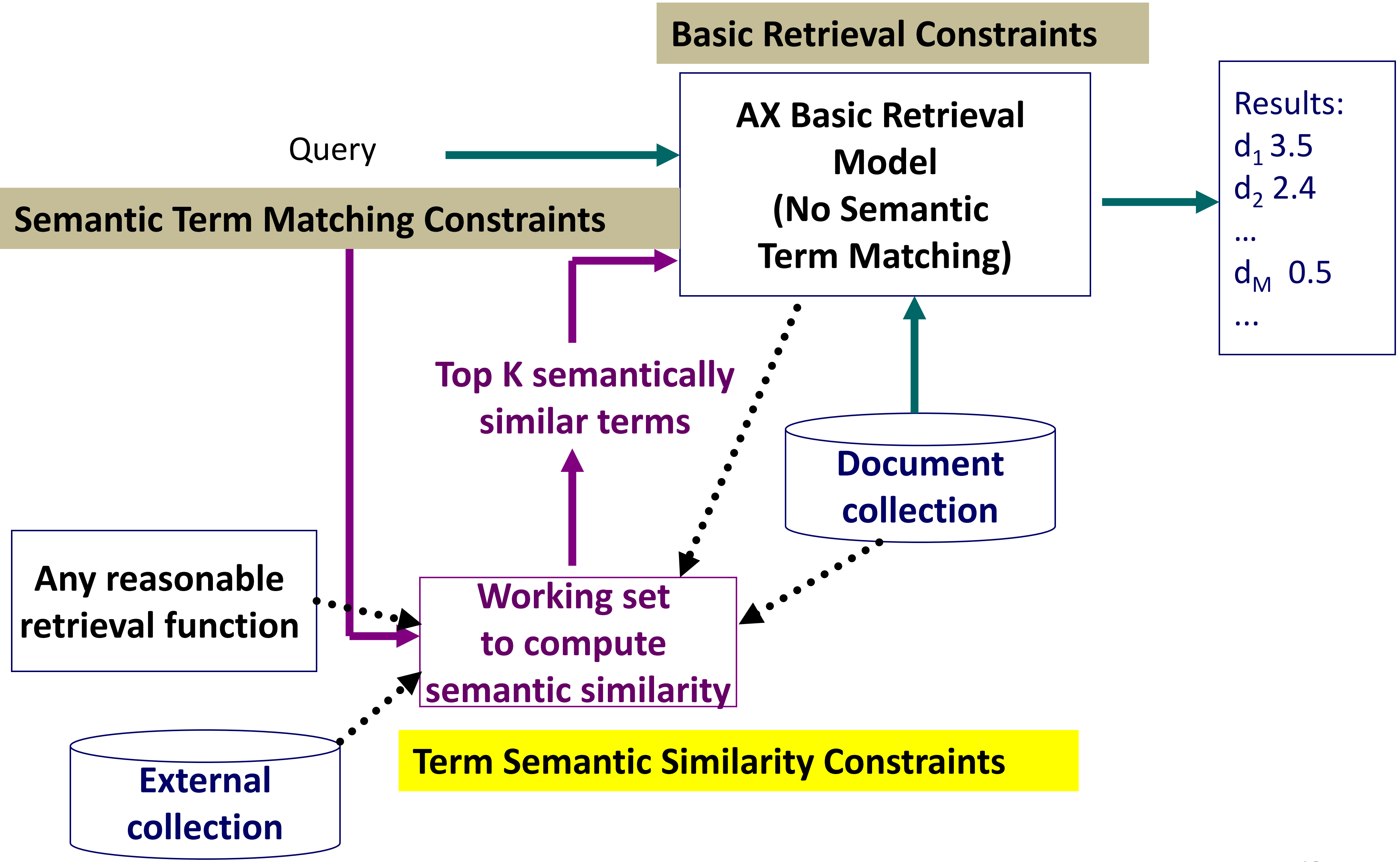
$$S(\{q\},\{d\}) = \begin{cases} weight(q) & q = d \\ 0 & q \neq d \end{cases}$$

$$\Downarrow$$

$$S(\{q\},\{d\}) = \begin{cases} weight(q) \times \dfrac{\boxed{s(q,d)}}{s(q,q)} \times \beta & d \in TopKSim(q) \\ 0 & otherwise \end{cases}$$

STMCs can provide an upper bound and a lower bound.

Semantic Term Matching in AX



Term Semantic Similarity Function

Term co-occurrence → semantic relationships

$$s(q,d) \approx MI(q,d;\boxed{C})$$

TSSC2

C mostly contains correct sense of q

TSSC1

q should not occur in every document of C

Top M ranked documents + $r \times M$ random documents

Documents can be either from collection or external resource (i.e. Web).

Top ranked documents can be returned by any retrieval function.

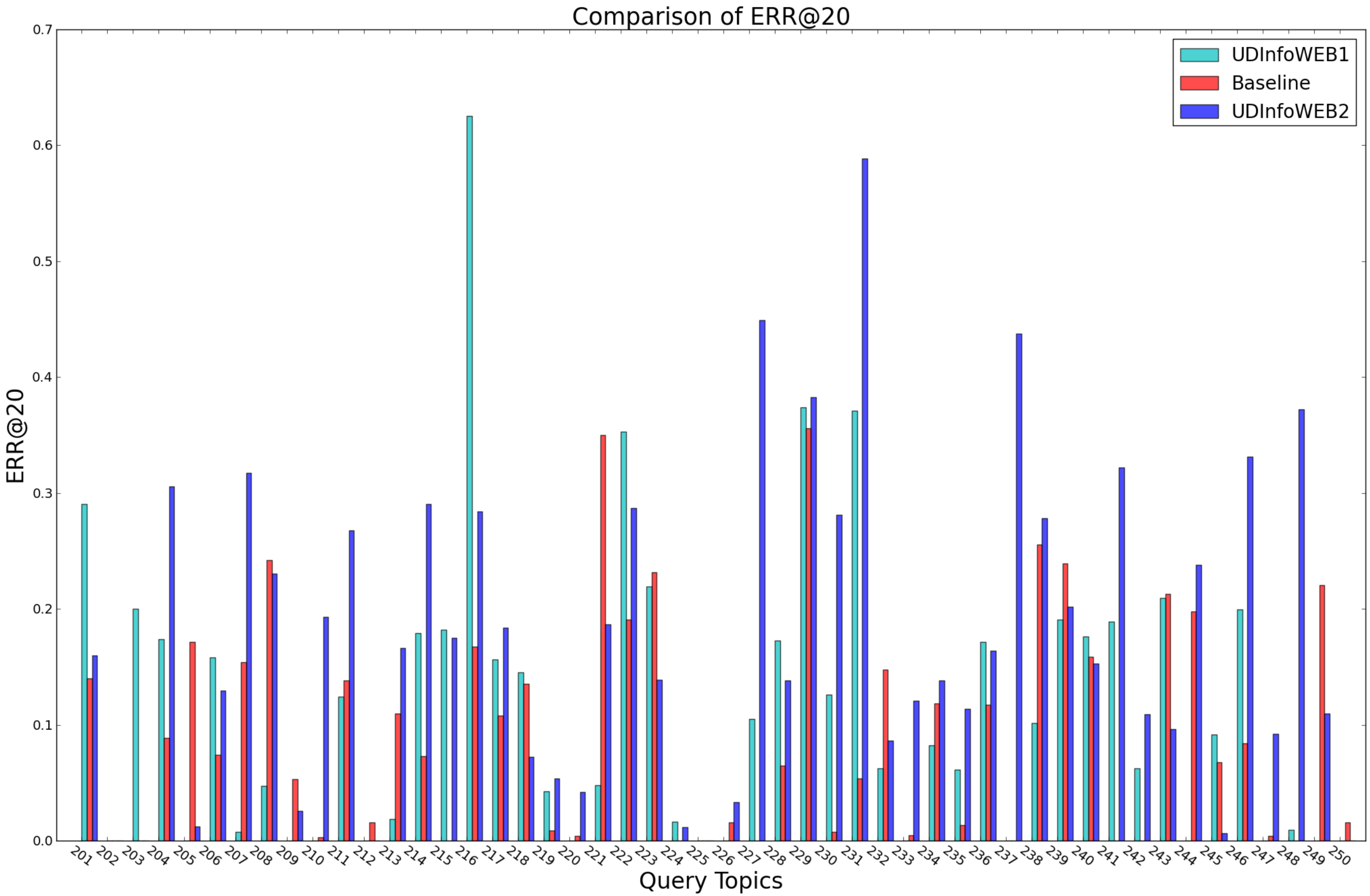
TSSCs provide an upper bound and a lower bound for r

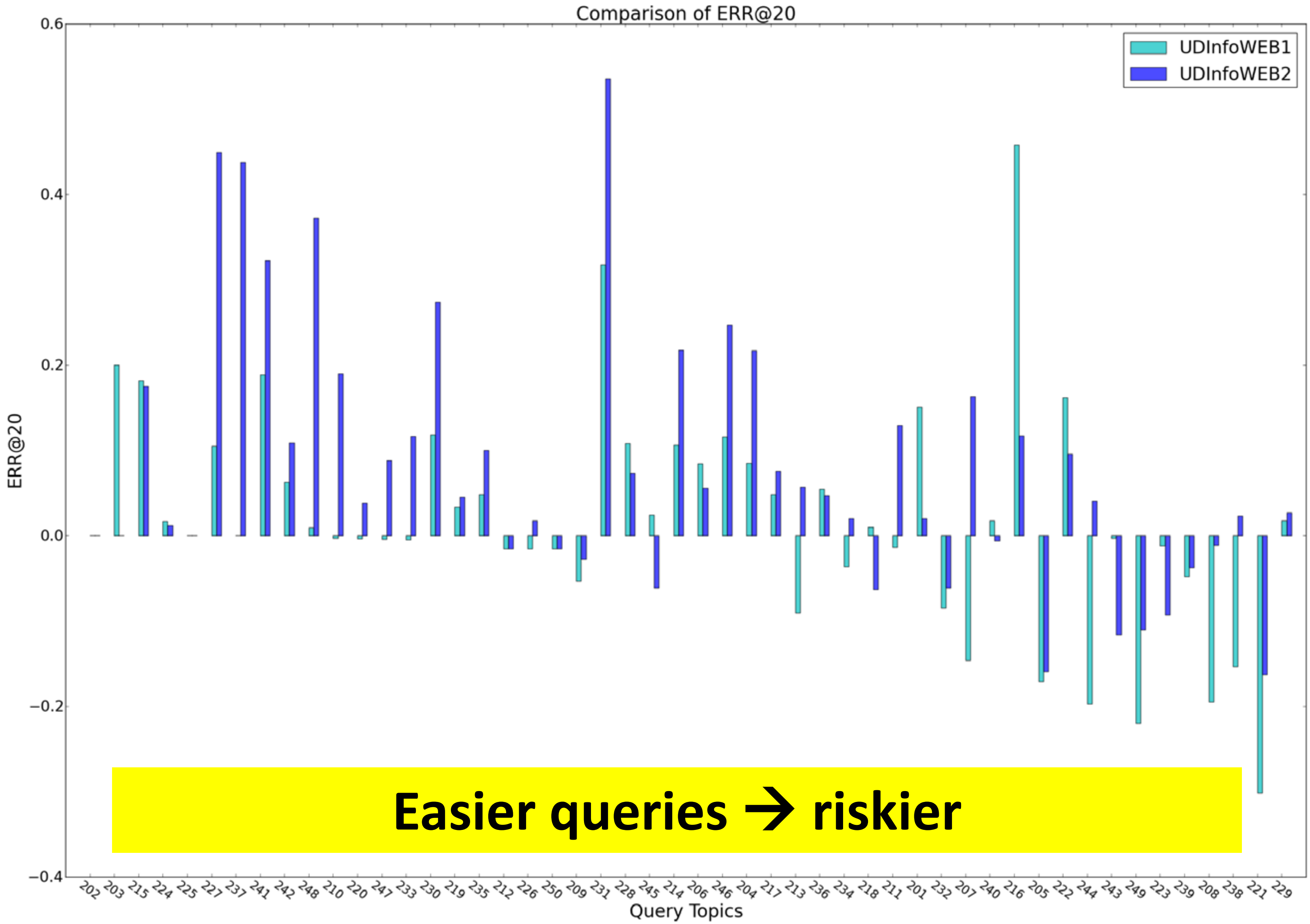
Results of Submitted Runs

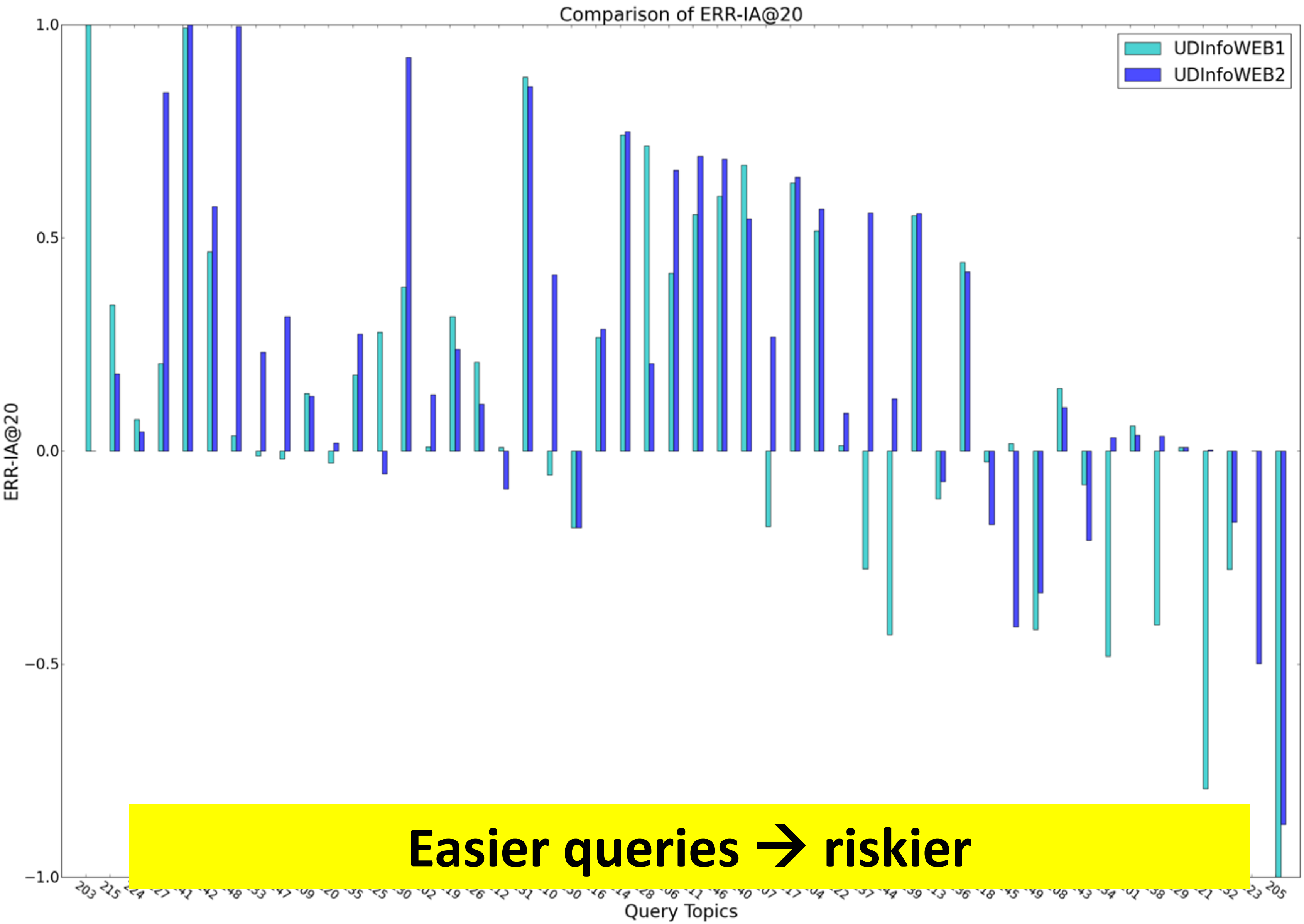
Run IDs	Expansion Source	beta	ERR@20	ERR-IA@20
UDInfolabWEB1	Internal	0.1	0.1149	0.4943
UDInfolabWEB2	External	1.7	**0.1755**	**0.5819**

	UDInfolabWEB1	UDInfolabWEB2
ERR (alpha=0)	0.0185	0.0793
ERR (alpha=1)	-0.0172	0.0604
ERR (alpha=5)	-0.1606	-0.0149
ERR (alpha=10)	-0.3399	-0.1090

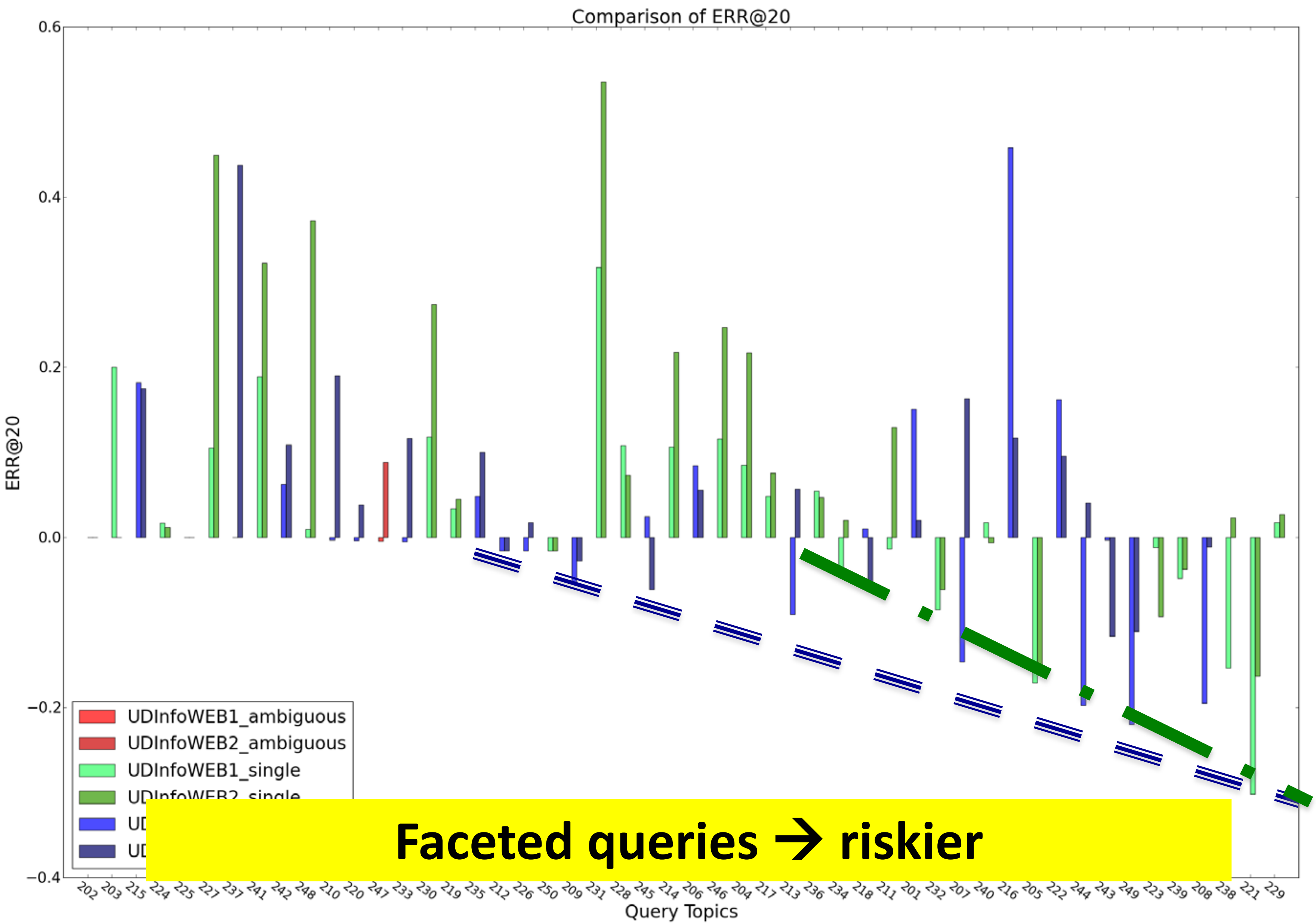
	UDInfolabWEB1	UDInfolabWEB2
ERR-IA (alpha=0)	0.1419	0.2295
ERR-IA (alpha=1)	0.0465	0.1682
ERR-IA (alpha=5)	-0.3352	-0.0771
ERR-IA (alpha=10)	-0.8123	-0.3837

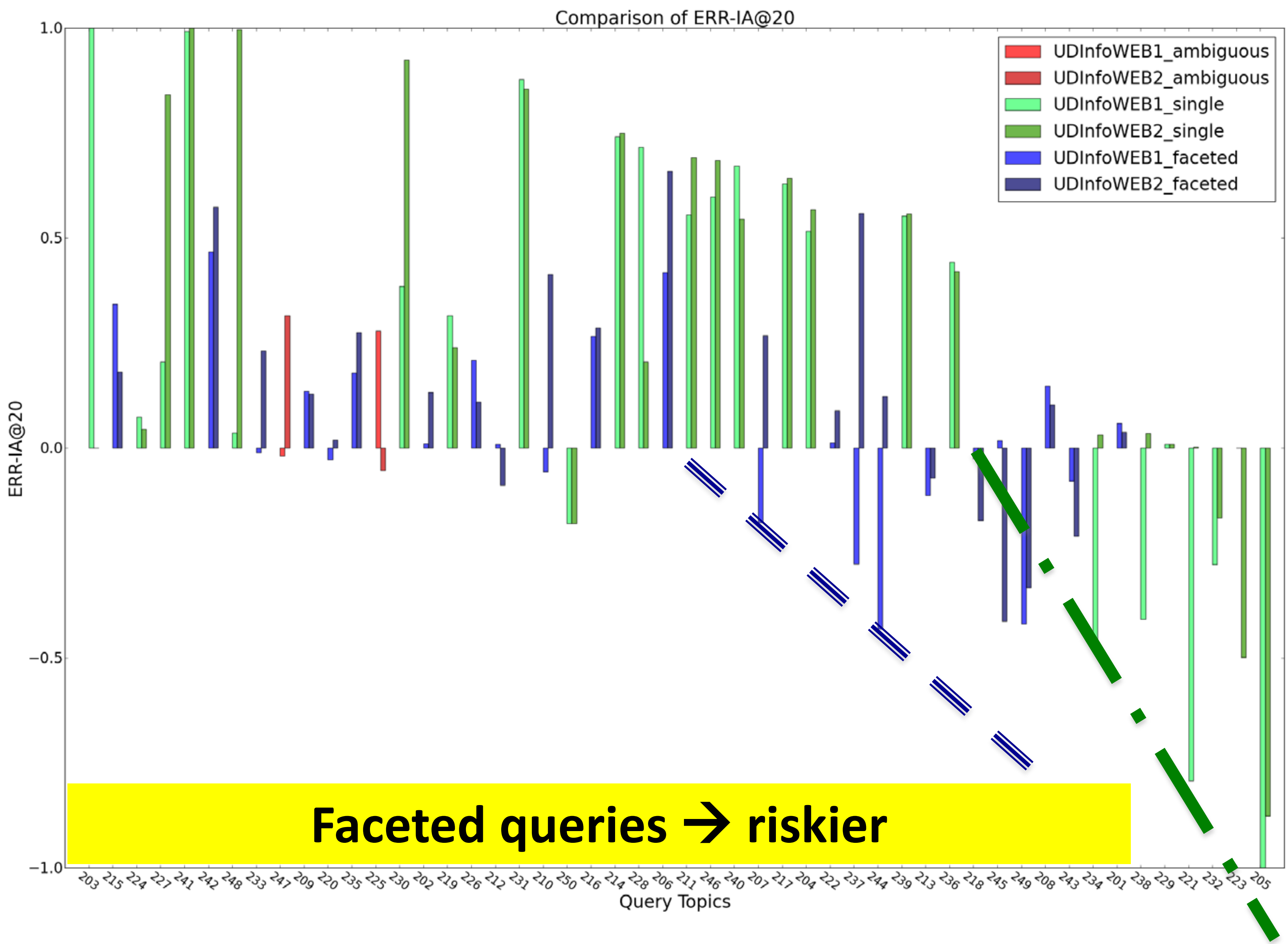

Comparison of ERR@20
UDInfoWEB1
Baseline
UDInfoWEB2
ERR@20
Query Topics
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250


Comparison of ERR@20
UDInfoWEB1
UDInfoWEB2
ERR@20
0.6
0.4
0.2
0.0
−0.2
−0.4
202 203 215 224 225 227 237 241 242 248 210 220 247 233 230 219 235 212 226 250 209 231 228 245 214 206 246 204 217 213 236 234 218 211 201 232 207 240 216 205 222 244 243 249 223 239 208 238 221 229
Query Topics
Easier queries → riskier



Easier queries → riskier


Comparison of ERR@20
ERR@20
0.6
0.4
0.2
0.0
−0.2
−0.4
UDInfoWEB1_ambiguous
UDInfoWEB2_ambiguous
UDInfoWEB1_single
UDInfoWEB2_single
Faceted queries → riskier
202 203 215 224 225 227 237 241 242 248 210 220 247 233 230 219 235 212 226 250 209 231 228 245 214 206 246 204 217 213 236 234 218 211 201 232 207 240 216 205 222 244 243 249 223 239 208 238 221 229
Query Topics


Comparison of ERR-IA@20
ERR-IA@20
1.0
0.5
0.0
−0.5
−1.0
UDInfoWEB1_ambiguous
UDInfoWEB2_ambiguous
UDInfoWEB1_single
UDInfoWEB2_single
UDInfoWEB1_faceted
UDInfoWEB2_faceted
203 215 224 227 241 242 248 233 247 209 220 235 225 230 202 219 226 212 231 210 250 216 214 228 206 211 246 240 207 217 204 222 237 244 239 213 236 218 245 249 208 243 234 201 238 229 221 232 223 205
Query Topics
Faceted queries → riskier

Conclusions and Future Work

- Conclusions
  - Axiomatic approaches are effective.
  - But its effective varies for different queries.
    - Easier → riskier
    - Faceted queries → riskier
- Future work
  - Applying different sets of constraints for different queries for risk minimization

Thank you!

Questions?