Axiomatic Result Re-Ranking

Matthias Hagen Michael Völske Steve Göring Benno Stein

Bauhaus-University Weimar

www.webis.de

CIKM 2016
Indianapolis
October 24-28, 2016

25th ACM International Conference on Information and Knowledge Management (CIKM); Indianapolis, Indiana, USA, 2016

Axiomatic Result Re-Ranking

Abstract

- Axiomatic IR has identified a diverse set of constraints that retrieval models should fulfill, but so far these have been limited to theoretical analysis

- We incorporate axioms into the retrieval process via re-ranking

- Large-scale study on Clueweb corpora to show feasibility

A Brief Tour of Axiomatic IR

A Brief Tour of Axiomatic IR

Observations

- Common strong baseline retrieval models perform similarly well, although derived very differently (BM25, PL2, Query Likelihood...)
- Even minor variations tend to fail in some way or another; why?

Axiomatic IR Answer: these models share beneficial properties, independently of how they are derived

Research Goal: identify and formalize these properties as axioms.

A Brief Tour of Axiomatic IR

Axioms

Successful retrieval functions share similar heuristics:

Example:

$$BM25(Q,D) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i,D) \cdot (k_1+1)}{TF(q_i,D) \; + \; k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

A Brief Tour of Axiomatic IR

Axioms

Successful retrieval functions share similar heuristics:

- TF weighting

Example:

$$BM25(Q,D) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i, D) \cdot (k_1 + 1)}{TF(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

A Brief Tour of Axiomatic IR

Axioms

Successful retrieval functions share similar heuristics:

- TF weighting
- IDF weighting

Example:

$$BM25(Q, D) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i, D) \cdot (k_1 + 1)}{TF(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

A Brief Tour of Axiomatic IR

Axioms

Successful retrieval functions share similar heuristics:

- TF weighting
- IDF weighting
- Length normalization

Example:

$$BM25(Q, D) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i, D) \cdot (k_1 + 1)}{TF(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

A Brief Tour of Axiomatic IR

Axioms

Successful retrieval functions share similar heuristics:

- TF weighting
- IDF weighting
- Length normalization

Example:

$$BM25(Q, D) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i, D) \cdot (k_1 + 1)}{TF(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

Axioms formally capture these heuristics, and how they should be used.

A Brief Tour of Axiomatic IR

Axioms

Purpose	Acronyms	Source
Term frequency	TFC1–TFC3	[Fang, Tao, Zhai; SIGIR’04]
	TDC	[Fang, Tao, Zhai; SIGIR’04]
Document length	LNC1 + LNC2	[Fang, Tao, Zhai; SIGIR’04]
	TF-LNC	[Fang, Tao, Zhai; SIGIR’04]
	QLNC	[Cummins, O’Riordan; CIKM’12]
Lower bound	LB1 + LB2	[Lv, Zhai; CIKM’11]
Query aspects	REG	[Zheng, Fang; ECIR’10]
	DIV	[Gollapurdi, Sharma; WWW’09]
Semantic similarity	STMC1 + STMC2	[Fang, Zhai; SIGIR’06]
	STMC3	[Fang, Zhai; SIGIR’06]
	TSSC1 + TSSC2	[Fang, Zhai; SIGIR’06]
Term proximity	PHC + CCC	[Tao, Zhai; SIGIR’07]

A Brief Tour of Axiomatic IR

Term Frequency Constraints

TFC1	Give a higher score to a document with more occurrences of a query term.
TFC2	The amount of increase in the score due to adding a query term must decrease as we add more terms.
TFC3	Favor a document with more distinct query terms.

Length Normalization Constaints

LNC1	Penalize long documents.
LNC2	Avoid over-penalizing long documents.
TF-LNC	Regularize the interaction of TF and document length.

Lower-bounding Term Frequency Constraints

LB1	The presence-absence gap shouldn't be closed due to length normalization.
LB2	Repeated occurrence isn't as important as first occurrence.

A Brief Tour of Axiomatic IR

Term Frequency Constraints

TFC1	Give a higher score to a document with more occurrences of a query term.
TFC2	The amount of increase in the score due to adding a query term must decrease as we add more terms.
TFC3	Favor a document with more distinct query terms.

Length Normalization Constaints

LNC1	Penalize long documents.
LNC2	Avoid over-penalizing long documents.
TF-LNC	Regularize the interaction of TF and document length.

Lower-bounding Term Frequency Constraints

LB1	The presence-absence gap shouldn't be closed due to length normalization.
LB2	Repeated occurrence isn't as important as first occurrence.

A Brief Tour of Axiomatic IR

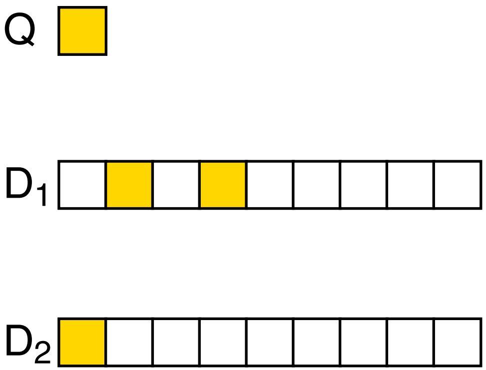
Axiom Examples: TFC1

TFC1 Give a higher score to a document with more occurrences of a query term.

Given:

- Single-term query $Q = \{q\}$
- Documents D_1, D_2 with $|D_1| = |D_2|$

IF $TF(q, D_1) > TF(q, D_2)$ THEN $Score(Q, D_1) > Score(Q, D_2)$



A Brief Tour of Axiomatic IR

Axiom Examples: LB2

LB2 Repeated occurrence isn't as important as first occurrence.

Given:

- Two-term query $Q = \{q_1, q_2\}$

Q 

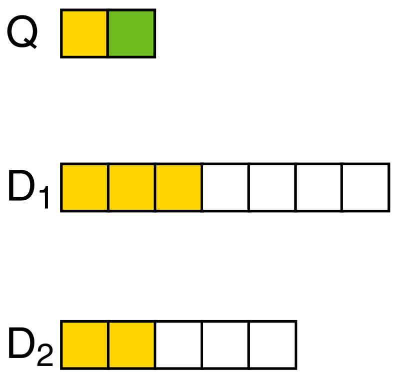
A Brief Tour of Axiomatic IR

Axiom Examples: LB2

LB2 Repeated occurrence isn't as important as first occurrence.

Given:

- Two-term query $Q = \{q_1, q_2\}$
- Documents D_1, D_2 with $TF(q_1, D_1) > 0$ and $TF(q_1, D_2) > 0$ and $TF(q_2, D_1) = TF(q_2, D_2) = 0$



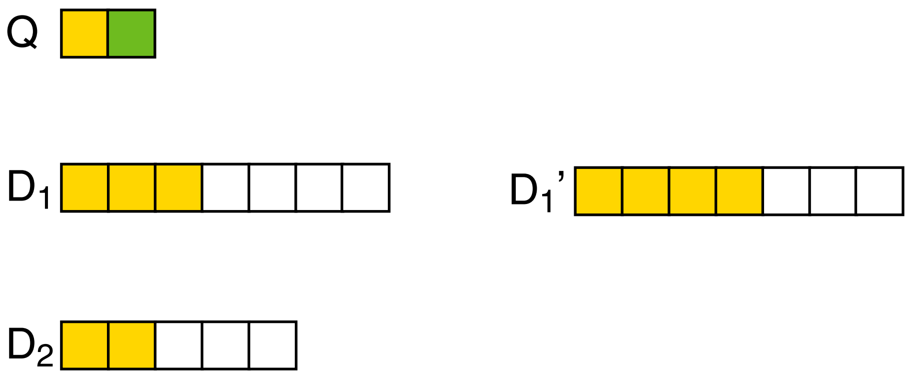
A Brief Tour of Axiomatic IR

Axiom Examples: LB2

LB2 Repeated occurrence isn't as important as first occurrence.

Given:

- Two-term query $Q = \{q_1, q_2\}$
- Documents D_1, D_2 with $TF(q_1, D_1) > 0$ and $TF(q_1, D_2) > 0$ and $TF(q_2, D_1) = TF(q_2, D_2) = 0$
- Document $D_1' = D_1 \cup \{q_1\} \setminus \{t_1\}$ for any $t_1 \in D_1$, $t_1 \notin Q$



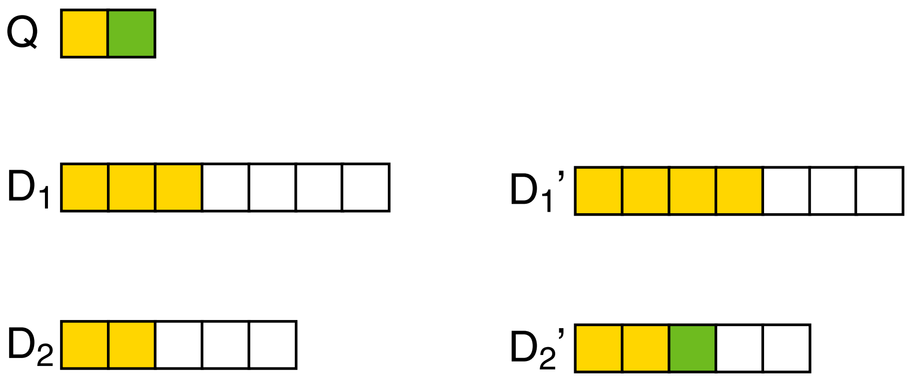
A Brief Tour of Axiomatic IR

Axiom Examples: LB2

LB2 Repeated occurrence isn't as important as first occurrence.

Given:

- Two-term query $Q = \{q_1, q_2\}$
- Documents D_1, D_2 with $TF(q_1, D_1) > 0$ and $TF(q_1, D_2) > 0$ and $TF(q_2, D_1) = TF(q_2, D_2) = 0$
- Document $D_1' = D_1 \cup \{q_1\} \setminus \{t_1\}$ for any $t_1 \in D_1$, $t_1 \notin Q$
- Document $D_2' = D_2 \cup \{q_2\} \setminus \{t_2\}$ for any $t_2 \in D_2$, $t_2 \notin Q$



A Brief Tour of Axiomatic IR

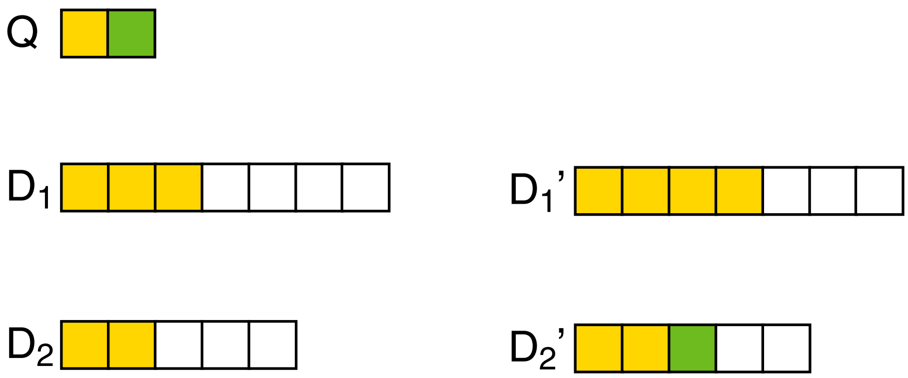
Axiom Examples: LB2

LB2 Repeated occurrence isn't as important as first occurrence.

Given:

- Two-term query $Q = \{q_1, q_2\}$
- Documents D_1, D_2 with $TF(q_1, D_1) > 0$ and $TF(q_1, D_2) > 0$ and $TF(q_2, D_1) = TF(q_2, D_2) = 0$
- Document $D_1' = D_1 \cup \{q_1\} \setminus \{t_1\}$ for any $t_1 \in D_1$, $t_1 \notin Q$
- Document $D_2' = D_2 \cup \{q_2\} \setminus \{t_2\}$ for any $t_2 \in D_2$, $t_2 \notin Q$

IF $Score(Q, D_1) = Score(Q, D_2)$ THEN $Score(Q, D_1') < Score(Q, D_2')$



- BM25 (no matter the parameter setting) violates the LB2 constraint
- A minor modification corrects this, for consistently better performance

  [Lv and Zhai, CIKM'11]

$$BM25\ \ (Q,D)=\sum_{i=1}^{n} IDF(q_i)\ \cdot \quad \frac{TF(q_i,D)\cdot(k_1+1)}{TF(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)}$$

- BM25 (no matter the parameter setting) violates the LB2 constraint
- A minor modification corrects this, for consistently better performance
  [Lv and Zhai, CIKM'11]

$$BM25^{+}(Q,D) = \sum_{i=1}^{n} IDF(q_i) \cdot \left(\frac{TF(q_i,D)\cdot(k_1+1)}{TF(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)} + \delta \right)$$

A Brief Tour of Axiomatic IR

Axiomatic Analysis

- BM25 (no matter the parameter setting) violates the LB2 constraint
- A minor modification corrects this, for consistently better performance
  [Lv and Zhai, CIKM'11]

$$BM25^{+}(Q,D) = \sum_{i=1}^{n} IDF(q_i) \cdot \left(\frac{TF(q_i,D) \cdot (k_1+1)}{TF(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)} + \delta \right)$$

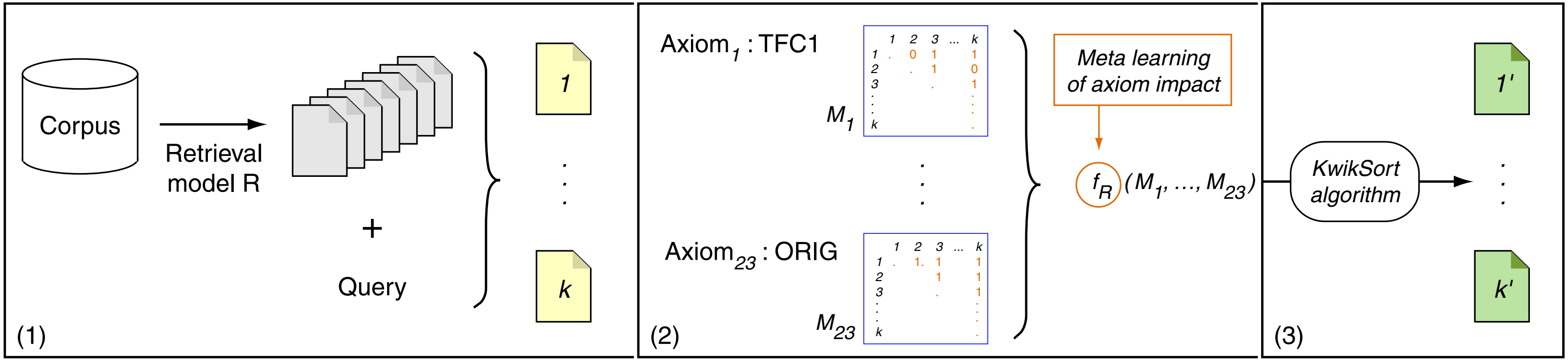
Our research question:

How can we automate the "axiomatization" of retrieval models?

Axiomatic Result Re-ranking

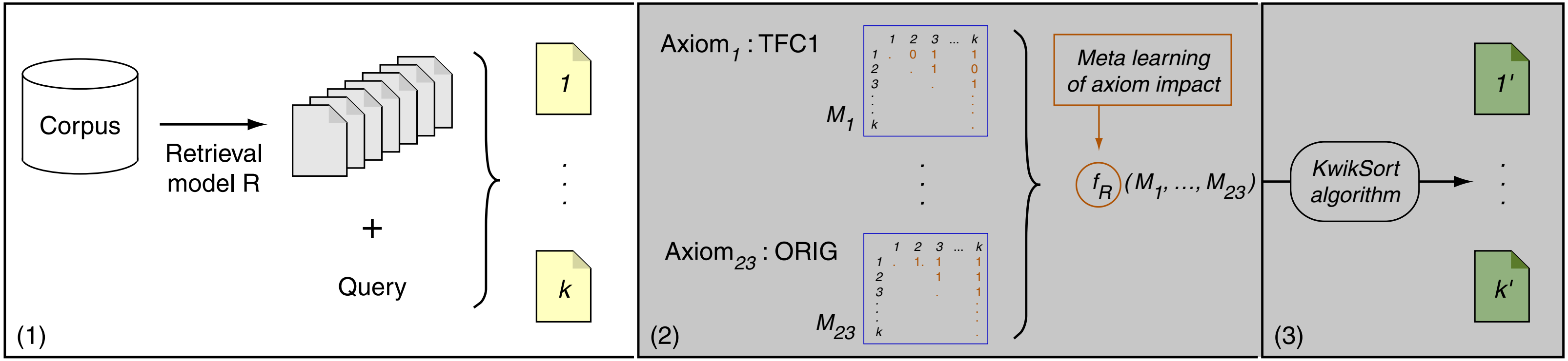
Axiomatic Result Re-ranking

Axiomatic Re-ranking Pipeline



Axiomatic Result Re-ranking

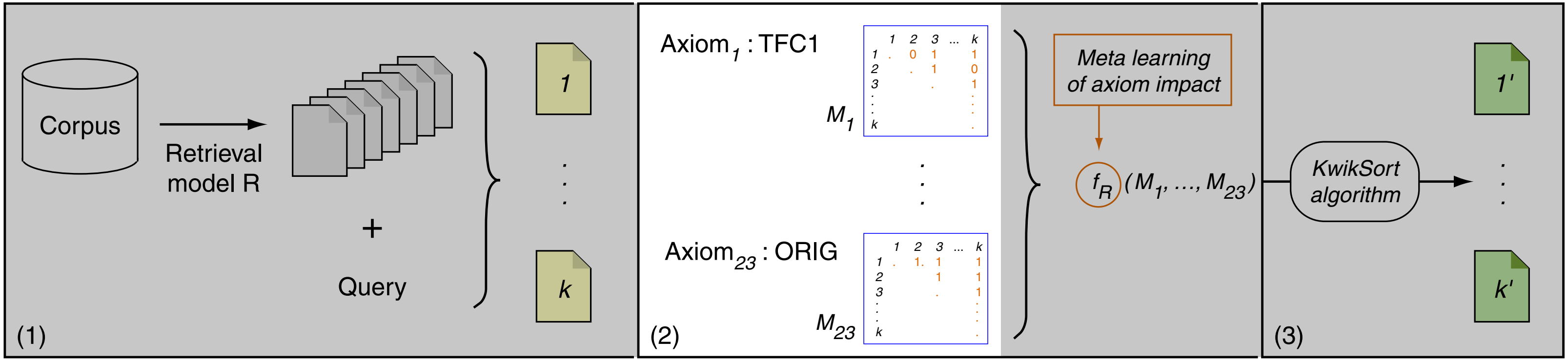
Axiomatic Re-ranking Pipeline



1. Retrieve an initial top-k result set.

Axiomatic Result Re-ranking

Axiomatic Re-ranking Pipeline

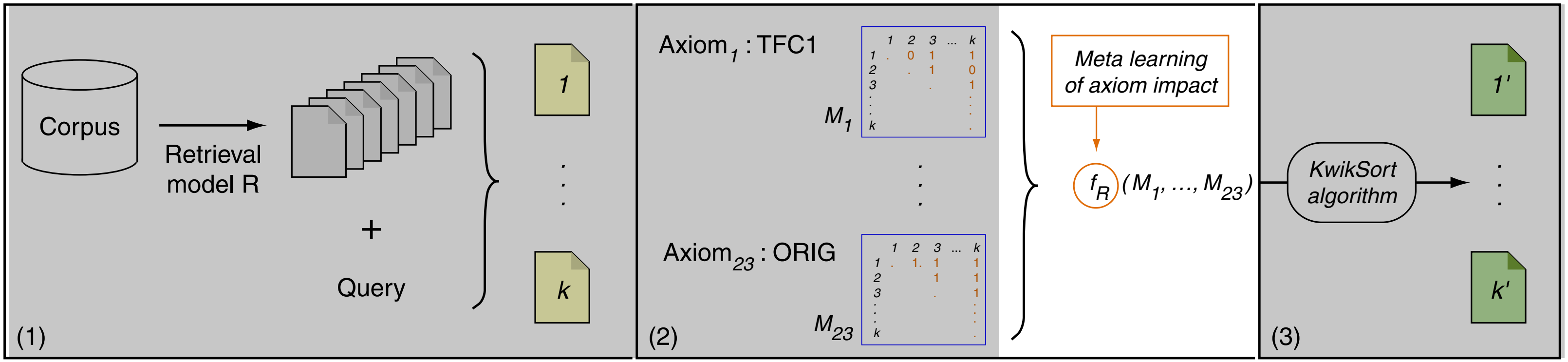


1. Retrieve an initial top-k result set.

2. (a) Compute re-ranking preferences of various axioms.

Axiomatic Result Re-ranking

Axiomatic Re-ranking Pipeline

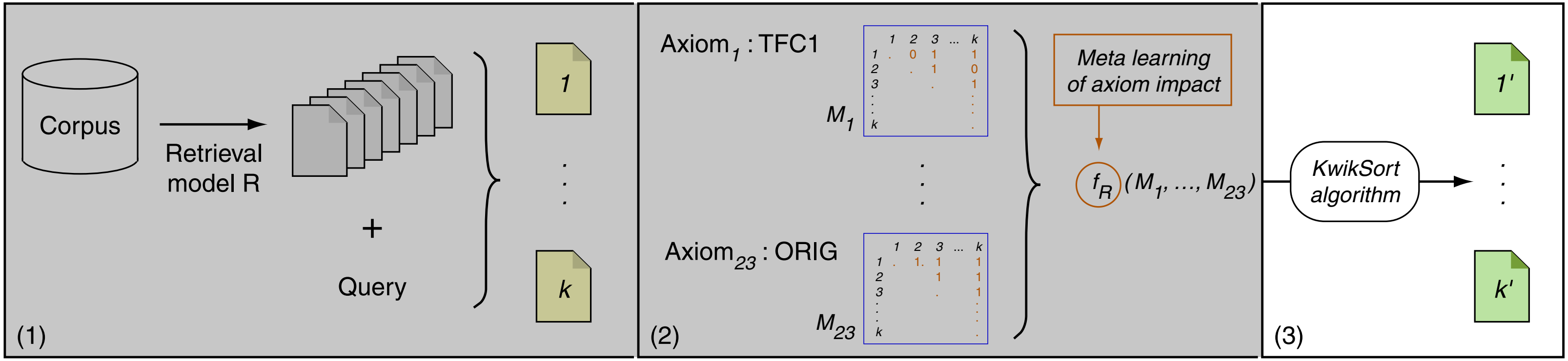


1. Retrieve an initial top-k result set.

2. (a) Compute re-ranking preferences of various axioms.
   (b) Aggregate re-ranking preferences.

Axiomatic Result Re-ranking

Axiomatic Re-ranking Pipeline



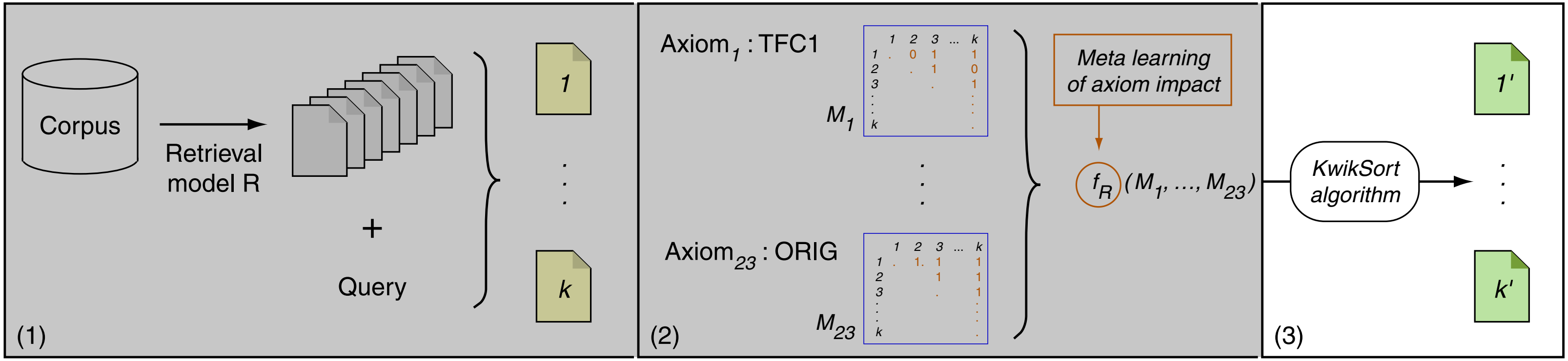
1. Retrieve an initial top-k result set.

2. (a) Compute re-ranking preferences of various axioms.
   (b) Aggregate re-ranking preferences.

3. Re-rank the initial result set.

Axiomatic Result Re-ranking

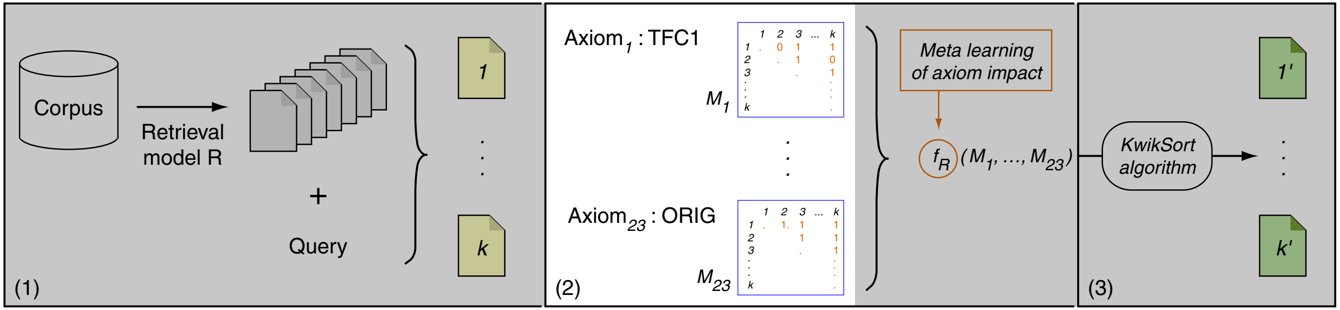
Axiomatic Re-ranking Pipeline



1. Retrieve an initial top-k result set.
2. (a) Compute re-ranking preferences of various axioms.
   (b) Aggregate re-ranking preferences.
3. Re-rank the initial result set.

Axiomatic Result Re-ranking

Requirements on Axioms

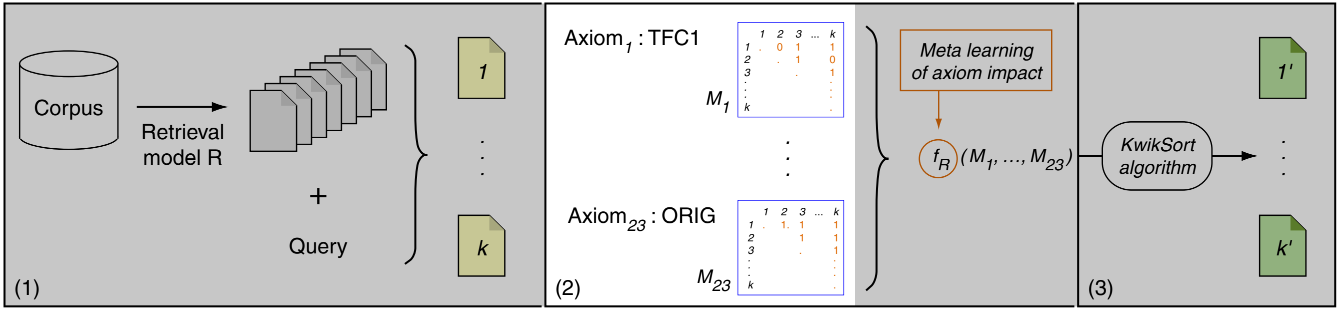


Re-state each axiom as a triple:

$$A = (precondition, filter, conclusion)$$

Axiomatic Result Re-ranking

Requirements on Axioms



Re-state each axiom as a triple:

$$A = (precondition, filter, conclusion)$$

Given axiom A and document pair D_1, D_2:

- The *precondition* evaluates whether or not A can be applied to D_1, D_2
- If the *filter* condition is satisfied...
- The *conclusion* is a ranking preference of the form $D_1 >_A D_2$

Axiomatic Result Re-ranking

Adapting Existing Axioms

1. Convert to our triple formulation
2. Relax equality constraints and tighten inequality constraints
3. Modify conclusion to express a ranking preference

Axiomatic Result Re-ranking

Adapting Existing Axioms

1. Convert to our triple formulation
2. Relax equality constraints and tighten inequality constraints
3. Modify conclusion to express a ranking preference

Example: TFC1

Given:

- Single-term query $Q = \{q\}$
- Documents D_1, D_2 with $|D_1| = |D_2|$

IF $TF(q, D_1) > TF(q, D_2)$ THEN $Score(Q, D_1) > Score(Q, D_2)$

Axiomatic Result Re-ranking

Adapting Existing Axioms

1. Convert to our triple formulation
2. Relax equality constraints and tighten inequality constraints
3. Modify conclusion to express a ranking preference

Example: TFC1

Given:

- Single-term query $Q = \{q\}$
- Documents D_1, D_2 with $|D_1| = |D_2|$

$$\text{IF } TF(q, D_1) > TF(q, D_2) \text{ THEN } Score(Q, D_1) > Score(Q, D_2)$$

Precondition := $|D_1| \approx_{10\%} |D_2|$

Filter := $TF(q, D_1) >_{10\%} TF(q, D_2)$

Conclusion := $D_1 >_{TFC1} D_2$

Axiomatic Result Re-ranking

Adapting Existing Axioms

Purpose	Acronyms	Adapted
Term frequency	TFC1–TFC3	✓
	TDC	✓
Document length	LNC1 + LNC2	✓
	TF-LNC	✓
	QLNC	✗
Lower bound	LB1 + LB2	✓
Query aspects	REG	✓
	DIV	✓
Semantic similarity	STMC1 + STMC2	✓
	STMC3	✗
	TSSC1 + TSSC2	✗
Term proximity	PHC + CCC	✗
	QPHRA	New
	PROX1–5	New
Other	ORIG	New

Axiomatic Result Re-ranking

New Term Proximity Axioms

Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

Precondition: both documents contain all query terms.

Give preference to the document where:

PROX1 Query term pairs are closer together on average.

PROX2 Query terms first occur earlier in the document.

PROX3 The whole query as a phrase occurs earlier in the document.

PROX4 The number of non-query terms in the closest grouping of all query terms is smaller.

PROX5 The average shortest text span containing all query terms is smaller.

Axiomatic Result Re-ranking

New Term Proximity Axioms

Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

PROX1 Query term pairs are closer together on average.

Q

Axiomatic Result Re-ranking

New Term Proximity Axioms

Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

PROX1 Query term pairs are closer together on average.

Consider query term pairs $P = \{(i, j) \mid q_i, q_j \in Q, i < j\}$

Q 1 2 3

P = { (1,2), (1,3), (2,3) }

Axiomatic Result Re-ranking

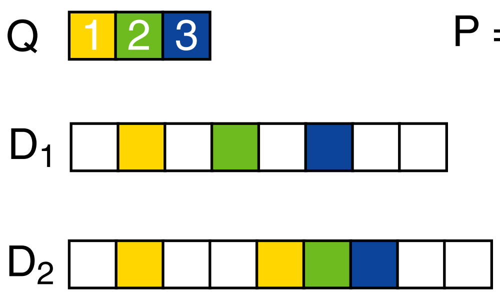
New Term Proximity Axioms

Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

PROX1 Query term pairs are closer together on average.

Consider query term pairs $P = \{(i, j) \mid q_i, q_j \in Q, i < j\}$

Precondition := both documents contain all query terms



P = { (1,2), (1,3), (2,3) }

Axiomatic Result Re-ranking

New Term Proximity Axioms

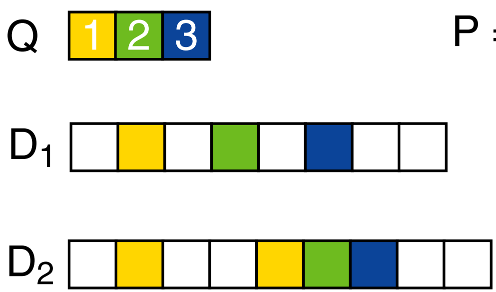
Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

PROX1 Query term pairs are closer together on average.

Consider query term pairs $P = \{(i,j) \mid q_i, q_j \in Q, i < j\}$ and count $\delta(D,i,j)$ average number of words between q_i, q_j in D

Precondition := both documents contain all query terms

Filter := $\pi(Q, D_1) < \pi(Q, D_2)$ where $\pi(Q,D) = \frac{1}{|P|} \sum_{(i,j) \in P} \delta(D,i,j)$



P = { (1,2), (1,3), (2,3) }

$\pi(Q,D_i)$

1/3 (1 + 3 + 1) = 5/3

Axiomatic Result Re-ranking

New Term Proximity Axioms

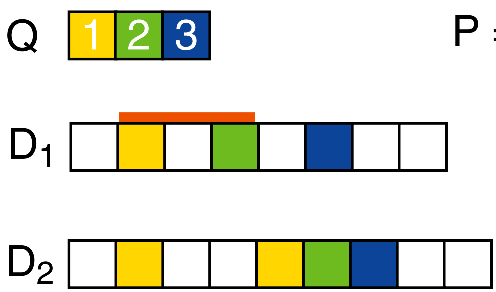
Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

PROX1 Query term pairs are closer together on average.

Consider query term pairs $P = \{(i,j) \mid q_i, q_j \in Q, i < j\}$ and count $\delta(D,i,j)$ average number of words between q_i, q_j in D

Precondition := both documents contain all query terms

Filter := $\pi(Q, D_1) < \pi(Q, D_2)$ where $\pi(Q,D) = \frac{1}{|P|}\sum_{(i,j)\in P}\delta(D,i,j)$



P = { (1,2), (1,3), (2,3) }

$\pi(Q,D_i)$

1/3 (1 + 3 + 1) = 5/3

Axiomatic Result Re-ranking

New Term Proximity Axioms

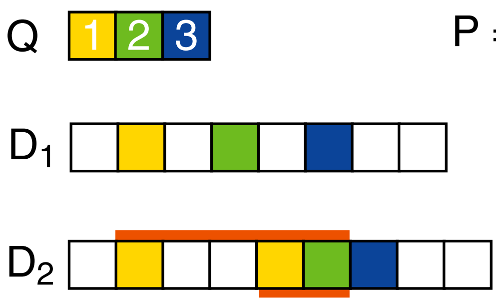
Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \dots, q_n\}$

PROX1   Query term pairs are closer together on average.

Consider query term pairs $P = \{(i,j) \mid q_i, q_j \in Q, i < j\}$ and count $\delta(D,i,j)$ average number of words between q_i, q_j in D

Precondition := both documents contain all query terms

Filter := $\pi(Q,D_1) < \pi(Q,D_2)$ where $\pi(Q,D) = \frac{1}{|P|}\sum_{(i,j)\in P}\delta(D,i,j)$



P = { (1,2), (1,3), (2,3) }

π(Q,D$_i$)

1/3 (1 + 3 + 1) = 5/3

1/3 ((3+0)/2 + (4+1)/2 + 0) = 4/3

Axiomatic Result Re-ranking

New Term Proximity Axioms

Given two documents D_1, D_2 and multi-term query $Q = \{q_1, q_2, \ldots, q_n\}$

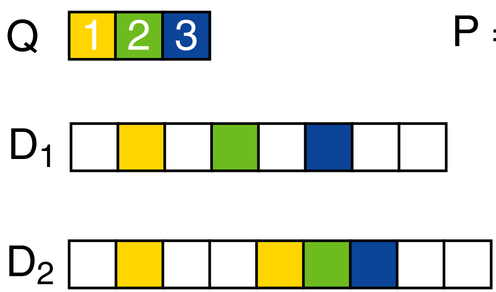
PROX1 Query term pairs are closer together on average.

Consider query term pairs $P = \{(i,j) \mid q_i, q_j \in Q, i < j\}$ and count $\delta(D,i,j)$ average number of words between q_i, q_j in D

Precondition := both documents contain all query terms

Filter := $\pi(Q, D_1) < \pi(Q, D_2)$ where $\pi(Q, D) = \frac{1}{|P|} \sum_{(i,j) \in P} \delta(D,i,j)$

Conclusion := $D_1 >_{PROX1} D_2$



P = { (1,2), (1,3), (2,3) }

$\pi(Q,D_i)$

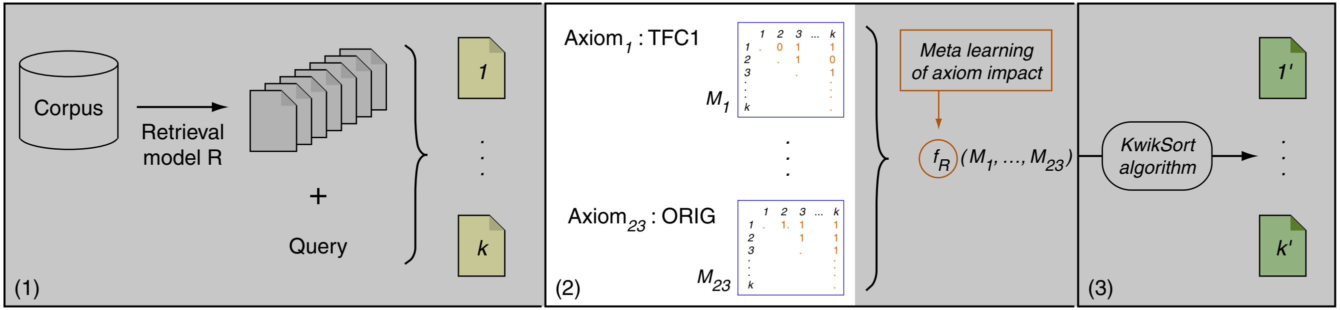
D_1: 1/3 (1 + 3 + 1) = 5/3

D_2: 1/3 ((3+0)/2 + (4+1)/2 + 0) = 4/3

$D_1 >_{PROX1} D_2 = 0$ $D_2 >_{PROX1} D_1 = 1$

Axiomatic Result Re-ranking

Axiom Preference Aggregation

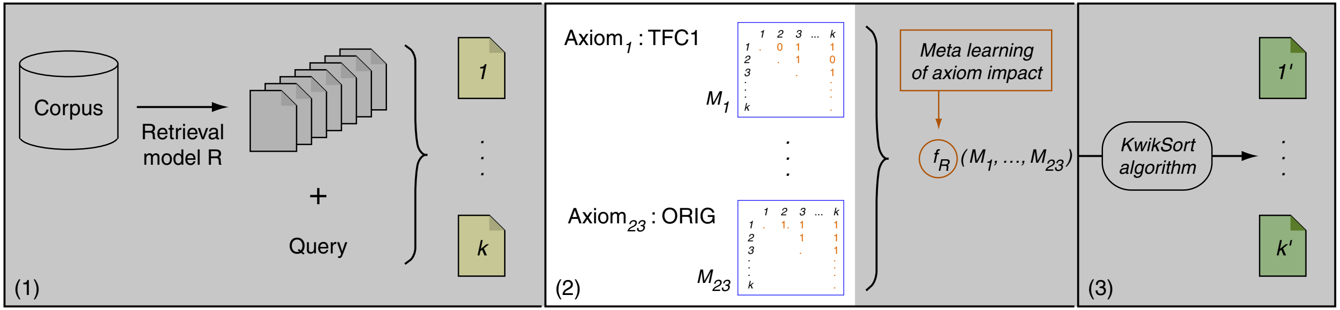


An initial result set $\{D_1, \ldots, D_k\}$ and axiom A yield a k-by-k preference matrix

$$M_A[i,j] = \begin{cases} 1 & \text{if } D_i >_A D_j, \\ 0 & \text{otherwise.} \end{cases}$$

Axiomatic Result Re-ranking

Axiom Preference Aggregation



An initial result set $\{D_1, \ldots, D_k\}$ and axiom A yield a k-by-k preference matrix

$$M_A[i,j] = \begin{cases} 1 & \text{if } D_i >_A D_j, \\ 0 & \text{otherwise.} \end{cases}$$

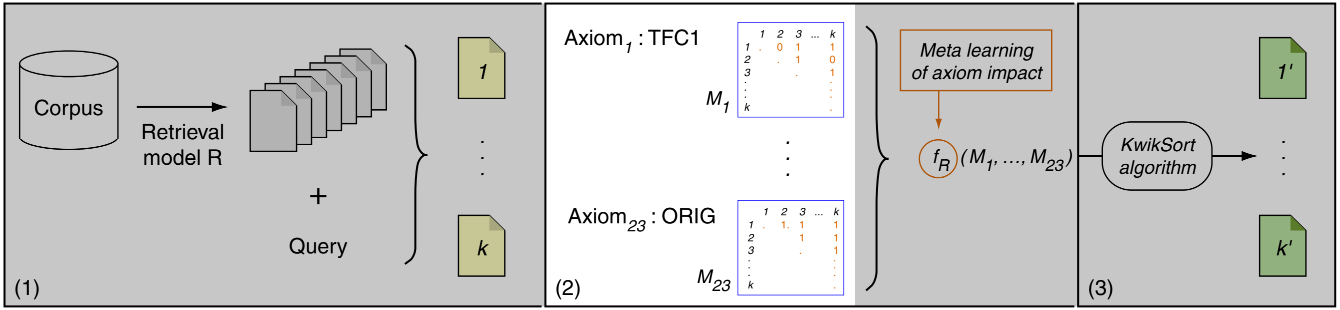
A_1 : TFC1

M_1

	1	2	3	...	k
1	.	0	1		1
2		.	1		0
3			.		1
⋮					⋮
k					.

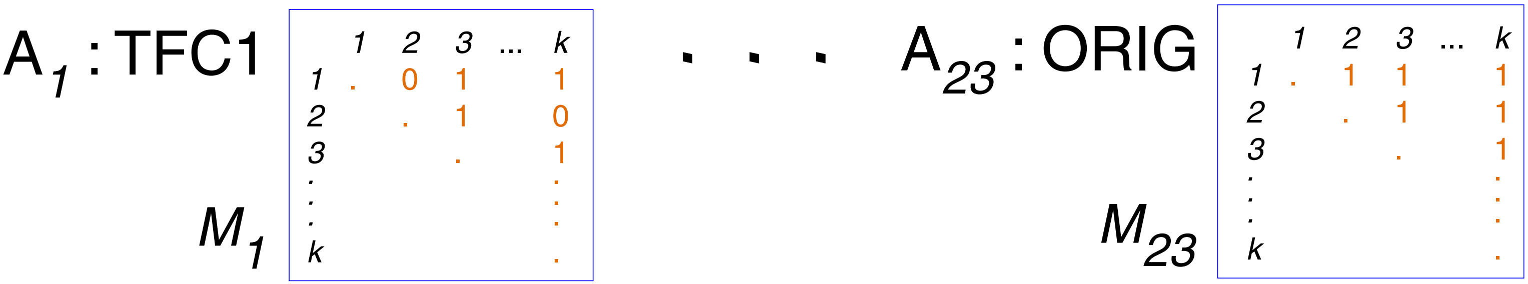
Axiomatic Result Re-ranking

Axiom Preference Aggregation



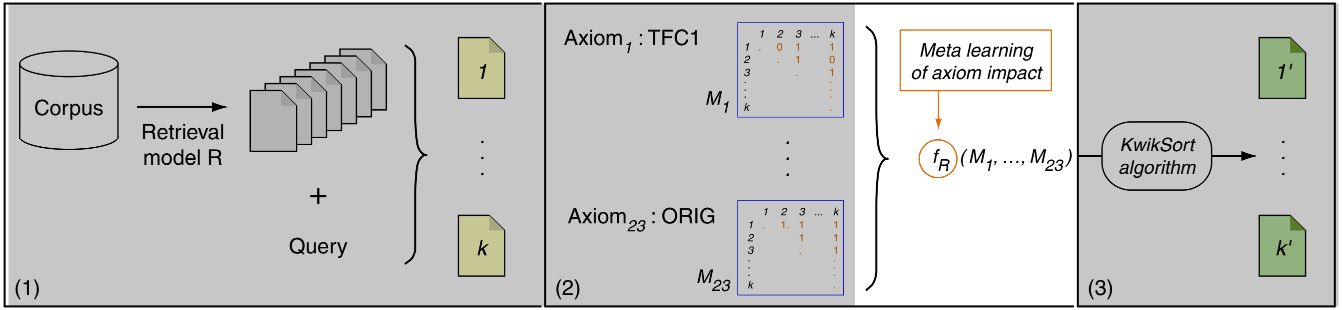
An initial result set $\{D_1, \ldots, D_k\}$ and axiom A yield a k-by-k preference matrix

$$M_A[i,j] = \begin{cases} 1 & \text{if } D_i >_A D_j, \\ 0 & \text{otherwise.} \end{cases}$$



Axiomatic Result Re-ranking

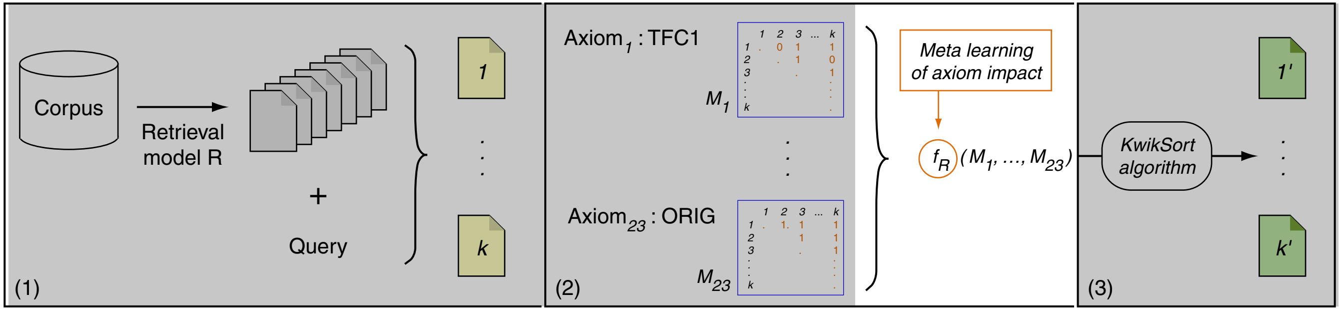
Axiom Preference Aggregation



For a set of m axioms $\{A_1, \ldots A_m\}$, aggregate the individual preference matrices and use the aggregate preference matrix for re-ranking.

Axiomatic Result Re-ranking

Axiom Preference Aggregation

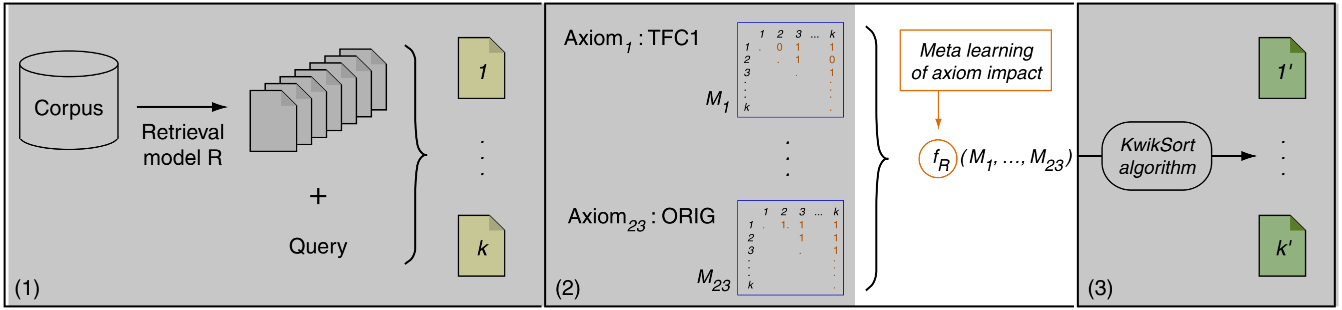


For a set of m axioms $\{A_1, \ldots A_m\}$, aggregate the individual preference matrices and use the aggregate preference matrix for re-ranking.

Hypothesis: Different basis retrieval models will deviate from the axiomatic constraints in different ways.

Axiomatic Result Re-ranking

Axiom Preference Aggregation



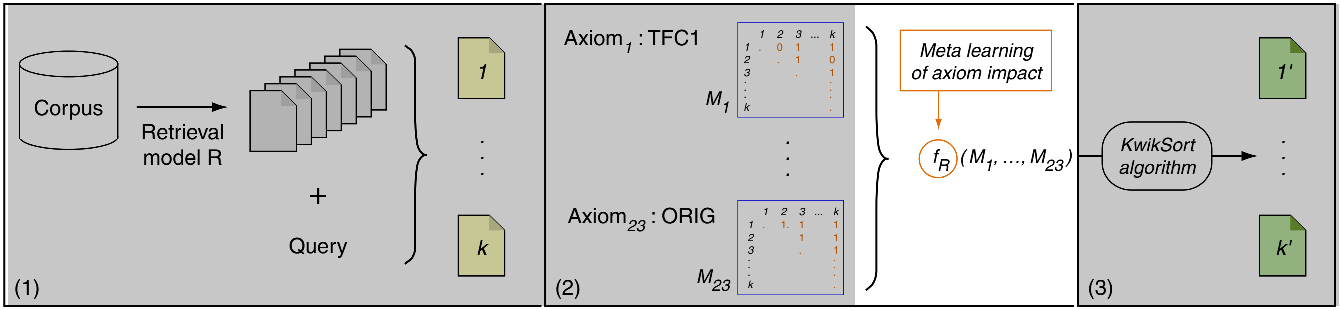
For a set of m axioms $\{A_1, \ldots A_m\}$, aggregate the individual preference matrices and use the aggregate preference matrix for re-ranking.

Hypothesis: Different basis retrieval models will deviate from the axiomatic constraints in different ways.

Approach: Given a set of queries with known relevance judgments, learn a retrieval-model-specific aggregation function that optimizes the average retrieval performance of the re-ranking.

Axiomatic Result Re-ranking

Axiom Preference Aggregation



Frame as Classification Problem:

- Individual axiom preferences as predictors

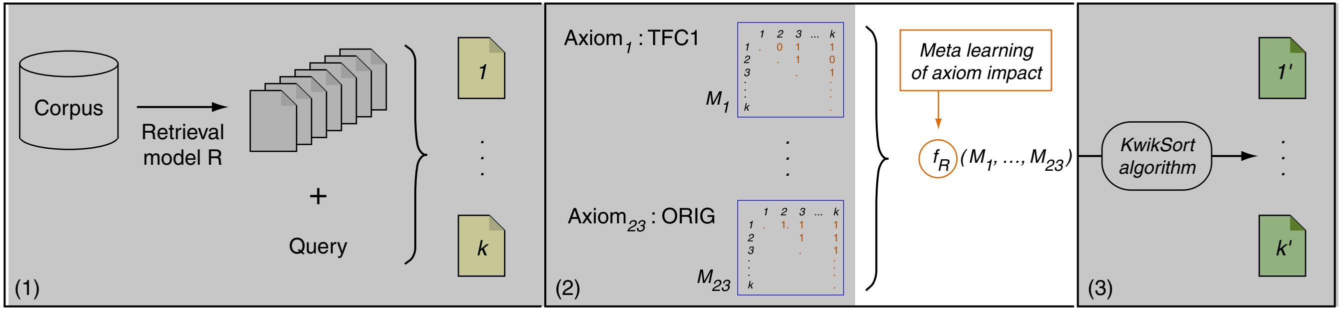
	1	2	3	...	k
1	.	0	1		1
2	0	.	1		0
3			.		1
⋮					⋮
k					.

	1	2	3	...	k
1	.	1	1		0
2	0	.	0		0
3			.		1
⋮					⋮
k					.

	1	2	3	...	k
1	.	1	1		1
2		.	1		1
3			.		1
⋮					⋮
k					.

Axiomatic Result Re-ranking

Axiom Preference Aggregation



Frame as Classification Problem:

- Individual axiom preferences as predictors
- Relative document relevance as response

	1	2	3	...	k
1	.	0	1		1
2	0	.	1		0
3			.		1
⋮					⋮
k					.

	1	2	3	...	k
1	.	1	1		0
2	0	.	0		0
3			.		1
⋮					⋮
k					.

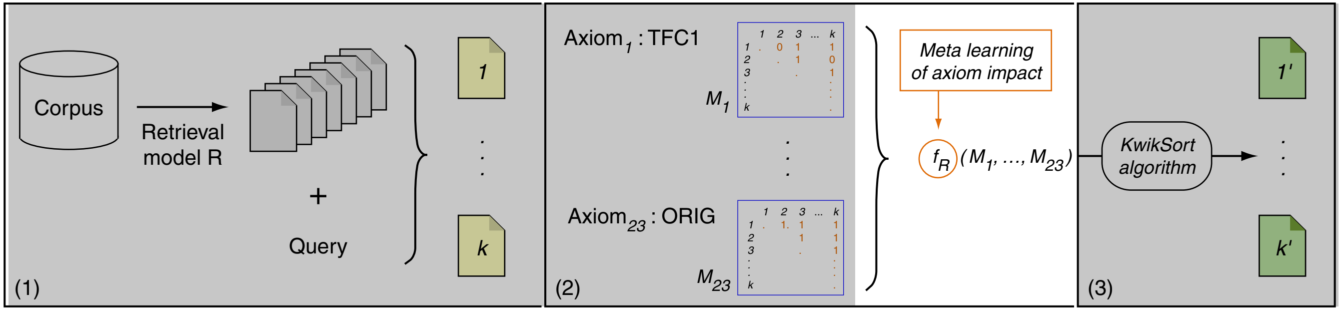
	1	2	3	...	k
1	.	1	1		1
2		.	1		1
3			.		1
⋮					⋮
k					.

→

	1	2	3	...	k
1	.	0	1		1
2	1	.	1		1
3			.		0
⋮					⋮
k					.

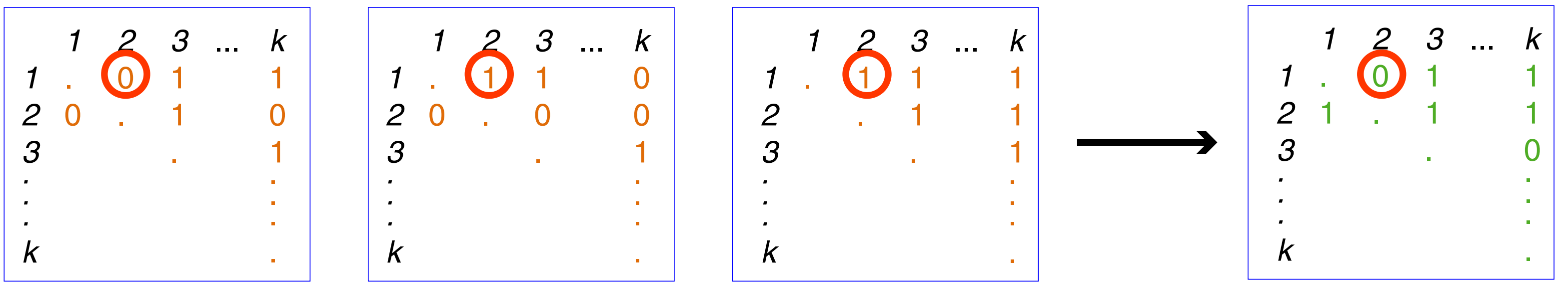
Axiomatic Result Re-ranking

Axiom Preference Aggregation



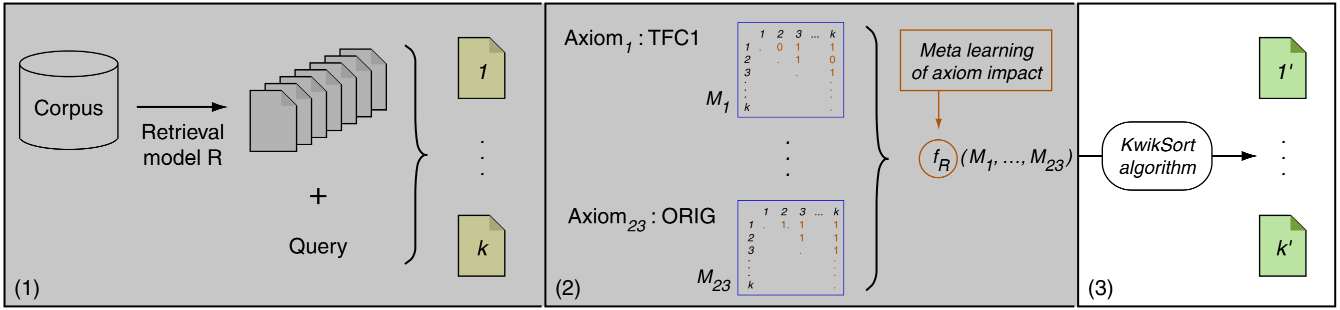
Frame as Classification Problem:

- Individual axiom preferences as predictors
- Relative document relevance as response
- One training example per document pair



Axiomatic Result Re-ranking

Re-ranking with Aggregated Preferences



- The aggregated preference matrix may contain contradictions
- E.g. $M[i, j] = M[j, i]$
- Need to solve rank-aggregation problem at this step
- We use a Kemeny rank-aggregation scheme [Kemeny; 1959]
- Solve using the KwikSort approximation algorithm [Ailon, Charikar, Newman; 2008]

Experimental Evaluation

Experimental Evaluation

Impact of Axiomatic Reranking on Web Track Performance

- Consider 16 basis retrieval models implemented in the Terrier[1] framework
- Index the ClueWeb09 corpus using each basis retrieval model
- Retrieve top 50 results and re-rank
- 120 queries from TREC Web tracks 2009–2014 as training set
- 60 queries as test set
- Measure difference in nNDCG@10 using
  - Axiomatic reranking (AX)
  - Markov Random Field term dependency (MRF)
  - Both (MRF+AX)

[1] http://terrier.org

Experimental Evaluation

Average nDCG@10 on Test Set (n=60) Using Top-50 Results

Model	Basis
DPH	0.273
DFRee	0.205
In_expC2	0.205
TF_IDF	0.202
In_expB2	0.201
DFReeKLIM	0.199
BM25	0.198
InL2	0.197
BB2	0.195
DFR_BM25	0.194
LemurTF_IDF	0.187
DLH13	0.164
PL2	0.160
DLH	0.153
DirichletLM	0.139
Hiemstra_LM	0.107

Experimental Evaluation

Average nDCG@10 on Test Set (n=60) Using Top-50 Results

Model	Basis	AX
DPH	0.273	**0.291**
DFRee	0.205	**0.236**
In_expC2	0.205	**0.214**
TF_IDF	0.202	**0.228**
In_expB2	0.201	**0.202**
DFReeKLIM	0.199	**0.213**
BM25	0.198	0.188
InL2	0.197	0.197
BB2	0.195	**0.197**
DFR_BM25	0.194	**0.206**
LemurTF_IDF	0.187	**0.224**
DLH13	0.164	**0.187**
PL2	0.160	**0.213**
DLH	0.153	**0.187**
DirichletLM	0.139	**0.242**
Hiemstra_LM	0.107	**0.167**
(# higher)		14

Experimental Evaluation

Average nDCG@10 on Test Set (n=60) Using Top-50 Results

Model	Basis	AX	MRF
DPH	0.273	**0.291**	**0.307**
DFRee	0.205	**0.236**	0.230
In_expC2	0.205	**0.214**	**0.229**
TF_IDF	0.202	**0.228**	**0.239**
In_expB2	0.201	**0.202**	**0.234**
DFReeKLIM	0.199	**0.213**	**0.224**
BM25	0.198	0.188	**0.229**
InL2	0.197	0.197	**0.235**
BB2	0.195	**0.197**	**0.236**
DFR_BM25	0.194	**0.206**	**0.236**
LemurTF_IDF	0.187	**0.224**	0.221
DLH13	0.164	**0.187**	0.184
PL2	0.160	**0.213**	0.190
DLH	0.153	**0.187**	0.181
DirichletLM	0.139	**0.242**	0.192
Hiemstra_LM	0.107	**0.167**	0.161
(# higher)		14	9 (16)

Experimental Evaluation

Average nDCG@10 on Test Set (n=60) Using Top-50 Results

Model	Basis	AX	MRF	MRF+AX
DPH	0.273	**0.291**	**0.307**	**0.314**
DFRee	0.205	**0.236**	0.230	**0.245**
In_expC2	0.205	**0.214**	**0.229**	**0.238**
TF_IDF	0.202	**0.228**	**0.239**	0.200
In_expB2	0.201	**0.202**	**0.234**	**0.237**
DFReeKLIM	0.199	**0.213**	**0.224**	0.224
BM25	0.198	0.188	**0.229**	0.216
InL2	0.197	0.197	**0.235**	0.212
BB2	0.195	**0.197**	**0.236**	0.234
DFR_BM25	0.194	**0.206**	**0.236**	0.220
LemurTF_IDF	0.187	**0.224**	0.221	**0.237**
DLH13	0.164	**0.187**	0.184	**0.201**
PL2	0.160	**0.213**	0.190	**0.211**
DLH	0.153	**0.187**	0.181	**0.197**
DirichletLM	0.139	**0.242**	0.192	**0.253**
Hiemstra_LM	0.107	**0.167**	0.161	**0.163**
(# higher)		14	9 (16)	10 (15)

Experimental Evaluation

Average nDCG@10 on Test Set (n=60) Using Top-50 Results

Model	Basis	AX	MRF	MRF+AX	max
DPH	0.273	**0.291**	**0.307**	**0.314**	0.642
DFRee	0.205	**0.236**	0.230	**0.245**	0.599
In_expC2	0.205	**0.214**	**0.229**	**0.238**	0.591
TF_IDF	0.202	**0.228**	**0.239**	0.200	0.589
In_expB2	0.201	**0.202**	**0.234**	**0.237**	0.592
DFReeKLIM	0.199	**0.213**	**0.224**	0.224	0.591
BM25	0.198	0.188	**0.229**	0.216	0.587
InL2	0.197	0.197	**0.235**	0.212	0.593
BB2	0.195	**0.197**	**0.236**	0.234	0.587
DFR_BM25	0.194	**0.206**	**0.236**	0.220	0.591
LemurTF_IDF	0.187	**0.224**	0.221	**0.237**	0.576
DLH13	0.164	**0.187**	0.184	**0.201**	0.499
PL2	0.160	**0.213**	0.190	**0.211**	0.550
DLH	0.153	**0.187**	0.181	**0.197**	0.470
DirichletLM	0.139	**0.242**	0.192	**0.253**	0.564
Hiemstra_LM	0.107	**0.167**	0.161	**0.163**	0.397
(# higher)		14	9 (16)	10 (15)	16

Conclusions

Conclusions

Summary

- Axiom-based framework for re-ranking
- Incorporating axiomatic ideas into the retrieval process directly
- New axioms for modeling term proximity preferences

Conclusions

Summary

- Axiom-based framework for re-ranking
- Incorporating axiomatic ideas into the retrieval process directly
- New axioms for modeling term proximity preferences

Future Work

- Formulate further axiomatic ideas in a way suitable for re-ranking (e.g. for near-duplicates)
- Comparison with learning-to-rank algorithms
- Improve efficiency of the re-ranking process

Conclusions

Summary

- Axiom-based framework for re-ranking
- Incorporating axiomatic ideas into the retrieval process directly
- New axioms for modeling term proximity preferences

Future Work

- Formulate further axiomatic ideas in a way suitable for re-ranking (e.g. for near-duplicates)
- Comparison with learning-to-rank algorithms
- Improve efficiency of the re-ranking process

Thank you!