Axiomatic Information Retrieval Experimentation

Grenoble, November 20, 2025

Jan Heinrich Merker, Alexander Bondarenko, Maik Fröbe, Matthias Hagen,
Benno Stein, Michael Völske, Martin Potthast

Friedrich-Schiller-Universität Jena, Leipzig University,
Bauhaus-Universität Weimar, University of Kassel

https://webis.de

Axiomatic Information Retrieval Experimentation

- Axiomatic Constraints for Retrieval Models
  - Levels of Evaluation
  - Properties of Retrieval Models
  - Axiomatic Framework for IR
  - Practical Considerations

- Applications for Retrieval Axioms

- Hands-on: Axiomatic Experiments with `ir_axioms`

- Open Topics and Discussion

Axiomatic Constraints for Retrieval Models

Analogy: Levels of Software Testing

Goal: Check if software works as expected

- System tests
  - End-to-end user interaction, complete system
  - → Macroscopic
- Integration tests: part of system, no user, mesoscopic
- Unit tests
  - Lightweight, single component
  - → Microscopic

Axiomatic Constraints for Retrieval Models

Analogy: Levels of Software Testing

Goal: Check if software works as expected

- System tests
  - – End-to-end user interaction, complete system
  - → Macroscopic: realistic, expensive, broad
- Integration tests: part of system, no user, mesoscopic
- Unit tests
  - – Lightweight, single component
  - → Microscopic: cheap, limited scope

Choose the right tool for the job!

Axiomatic Constraints for Retrieval Models

Analogy: Levels of Retrieval Evaluation

Goal: Check if retrieval system works as expected

- Online Evaluation
  - Example: A/B testing
  - → Macroscopic
- Offline Evaluation
  - Example: nDCG
  - → Mesoscopic
- Unit tests?

Axiomatic Constraints for Retrieval Models

Analogy: Levels of Retrieval Evaluation

Goal: Check if retrieval system works as expected

- Online Evaluation
  - Example: A/B testing
  - → Macroscopic: realistic, very expensive, broad
- Offline Evaluation
  - Example: nDCG
  - → Mesoscopic: still expensive, broad
- Unit tests?

Axiomatic Constraints for Retrieval Models

Analogy: Levels of Retrieval Evaluation

Goal: Check if retrieval system works as expected

- Online Evaluation
  - Example: A/B testing
  - → Macroscopic: realistic, very expensive, broad
- Offline Evaluation
  - Example: nDCG
  - → Mesoscopic: still expensive, broad
- **Axiomatic constraints**
  - Example: TFC1
  - → Microscopic

Axiomatic Constraints for Retrieval Models

Analogy: Levels of Retrieval Evaluation

Goal: Check if retrieval system works as expected

- Online Evaluation
  - Example: A/B testing
  - → Macroscopic: realistic, very expensive, broad
- Offline Evaluation
  - Example: nDCG
  - → Mesoscopic: still expensive, broad
- **Axiomatic constraints**
  - Example: TFC1
  - → Microscopic: cheap, limited scope, explainable

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations
- But: Small changes can make them ineffective ➔ Why?

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations
- But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such “desirable” properties

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations

- But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such “desirable” properties

Example: Okapi BM25 [Robertson 1994]

$$\rho_{\text{BM25}}(q,d) = \sum_{t \in q} \mathrm{IDF}(t) \cdot \frac{\mathrm{TF}(t,d) \cdot (k_1 + 1)}{\mathrm{TF}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)}$$

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations
- But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such “desirable” properties:

- TF weighting

Example: Okapi BM25 [Robertson 1994]

$$\rho_{\text{BM25}}(q, d) = \sum_{t \in q} \mathrm{IDF}(t) \cdot \frac{\boxed{\mathrm{TF}(t, d) \cdot (k_1 + 1)}}{\boxed{\mathrm{TF}(t, d)} + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)}$$

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations

- But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such "desirable" properties:
  - TF weighting
  - IDF weighting

Example: Okapi BM25 [Robertson 1994]

$$\rho_{\text{BM25}}(q,d) = \sum_{t \in q} \mathrm{IDF}(t) \cdot \frac{\mathrm{TF}(t,d) \cdot (k_1 + 1)}{\mathrm{TF}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)}$$

Axiomatic Constraints for Retrieval Models

Observations

- ❑ Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations
- ❑ But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such “desirable” properties:
- TF weighting
- IDF weighting
- Length normalization

Example: Okapi BM25 [Robertson 1994]

$$\rho_{\text{BM25}}(q,d) = \sum_{t \in q} \mathrm{IDF}(t) \cdot \frac{\mathrm{TF}(t,d) \cdot (k_1 + 1)}{\mathrm{TF}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)}$$

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations

- But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such "desirable" properties:
  - TF weighting
  - IDF weighting
  - Length normalization

Example: Query Likelihood [Ponte and Croft, 1998]

$$\rho_{\text{QL}}(q,d) := \; p(q|d) = \prod_{t \in q} p(t|d) \;\cdot\; \prod_{t \notin q} (1 - p(t|d)) \qquad \text{with:}$$

$$p(t|d) = \left(\frac{\text{TF}(t,d)}{|d|}\right)^{1 - R_{t,d}} \cdot \left(\frac{\sum_{d_{(t \in q)}} \text{TF}(t,d) / |d|}{\text{DF}(t)}\right)^{R_{t,d}}$$

Axiomatic Constraints for Retrieval Models

Observations

- Baseline retrieval models (e.g., BM25, Query Likelihood, ...) similarly effective despite different formulations

- But: Small changes can make them ineffective → similar properties modeled

→ Axiomatic IR: Identify and formalize such "desirable" properties:

- TF weighting
- IDF weighting
- Length normalization
- Term "risk" normalization

Example: Query Likelihood [Ponte and Croft, 1998]

$$\rho_{\mathsf{QL}}(q,d) := p(q|d) = \prod_{t \in q} p(t|d) \cdot \prod_{t \notin q} (1 - p(t|d)) \qquad \text{with:}$$

$$p(t|d) = \left(\frac{\mathrm{TF}(t,d)}{|d|}\right)^{1 - R_{t,d}} \cdot \left(\frac{\sum_{d_{(t \in q)}} \mathrm{TF}(t,d) / |d|}{\mathrm{DF}(t)}\right)^{R_{t,d}}$$

Axiomatic Constraints for Retrieval Models

Axiom Examples: TFC1 [Fang, Tao, Zhai 2004]

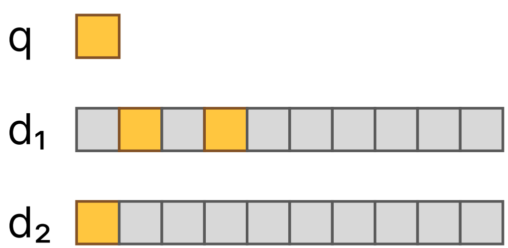
Property: Term frequency

Intuition: Higher score to document with more occurrences of query term.

Formalization: Given a single-term query $q = \{t\}$ and two documents d_1, d_2 with $|d_1| = |d_2|$.

If $\mathrm{TF}(t, d_1) > \mathrm{TF}(t, d_2)$ then $\rho(q, d_1) > \rho(q, d_2)$

Visualization:



Axiomatic Constraints for Retrieval Models

Axiom Examples: TFC1 (practical)

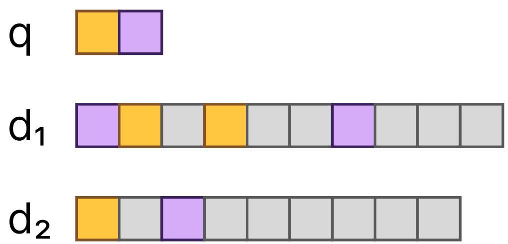
Property: Term frequency

Intuition: Prefer documents with more occurrences of the query terms.

Formalization: Given a multi-term query $q = \{t_1, ..., t_n\}$ and two documents d_1, d_2 with $|d_1| \approx |d_2|$.

If $\sum_{t \in q} \mathrm{TF}(t, d_1) > \sum_{t \in q} \mathrm{TF}(t, d_2)$ then $\rho(q, d_1) > \rho(q, d_2)$

Visualization:



Axiomatic Constraints for Retrieval Models

Axiom Examples: TFC1 (practical)

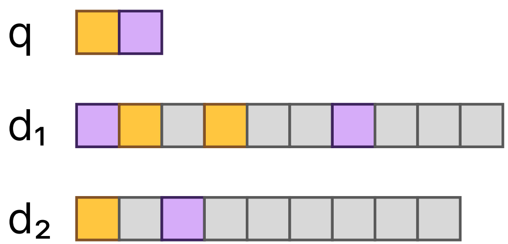
Property: Term frequency

Intuition: Prefer documents with more occurrences of the query terms.

Formalization: Given a multi-term query $q = \{t_1, ..., t_n\}$ and two documents d_1, d_2 with $|d_1| \approx |d_2|$.

If $\sum_{t \in q} \mathrm{TF}(t, d_1) > \sum_{t \in q} \mathrm{TF}(t, d_2)$ then $d_1 >_{\text{TFC1}} d_2$

Visualization:



Axiomatic Constraints for Retrieval Models

Axiom Definitions

Property: <a “desirable” property>

Intuition: Prefer <documents> with <. . . >

Formalization: Given a <query> q and
two <documents> d_1, d_2 with <precondition>.

If <rule> then $d_1 >_A d_2$

Axiomatic Constraints for Retrieval Models

Axiom Definitions

Property: <a “desirable” property>

Intuition: Prefer <outputs> with <...>

Formalization: Given an <input> q and
two <outputs> d_1, d_2 with <precondition>.

If <rule> then $d_1 >_A d_2$

Axiomatic Constraints for Retrieval Models

Axiom Definitions

Property: <a “desirable” property>

Intuition: Prefer <outputs> with <. . . >

Formalization: Given an <input> q and two <outputs> d_1, d_2 with <precondition>.

If <rule> then $d_1 >_A d_2$

Remarks

- Simple if-then rules
  - → Easy to explain axiom preferences
- Formal mathematical definition
  - → Prove ranking model “errors” formally
  - → Apply simple algorithms

Axiomatic Constraints for Retrieval Models

Axioms

Term Frequency Constraints [Fang, Tao, Zhai 2004; 2011]

TFC1	Prefer documents with more query term occurrences.
TFC2	Additional query term occurrences yield smaller score improvements.
TFC3	Prefer documents with occurrences of more distinct query terms.
TDC	Prefer documents with more discriminative query terms.

Length Normalization Constraints [Fang, Tao, Zhai 2004]

LNC1	Penalize longer documents for non-relevant terms.
LNC2	Avoid over-penalizing long documents.
TF-LNC	Reward additional query terms more than document length is penalized.

Lower-bounding Term Frequency Constraints [Lv and Zhai 2011]

LB1	Do not override the term presence–absence gap with length normalization.
LB2	Repeated query term occurrence is less important than first occurrence.

Query Aspect-based Constr. [Gollapurdi and Sharma 2009; Zheng and Fang 2010; Wu and Fang 2012]

REG	Prefer documents covering more different query aspects.
AND	Prefer documents containing all query terms.
DIV	Prefer documents with larger vocabulary overlap with the query.

Axiomatic Constraints for Retrieval Models

Axioms

Query Aspect-based Constr. [Gollapurdi and Sharma 2009; Zheng and Fang 2010; Wu and Fang 2012]

REG	Prefer documents covering more different query aspects.
AND	Prefer documents containing all query terms.
DIV	Prefer documents with larger vocabulary overlap with the query.

Semantic Similarity Constraints [Fang and Zhai 2006]

STMC1	Prefer documents with terms more similar to query terms.
STMC2	Do not reward similar terms more than exact matches.
STMC3	Prefer documents with more distinct query terms

Term Proximity Constraints [Tao and Zhai 2007; Hagen et al. 2016]

PHC	Prefer documents with query terms closer together.
CCC	Make the proximity-based score increase convex.
PROX1	Prefer documents with shorter distance between query term pairs.
PROX2	Prefer documents with earlier query term occurrences.
PROX3	Prefer documents where the query occurs earlier as a phrase.
PROX4	Prefer documents that contain all query terms in a shorter substring.
PROX5	Prefer documents where the query terms are closer together on average.

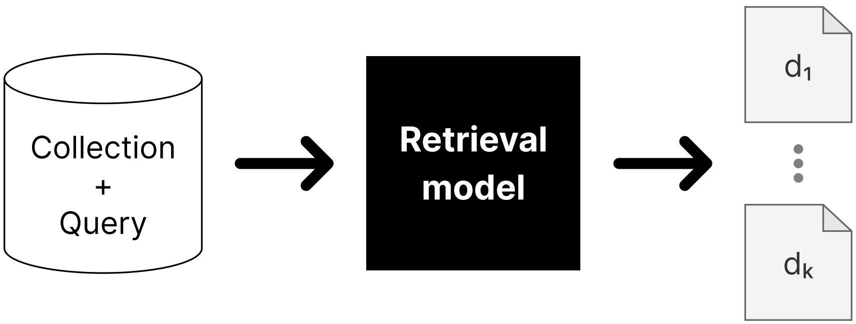
. . . and many more . . .

Axiomatic Information Retrieval Experimentation

- Axiomatic Constraints for Retrieval Models
- Applications for Retrieval Axioms
  - Overview
  - Explain Ranking Decisions
  - Axiomatic Re-Ranking
  - Axioms for RAG
- Hands-on: Axiomatic Experiments with `ir_axioms`
- Open Topics and Discussion

Applications for Retrieval Axioms

Overview



- Analyze and explain neural rankers
  [Rennings et al. 2019; Camâra, Hauff 2020; Völske et al. 2021; Formal et al. 2021; MacAvaney et al. 2022]

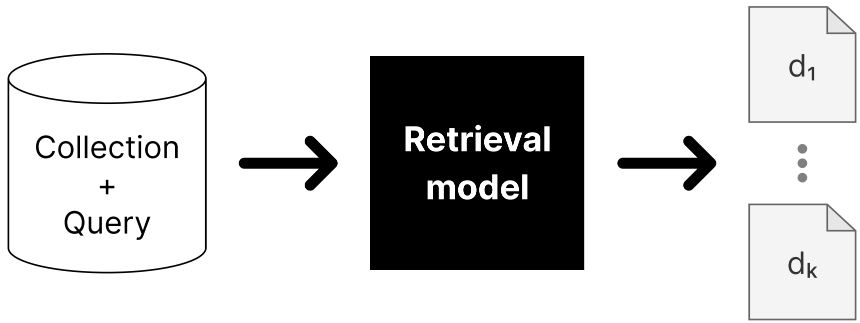
- Improve effectiveness by re-ranking [Hagen et al. 2016]

- Improve (neural) model training [Rosset et al. 2019; Arora and Yates 2019]

- New: Analyze and explain RAG [Merker et al. 2025]

Applications for Retrieval Axioms

Overview



- Analyze and explain neural rankers

  [Rennings et al. 2019; Camâra, Hauff 2020; Völske et al. 2021; Formal et al. 2021; MacAvaney et al. 2022]

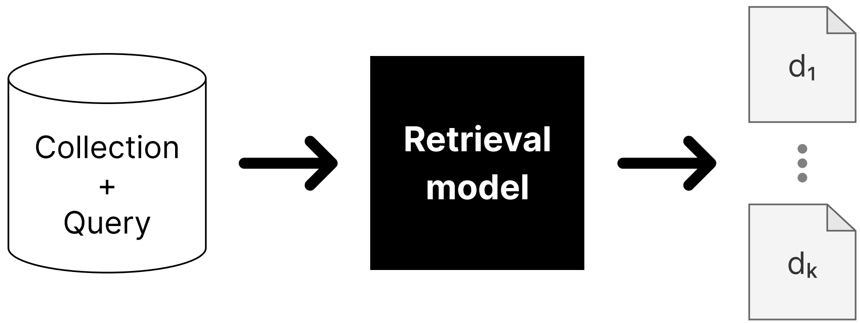
- Improve effectiveness by re-ranking [Hagen et al. 2016]

- Improve (neural) model training [Rosset et al. 2019; Arora and Yates 2019]

- New: Analyze and explain RAG [Merker et al. 2025]

Applications for Retrieval Axioms

Overview

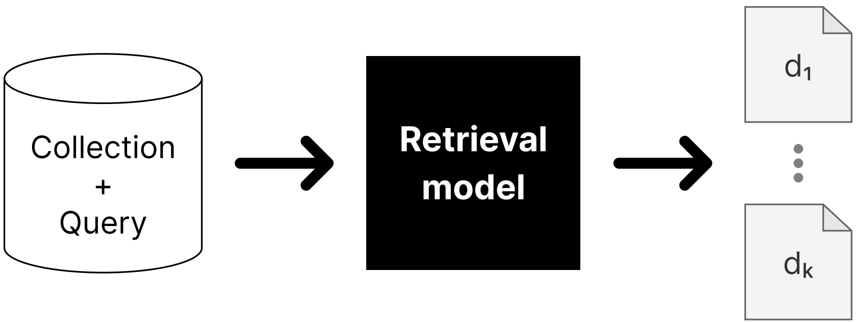


- Analyze and explain neural rankers
  [Rennings et al. 2019; Camâra, Hauff 2020; Völske et al. 2021; Formal et al. 2021; MacAvaney et al. 2022]
- Improve effectiveness by re-ranking [Hagen et al. 2016]
- Improve (neural) model training [Rosset et al. 2019; Arora and Yates 2019]
- New: Analyze and explain RAG [Merker et al. 2025]

→ How to run axiomatic experiments?

Applications for Retrieval Axioms

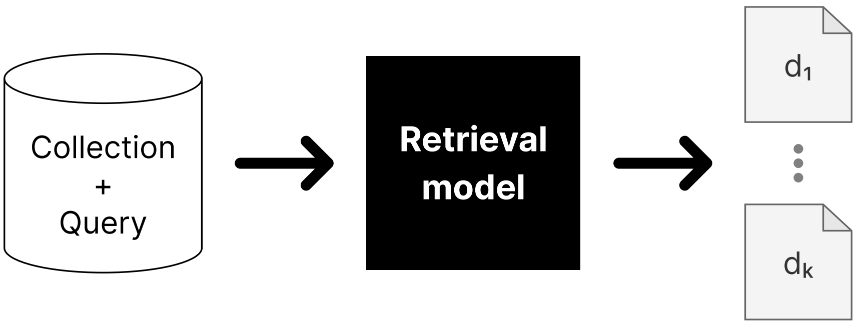
Explain Ranking Decisions



- Many retrieval models too complex to interpret
- Do complex models still "obey" basic properties?

Applications for Retrieval Axioms

Explain Ranking Decisions



- Many retrieval models too complex to interpret
- Do complex models still “obey” basic properties?

Axiomatic Relevance Hypothesis [Zhai and Fang 2013]

- Relevance modeled by constraints on retrieval function (i.e., axioms)
- System A satisfies many axioms → good effectiveness
- System A satisfies more than system B → system A better than B

Approach: Compare ranking preferences against axioms

Applications for Retrieval Axioms

Explain Ranking Decisions: Empirical Model

Prerequisite: Original ranking preferences [Hagen et al. 2016]

$$d_1 >_{\textsf{ORIG}} d_2 \quad\Leftrightarrow\quad \rho(q, d_1) > \rho(q, d_2)$$
$$d_1 <_{\textsf{ORIG}} d_2 \quad\Leftrightarrow\quad \rho(q, d_1) < \rho(q, d_2)$$

Applications for Retrieval Axioms

Explain Ranking Decisions: Empirical Model

Prerequisite: Original ranking preferences [Hagen et al. 2016]

$$
\begin{aligned}
d_1 >_{\textsf{ORIG}} d_2 &\;\Leftrightarrow\; \rho(q, d_1) > \rho(q, d_2) \\
d_1 <_{\textsf{ORIG}} d_2 &\;\Leftrightarrow\; \rho(q, d_1) < \rho(q, d_2)
\end{aligned}
$$

Prerequisite: Preference function for axiom A

$$
\begin{aligned}
\mathrm{pref}_A(q, d_1, d_2) &= 1 \;\Leftrightarrow\; d_1 >_A d_2 \\
\mathrm{pref}_A(q, d_1, d_2) &= -1 \;\Leftrightarrow\; d_1 <_A d_2 \\
\mathrm{pref}_A(q, d_1, d_2) &= 0 \;\Leftrightarrow\; d_1 \not>_A d_2 \;\wedge\; d_1 \not<_A d_1
\end{aligned}
$$

Applications for Retrieval Axioms

Explain Ranking Decisions: Empirical Model

Prerequisite: Original ranking preferences [Hagen et al. 2016]

$$d_1 >_{\text{ORIG}} d_2 \quad \Leftrightarrow \quad \rho(q, d_1) > \rho(q, d_2)$$
$$d_1 <_{\text{ORIG}} d_2 \quad \Leftrightarrow \quad \rho(q, d_1) < \rho(q, d_2)$$

Prerequisite: Preference function for axiom A

$$\begin{aligned} \text{pref}_A(q, d_1, d_2) &= 1 \quad \Leftrightarrow \quad d_1 >_A d_2 \\ \text{pref}_A(q, d_1, d_2) &= -1 \quad \Leftrightarrow \quad d_1 <_A d_2 \\ \text{pref}_A(q, d_1, d_2) &= 0 \quad \Leftrightarrow \quad d_1 \not>_A d_2 \,\wedge\, d_1 \not<_A d_1 \end{aligned}$$

Prerequisite: Preference matrix M_A for axiom A
(applied to query q and ranking $D = [d_1, \ldots, d_n]$)

$$M_A(q, D)[i, j] = M_A[i, j] = \text{pref}_A(q, d_i, d_j)$$

Example:

$$M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad M_A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Explain Ranking Decisions: Empirical Model

Prerequisite: Original ranking preferences [Hagen et al. 2016]

$$d_1 >_{\text{ORIG}} d_2 \quad \Leftrightarrow \quad \rho(q, d_1) > \rho(q, d_2)$$
$$d_1 <_{\text{ORIG}} d_2 \quad \Leftrightarrow \quad \rho(q, d_1) < \rho(q, d_2)$$

Prerequisite: Preference function for axiom A

$$\text{pref}_A(q, d_1, d_2) = 1 \quad \Leftrightarrow \quad d_1 >_A d_2$$
$$\text{pref}_A(q, d_1, d_2) = -1 \quad \Leftrightarrow \quad d_1 <_A d_2$$
$$\text{pref}_A(q, d_1, d_2) = 0 \quad \Leftrightarrow \quad d_1 \not>_A d_2 \ \wedge \ d_1 \not<_A d_1$$

Prerequisite: Preference matrix M_A for axiom A
(applied to query q and ranking $D = [d_1, \ldots, d_n]$)

$$M_A(q, D)[i, j] = M_A[i, j] = \text{pref}_A(q, d_i, d_j)$$

Example:

$$M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad M_A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Explain Ranking Decisions: Empirical Model

Prerequisite: Original ranking preferences [Hagen et al. 2016]

$$d_1 >_{\text{ORIG}} d_2 \;\Leftrightarrow\; \rho(q, d_1) > \rho(q, d_2)$$
$$d_1 <_{\text{ORIG}} d_2 \;\Leftrightarrow\; \rho(q, d_1) < \rho(q, d_2)$$

Prerequisite: Preference function for axiom A

$$\begin{aligned}
\mathrm{pref}_A(q, d_1, d_2) &= 1 \;\Leftrightarrow\; d_1 >_A d_2 \\
\mathrm{pref}_A(q, d_1, d_2) &= -1 \;\Leftrightarrow\; d_1 <_A d_2 \\
\mathrm{pref}_A(q, d_1, d_2) &= 0 \;\Leftrightarrow\; d_1 \not>_A d_2 \;\wedge\; d_1 \not<_A d_1
\end{aligned}$$

Prerequisite: Preference matrix M_A for axiom A
(applied to query q and ranking $D = [d_1, \ldots, d_n]$)

$$M_A(q, D)[i, j] = M_A[i, j] = \mathrm{pref}_A(q, d_i, d_j)$$

Example:

$$M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad M_A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Explain Ranking Decisions

How often does a model satisfy the axiomatic constraints?

$$\text{Consistency}_A(q, D) = \frac{\sum_{i,j;j>i} M_{\text{ORIG}}[i,j] = M_A[i,j]}{(n^2 - n)/2}$$

Applications for Retrieval Axioms

Explain Ranking Decisions

Example:

$$\text{Consistency}_A(q, D) = \frac{\sum_{i,j;j>i} M_{\text{ORIG}}[i,j] = M_A[i,j]}{(n^2 - n)/2}$$

$$M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad M_A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Explain Ranking Decisions

Example:

$$\text{Consistency}_A(q, D) = \frac{\sum_{i,j;j>i} M_{\text{ORIG}}[i,j] = M_A[i,j]}{(n^2 - n)/2}$$

$$M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad M_A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Explain Ranking Decisions

Example:

$$\text{Consistency}_A(q, D) = \frac{2}{3} = 67\%$$

$$M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad M_A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Explain Ranking Decisions: TREC Deep Learning Track 2019

Step 1: Compare axiom consistency

Axiom	Documents			Passages		
	LLM	Neural	Trad.	LLM	Neural	Trad.
TFC1	48%	54%	66%	60%	56%	56%
STMC1	54%	52%	55%	57%	53%	55%
PROX1	67%	60%	59%	61%	57%	58%

Applications for Retrieval Axioms

Explain Ranking Decisions: TREC Deep Learning Track 2019

Step 1: Compare axiom consistency

Axiom	**Documents**			**Passages**		
	LLM	Neural	Trad.	LLM	Neural	Trad.
TFC1	48%	54%	66%	60%	56%	56%
STMC1	54%	52%	55%	57%	53%	55%
PROX1	67%	60%	59%	61%	57%	58%

Step 2: Debug individual violations (most effective run at TREC 2019 DL passage retrieval)

Query	`how are some sharks warm blooded`		**Axioms**		
Rank	**Rel.**	**Content**	TFC1	STMC1	PROX1
3	1	Great white sharks are some of the only warm blooded sharks. This allows them to swim in colder waters in addition to warm, tropical waters. Great White sharks [...] exist worldwide [...].			
5	2	These sharks can raise their temperature about the temperature of the water; they need to have occasional short bursts of speed in hunting Cold blooded [...]. Actually the Salmon Shark is a warm blooded shark.			

Applications for Retrieval Axioms

Explain Ranking Decisions: TREC Deep Learning Track 2019

Step 1: Compare axiom consistency

Axiom	Documents			Passages		
	LLM	Neural	Trad.	LLM	Neural	Trad.
TFC1	48%	54%	66%	60%	56%	56%
STMC1	54%	52%	55%	57%	53%	55%
PROX1	67%	60%	59%	61%	57%	58%

Step 2: Debug individual violations (most effective run at TREC 2019 DL passage retrieval)

Query `how are some sharks warm blooded`			Axioms		
Rank	**Rel.**	**Content**	TFC1	STMC1	PROX1
3	1	Great white sharks are some of the only warm blooded sharks. This allows them to swim in colder waters in addition to warm, tropical waters. Great White sharks [...] exist worldwide [...].	👎	👎	👍
5	2	These sharks can raise their temperature about the temperature of the water; they need to have occasional short bursts of speed in hunting Cold blooded [...]. Actually the Salmon Shark is a warm blooded shark.	👍	👍	👎

Step 3: Improve retrieval model based on findings → How?

Applications for Retrieval Axioms

Axiomatic Re-Ranking: Motivation

- BM25 (no matter the parameter setting) violates the LB2 constraint
- Minor modification corrects it → better effectiveness [Lv and Zhai 2011]

$$\rho_{\text{BM25}}(q,d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t,d) \cdot (k_1 + 1)}{\text{TF}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)}$$

Applications for Retrieval Axioms

Axiomatic Re-Ranking: Motivation

- BM25 (no matter the parameter setting) violates the LB2 constraint
- Minor modification corrects it → better effectiveness [Lv and Zhai 2011]

$$\rho^{+}_{\text{BM25}}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \left(\frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)} + \delta \right)$$

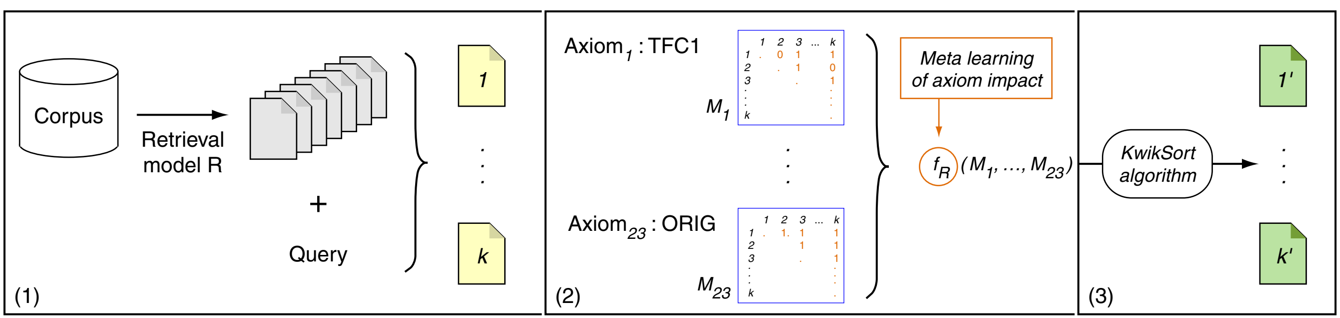
Applications for Retrieval Axioms

Axiomatic Re-Ranking: Motivation

- BM25 (no matter the parameter setting) violates the LB2 constraint
- Minor modification corrects it → better effectiveness [Lv and Zhai 2011]

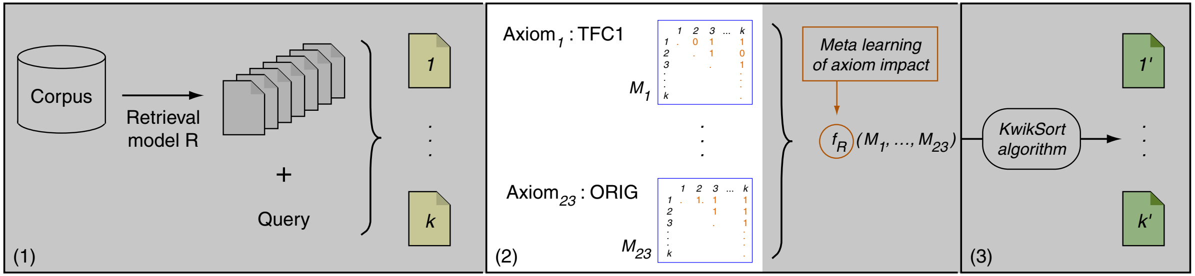
$$\rho^{+}_{\text{BM25}}(q,d) = \sum_{t \in q} \text{IDF}(t) \cdot \left(\frac{\text{TF}(t,d) \cdot (k_1 + 1)}{\text{TF}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)} + \delta \right)$$

→ Goal: Automate "axiomatization" of retrieval models with axiomatic re-ranking



Applications for Retrieval Axioms

Axiomatic Re-Ranking: Preference Matrices

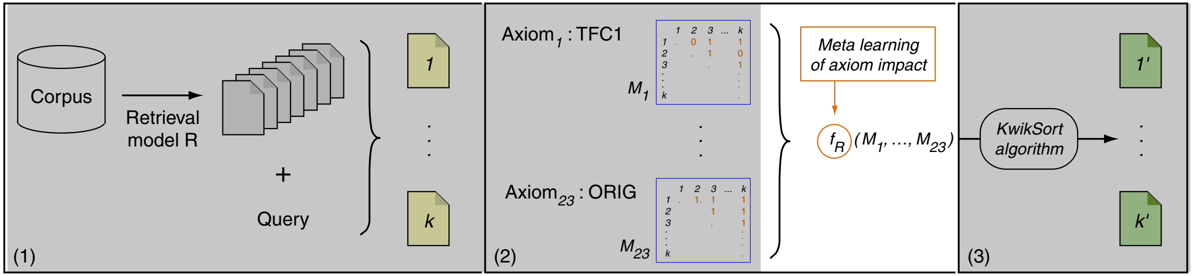


Step 1: From axioms and original ranking, compute preference matrices

$$\overset{M_{\text{TFC1}}}{\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}} \quad \overset{M_{\text{PROX1}}}{\begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}} \quad \cdots \quad \overset{M_{\text{ORIG}}}{\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}}$$

Applications for Retrieval Axioms

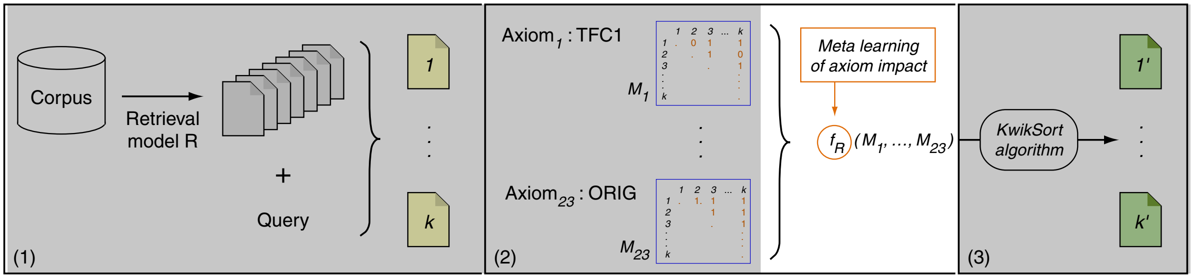
Axiomatic Re-Ranking: Preference Aggregation



Step 2: Aggregate preference matrices to use aggregate preference for re-ranking.

Applications for Retrieval Axioms

Axiomatic Re-Ranking: Preference Aggregation

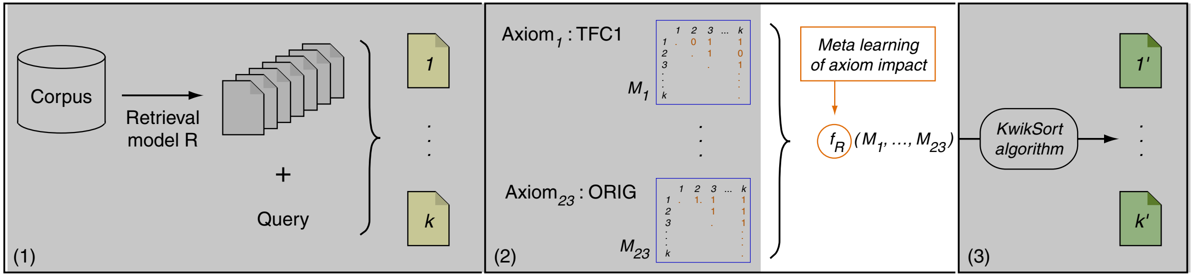


Step 2: Aggregate preference matrices to use aggregate preference for re-ranking.

Assumption: Axiomatic Relevance Hypothesis [Zhai and Fang 2013]

Applications for Retrieval Axioms

Axiomatic Re-Ranking: Preference Aggregation



Step 2: Aggregate preference matrices to use aggregate preference for re-ranking.

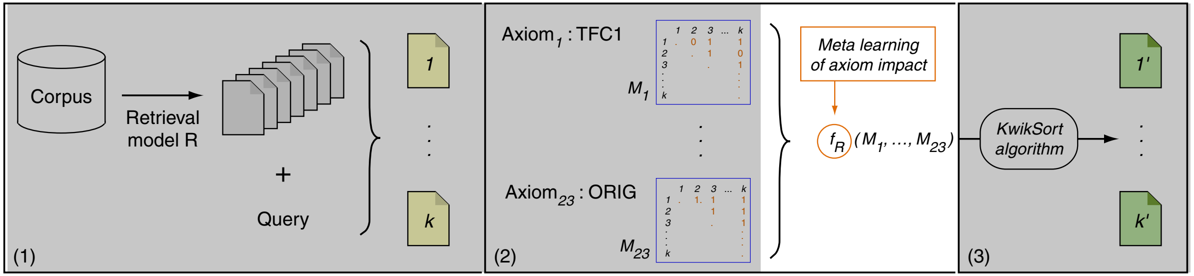
Assumption: Axiomatic Relevance Hypothesis [Zhai and Fang 2013]

Approach: Learn to estimate ground-truth preferences from different axiom's preferences.

$$\underset{M_{\text{TFC1}}}{\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}} \quad \underset{M_{\text{PROX1}}}{\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}} \quad \cdots \quad \underset{M_{\text{ORIG}}}{\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}} \xrightarrow{\text{learn}} \underset{M_{\text{ORACLE}}}{\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}}$$

Applications for Retrieval Axioms

Axiomatic Re-Ranking: Preference Aggregation



Step 2: Aggregate preference matrices to use aggregate preference for re-ranking.

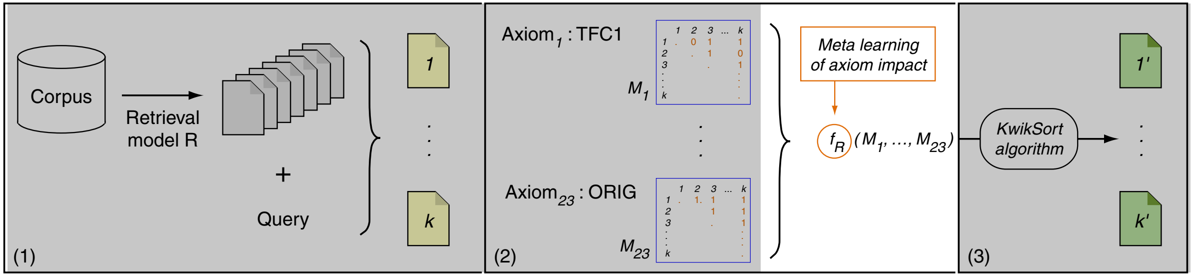
Assumption: Axiomatic Relevance Hypothesis [Zhai and Fang 2013]

Approach: Learn to estimate ground-truth preferences from different axiom's preferences.

$$M_{\text{TFC1}} = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \quad M_{\text{PROX1}} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \quad \cdots \quad M_{\text{ORIG}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \xrightarrow{\text{learn}} M_{\text{ORACLE}} = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

Applications for Retrieval Axioms

Axiomatic Re-Ranking: Preference Aggregation



Step 2: Aggregate preference matrices to use aggregate preference for re-ranking.

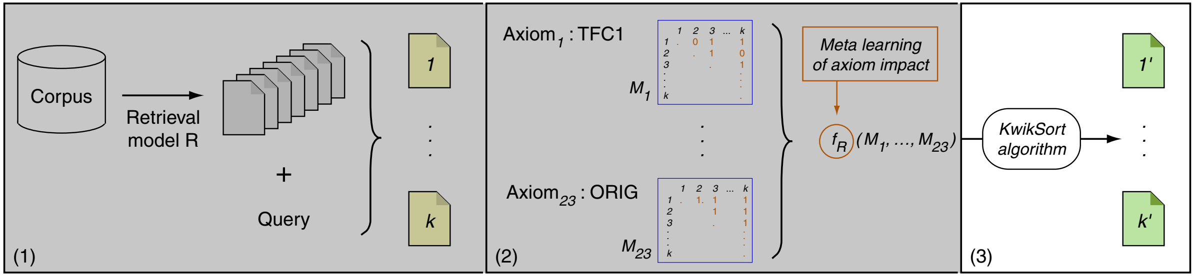
Assumption: Axiomatic Relevance Hypothesis [Zhai and Fang 2013]

Approach: Learn to estimate ground-truth preferences from different axiom's preferences.

$$\underset{M_{\text{TFC1}}}{\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}} \quad \underset{M_{\text{PROX1}}}{\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}} \quad \cdots \quad \underset{M_{\text{ORIG}}}{\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}} \xrightarrow{\text{learn}} \underset{M_{\text{ORACLE}}}{\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}}$$

Applications for Retrieval Axioms

Axiomatic Re-Ranking: KwikSort



Step 3: From aggregated preference matrix, derive final ranking.

- Aggregated preference matrix can contain contradictions, e.g. $M[i,j] = M[j,i]$
- Rank-aggregation to resolve contradictions [Kemeny 1959]
- Algorithm: KwikSort [Ailon, Charikar, Newman 2008] (works similar to QuickSort)

Applications for Retrieval Axioms

RAG Axioms: Motivation

- Problem: Utility of RAG responses not just topical relevance
- Ground-truth-based evaluation
- Ground-truth-free approaches

Applications for Retrieval Axioms

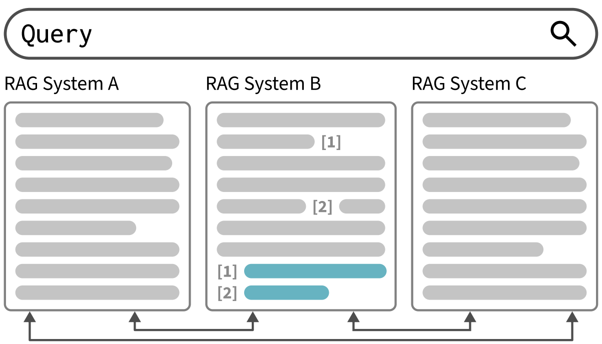
RAG Axioms: Motivation

- Problem: Utility of RAG responses not just topical relevance
- ~~Ground-truth-based evaluation~~ → unavailable/expensive
- Ground-truth-free approaches:
  - Information nuggets: SWAN, LLM-Rubric, TREC RAG
  - Question answering-based: EXAM, RUBRIC
  - Direct assessment by LLMs: RAGAs, ARES

Applications for Retrieval Axioms

RAG Axioms: Motivation

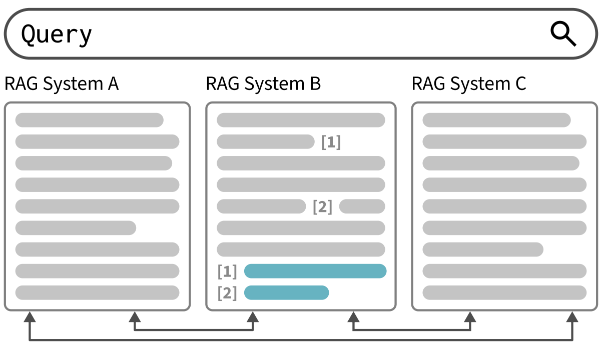
- Problem: Utility of RAG responses not just topical relevance
- ~~Ground-truth-based evaluation~~ → unavailable/expensive
- Ground-truth-free approaches:
  - Information nuggets: SWAN, LLM-Rubric, TREC RAG
  - Question answering-based: EXAM, RUBRIC
  - Direct assessment by LLMs: RAGAs, ARES



Applications for Retrieval Axioms

RAG Axioms: Motivation

- Problem: Utility of RAG responses not just topical relevance
- ~~Ground-truth-based evaluation~~ → unavailable/expensive
- Ground-truth-free approaches:
  - Information nuggets: SWAN, LLM-Rubric, TREC RAG
  - Question answering-based: EXAM, RUBRIC
  - Direct assessment by LLMs: RAGAs, ARES



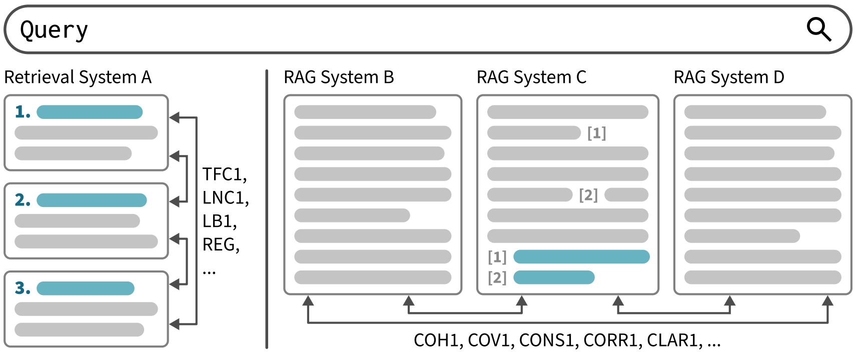
Limitations

- Scalability: Cost of manual judgments (or LLM inference)
- Explainability: Opaque, biased LLMs used in evaluation

Applications for Retrieval Axioms

RAG Axioms: Approach

- Traditional axioms: "Vertical" preferences on a single system's ranking
- RAG axioms: "Horizontal" preferences between different systems



Adapt Traditional Axioms

- Reuse from `ir_axioms`
- Limitation: index statistics for "infinite index"
- 18 of 25 traditional axioms adapted for RAG

New RAG-specific Axioms

- Utility dimensions [Gienapp et al. 2024]: Coherence, correctness, coverage, consistency, clarity
- 11 new RAG axioms
- Integrated into `ir_axioms`

Applications for Retrieval Axioms

RAG Axioms

Coherence-based Constraints

COH1	Prefer responses with less variance in avg. word length across sentences.
COH2	Prefer responses with subject–verb pairs closer together.

Coverage-based Constraints

COV1	Prefer responses with more extracted aspects in response.
COV2	Prefer responses with less redundant extracted aspects.
COV3	Prefer responses with more sentences covering aspects from the query.

Consistency-based Constraints

CONS1	Prefer responses with more sentences covering aspects from the context.
CONS2	Prefer responses with higher text overlap with contexts.
CONS3	Penalize aspects mentioned in contradictory phrases.

Correctness-based Constraints

CORR1	Prefer responses with more sentences with references to sources.

Clarity-based Constraints

CLAR1	Prefer responses with fewer grammar errors.
CLAR2	Prefer responses with better readability.

. . . **not complete** → Contribute!

Applications for Retrieval Axioms

RAG Axioms: Experiments

- TREC 2025 RAG: Information nuggets recall / coverage → LLM-based
- Webis CrowdRAG-25: Crowd-sourced judgments, 5 utility dims. → Manual

Method

- Consistency with oracle preferences
- Decisiveness (How often does the axiom yield a preference?)
- Use cases: Inspect LM generation preferences, aid annotation, etc.

Query `good morning accenture`

#	Response	TFC1	STMC1	PROX2	PROX3	PROX4	PROX5	COH1	COV1	COV2	COV3	CONS1	CONS2	CORR1	CLAR2
1	“The question can’t be answered using the references provided. Please try with shorter phrases [...] to find out relevant results.”	👍	👍	👍	👎	👍	👍	👍	👎	👎	👎	👎	👎	👍	👎
2	“The "Good Morning Accenture" initiative [...] has significantly impacted the company by repositioning and invigorating [...] consultation and technological advancement [3, 4].”	👎	👎	👎	👍	👎	👎	👎	👍	👍	👍	👍	👍	👎	👍

Axiomatic Information Retrieval Experimentation

- Axiomatic Constraints for Retrieval Models
- Applications for Retrieval Axioms
- Hands-on: Axiomatic Experiments with `ir_axioms`
  - The `ir_axioms` Framework
  - Post-hoc Axiomatic Analyses
  - Axiomatic Re-Ranking
  - Developing New Retrieval Axioms
- Open Topics and Discussion

Hands-on: Axiomatic Experiments with `ir_axioms`

Practical Axiomatic Experiments

- Many IR toolkits: Terrier, Anserini, etc.
- But: None includes components for retrieval axioms

Hands-on: Axiomatic Experiments with `ir_axioms`

Practical Axiomatic Experiments

- Many IR toolkits: Terrier, Anserini, etc.
- But: None includes components for retrieval axioms

The `ir_axioms` Framework

- Python library adds axiom components to IR toolkits
- 25 retrieval axioms and 11 RAG axioms included
  with practical relaxations, e.g., $|d_1| \approx_{10\%} |d_2|$ for TFC1
- Access to retrieval models and test collections in PyTerrier and `ir_datasets`

Hands-on: Axiomatic Experiments with `ir_axioms`

Practical Axiomatic Experiments

- Many IR toolkits: Terrier, Anserini, etc.
- But: None includes components for retrieval axioms

The `ir_axioms` Framework

- Python library adds axiom components to IR toolkits
- 25 retrieval axioms and 11 RAG axioms included
  with practical relaxations, e.g., $|d_1| \approx_{10\%} |d_2|$ for TFC1
- Access to retrieval models and test collections in PyTerrier and `ir_datasets`

Design Goals

1. Usable: Supports many axiomatic applications
2. Extensible: Easy to define new axioms
   ... by extending a Python class
3. Composable: “Remix” axioms to build more complex constraints
   ... by using Python operators

Hands-on: Axiomatic Experiments with `ir_axioms`

Showcase

Jupyter Notebook:

https://github.com/webis-de/ir_axioms/blob/main/experiments/grenoble2025_showcase.ipynb

Hands-on: Axiomatic Experiments with `ir_axioms`

Developing New (Retrieval) Axioms

Recall: Axiom Definitions

Property: <a “desirable” property>

Intuition: Prefer <documents> with <...>

Formalization: Given an <input> q and two <documents> d_1, d_2 with <precondition>.

If <rule> then $\rho(q, d_1) > \rho(q, d_2)$

Hands-on: Axiomatic Experiments with `ir_axioms`

Developing New (Retrieval) Axioms

Recall: Axiom Definitions

Property: <a “desirable” property>

Intuition: Prefer <documents> with <…>

Formalization: Given an <input> q and two <documents> d_1, d_2 with <precondition>.

If <rule> then $\rho(q, d_1) > \rho(q, d_2)$

Steps to develop an axiom:

1. Identify desirable property
2. Formalize as pairwise preference between two documents
3. Implement the axiom in `ir_axioms`
4. Run experiments

Axiomatic Information Retrieval Experimentation

Summary

- Formally analyze and explain retrieval and RAG
- More than 30 axioms implemented in `ir_axioms`
  - Post-hoc analyses
  - Axiomatic re-ranking
  - Easy to define new axioms
- Axioms support not replace typical evaluation

Software and examples → Contributions are welcome!:

Code

`webis-de/ir_axioms`

`pip install ir_axioms>=1.0`

Future Work: More axioms for RAG, other domains/modalities, regularize LLM's, ...

Axiomatic Information Retrieval Experimentation

Summary

- Formally analyze and explain retrieval and RAG
- More than 30 axioms implemented in `ir_axioms`
  - Post-hoc analyses
  - Axiomatic re-ranking
  - Easy to define new axioms
- Axioms support not replace typical evaluation

Software and examples → Contributions are welcome!:

Code

`webis-de/ir_axioms`

`pip install ir_axioms>=1.0`

Future Work: More axioms for RAG, other domains/modalities, regularize LLM's, ...

Thank you!