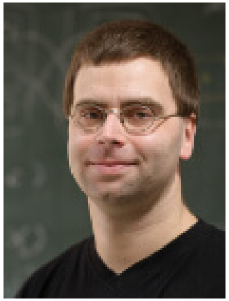
Axiomatic Retrieval Experimentation with **`ir_axioms`**

July 11–15, 2022

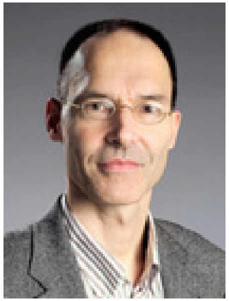
J. Heinrich
Reimer

Alexander
Bondarenko

Maik
Fröbe



Matthias
Hagen



Benno
Stein

Michael
Völske

Martin-Luther-Universität Halle-Wittenberg

Bauhaus-Universität Weimar

`www.webis.de`

SIGIR 2022
Madrid

45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022)

Axiomatic Retrieval Experimentation with `ir_axioms`

A Brief Tour into Axiomatic IR

Successful retrieval scoring functions share similar properties:

- TF weighting
- IDF weighting
- Length normalization

Example:

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

“Axioms” formally capture such properties / constraints of relevance.

© Bondarenko, Fröbe, Reimer, Stein, Völske, Hagen 2022

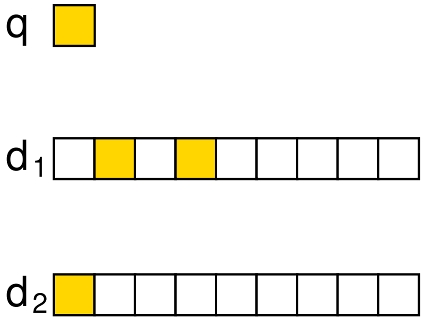
Axiomatic Retrieval Experimentation with `ir_axioms`

Axiom Example: TFC1

TFC1 Favor documents with more occurrences of query terms [Fang et al., SIGIR'04]

Given:

- Single-term query $q = \{t\}$
- Documents d_1, d_2 with $|d_1| = |d_2|$



© Bondarenko, Fröbe, Reimer, Stein, Völske, Hagen 2022

Axiomatic Retrieval Experimentation with `ir_axioms`

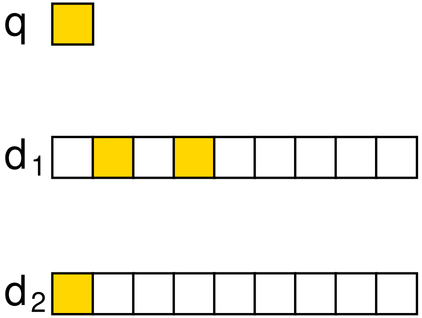
Axiom Example: TFC1

TFC1 Favor documents with more occurrences of query terms [Fang et al., SIGIR'04]

Given:

- Single-term query $q = \{t\}$
- Documents d_1, d_2 with $|d_1| = |d_2|$

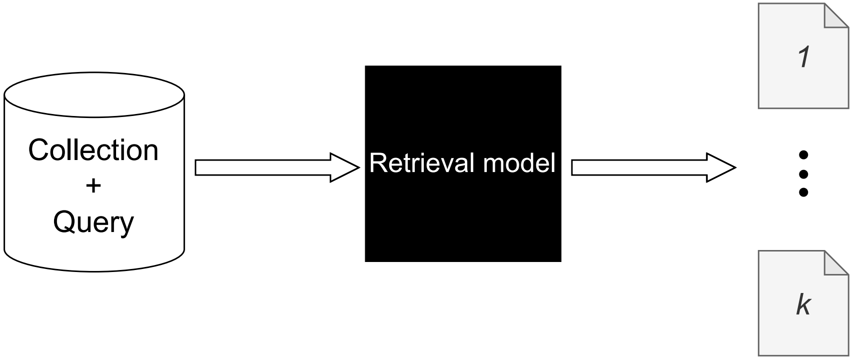
IF $TF(t, d_1) > TF(t, d_2)$ THEN $d_1 >_{\text{TFC1}} d_2$



Axiomatic Retrieval Experimentation with `ir_axioms`

Axiom Applications

- ❑ Improving an initial retrieval result via re-ranking [Hagen et al., CIKM’16]
- ❑ Using axioms as regularization loss in neural models [Rosset et al., SIGIR’19]
- ❑ Learning how to combine different retrieval models [Arora and Yates, AMIR@ECIR’19]
- ❑ Analyzing and explaining neural rankers [Völske et al., ICTIR’21; Formal et al., ECIR’21]



Axiomatic Retrieval Experimentation with `ir_axioms`

Motivation

- Many IR toolkits: Terrier, Anserini, Capreolus, PyTerrier, etc.
- They do not include components for IR axioms

Axiomatic Retrieval Experimentation with ir_axioms

Motivation

- Many IR toolkits: Terrier, Anserini, Capreolus, PyTerrier, etc.
- They do not include components for IR axioms

ir_axioms

- Python framework for experiments with IR axioms
- Implements 25 axioms (relax $|d_i| \approx_{10\%} |d_j|$, multi-term queries, etc.)
- Access to retrieval models and test collections in PyTerrier and ir_datasets
- Extensibility: simple to define new axioms

```python
class TFC1(Axiom):
  name = "TFC1"
  def preference(self, c, q, d_i, d_j) -> float:
    if not approximately_same_length(c, d_i, d_j, 0.1):
      return 0
    tf_i = sum(c.term_frequency(d_i, t) for t in c.terms(q))
    tf_j = sum(c.term_frequency(d_j, t) for t in c.terms(q))
    if approximately_equal(tf_i, tf_j, 0.1):
      return 0
    return 1 if tf_i > tf_j else -1
```

Axiomatic Retrieval Experimentation with `ir_axioms`

Showcase

Jupyter Notebook:

`https://github.com/webis-de/ir_axioms/blob/main/examples/sigir2022_showcase.ipynb`

Google Colab:

`https://colab.research.google.com/github/webis-de/ir_axioms/blob/main/examples/sigir2022_showcase.ipynb`

© Bondarenko, Fröbe, Reimer, Stein, Völske, Hagen 2022

Axiomatic Retrieval Experimentation with `ir_axioms`

Summary

`ir_axioms`:

- 25 axioms
- Post-hoc analysis of rankings
- Axiomatic re-ranking pipeline
- Axiom preferences as features for LTR
- Caching and parallelization

Software and examples (contributions are welcome):

```
webis-de/ir_axioms
pip install ir_axioms
```

Axiomatic Retrieval Experimentation with `ir_axioms`

Summary

`ir_axioms`:

- 25 axioms
- Post-hoc analysis of rankings
- Axiomatic re-ranking pipeline
- Axiom preferences as features for LTR
- Caching and parallelization

Software and examples (contributions are welcome):

- `webis-de/ir_axioms`
- `pip install ir_axioms`

thank you!

Thanks to SIGIR for a student travel grant.