Axiomatic Analysis and Optimization of Information Retrieval Models

SIGIR 2014 Tutorial

Hui Fang
Dept. of Electrical and Computer Engineering
University of Delaware
USA
http://www.eecis.udel.edu/~hfang

ChengXiang Zhai
Dept. of Computer Science
University of Illinois at Urbana-Champaign
USA
http://www.cs.illinois.edu/homes/zhai

Goal of Tutorial

- Axiomatic Approaches to IR
  - Review major research progress
  - Discuss promising research directions

You can expect to learn:
- Basic methodology of axiomatic analysis and optimization of retrieval models
- Novel retrieval models developed using axiomatic analysis

Organization of Tutorial

Motivation

Axiomatic Analysis and Optimization: Early Work

Axiomatic Analysis and Optimization: Recent Work

Summary

Search accuracy matters!

<table>
<thead>
<tr>
<th># Queries /Day</th>
<th>X 1 sec</th>
<th>X 10 sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>4,700,000,000</td>
<td>~1,300,000 hrs</td>
</tr>
<tr>
<td>Twitter</td>
<td>1,600,000,000</td>
<td>~440,000 hrs</td>
</tr>
<tr>
<td>PubMed</td>
<td>2,000,000</td>
<td>~550 hrs</td>
</tr>
</tbody>
</table>

How can we improve all search engines in a general way?

Sources:
Google, Twitter: http://www.statisticbrain.com/
Behind all the search boxes…

Scoring based on bag of words in general

Some are working very well (equally well)

Retrieval model = computational definition of ‘relevance’

Improving retrieval models is a long-standing challenge.

Some state of the art retrieval models
PIV, DIR, BM25 and PL2 tend to perform similarly.

Performance Comparison (MAP)

<table>
<thead>
<tr>
<th>Method</th>
<th>AP88-89</th>
<th>DOE</th>
<th>FR88-89</th>
<th>Wt2g</th>
<th>Trec7</th>
<th>trec8</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIV</td>
<td>0.23</td>
<td>0.18</td>
<td>0.19</td>
<td>0.29</td>
<td>0.18</td>
<td>0.24</td>
</tr>
<tr>
<td>DIR</td>
<td>0.22</td>
<td>0.18</td>
<td>0.21</td>
<td>0.30</td>
<td>0.19</td>
<td>0.26</td>
</tr>
<tr>
<td>BM25</td>
<td>0.23</td>
<td>0.19</td>
<td>0.23</td>
<td>0.31</td>
<td>0.19</td>
<td>0.25</td>
</tr>
<tr>
<td>PL2</td>
<td>0.22</td>
<td>0.19</td>
<td>0.22</td>
<td>0.31</td>
<td>0.18</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Why do they tend to perform similarly even though they were derived in very different ways?

Why does it seem to be hard to beat these strong baseline methods?

• "Ad Hoc IR – Not Much Room for Improvement" [Trotman & Keeler 2011]
• "Has Ad Hoc Retrieval Improved Since 1994?" [Armstrong et al. 2009]

Are they hitting the ceiling of bag-of-words assumption?
• If yes, how can we prove it?
• If not, how can we find a more effective one?

Additional Observations

• PIV (vector space model) 1996
• DIR (language modeling approach) 2001
• BM25 (classic probabilistic model) 1994
• PL2 (divergence from randomness) 2002

Why does it seem to be hard to beat these strong baseline methods?

• “Ad Hoc IR – Not Much Room for Improvement” [Trotman & Keeler 2011]
• “Has Ad Hoc Retrieval Improved Since 1994?” [Armstrong et al. 2009]

Are they hitting the ceiling of bag-of-words assumption?
• If yes, how can we prove it?
• If not, how can we find a more effective one?

Suggested Answers: Axiomatic Analysis

• Why do these methods tend to perform similarly even though they were derived in very different ways?
  They share some nice common properties
  These properties are more important than how each is derived

• Why are they better than many other variants?
  Other variants don’t have all the “nice properties”

• Why does it seem to be hard to beat these strong baseline methods?
  We don’t have a good knowledge about their deficiencies

• Are they hitting the ceiling of bag-of-words assumption?
  – If yes, how can we prove it?
  – If not, how can we find a more effective one?
  Need to formally define “the ceiling” (i.e. complete set of “nice properties”)

Axiomatic Analysis

Axiomatic Relevance Hypothesis (ARH)

• Relevance can be modeled by a set of formally defined constraints on a retrieval function.
  – If a function satisfies all the constraints, it will perform well empirically.
  – If function $F'$ satisfies more constraints than function $F$, $F'$ would perform better than $F$, empirically.

• Analytical evaluation of retrieval functions
  – Given a set of relevance constraints $C = \{ \neg c_1, \ldots, \neg c_r \}$
  – Function $F'$ is analytically more effective than function $F$, if the set of constraints satisfied by $F'$ is a proper subset of those satisfied by $F$.
  – A function $F'$ is optimal iff it satisfies all the constraints in $C$. 

Performance sensitive to small variations in a formula

$$ PIV: S(Q, D) = \sum_{t \in Q} c(t, Q) \times \log \left( \frac{N + 1}{df(t)} \right) \frac{1}{1 - s} \times \frac{DF(t)}{avdl} $$

Why is a state of the art retrieval function better than many other variants?
Different functions, but similar heuristics

- PIV (vector space model)
  \[ \sum_{d \in \mathcal{D}} \log(1 + \text{freq}(q,d)) \cdot \text{tf}(q,d) \cdot \text{idf}(q) \]

- DIR (language modeling approach)
  \[ \sum_{d \in \mathcal{D}} \frac{\text{freq}(q,d)}{(1 + \text{freq}(q,d))^{0.75}} \cdot \text{idf}(q) \]

- BM25 (classic probabilistic model)
  \[ \sum_{d \in \mathcal{D}} \frac{\text{freq}(q,d)}{1 + \text{freq}(q,d)} \cdot \frac{10 \cdot \text{idf}(q)}{1 + \text{freq}(q,d)} \]

- PL2 (divergence from randomness)
  \[ \left( \sum_{d \in \mathcal{D}} \frac{\text{freq}(q,d)}{1 + \text{freq}(q,d)} \right)^2 \cdot \text{idf}(q) \]

TF weighting
- IDF weighting
- Length Norm.

Term Frequency Constraints (TFC1)

TF weighting heuristic I:
- Give a higher score to a document with more occurrences of a query term.

- TFC1
  - Let \( Q \) be a query and \( D \) be a document.
  - If \( q \in Q \) and \( d \notin Q \)
  - then \( S(D, Q \cup \{q\}) > S(D \cup \{q\}) \)
  - \( S(D, Q) \) is the score of \( Q \) on \( D \)

\[ S(D, Q) = \frac{1}{1 + \text{freq}(q,d)} \cdot \text{idf}(q) \]

Term Frequency Constraints (TFC2)

TF weighting heuristic II:
- Require that the amount of increase in the score due to adding a query term must decrease as we add more terms.

- TFC2
  - Let \( Q \) be a query with only one query term \( q \).
  - Let \( D_q \) be a document.
  - then \( S(D_q \cup \{q\}, Q) > S(D_q, Q) \) and \( S(D_q \cup \{q\}, Q) - S(D_q, Q) = S(D_q \cup \{q\}, Q) - S(D_q, Q) \)

\[ S(D_q, Q) = \frac{1}{1 + \text{freq}(q,d)} \cdot \text{idf}(q) \]
**Term Frequency Constraints (TFC3)**

TF weighting heuristic III:
Favor a document with more distinct query terms.

- **TFC3**
  - Let \( q \) be a query and \( w_1, w_2 \) be two query terms.
  - Assume \( idf_t(w_1) = idf_t(w_2) \) and \( |d_1| = |d_2| \)
  - if \( c(w_1, d_1) = c(w_1, d_2) = c(w_2, d_1) \)
  - and \( c(w_2, d_2) = 0 \), \( c(w_1, d_1) \neq 0, c(w_2, d_2) \neq 0 \)
  - then \( S(d_1, q) > S(d_2, q) \)

**Length Normalization Constraints (LNCs)**

Document length normalization heuristic:
Penalize long documents (LNC1);
Avoid over-penalizing long documents (LNC2).

- **LNC1**
  - Let \( Q \) be a query and \( D \) be a document.
  - If \( |q| < |D| \), \( Q \cap D \neq \emptyset \)
  - then \( S(D \cup |q|, Q) > S(D, Q) \)

- **LNC2**
  - Let \( Q \) be a query and \( D \) be a document.
  - If \( D \cap q = \emptyset \) and \( D \) is constructed by concatenating \( D \) with itself \( k \) times,
  - then \( S(D, Q) \neq S(D, Q) \)

**TF & Length normalization Constraint (TF-LNC)**

TF-LN heuristic:
Regularize the interaction of TF and document length.

- **TF-LNC**
  - Let \( Q \) be a query and \( D \) be a document.
  - If \( q \) is a query term,
  - then \( S(D \cup \{q\}, Q) > S(D, Q) \)
  - \( S(Q, D) > S(Q, D) \)

**Seven Basic Relevance Constraints**
[Fang et al. 2011]

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

**Disclaimers**

- Given a retrieval heuristic, there could be multiple ways of formalizing it as constraints.

- When formalizing a retrieval constraint, it is necessary to check its dependency on other constraints.

**Weak or Strong Constraints?**

The Heuristic captured by TDC:
To penalize the words popular in the collection and assign higher weights to discriminative terms.

- **Our first attempt:**
  - Let \( Q = \{q_1, q_2\} \). Assume \( |D_1| = |D_2| \) and \( c(q_1, D_1) + c(q_2, D_1) = c(q_1, D_2) + c(q_2, D_2) \). If \( td(q_1) = td(q_2) \) and \( c(q_1, D_1) = c(q_2, D_2) \), we have \( S(Q, D_1) > S(Q, D_2) \).

- **Our second attempt (a relaxed version):**
  - Let \( Q = \{q_1, q_2\} \). Assume \( |D_1| = |D_2| \) and \( D_1 \) contains only \( q_1 \) and \( D_2 \) contains only \( q_2 \).
  - If \( td(q_1) = td(q_2) \), we have \( S(Q, D_1 \cup D_2) > S(Q, D_1 \cup D) \).
Key Steps of Constraint Formalization

• Identify desirable retrieval heuristics

• Formalize a retrieval heuristic as reasonable retrieval constraints.

• After formalizing a retrieval constraint, check how it is related to other retrieval constraints.
  – Properties of a constraint set
    • Completeness
    • Redundancy
    • Conflict

Axiomatic Analysis and Optimization: Early Work

– Outline

• Formalization of Information Retrieval Heuristics

• Analysis of Retrieval Functions with Constraints

• Development of Novel Retrieval Functions

An Example of Constraint Analysis

PIV:

\[
S(d,q) = \sum_{w \in q} \frac{1 + \log(1 + \log(c(w, d)))}{1 - s + \frac{\log(|d|)}{\log(n + 1)}} \cdot c(w, q) \cdot \log(N + 1) + \frac{d_1 \cdot \log(|d|)}{\log(n + 1)}
\]

LNC2:

Let \( q \) be a query.

If \( \forall d_2, \|d_2\| = k \cdot \|d_1\| \) and \( c(w, d_2) = k \cdot c(w, d_1) \),
then \( S(d_2, q) = S(d_1, q) \)

\[ f(d_2, q) = f(d_1, q) \]

Does PIV satisfy LNC2?

An Example of Constraint Analysis

\[
LNC2: \quad \text{Let } q \text{ be a query.}
\]

If \( \forall d_2, \|d_2\| = k \cdot \|d_1\| \) and \( c(w, d_2) = k \cdot c(w, d_1) \),
then \( S(d_2, q) = S(d_1, q) \)

\[ f(d_2, q) = f(d_1, q) \]

An Example of Constraint Analysis

\[
S(d,q) = \sum_{w \in q} \frac{1 + \log(1 + \log(c(w, d)))}{1 - s + \frac{\log(|d|)}{\log(n + 1)}} \cdot c(w, q) \cdot \log(N + 1) + \frac{d_1 \cdot \log(|d|)}{\log(n + 1)}
\]

\[ f(d_2, q) = f(d_1, q) \]

An Example of Constraint Analysis

\[
S(d,q) = \sum_{w \in q} \frac{1 + \log(1 + \log(c(w, d)))}{1 - s + \frac{\log(|d|)}{\log(n + 1)}} \cdot c(w, q) \cdot \log(N + 1) + \frac{d_1 \cdot \log(|d|)}{\log(n + 1)}
\]

\[ f(d_2, q) = f(d_1, q) \]
Review: Axiomatic Relevance Hypothesis

- Relevance can be modeled by a set of formally defined constraints on a retrieval function.
  - If a function satisfies all the constraints, it will perform well empirically.
  - If function \( f_2 \) satisfies more constraints than function \( f_1 \), \( f_2 \) would perform better than \( f_1 \) empirically.
- Analytical evaluation of retrieval functions
  - Given a set of relevance constraints \( C = (c_1, \ldots, c_n) \)
  - Function \( f_2 \) is analytically more effective than function \( f_1 \)
    iff the set of constraints satisfied by \( f_2 \) is a proper subset of those satisfied by \( f_1 \).
  - A function \( F \) is optimal iff it satisfies all the constraints in \( C \).

Testing the Axiomatic Relevance Hypothesis

- Is the satisfaction of these constraints correlated with good empirical performance of a retrieval function?
- Can we use these constraints to analytically compare retrieval functions without experimentation?
  - “Yes!” to both questions
    - When a formula does not satisfy the constraint, it often indicates non-optimality of the formula.
    - Violation of constraints may pinpoint where a formula needs to be improved.
    - Constraint analysis reveals optimal ranges of parameter values

Violation of Constraints \( \rightarrow \) Poor Performance

- Okapi BM25

\[
\log \left( \frac{N - d_f(t) + 0.5}{d_f(t)} \right) \cdot \frac{(k_1 + 1) \cdot c(t, D)}{(k_1 + 1) \cdot c(t, D) + b \cdot \frac{1}{|D|}} - \frac{k_1 \cdot c(t, D)}{(k_1 + 1) \cdot c(t, D) + b \cdot \frac{1}{|D|}}
\]

Negative \( \rightarrow \) Violates the constraints

Constraints Analysis \( \rightarrow \) Guidance for Improving an Existing Retrieval Function

- Modified Okapi BM25

\[
\log \left( \frac{N - d_f(t) + 0.5}{d_f(t)} \right) \cdot \frac{(k_1 + 1) \cdot c(t, D)}{(k_1 + 1) \cdot c(t, D) + b \cdot \frac{1}{|D|}} - \frac{k_1 \cdot c(t, D)}{(k_1 + 1) \cdot c(t, D) + b \cdot \frac{1}{|D|}}
\]

Make it satisfy constraints; expected to improve performance

Conditional Satisfaction of Constraints \( \rightarrow \) Parameter Bounds

- PIV

\[
\text{LCN2} \Rightarrow c < 0.4
\]

Systematic Analysis of 4 State of the Art Models

- Parameter \( s \) must be small
  - C1
  - C2
  - C3
  - C4
- Problematic when a query term occurs less frequently in a doc than expected
  - Parameter \( c \) must be large
  - C5
  - C6
  - C7
  - C8
- Problematic with common terms
  - Negative IDF
  - C1
  - C2
  - C3
  - C4
- (Modified)
  - PL2 (Original)
    - C5
    - C6
    - C7
    - C8
  - PL2 (modified)
    - Yes
    - C6
    - Yes
    - C8
  - C8

Parameters:

- C1
- C2
- C3
- C4
- C5
- C6
- C7
- C8

Systematic Analysis of 4 State of the Art Models

- [Fang et al. 2011]
Perturbation tests:
An empirical way of analyzing the constraints

For details, see

What if constraint analysis is NOT sufficient?

Medical Diagnosis Analogy

Non-optimal retrieval function
Better performed retrieval function

observe symptoms
Design tests with available instruments
provide treatments

How to find available instruments?
How to design diagnostic tests?

Relevance-Preserving Perturbations

<table>
<thead>
<tr>
<th>Name</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevance addition</td>
<td>Add a query term to a relevant document</td>
</tr>
<tr>
<td>Noise addition</td>
<td>Add a noisy term to a document</td>
</tr>
<tr>
<td>Internal term growth</td>
<td>Add a term to a document that original contains the term</td>
</tr>
<tr>
<td>Document scaling</td>
<td>Concatenate D with itself K times</td>
</tr>
<tr>
<td>Relevance document concatenation</td>
<td>Concatenate two relevant documents K times</td>
</tr>
<tr>
<td>Non-relevant document concatenation</td>
<td>Concatenate two non-relevant documents K times</td>
</tr>
<tr>
<td>Noise deletion</td>
<td>Delete a term from a non-relevant document</td>
</tr>
<tr>
<td>Document addition</td>
<td>Add a document to the collection</td>
</tr>
<tr>
<td>Document deletion</td>
<td>Delete a document from the collection</td>
</tr>
</tbody>
</table>

Relevance-Preserving Perturbations

- Perturb term statistics
- Keep relevance status

Document scaling perturbation:
concatenate every document with itself K times

Length Scaling Test (LV3)

1. Identify the aspect to be diagnosed
2. Choose appropriate perturbations
3. Perform the test and interpret the results

Dirichlet over-penalizes long documents!
Summary of All Tests

<table>
<thead>
<tr>
<th>Tests</th>
<th>What to measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length variance reduction (LV1)</td>
<td>The gain on length normalization</td>
</tr>
<tr>
<td>Length variance amplification (LV2)</td>
<td>The robustness to larger document variance</td>
</tr>
<tr>
<td>Length scaling (LV3)</td>
<td>The ability at avoid over-penalizing long documents</td>
</tr>
<tr>
<td>Term noise addition (TN)</td>
<td>The ability to penalize long documents</td>
</tr>
<tr>
<td>Single query term growth (TG1)</td>
<td>The ability to favor docs with more distinct query terms</td>
</tr>
<tr>
<td>Majority query term growth (TG2)</td>
<td>Favor documents with more query terms</td>
</tr>
<tr>
<td>All query term growth (TG3)</td>
<td>Balance TF and LN more appropriately</td>
</tr>
</tbody>
</table>

Diagnostic Results for DIR

\[
S(Q,D) = \sum_{t \in Q \cap D} c(t,Q) \log \left( 1 + \frac{c(t,D)}{\mu \cdot p(t|C)} \right) - \log(1 + \frac{|D|}{\mu})Q
\]

- **Weaknesses**
  - over-penalizes long documents (TN, LV3)
  - fails to implement one desirable property of TF (TG1)

- **Strengths**
  - performs better in a document with higher document length variance (LV2)
  - implements another desirable property of TF (TG2)

Identifying the weaknesses makes it possible to improve the performance

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>P@30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>trec8</td>
<td>w12g</td>
</tr>
<tr>
<td>DIR</td>
<td>0.257</td>
<td>0.302</td>
</tr>
<tr>
<td>Imp.D.</td>
<td>0.263</td>
<td>0.323</td>
</tr>
</tbody>
</table>

Axiomatic Analysis and Optimization: Early Work – Outline

- Formalization of Information Retrieval Heuristics
- Analysis of Retrieval Functions with Constraints
- Development of Novel Retrieval Functions

Basic Idea of Axiomatic Approach

- How to define the constraints?
  - We’ve talked about that; more later
- How to define the function space?
  - One possibility: leverage existing state of the art functions
- How to search in the function space?
  - One possibility: search in the neighborhood of existing state of the art functions

For details, see:
- Hui Fang and ChengXiang Zhai: An Exploration of Axiomatic Approaches to Information Retrieval, SIGIR’05.
Derivation of New Document Growth Function

Inductive Definition of Function Space

\[ S : Q \times D \to N \]

Define the function space \textit{inductively}

- Primitive weighting function \( f \)
  \[ S(Q,D) = S(Q,D) = f(Q,D) \]
- Query growth function \( h \)
  \[ S(Q,D) = S(Q,D) = h(Q,D) \]
- Document growth function \( g \)
  \[ S(Q,D) = S(Q,D) = g(Q,D) \]

A Sample Derived Function based on BM25

Fang & Zhai 2005

The derived function is less sensitive to the parameter setting

\[ S(Q,D) = \sum_{d \in D} \left( c \cdot \frac{N \cdot df(d)}{df(Q)} \right) \cdot \frac{c \cdot df(d)}{df(Q)} \]

length normalization
A Recent Success of Axiomatic Analysis: Lower Bounding TF Normalization

![Diagram of Existing retrieval functions lack a lower bound for normalized TF with document length.](image)

**Long documents are overly penalized!**

**A very long document matching two query terms can have a lower score than a short document matching only one query term.**

**Lower Bounding TF Constraints (LB1)**

*The presence–absence gap (0-1 gap) shouldn’t be closed due to length normalization.*

\[ S(Q, D_2) = S(Q, D_1) \]

\[ S(Q \cup \{q\}, D_2) < S(Q \cup \{q\}, D_1) \]

**LB1**: Let \( Q \) be a query. Assume \( D_1 \) and \( D_2 \) are two documents such that \( S(Q, D_1) = S(Q, D_2) \). If we reformulate the query by adding another term \( q \notin Q \) into \( Q \), where \( c(q, D_1) = 0 \) and \( c(q, D_2) > 0 \), then \( S(Q \cup \{q\}, D_1) < S(Q \cup \{q\}, D_2) \).
**Lower Bounding TF Constraints (LB2)**

Repeated occurrence of an already matched query term isn’t as important as the first occurrence of an otherwise absent query term.

**LB2:** Let $Q = \{q_1, q_2\}$ be a query with two terms $q_1$ and $q_2$. Assume $bd(q_1) = bd(q_2)$, where $bd(t)$ can be any reasonable measure of term discrimination value. If $D_1$ and $D_2$ are two documents such that $c(q_1, D_1) = c(q_2, D_2) = 0$, $c(q_1, D_1) > 0$, $c(q_1, D_2) > 0$, and $S(Q, D_1) = S(Q, D_2)$, then $S(Q, D_1 \cup \{q_1\} \setminus \{t_i\}) < S(Q, D_2 \cup \{q_2\} \setminus \{t_i\})$, for all $t_i$ and $t_j$ such that $t_i \in D_1$, $t_j \in D_2$, $t_i \notin Q$ and $t_j \notin Q$.

---

**Constraint Comparison (1)**

- **LB1**
  - $Q'$: $Q \cup \{q'_1\}$
  - $D_1$
  - $D_2$
  
- **TFLNC**
  - $W \in Q, (c(q, D_1) = c(q, D_2)) \Rightarrow (c(q, D_1) = c(q, D_2))$

Both constraints are designed to avoid over-penalizing long documents. However, LB1 is more general since it puts less restriction on the document length.

---

**No retrieval model satisfies both LB constraints**

<table>
<thead>
<tr>
<th>Model</th>
<th>LB1</th>
<th>LB2</th>
<th>Parameter and/or query restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>Yes</td>
<td>No</td>
<td>$b$ and $k_f$ should not be too large</td>
</tr>
<tr>
<td>PIV</td>
<td>Yes</td>
<td>No</td>
<td>$s$ should not be too large</td>
</tr>
<tr>
<td>PL2</td>
<td>No</td>
<td>No</td>
<td>$c$ should not be too small</td>
</tr>
<tr>
<td>Dir</td>
<td>No</td>
<td>Yes</td>
<td>$\mu$ should not be too large; query terms should be discriminative</td>
</tr>
</tbody>
</table>

---

**Solution: a general approach to lower-bounding TF normalization**

- **Current retrieval model:**
  
  \[ F(c(t, D), |D|, ...) \]

- **Lower-bounded retrieval model:**
  
  \[ \begin{cases} 
  F(c(t, D), |D|, ...) + F(0, l, ...) & \text{if } c(t, D) > 0 \\
  F(c(t, D), |D|, ...) + F(0, l, ...) & \text{Otherwise} 
  \end{cases} \]

**Example: Dir+, a lower-bounded version of the query likelihood function**

\[ \begin{align*}
\text{Dir:} & \quad \sum_{c(t, D)} c(q, Q) \cdot \log \left( 1 + \frac{c(q, D)}{\mu \cdot p(w | C)} \right) + |Q| \cdot \log \left( 1 + \frac{\delta}{\mu + |D|} \right) \\
\text{Dir+:} & \quad \sum_{c(t, D)} c(q, Q) \cdot \log \left( 1 + \frac{c(q, D)}{\mu \cdot p(w | C)} \right) + |Q| \cdot \log \left( 1 + \frac{\delta}{\mu + |D|} \right) \\
\end{align*} \]

Dir+ incurs almost no additional computational cost.
Example: BM25+, a lower-bounded version of BM25

BM25: \[ \sum_{q \in D} \frac{k_1 + 1}{k_1 + k} \cdot c(t, Q) \cdot \log \frac{N + 1}{df(t)} \]

BM25+: \[ \sum_{q \in D} \frac{k_1 + 1}{k_1 + k} \cdot c(t, Q) \cdot \log \frac{N + 1}{df(t)} \]

BM25+ incurs almost no additional computational cost

Axiomatic Analysis and Optimization: Recent Work – Outline

- Lower-bounding TF Normalization

- Axiomatic Analysis of Pseudo-Relevance Feedback Models

- Axiomatic Analysis of Translational Model

Existing PRF Methods

- Mixture model [Zhai & Lafferty 2001b]
- Divergence minimization [Zhai & Lafferty 2001b]
- Geometric relevance model [Lavenko et al. 2001]
- eDCM (extended dirichlet compound multinomial) [Xu & Akella 2008]
- DRF Bo2 [Amati et al. 2003]
- Log-logistic model [Cinchant et al. 2010]

Motivation for the PRF Constraints

<table>
<thead>
<tr>
<th>Robust-A</th>
<th>Mixture model</th>
<th>Log-logistic model</th>
<th>Divergence minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.280</td>
<td>0.292</td>
<td>0.263</td>
<td></td>
</tr>
<tr>
<td>0.263</td>
<td>0.284</td>
<td>0.254</td>
<td></td>
</tr>
<tr>
<td>0.282</td>
<td>0.285</td>
<td>0.259</td>
<td></td>
</tr>
<tr>
<td>0.273</td>
<td>0.294</td>
<td>0.257</td>
<td></td>
</tr>
</tbody>
</table>

Log-logistic model is more effective because it selects terms
- that are not too common (high IDF and small TF)
- that still occur in sufficient number of feedback documents (average DF)
PRF Heuristic Constraints

[ Clinchant and Gaussier, 2013 ]

• TF effect
  - The feedback weight should increase with the term frequency.

• Concavity effect
  - The above increase should be less marked in high frequency ranges.

• IDF effect
  - When all other things being equal, the feedback weight of a term with higher IDF value should be larger.

• Document length effect
  - The number of occurrences of feedback terms should be normalized by the length of documents they appear in.

• DF effect
  - When all other things being equal, terms occurring in more feedback documents should receive higher feedback weights.

Summary of Constraint Analysis

<table>
<thead>
<tr>
<th></th>
<th>TF</th>
<th>Concave</th>
<th>IDF</th>
<th>Doc Len</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture</td>
<td>Y</td>
<td>Cond.</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Div Min</td>
<td>Y</td>
<td>Y</td>
<td>Cond.</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>G. Rel.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Bo</td>
<td>Y</td>
<td>N</td>
<td>Cond.</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

The authors also discussed how to revise the mixture model and geometric relevance model to improve the performance.

Axiomatic Analysis and Optimization: Recent Work

– Outline

• Lower-bounding TF Normalization

• Axiomatic Analysis of Pseudo-Relevance Feedback Models

• Axiomatic Analysis of Translational Model

For details, see

Maryam Karimzadehgan and ChengXiang Zhai: Axiomatic Analysis of Translation Language Model for Information Retrieval, ECIR'12

The Problem of Vocabulary Gap

Query = auto wash

auto

wash

…

car

wash

vehicle

How to support inexact matching?

("car", "vehicle") ←→ "auto"

"buy" ←→ "wash"

The Problem of Vocabulary Gap

Query = auto wash

auto

wash

…

car

wash

vehicle

P("auto")

P("wash")

How to estimate?

p(w | d) = \sum_{u} P_{ml}(u | d) p_{t}(w | u)

Translation Language Models for IR

[ Berger & Callibrary, 2009 ]

Estimation of translation model

p_{t}(w | u) = Pr(d mentions u | d is about w)

• How do we know whether one estimation method is better than another one?

• Is there any better way than pure empirical evaluation?

• Can we analytically prove the optimality of a translation language model?
**General Constraint 1: Constant Self-Trans. Prob.**

**C1:** In order to have a reasonable retrieval behavior, for all translation language models, the self-translation probability should be the same (constant).

\[ \forall v \text{ and } w, p(w|w) = p(v|v) \]

**Q:**

D1: 

D2: 

\[ \frac{p(w|w)}{p(v|v)} \]

If \( p(w|w) > p(v|v) \), D1 would be (unfairly) favored.

**General Constraint 2**

**C2:** Self-translation probability should be larger than translating any other words to this word. \( \forall u \text{ and } w, p(w|w) > p(w|u) \)

**Q:**

D1: 

D2: 

\[ p(w|D_1) = p(w|D_2) \times p(w|w) \]

\[ p(w|D_2) = p(u|D_2) \times p(w|u) \]

Since \( p(w|D_1) = p(u|D_2) \)

The constraint must be satisfied to ensure a document with exact matching gets higher score.

**General Constraint 3**

**C3:** A word is more likely to be translated to itself than translating into any other words. \( \forall u \text{ and } w, p(w|w) > p(u|w) \)

Again to avoid over-rewarding inexact matches.

**Constraint 4 – Co-occurrence**

**C4:** If word \( u \) occurs more times than word \( v \) in the context of word \( w \) and both words \( u \) and \( v \) co-occur with other words similarly, the probability of translating word \( u \) to word \( w \) should be higher.

If \( c(w, u) > c(w, v) \) and \( \sum_{w'} c(w', u) = \sum_{w'} c(w', v) \)

\[ p(w|u) > p(w|v) \]

**Q:** “Australia”

D: “Brisbane”

D': “Chicago”

p(Australia | Brisbane) > p(Australia | Brisbane)

**Analysis of Mutual Information-based Translation Language Model**

\[ I(w; u) = \sum_{k=0}^{3} \sum_{n=0}^{3} p(X_{w}, X_{u}) \log \frac{p(X_{w}, X_{u})}{p(X_{w})p(X_{u})} \]

\[ p_{wu}(v|w) = \frac{I(w; u)}{\sum_{v} I(w; v)} \]

It only satisfies C3:

\( \forall u \text{ and } w, p(w|w) > p(u|w) \)

Can we design a method to better satisfy the constraints?
Heuristic Adjustment of Self-Translation Probability

Old way (non-constant self translation)
\[ p(w|u) = \begin{cases} \alpha + (1 - \alpha) p(u|u) & w = u \\ (1 - \alpha) p(w|u) & w \neq u \end{cases} \]

New way (constant self translation)
\[ p'(w|u) = \begin{cases} s & s \geq 0.5 \\ (1 - s) p(w|u) & \sum_{w' \in u} p(w'|u) \end{cases} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]

Heuristic Adjustment of Self-Translation Probability

Spain \( \rightarrow \) Europe
Europe \( \rightarrow \) Spain

Main Idea:
- ... Europe ... Spain ...
- ... Europe ... Spain ...
- ... Europe ... France ...
- ... Europe ... France ...
- ... ...

New (constant self translation)
\[ p'(Europe|Spain) = \text{high} \]
\[ p'(Spain|Europe) = \text{low} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]

Heuristic Adjustment of Self-Translation Probability

Spain \( \rightarrow \) Europe
Europe \( \rightarrow \) Spain

Main Idea:
- ... Europe ... Spain ...
- ... Europe ... Spain ...
- ... Europe ... France ...
- ... Europe ... France ...
- ... ...

New (constant self translation)
\[ p'(Europe|Spain) = \text{high} \]
\[ p'(Spain|Europe) = \text{low} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]

Heuristic Adjustment of Self-Translation Probability

Spain \( \rightarrow \) Europe
Europe \( \rightarrow \) Spain

Main Idea:
- ... Europe ... Spain ...
- ... Europe ... Spain ...
- ... Europe ... France ...
- ... Europe ... France ...
- ... ...

New (constant self translation)
\[ p'(Europe|Spain) = \text{high} \]
\[ p'(Spain|Europe) = \text{low} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]

Heuristic Adjustment of Self-Translation Probability

Spain \( \rightarrow \) Europe
Europe \( \rightarrow \) Spain

Main Idea:
- ... Europe ... Spain ...
- ... Europe ... Spain ...
- ... Europe ... France ...
- ... Europe ... France ...
- ... ...

New (constant self translation)
\[ p'(Europe|Spain) = \text{high} \]
\[ p'(Spain|Europe) = \text{low} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]

Heuristic Adjustment of Self-Translation Probability

Spain \( \rightarrow \) Europe
Europe \( \rightarrow \) Spain

Main Idea:
- ... Europe ... Spain ...
- ... Europe ... Spain ...
- ... Europe ... France ...
- ... Europe ... France ...
- ... ...

New (constant self translation)
\[ p'(Europe|Spain) = \text{high} \]
\[ p'(Spain|Europe) = \text{low} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]

Heuristic Adjustment of Self-Translation Probability

Spain \( \rightarrow \) Europe
Europe \( \rightarrow \) Spain

Main Idea:
- ... Europe ... Spain ...
- ... Europe ... Spain ...
- ... Europe ... France ...
- ... Europe ... France ...
- ... ...

New (constant self translation)
\[ p'(Europe|Spain) = \text{high} \]
\[ p'(Spain|Europe) = \text{low} \]

Conditional Context Analysis: Detail

- Use the frequency of seeing word \( w \) in the context of word \( u \) to estimate \( p(w|u) \).
- See \( w \) often in the context of \( u \rightarrow \) high \( p(w|u) \)
\[
p(w|u) = \frac{\sum_{w' \in u} c(w', u) + 1}{\sum_{w' \in u} |V| + 1}
\]

Satisfies more constraints than MI
However, C1 is not satisfied by either method
\[ \forall v \text{ and } w, p(w|v) = p(v|v) \]
Summary: Axiomatic Relevance Hypothesis

• Formal retrieval function constraints for modeling relevance
• Axiomatic analysis as a way to assess optimality of retrieval models
• Inevitability of heuristic thinking in developing retrieval models for bridging the theory-effectiveness gap
• Possibility of leveraging axiomatic analysis to improve the state of the art models
• Axiomatic Framework = constraints + constructive function space based on existing or new models and theories

What we’ve achieved so far

• A large set of formal constraints on retrieval functions
• A number of new functions that are more effective than previous ones
• Some specific questions about existing models that may potentially be addressed via axiomatic analysis
• A general axiomatic framework for developing new models
  – Definition of formal constraints
  – Analysis of constraints (analytical or empirical)
  – Improve a function to better satisfy constraints

Inevitability of heuristic thinking and necessity of axiomatic analysis

• The “theory-effectiveness gap”
  – Theoretically motivated models don’t automatically perform well empirically
  – Heuristic adjustment seems always necessary
  – Cause: inaccurate modeling of relevance
• How can we bridge the gap?
  – The answer lies in axiomatic analysis
  – Use constraints to help identify the error in modeling relevance, thus obtaining insights about how to improve a model

Two unanswered “why questions” that may benefit from axiomatic analysis

• The derivation of the query likelihood retrieval function relies on 3 assumptions: (1) query likelihood scoring; (2) independency of query terms; (3) collection LM for smoothing; however, it can’t explain why some apparently reasonable smoothing methods perform poorly
• No explanation why other divergence-based similarity function doesn’t work well as the asymmetric KL-divergence function D(Q||D)

Open Challenges

• Does there exist a complete set of constraints?
  – If yes, how can we define them?
  – If no, how can we prove it?
• How do we evaluate the constraints?
  – How do we evaluate a constraint? (e.g., should the score contribution of a term be bounded? In BM25, it is.)
  – How do we evaluate a set of constraints?
• How do we define the function space?
  – Search in the neighborhood of an existing function?
  – Search in a new function space?
Open Challenges

- How do we check a function w.r.t. a constraint?
  - How can we quantify the degree of satisfaction?
  - How can we put constraints in a machine learning framework? Something like maximum entropy?
- How can we go beyond bag of words? Model pseudo feedback? Cross-lingual IR?
- Conditional constraints on specific type of queries? Specific type of documents?

Possible Future Scenario 1:
Impossibility Theorems for IR

- We will find inconsistency among constraints
- Will be able to prove impossibility theorems for IR
  - Similar to Kleinberg’s impossibility theorem for clustering


Future Scenario 2:
Sufficiently Restrictive Constraints

- We will be able to propose a comprehensive set of constraints that are sufficient for deriving a unique (optimal) retrieval function
  - Similar to the derivation of the entropy function

Future Scenario 3 (most likely):
Open Set of Insufficient Constraints

- We will have a large set of constraints without conflict, but insufficient for ensuring good retrieval performance
- Room for new constraints, but we’ll never be sure what they are
- We need to combine axiomatic analysis with a constructive retrieval functional space and supervised machine learning

Generalization of the axiomatic analysis process (beyond IR)

1. Set an objective function, e.g.,
   - Ranking: S(Q,D)
   - Diversification: f(D, q, w(), dsim())
2. Identify variables that have impacts to the objective function
3. Formalize constraints based on the variables
   - For each variable, figure out its desirable behavior with respect to the objective function, and these desirable properties would be formalized as axioms (i.e., constraints).
   - Exploratory data analysis
   - Study the relations among multiple variables and formalize the desirable properties of these relations as additional constraints.
4. For all the formalized constraints, study their dependencies and conflicts, and remove redundant constraints.
5. Function Derivation
   - If no conflict constraints, find instantiations of the objective function that can satisfy all constraints.
     - Derive new functions
     - Modify existing ones
   - If there are conflict constraints, study the trade-off and identify scenarios that requires a subset of non-conflict constraints, and then derive functions based on these constraints.
Towards General Axiomatic Thinking

- Given a task of designing a function to solve a problem: Y=f(X)
  - Identify properties function f should satisfy
  - Formalize such properties with mathematically well defined constraints
  - Use the constraints to help identify the best function
- Potentially helpful for designing any function
- Constraints can be of many different forms (inequality, equality, pointwise, listwise, etc)
  - Pointwise: For all “a” that satisfies a certain condition, f(a)=b
  - Pairwise: For all a and b that satisfy a certain condition, f(a)>f(b) (or f(a)=f(b))
  - Listwise: For all a1, a2, ... and ak that satisfy a certain condition, then f(a1)>f(a2)>...>f(ak) (or f(a1)=...=f(ak))

Some Examples of Axiomatic Thinking outside IR (1)

  - “Our idea is inspired by the heuristics in information retrieval such as TF-IDF weighting, and we adapt such heuristics into traffic analysis. ProWord uses a ranking algorithm that maps different dimensions of protocol feature heuristics into different word scoring functions and uses the aggregate score to rank the candidates.”

Some Examples of Axiomatic Thinking outside IR (2)

  - “In this paper, we present a formal study of Feature selection (FS) in text categorization. We first define three desirable constraints that any reasonable FS function should satisfy, then check these constraints on some popular FS methods ... Experimental results indicate that the empirical performance of a FS function is tightly related to how well it satisfies these constraints”

Some Examples of Axiomatic Thinking outside IR (3)

  - Using constraints to help generate test cases for schema matching
  - Cited [Fang & Zhai 2004] as a relevant work

Axiomatic Thinking & Machine Learning

- Learn f using supervised learning = constrain the choice of f with an empirical objective function (minimizing errors on training data)
- However, the learned functions may violate obvious constraints due to limited training data (the data is almost always limited!)
- Axiomatic thinking can help machine learning by regularizing the function space or suggesting a certain form of the functions
- For example, f(x)=a1*x1+a2*x2+...+ak*xk
  - A simple constraint can be if x2 increases, f(X) should increase (derivative w.r.t. x2 is positive) \( \Rightarrow a2>0 \)
  - Another constraint can be: the second derivative w.r.t. x2 is negative (i.e., “diminishing return”) \( \Rightarrow a2<0 \)

The End
References

Axiomatic Approaches (1)

Axiomatic Approaches (2)

Axiomatic Approaches (3)
- [Clinchant&Gaussier, 2011a] Is document frequency important for PRF? S. Clinchant and E. Gaussier. ICTIR 2011.
- [Shima Gerani, ChengXiang Zhai, Fabio Crestani: Score Transformation in Linear Combination for Multi-criteria Relevance Ranking. ECR 2012: 256-267

Axiomatic Approaches (4)

Other References (1)
Other References (2)


Other References (3)


Other References (4)