UNIVERSITY of DELAWARE

Axiomatic Approaches to IR

(CPEG 456/657: Search and Data Mining)

Hui Fang
Department of Electrical and Computer Engineering
University of Delaware

1

UNIVERSITY of DELAWARE

What is axiomatic thinking?

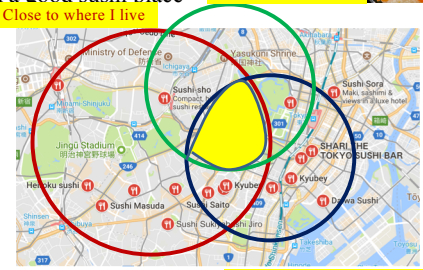
- A thinking process that helps you identify the optimal solution to problem (X)
  - Identify **desirable properties** of a solution
  - **Formalize** the properties as **axioms**.
  - **Use** the **axioms** to **guide** the search for the **optimal solution**.
- You might not have heard it, but it is likely that you used it.

UNIVERSITY of DELAWARE

X = "Find a good sushi place"

Good reviews

Close to where I live

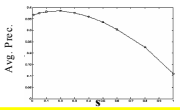


Open for lunch on Saturday

UNIVERSITY of DELAWARE

Empirical Observations

1. Retrieval performance is sensitive to parameter setting.



Parameter sensitivity of s in Pivoted Normalization Method

2. Optimal parameter setting depends on both queries and collections.

Optimal *s* in Pivoted Normalization Method

	AP	DOE	FR	ADF	Web	Trec 7	Trec 8
LK	0.2	0.2	0.05	0.2	---	---	---
SK	0.01	0.2	0.01	0.05	0.01	0.05	0.05
LV	0.3	0.3	0.1	0.2	0.2	0.2	0.2
SV	0.2	0.3	0.1	0.2	0.1	0.1	0.2

5

UNIVERSITY of DELAWARE

Problems with Coarse Modeling of Relevance

- Vector-space model
  - RelScore(Q,D) → sim(Q,D) → ?
  - Slight variation of the TF normalization formula can cause significant difference in performance
- Probabilistic model (classic & language models)
  - RelScore(Q,D) → p(Rel=1|Q,D) →?
  - Slight variation in smoothing causes significant difference in performance

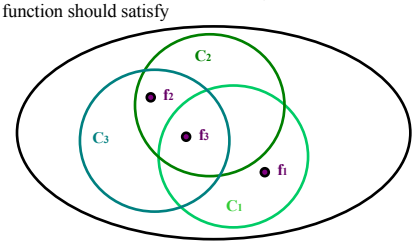
We need to model relevance directly at a finer granularity level of terms.

6

UNIVERSITY of DELAWARE

Basic Idea of Axiomatic Approach

A set of retrieval constraints that any reasonable retrieval function should satisfy



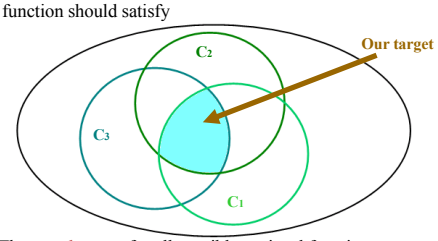
The search space for all possible retrieval functions

7

UNIVERSITY of DELAWARE UNIVERSITY of DE

Basic Idea of Axiomatic Approach

A set of retrieval constraints that any reasonable retrieval function should satisfy



The search space for all possible retrieval functions

8

UNIVERSITY of DELAWARE UNIVERSITY of DE

An Axiomatic Framework for Retrieval Functions

- Component 1: Constraints to be satisfied by an effective retrieval function
- Component 2: Function space that allows us to efficiently search for an effective function

10

UNIVERSITY of DELAWARE UNIVERSITY of DE

Implementation of Component 1:

Question:
How do we define retrieval constraints?

11

UNIVERSITY of DELAWARE UNIVERSITY of DE

The 3 Major Heuristics

- Pivoted Normalization Method

$$\sum_{w\in q\cap d} \frac{1+\ln(1+\ln c(w,d))}{(1-s)+s\frac{|d|}{avdl}} \cdot c(w,q) \ln\frac{N+1}{df(w)}$$

- Dirichlet Prior Method

$$\sum_{w\in q\cap d} c(w,q)\times\ln\left(1+\frac{c(w,d)}{\mu\cdot p(w|C)}\right)+|q|\ln\frac{\mu}{\mu+|d|}$$

- Okapi Method

$$\sum_{w\in q\cap d} \ln\frac{N-df(w)+0.5}{df(w)+0.5} \frac{(k_1+1)\times c(w,d)}{k_1((1-b)+b\frac{|d|}{avdl})+c(w,d)} \frac{(k_3+1)\times c(w,q)}{k_3+c(w,q)}$$

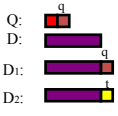
12

UNIVERSITY of DELAWARE UNIVERSITY of DE

Term Frequency Constraints (TFC1)

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

Let Q be a query and D be a document.
If $q \in Q$ and $t \notin Q$,
then $S(Q, D\cup\{q\}) > S(Q, D\cup\{t\})$



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

13

UNIVERSITY of DELAWARE UNIVERSITY of DE

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_1 be a document.
then $S(D_1\cup\{q\},Q)-S(D_1,Q) > S(D_1\cup\{q\}\cup\{q\},Q)-S(D_1\cup\{q\},Q)$

$S(D_2,Q)-S(D_1,Q) > S(D_3,Q)-S(D_2,Q)$

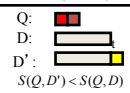
14

UNIVERSITY of DELAWARE

Length Normalization Constraints (LNCs)

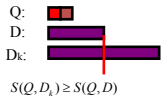
Document length normalization heuristic:
Penalize long documents(LNC1);
Avoid over-penalizing long documents (LNC2) .

- ***LNC1***
  Let Q be a query and D be a document.
  If t is a non-query term,
  then $S(D\cup\{t\},Q)<S(D,Q)$



$S(Q,D')<S(Q,D)$

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



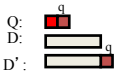
$S(Q,D_k)\geq S(Q,D)$

UNIVERSITY of DELAWARE

TF-LENGTH 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 $S(D\cup\{q\},Q)>S(D,Q)$.



$S(Q,D')>S(Q,D)$

16

UNIVERSITY of DELAWARE

Is constraint analysis related to the performance of a retrieval function?

18

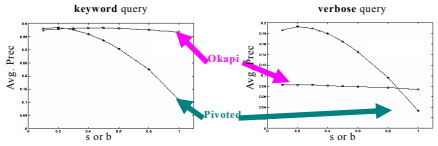
UNIVERSITY of DELAWARE

Violation of Constraints → Poor Performance

- Okapi Method

$$\sum_{w\in q\cap d}\ln\frac{N-df(w)+0.5}{df(w)+0.5}\cdot\frac{(k_1+1)\times c(w,d)}{k_1((1-b)+b\frac{|d|}{avdl})+c(w,d)}\cdot\frac{(k_3+1)\times c(w,q)}{k_3+c(w,q)}$$

Negative when df(w) is large → Violate many constraints



19

UNIVERSITY of DELAWARE

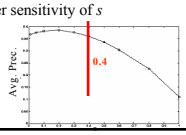
Conditional Satisfaction of Constraints → Parameter Bounds

- Pivoted Normalization Method LNC2 → s<0.4

Optimal s (for average precision)

	AP	DOE	FR	ADF	Web	Tree 7	Tree 8
LK	0.2	0.2	0.05	0.2	---	---	---
SK	0.01	0.2	0.01	0.05	0.01	0.05	0.05
LV	0.3	0.3	0.1	0.2	0.2	0.2	0.2
SV	0.2	0.3	0.1	0.2	0.1	0.1	0.2

Parameter sensitivity of s



20

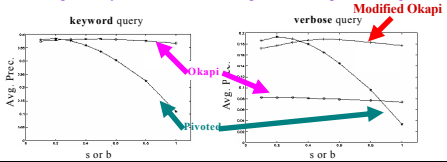
UNIVERSITY of DELAWARE

Constraints Analysis → Guidance for Improving an Existing Retrieval Function

- Modified Okapi Method

$$\sum_{w\in q\cap d}\ln\frac{N-df(w)+0.5}{df(w)+0.5}\cdot\ln\frac{N+1}{df}\cdot\frac{(k_1+1)\times c(w,d)}{k_1((1-b)+b\frac{|d|}{avdl})+c(w,d)}\cdot\frac{(k_3+1)\times c(w,q)}{k_3+c(w,q)}$$

Make Okapi satisfy more constraints; expected to help verbose queries



21

UNIVERSITY of DELAWARE UNIVERSITY of DE

Implementation of Component 2:

Question:
How can we define a function space that can be searched efficiently?

23

UNIVERSITY of DELAWARE UNIVERSITY of DE

Component 2: Function Space

$Q = \{q_1, q_2, ..., q_m\}$
$D = \{d_1, d_2, ..., d_n\}$ Bag of terms

$S : Q \times D \rightarrow \Re$ Function Space

Define the function space *inductively*

24

UNIVERSITY of DELAWARE UNIVERSITY of DE

Inductive Definition of Function Space

Primitive weighting function

Q: q
D: q
$S(Q, D) = weight(q)$

Q: q
D: d
$S(Q, D) = penalty(q, d)$

Query growth function

Q: q'
D:
$S(Q', D) = S(Q \cup \{q'\}, D) = h(S(Q, D), S(\{q'\}, D), q', D, Q)$

Document growth function

Q:
D': d'
$S(Q, D') = S(Q, D \cup \{d'\}) = g(S(Q, D), S(Q, \{d'\}), d', D, Q)$

Need to ensure that this indeed defines a function

25

UNIVERSITY of DELAWARE UNIVERSITY of DE

Derivation of New Retrieval Functions

Decompose the existing retrieval functions

⇩

Generalize each component function

⇩

Use constraints to find some alternative implementation of a component function

⇩

Assemble the new component functions to form a new retrieval function.

27

UNIVERSITY of DELAWARE UNIVERSITY of DE

Component Function 1: Primitive Weighting Function

Decompose the existing retrieval functions

⇩ Roughly IDF

Generalize the component function

⇩ Choice 1: $weight(q) = P(rel \mid occ)$

Choice 2: $weight(q) = \log \frac{P(occ \cap rel)}{P(occ)P(rel)}$

Use constraints to find some alternative implementation

Choice 1: $weight(q) = (\frac{N}{df(q)})^k$ EXP
Choice 2: $weight(q) = \log \frac{N}{df(q)}$ LOG

28

UNIVERSITY of DELAWARE UNIVERSITY of DE

Component Function 2: Query Growth Function

Follow the existing retrieval functions:

$$S(Q \cup \{q\}, D) = S(Q, D) + \alpha S(\{q\}, D)$$

$\alpha = 1$ (*same as Pivoted & Dirichlet*)

29

UNIVERSITY of DELAWARE UNIVERSITY of DE

Component Function 3: Document Growth Function

Decompose the existing functions Pivoted Normalization

$$\frac{1-s+s\frac{|D|}{avdl}}{1-s+s\frac{|D|+1}{avdl}}\cdot S(Q,D)+\frac{1-s+s\frac{1}{avdl}}{1-s+s\frac{|D|+1}{avdl}}\cdot[\ln(1+\ln(c(q,D)+1))-\ln(1+\ln(c(q,D)))]\cdot S(\{q\},D)$$

Generalize the component function of existing functions

$$\lambda_1(|D|)\cdot S(Q,D)+\lambda_2(|D|)\cdot z(c(q,D))\cdot S(\{q\},D)$$

Use constraints to find some alternative implementation

$$\lambda_1(k)=\frac{k+avdl/s}{k+1+avdl/s},\ \lambda_2(k)=\frac{1+avdl/s}{k+1+avdl/s}$$

30

UNIVERSITY of DELAWARE UNIVERSITY of DE

Summary of Derived Formulas

		Primitive Weighting Function	
		LOG	EXP
Doc. Growth Func.	PN-variation	$\sum_{i\in Q\cap D} c(t,Q)\cdot TF(c(t,D))\cdot LN(\lvert D\rvert)\cdot LW(t)$	$\sum_{i\in Q\cap D} c(t,Q)\cdot TF(c(t,D))\cdot LN(\lvert D\rvert)\cdot EW(t)$
	Okapi-variation	$\sum_{i\in Q\cap D} c(t,Q)\cdot TF\_LN(c(t,D),\lvert D\rvert)\cdot LW(t)$	$\sum_{i\in Q\cap D} c(t,Q)\cdot TF\_LN(c(t,D),\lvert D\rvert)\cdot EW(t)$
	DP-variation	$\sum_{i\in Q\cap D} c(t,Q)\cdot TF(c(t,D))\cdot LW(t)-\gamma(\lvert D\rvert,\lvert Q\rvert)$	$\sum_{i\in Q\cap D} c(t,Q)\cdot TF(c(t,D))\cdot EW(t)-\gamma(\lvert D\rvert,\lvert Q\rvert)$

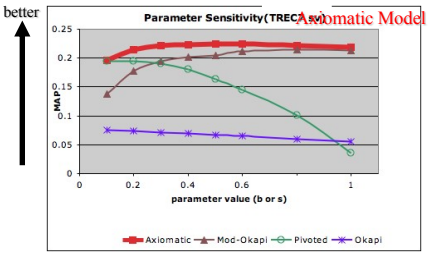
$$S(Q,D)=\sum_{i\in Q\cap D} c(t,Q)\cdot\left(\frac{N}{df(t)}\right)^{\gamma}\frac{c(t,D)}{c(t,D)+s+\frac{s\cdot|D|}{avdl}}$$

Re-parameterization of Okapi TF with one parameter

31

UNIVERSITY of DELAWARE UNIVERSITY of DE

The derived function is less sensitive to the parameter setting



32

UNIVERSITY of DELAWARE UNIVERSITY of DE

Any limitation?

34

UNIVERSITY of DELAWARE UNIVERSITY of DE

Semantic Term Matching is Missing

Q: car

D1: vehicle $S(D_1,Q)$? $S(D_2,Q)$

D2: fish

35

UNIVERSITY of DELAWARE UNIVERSITY of DE

Semantic Term Matching Constraints (STMC1)

Semantic Term Matching Heuristic I:
Give a higher score to a document with a term that is more semantically related to a query term

- ***STMC1***

Let $Q=\{q\}$ be a query.
Let $D_1=\{d_1\}$, $D_2=\{d_2\}$ be two documents.
If $s(q,d_1)>s(q,d_2)$,
then $S(Q,D_1)>S(Q,D_2)$

Q: q
D1: d1 $s(q,d_1)>s(q,d_2)$
D2: d2
$S(Q,D_1)>S(Q,D_2)$

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

36

UNIVERSITY of DELAWARE UNIVERSITY of DE

Semantic Term Matching Constraints

Semantic Term Matching heuristic II:
Favor semantically similar terms(SMTC2);
Avoid over-favoring semantic terms (SMTC3) .

- ***SMTC2***

Let $Q=\{q_1,q_2\}$ be a query and d be non-query term such that $s(q_2,d_2)>0$. $S(\{q_1\},\{q_1\})=S(\{q_2\},\{q_2\})$

If $|D_1|=|D_2|>1$, $c(q_1,D_1)=|D_1|$ and $c(q_1,D_2)=|D_2|-1$ then $S(Q,D_1)<S(Q,D_2)$

Q: D1: D2: $S(Q,D_1)<S(Q,D_2)$

- ***SMTC3***

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

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

Q: D1: D2: $S(Q,D_1)>S(Q,D_2)$

UNIVERSITY of DELAWARE UNIVERSITY of DE

Analyze Derived Function with SMTCs

$$S(Q,D)=\sum_{t\in Q\cap D} c(t,Q)\cdot\left(\frac{N}{df(t)}\right)^{0.35}\cdot\frac{c(t,D)}{c(t,D)+s+\frac{s\cdot|D|}{avdl}}$$

Primitive Weighting Function:

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

$$weight(q)=\left(\frac{N}{df(q)}\right)^{0.35}$$

Violate STMC1 constraint!

UNIVERSITY of DELAWARE UNIVERSITY of DE

Incorporate Semantic Term Matching

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

⇩ Generalization

$$f(q,d)=weight(q)\times s(q,d)$$

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

UNIVERSITY of DELAWARE UNIVERSITY of DE

What You Should Know

- The key difference between the axiomatic approach and traditional retrieval models is that it directly models relevance with term-based constraints.
- The axiomatic framework makes it possible to
  - Predict the performance of a formula analytically without doing experiments
  - Derive completely new retrieval formulas with some guarantee of optimality
- Some new retrieval formulas derived tend to be more robust than existing models
- **Challenge**: deriving a formula that is consistently better (more effective and more robust) than all the existing formulas