Retrieval and Feedback Models for Blog Distillation

CMU at the TREC 2007 Blog Track

Jonathan Elsas, Jaime Arguello,
Jamie Callan, Jaime Carbonell

CMU's Blog Distillation Focus

- Two Research Questions:
  - What is the appropriate unit of retrieval? Feeds or Entries?
  - How can we effectively do pseudo-relevance feedback for Blog Distillation?
- Our four submissions investigate these two dimensions.

Outline

- Corpus Preprocessing Tasks
- Two Feedback Models
- Two Retrieval Models
- Results
- Discussion

Corpus Preprocessing

- Used only FEED documents (vs. PERMALINK or HOMEPAGE documents).
- For each FEEDNO, extracted each new entry across all feed fetches
- Aggregated into 1-document-per-FEEDNO, retaining structural elements from the FEED documents:

  Title, description, entry, entry title, etc…
- Very helpful: Python Universal Feed Parser

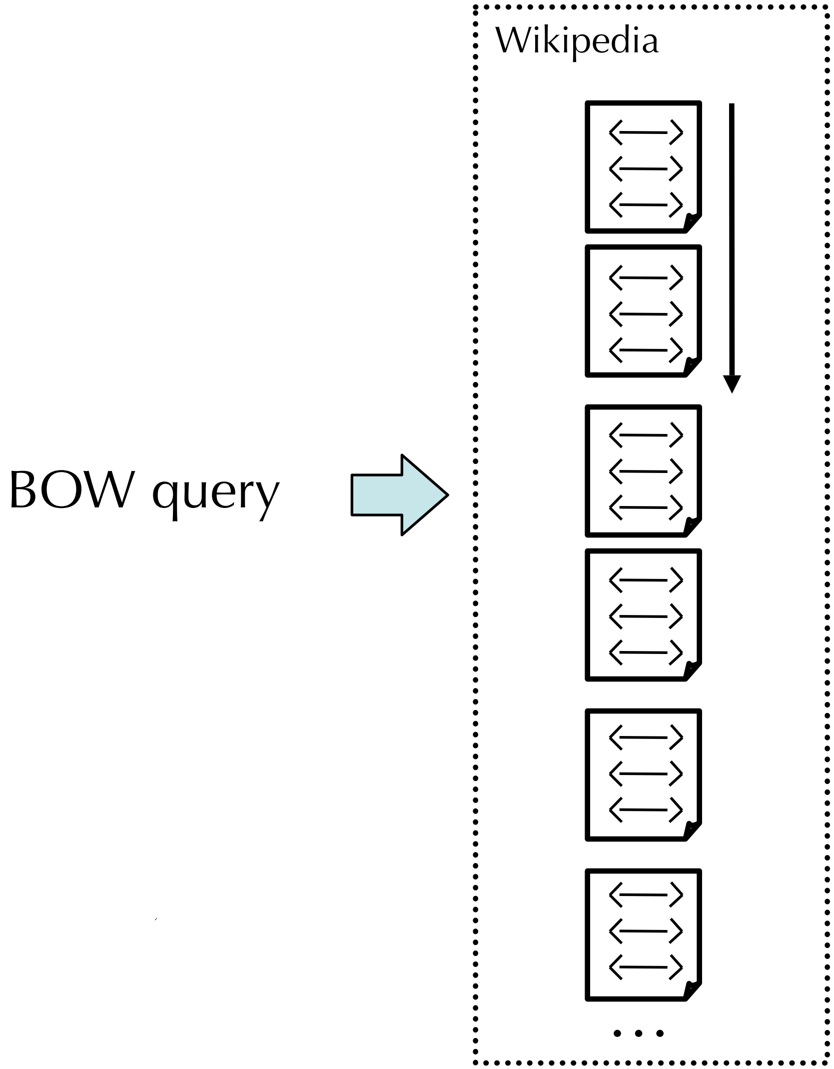
Two Pseudo-Relevance Feedback Models

- Indri's Built in Pseudo-Relevance Feedback (Lavrenko's Relevance Model)
  - Using Metzler's Dependence Model query on the full feed documents
  - Produces weighted unigram PRF query,

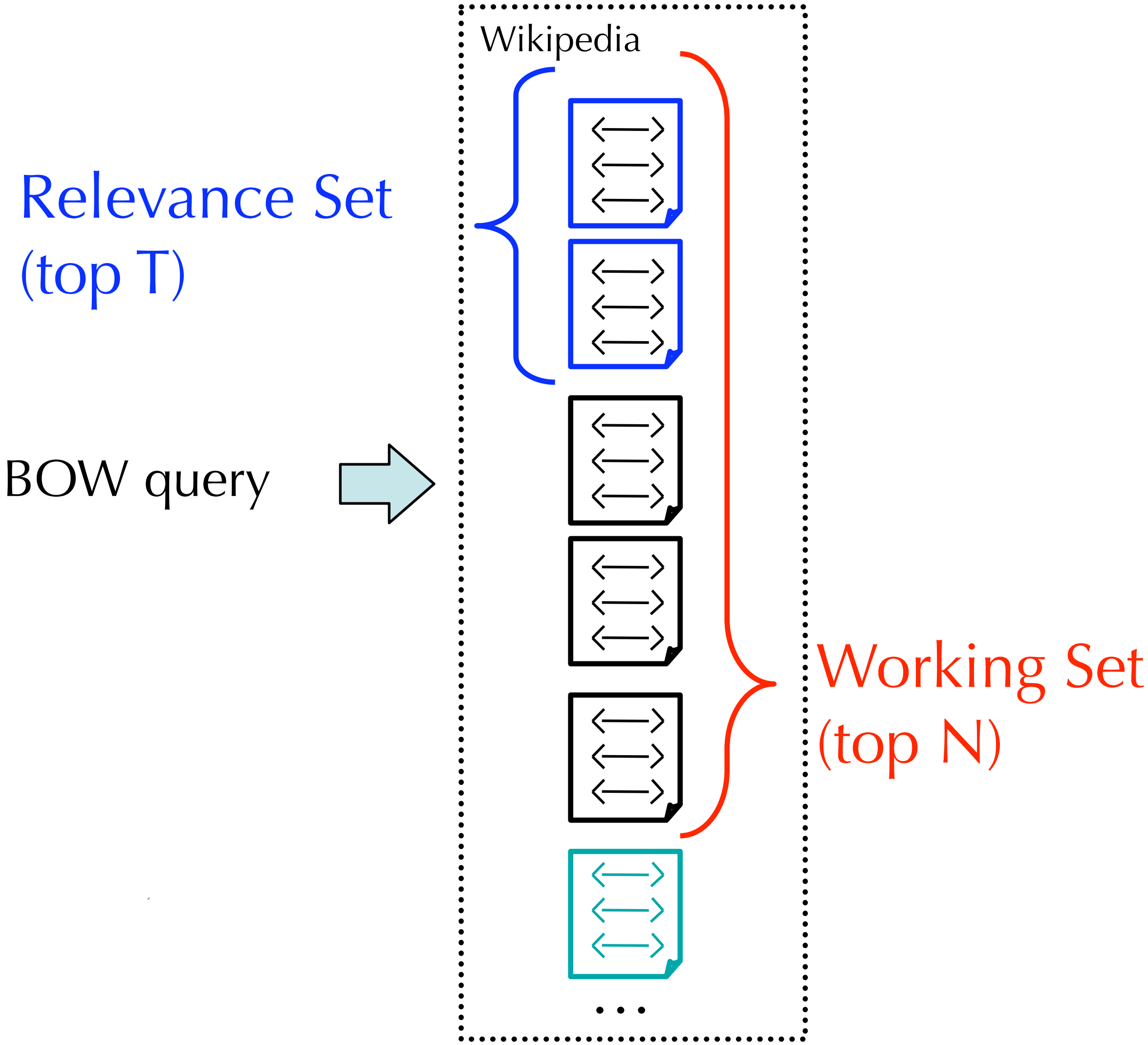
$$Q_{RM} = \#weight(\ w_1\ t_1 \quad w_2\ t_2 \quad \ldots \quad w_{50}\ t_{50})$$

- Wikipedia-based Pseudo-Relevance Feedback
  - Focus on anchor text linking to highly ranked documents wrt baseline BOW query

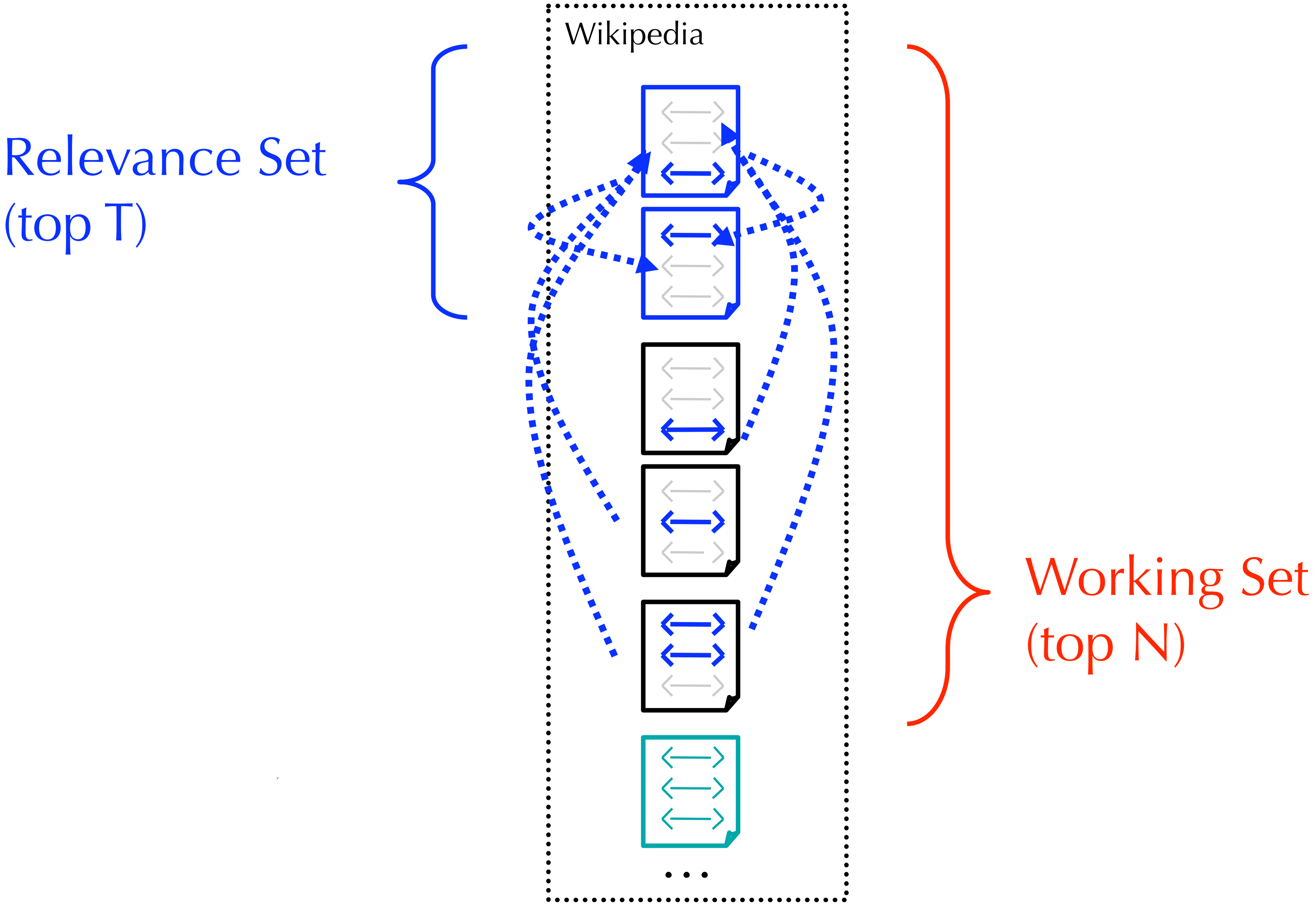
Wikipedia-based PRF



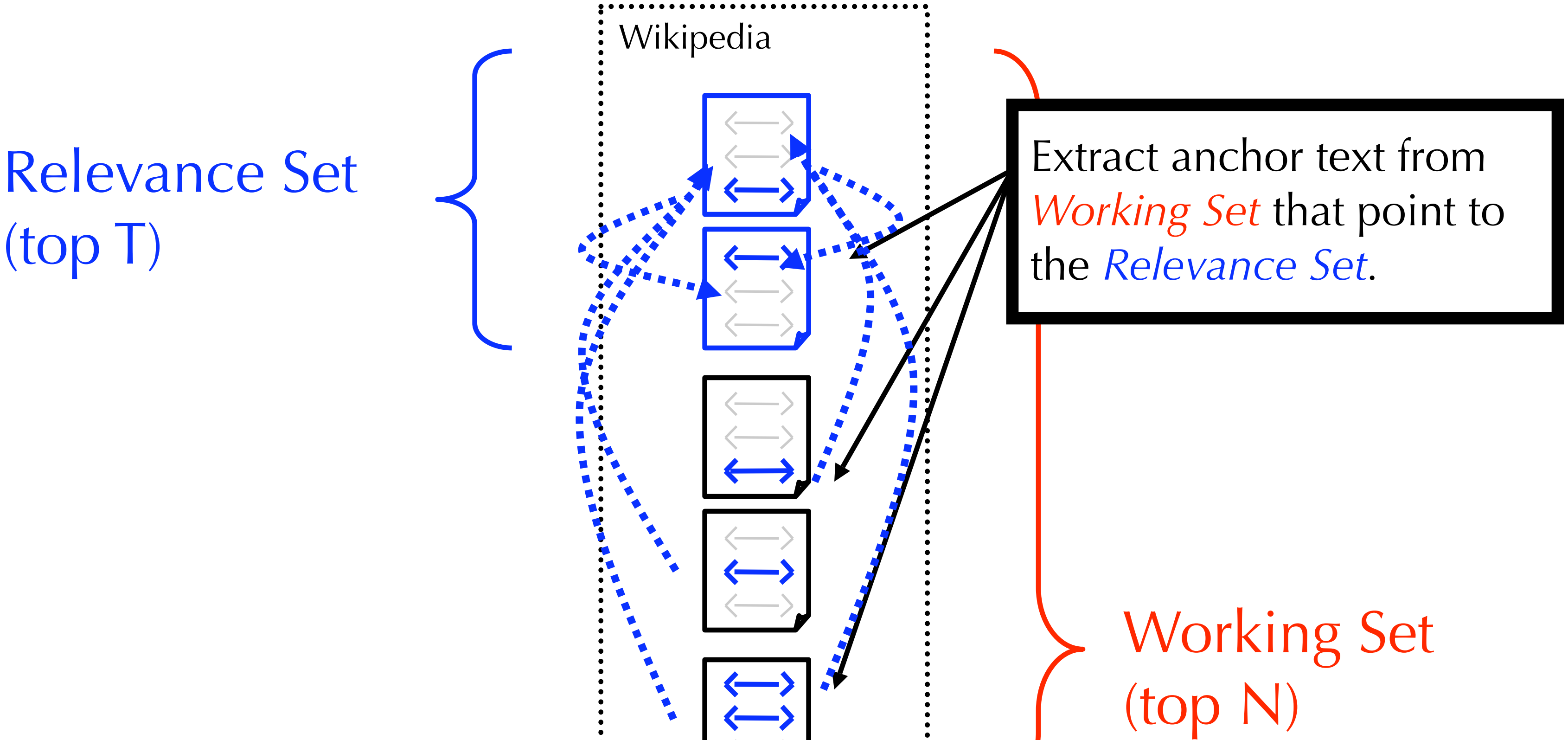
Wikipedia-based PRF



Wikipedia-based PRF



Wikipedia-based PRF



$$Q_{wiki} = \#weight(\ w_1\ \text{a-text}_1\quad w_2\ \text{a-text}_2\ \ \ldots\ \ w_{20}\ \text{a-text}_{20})$$

where $w_i \sim \text{sum}(\ T - \text{rank}(\text{target}(a_i))\ )$

Wikipedia-based PRF

Help us provide free content to the world by *donating today*!

article | discussion | edit this page | history

Help us spread knowledge worldwide. Donate to Wikimedia!

8,570 people have donated

"Wiki is Wonderful !!" – Jim Windgassen

WIKIPEDIA
The Free Encyclopedia

navigation
- Main page
- Contents
- Featured content
- Current events
- Random article

interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

search

Go | Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages

iPod mini

From Wikipedia, the free encyclopedia

The **iPod mini** is a mid-range iPod digital audio player designed and marketed by Apple Inc. It was announced on January 6, 2004, and released on February 20 of the same year; a second-generation version was announced on February 23, 2005. The device operates on Macintosh and Windows PCs, and has limited third-party support for Linux and other Unix workalikes. The iPod mini line was officially discontinued on September 7, 2005 and replaced by the iPod nano line.

The iPod mini retained the touch-sensitive scroll wheel of the third generation iPod; however, instead of the four touch buttons located above the wheel, the buttons were made mechanical beneath the wheel itself—hence the name *click wheel*.[specify] To use one of the four buttons, the user must physically push the edge of the wheel inward over one of the four labels. Like its predecessors, the wheel was developed for Apple by Synaptics. The click wheel is now also used in the fourth, fifth and sixth generation iPods and the iPod nano, from first generation through to third; however, in the nano and 5G iPods onwards, the clickwheel is developed by Apple.

Above the wheel is a monochrome LCD that displays a menu or information about the selected track. Newer-generation iPods have since adopted color displays.

Contents [hide]

Query: "Apple iPod"

Two Pseudo-Relevance Feedback Models

Q 983 “Photography”

Wikipedia PRF	Indri’s Relevance Model
photography	photography
photographer	aerial
depth of field	digital
camera	full
photograph	resource
pulitzer prize	stock
digital camera	free
photographic film	information
photojournalism	art
cinematography	wedding
shutter speed	great

Two Pseudo-Relevance Feedback Models

Q 995 “Ruby on Rails”

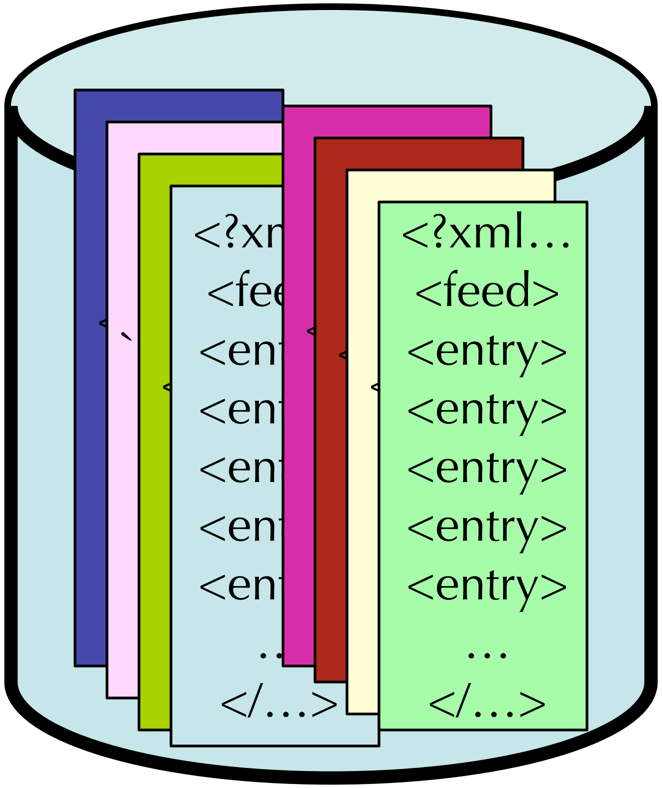
Wikipedia PRF	Indri’s Relevance Model
pokmon	rail
ruby	ruby
pokmon ruby and sapphire	kakutani
pokmon emerald	que
ruby programming language	article
php	weblog
pokmon firered and leafgreen	activestate
pokmon adventures	rubyonrail
pokmon video games	new
standard gauge	develop
tram	dontstopmusic

Two Retrieval Models

- Large Document model
  - Entire Feed is the unit of retrieval
- Small Document model
  - Individual entry is the unit of retrieval
  - Ranked Entries are aggregated into a Feed Ranking

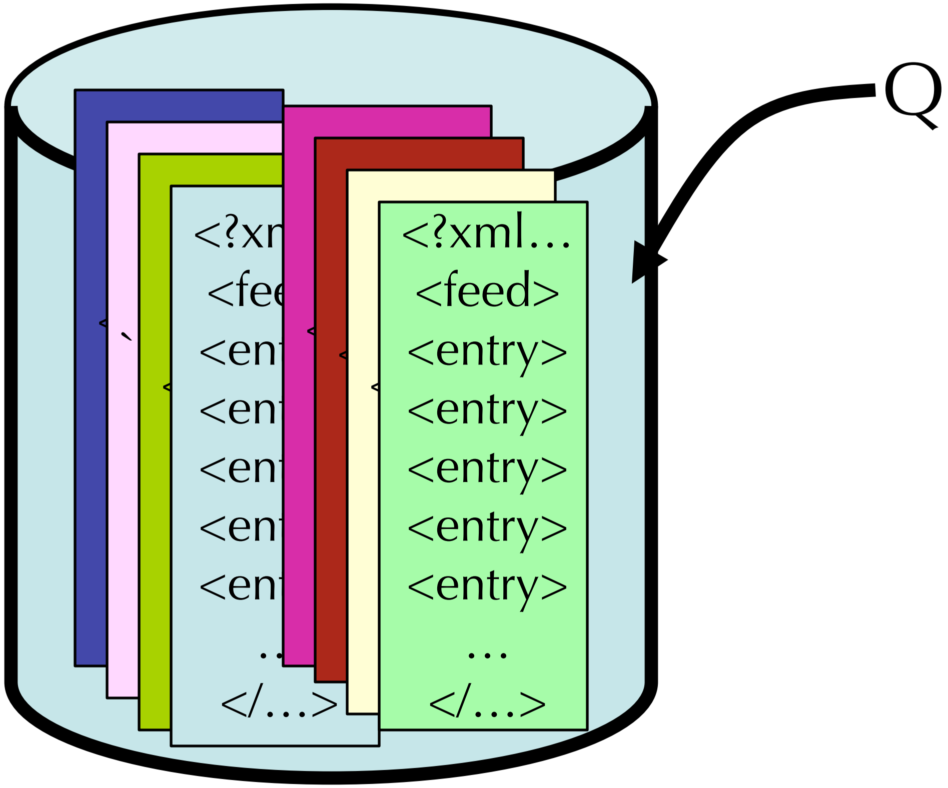
Large Document model

Feed Document
Collection



Large Document model

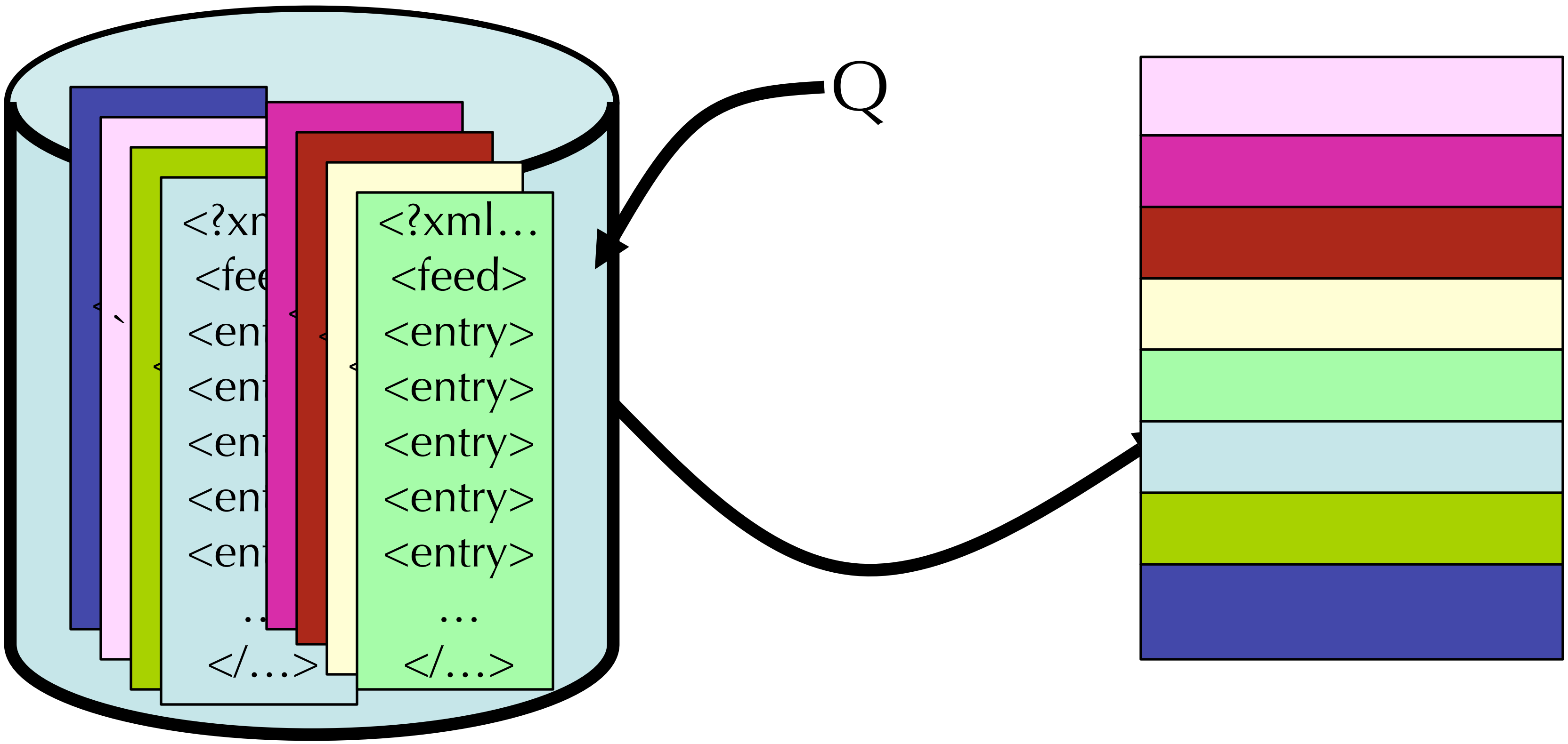
Feed Document
Collection



Large Document model

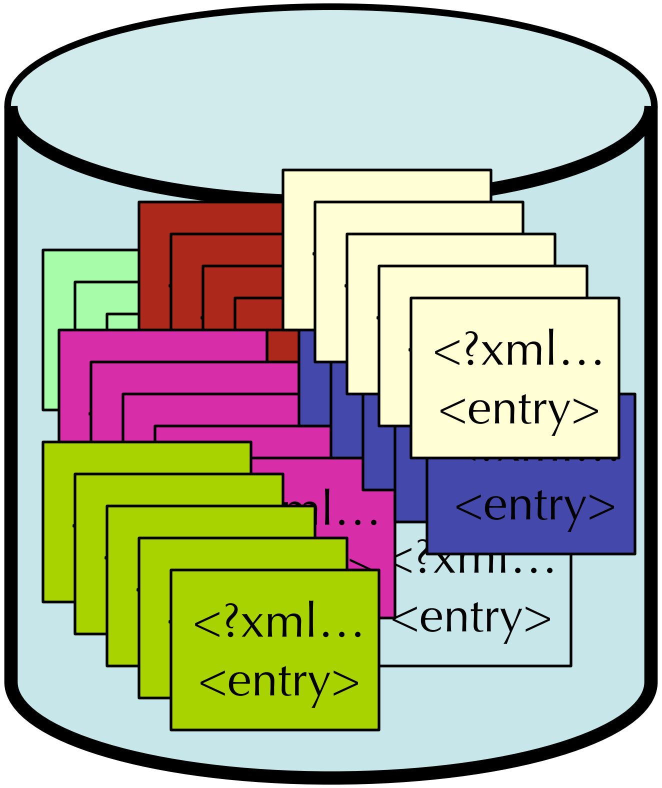
Feed Document Collection

Ranked Feeds

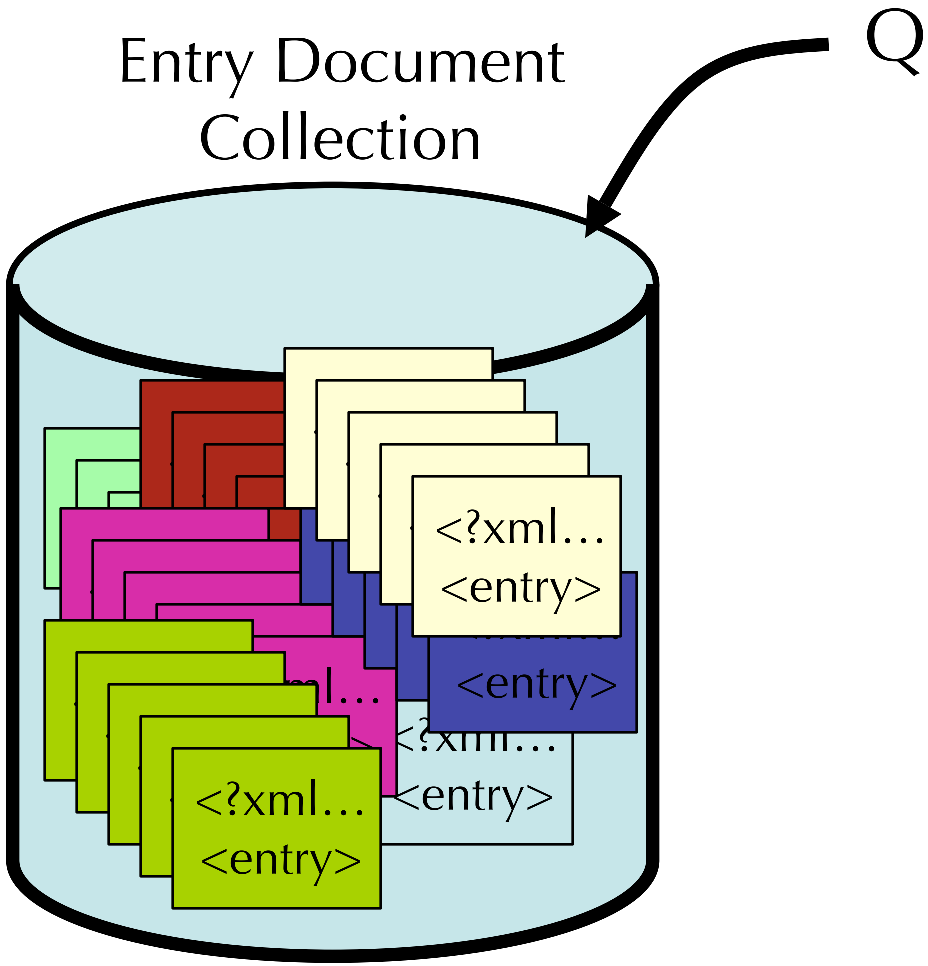


Small Document model

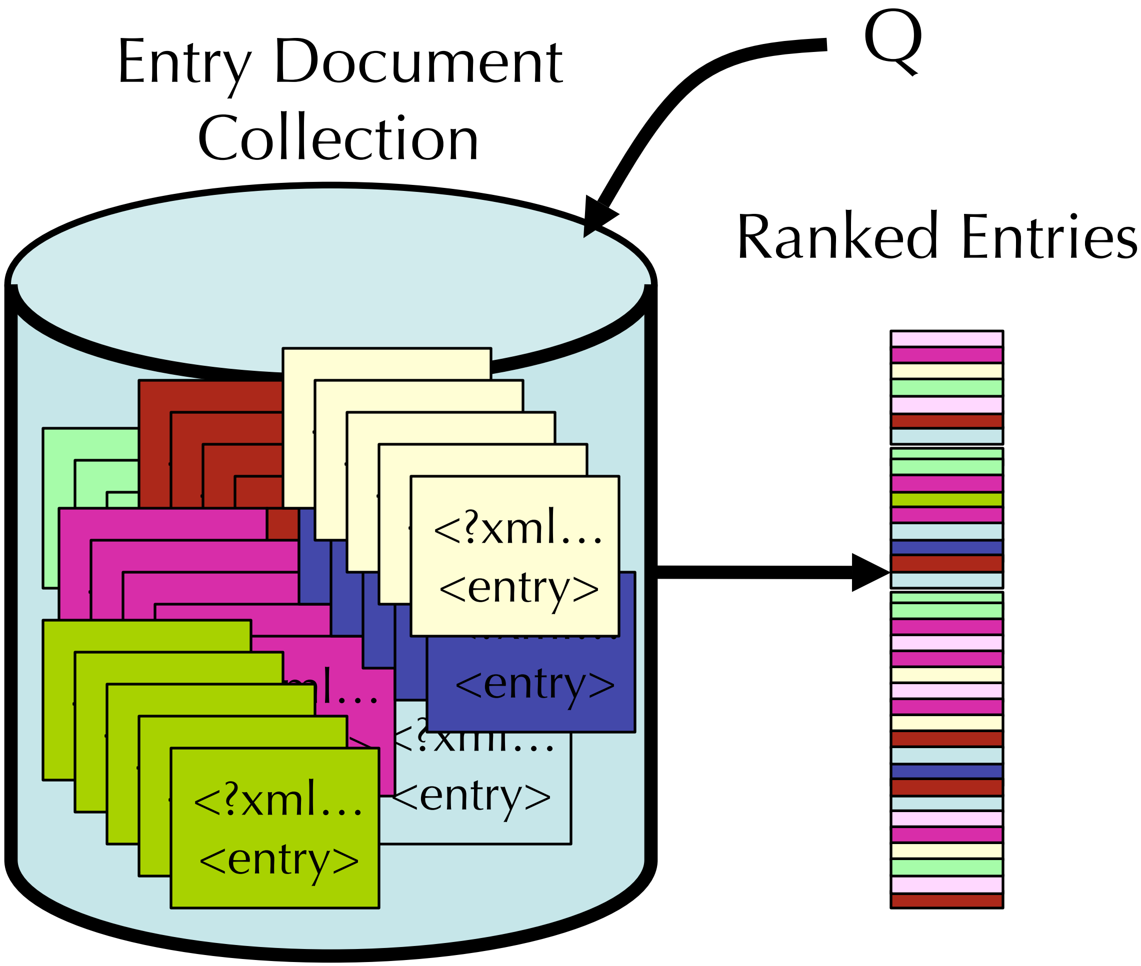
Entry Document
Collection



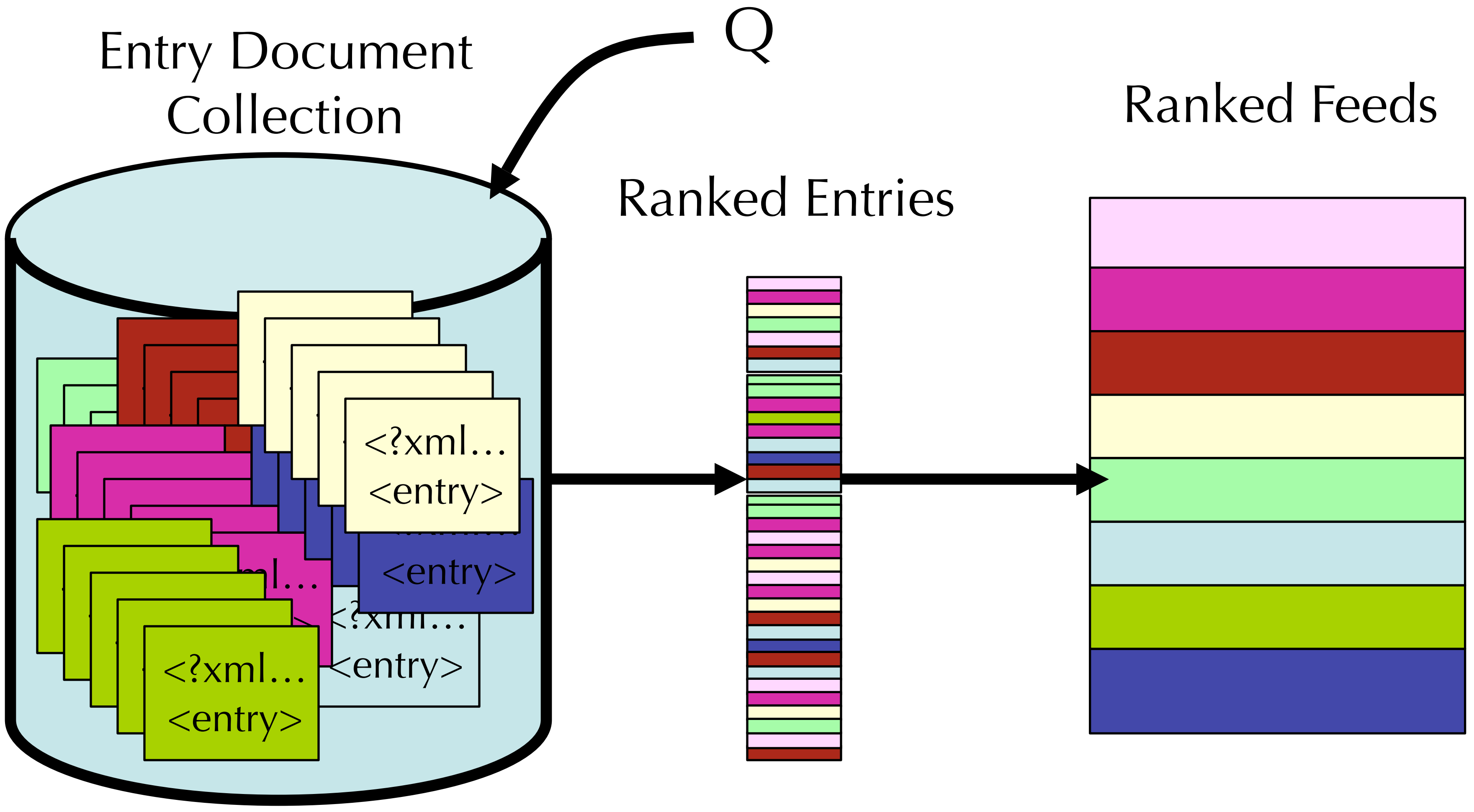
Small Document model



Small Document model



Small Document model



Large Document Model

- Language Modeling retrieval model using the feed document structure.
- Features used for Large Document model:
  - $P(\text{Feed} \mid Q_{\text{feed-title}})$
  - $P(\text{Feed} \mid Q_{\text{entry-text}})$
  - $P(\text{Feed} \mid Q_{RM})$
  - $P(\text{Feed} \mid Q_{\text{wiki}})$

Large Document Model

- Language Modeling retrieval model using the feed document structure.

Large Document Indri Query:

$$\texttt{\#weight(}\quad \lambda_{title}\, DM_{title} \quad \lambda_{entry}\, DM_{entry} \quad \lambda_{RM}\, Q_{RM} \quad \lambda_{wiki}\, Q_{wiki})$$

  - P(Feed | Q_{RM})
  - P(Feed | Q_{wiki})

Small Document Model

- Feed ranking in blog distillation is analogous to resource ranking in federated search

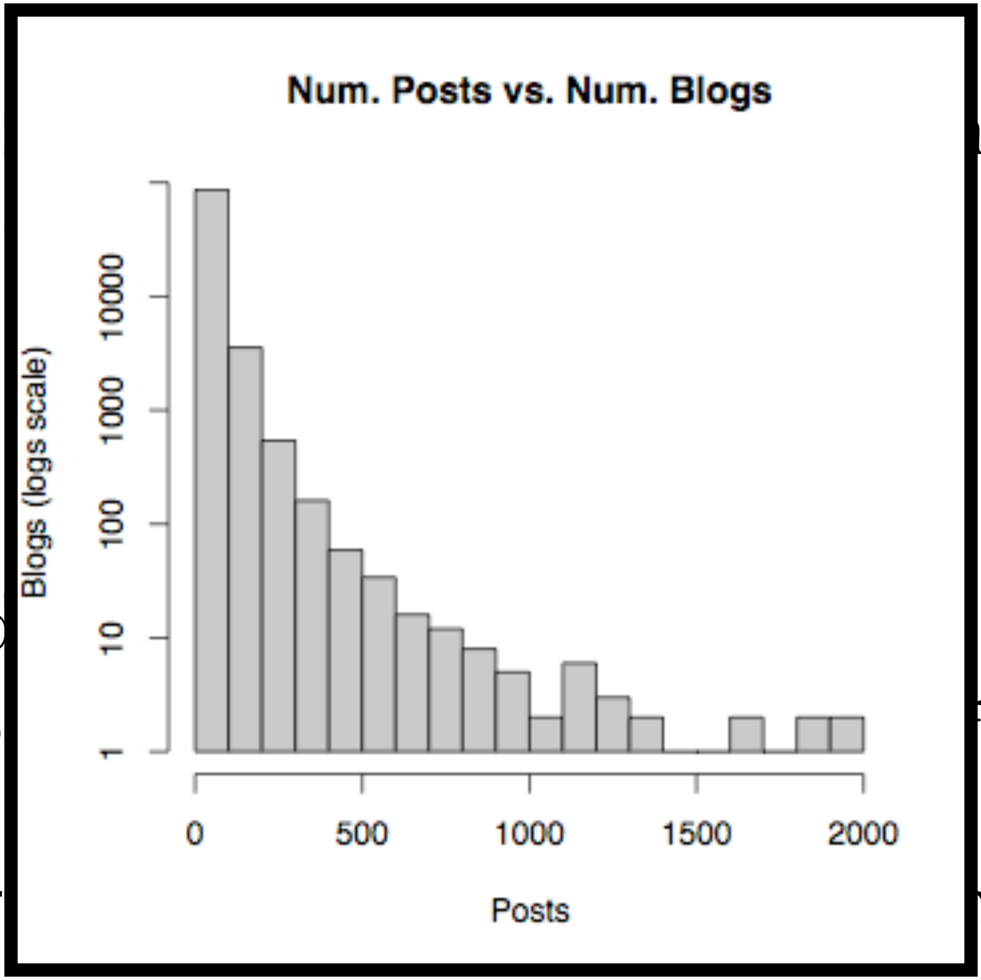
  Feed ~ Resource

  Entry ~ Document

- Created sampled collection
  - Sampled (with replacement) 100 entries from each feed
  - Control for dramatically different feed lengths

Small Document Model

- Feed ra logous to resourc



- Createc
  - Samp from each feed
  - Contr ngths

Small Document Model

Adapted Relevant Document Distribution Estimation (ReDDE) resource ranking.

ReDDE: well-known state-of-the-art federated search algorithm

$$\hat{Rel}_q(j) = \sum_{d_i \in C_j} P(rel|d_i) P(d_i|C_j) N_{C_j}$$

Small Document Model

Adapted Relevant Document Distribution Estimation (ReDDE) resource ranking.

Assuming uniform prior, equal feed length:

$$\hat{Rel}_q(j) = \sum_{d_i \in C_j} P(rel|d_i)P(d_i|C_j)N_{C_j}$$

Small Document Model

Adapted Relevant Document Distribution Estimation (ReDDE) resource ranking.

Assuming uniform prior, equal feed length:

$$\hat{Rel}_q(j) = \sum_{d_i \in C_j} P(rel|d_i)$$

Small Document Model

- Features used in the small document model :
  - P(Feed | $Q_{entry\text{-}text}$)
  - P(Feed | Q_{RM})
  - P(Feed | Q_{wiki})

Small Document Model

- Features used in the small document model :

Small Document Indri Query:

$$\texttt{\#wsum}(\ 1.0\ \texttt{\#combine[entry]}(\\ \texttt{\#weight}(\lambda_{entry} DM_{entry}\\ \lambda_{RM} Q_{RM}\\ \lambda_{wiki}\, Q_{wiki})\ ))$$

Parameter Setting

- Selecting feature weights (λ's) required training data
- Relevance judgments produced for a small subset of the queries (6+2)

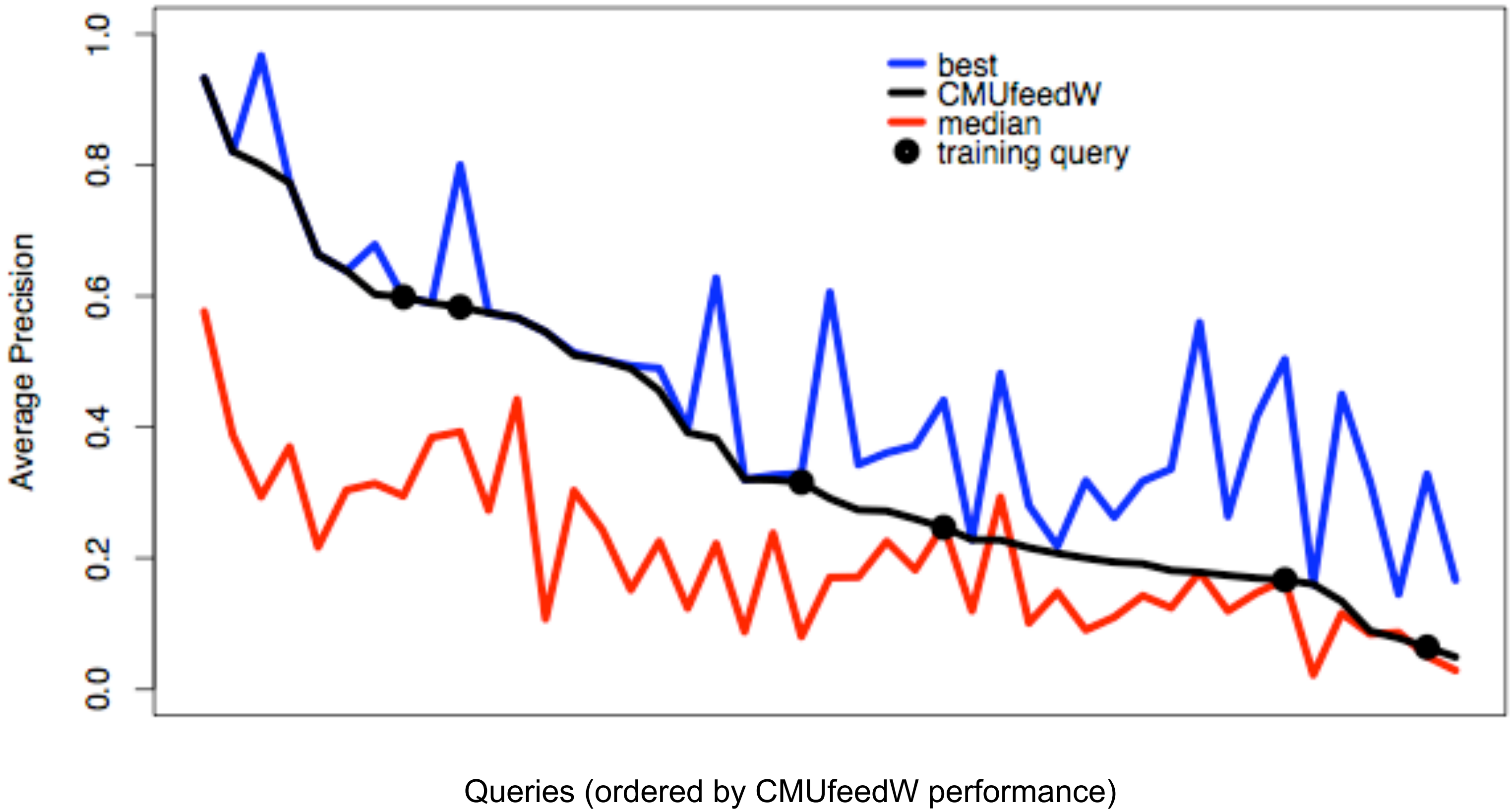
  BOW title query, 50 docs judged/query
- Simple grid search to choose parameters that maximized MAP

Results

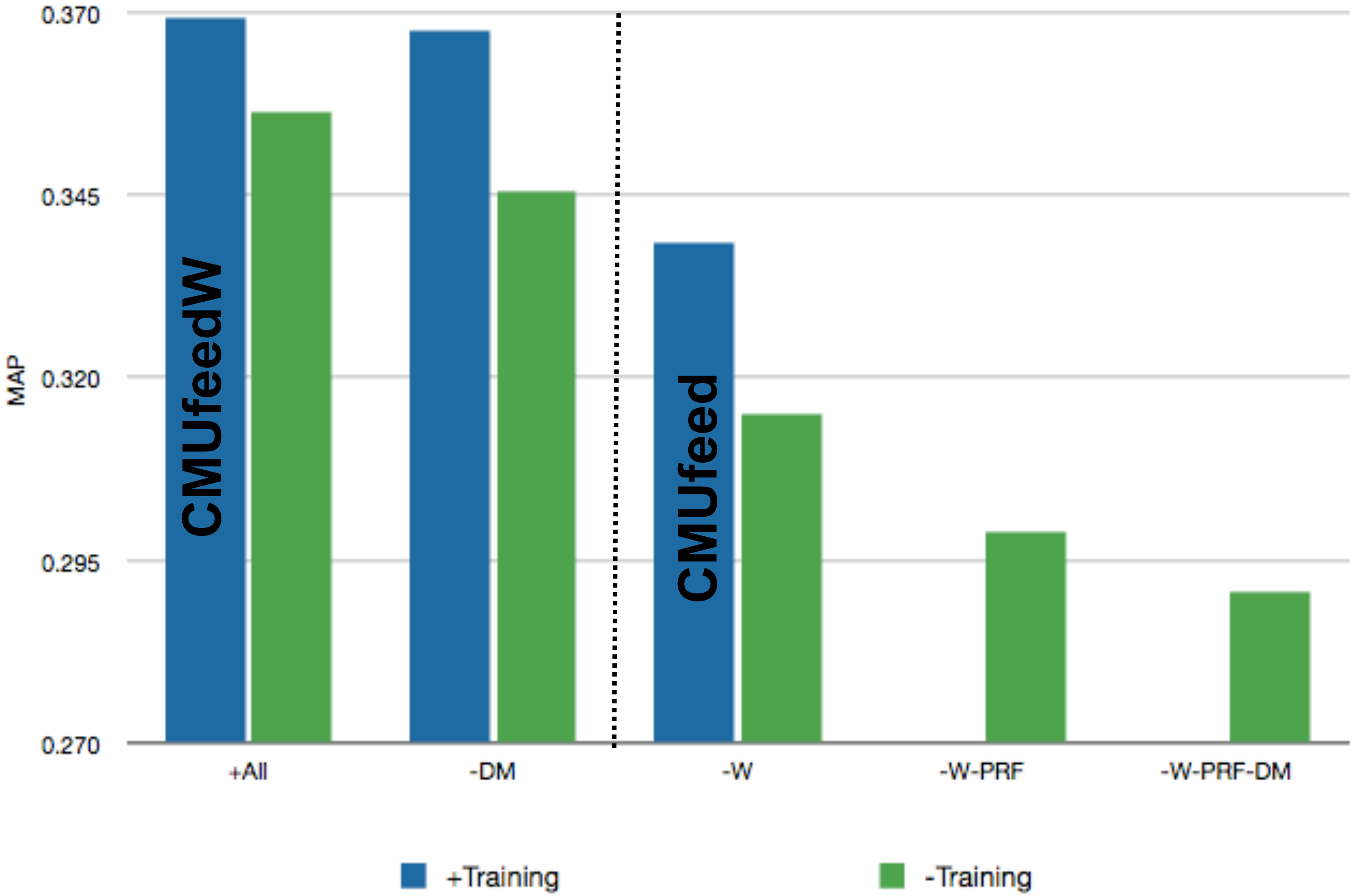
Run	MAP	R-prec	P10
CMUfeed	0.3385	0.4087	0.4733
CMUfeedW	**0.3695**	**0.4245**	**0.5356**
CMUentry	0.2453	0.3277	0.4089
CMUentryW	0.2552	0.3384	0.4267

Our best run:
Wikipedia expansion + Large Doc.

Results


best
CMUfeedW
median
training query
Average Precision
0.0
0.2
0.4
0.6
0.8
1.0
Queries (ordered by CMUfeedW performance)

Feature Elimination



Conclusions

- What worked
  - Preprocessing the corpus & using only feed XML
  - Simple retrieval model with appropriate features
  - Wikipedia expansion
  - (small amount of) training data
- What didn't (… or what we tried without success)
  - Small Document Model (but we think it can)
  - Spam/Splog detection, anchor text, URL segmentation

Thank You

Feature Elimination

	+Training	-Training
+All (**CMUfeedW**)	0.3695	0.3536
-Dependence Model	0.3676	0.3457
-Wiki (**CMUfeed**)	0.3385	0.3152
-Wiki, -Indri's PRF		0.2989
-Wiki, -Indri's PRF, -Dep Model		0.2907