11-442 / 11-642 / 11-742:
Search Engines

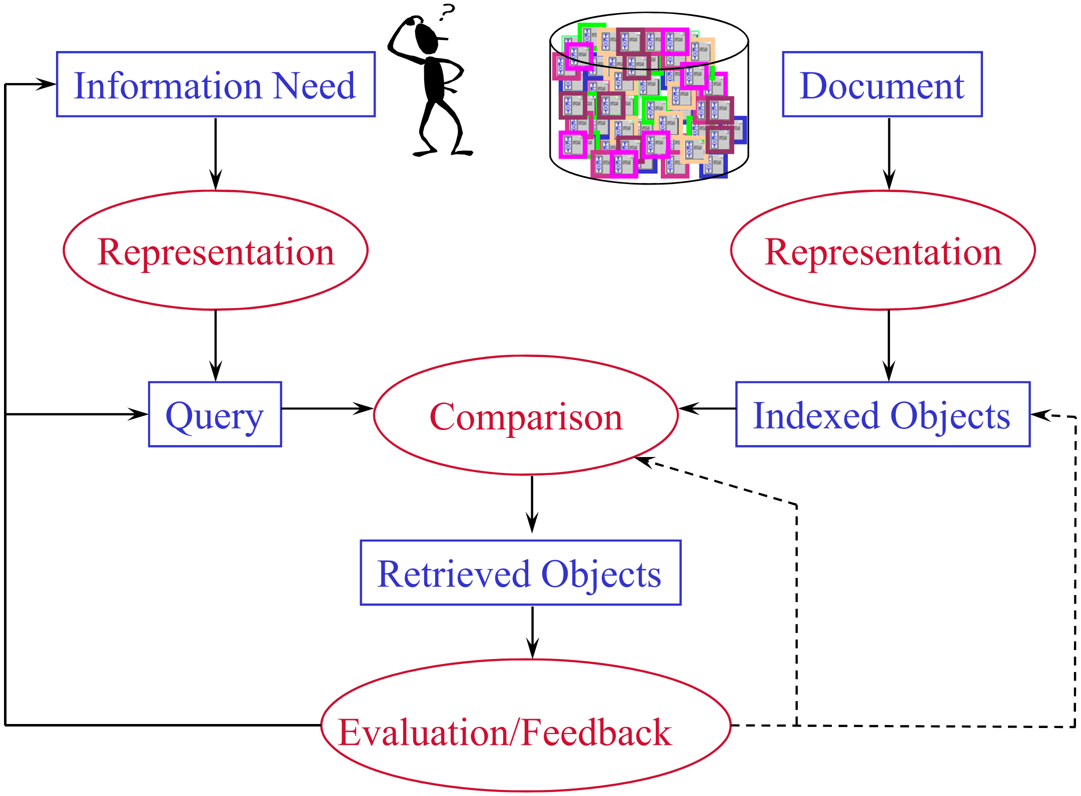
Relevance and
Pseudo Relevance Feedback

Jamie Callan

Carnegie Mellon University

callan@cs.cmu.edu

Overview of Information Retrieval Processes



Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Relevance Feedback

A query only approximates an information need

- Users often start with short queries (poor approximations)
- People can improve queries after seeing relevant and non-relevant documents
  - by adding and removing terms
    - » 'breast cancer' → 'breast cancer treatments'
  - by reweighing terms
    - » 'ipad sales' → '#wand(3.0 ipad 1.0 sales)'

Question: Can a better query be created automatically?

- Machine learning

Relevance Feedback

Overview of relevance feedback (supervised learning)

- Treat the initial query as a classifier
  - In theory: f (document) → {relevant, not relevant}
  - In practice: f (document) → score
- Obtain some labeled data
  - Ask the user to provide labels for some documents
    - » Usually documents retrieved by the initial query
- Use the labeled data to learn a better classifier / query
  - Use your favorite algorithm (many choices)
- Use the learned classifier / query to retrieve documents

Relevance Feedback:
Initial Query and Top Results

Initial query: New space satellite applications

1. Soviets May Adapt Parts of SS-20 Missile For Commercial… ✓, +
2. NASA Hasn't Scrapped Imaging Spectrometer ✓, +
3. When the Pentagon Launches a Secret Satellite, Space …
4. NASA Uses 'Warm' Superconductors For Fast Circuit
5. NASA Scratches Environment Gear From Satellite Plan ✓, +
6. Pentagon Lags in Race To Match the Soviets In Rocket Launchers
7. Rescue of Satellite By Space Agency To Cost $90M
8. Telecommunications Tale of Two Companies ✓, +

✓: Judged by the user **+: Relevant document**

1987-1992 Wall Street Journal, (173,252 documents, 533.2 MB)

Relevance Feedback: A Learned Query

```
#wand (
     2.08  new           15.11  space
    30.82  satellite      5.66  application
     5.99  nasa           5.20  eos
     4.20  launch         3.97  aster
     3.52  instrument     3.45  arianespace
     3.00  bundespost     2.81  ss
     2.79  rocket         2.05  scientist
     2.00  broadcast      1.17  earth
     0.84  oil            0.65  measure)
```

Original Query Terms

New Query Terms

Relevance Feedback:
Top Results Retrieved by Learned Query

Initial query: New space satellite applications

1. NASA Hasn't Scrapped Imaging Spectrometer ✓, +
2. NASA Scratches Environment Gear From Satellite Plan ✓, +
3. Science Panel Backs NASA Satellite Plan, But … +
4. A NASA Satellite Project Accomplishes Incredible Feat …
5. Scientist … Proposes Satellites for Climate Research +
6. Report Provides Support for the Critics Of Using Big Satellites to Study Climate +
7. Arianespace Receives Satellite Launch Pact From Telesat … +
8. Telecommunications Tale of Two Companies ✓, +

✓: Judged by the user **+: Relevant document**

1987-1992 Wall Street Journal, (173,252 documents, 533.2 MB)

Relevance Feedback

Many algorithms can be used for relevance feedback

- E.g., Naïve Bayes, Perceptron (Rocchio), SVM (with linear kernel), …

Typical characteristics

- Model each labeled document as a feature vector
  - Terms and weights
- Use feature vectors of labeled documents to create a new query
  - Select query term candidates from positive documents
  - Set weights based on frequency in + and – documents
  - Use just the *n* query term candidates with the highest weights
    - » To keep the query length / computational costs reasonable

Relevance Feedback

Use the initial query to rank documents

Query: a b

5 Documents

a a e b c a

a b d e f

b a d d b e

a g h b

b i a j

Relevance Feedback

Use the initial query to rank documents

Obtain document labels

- Usually from the user
  - + Relevant
  - – Not relevant

Query: a b

5 Documents	Labels
a a e b c a	+
a b d e f	–
b a d d b e	+
a g h b	–
b i a j	

Relevance Feedback

Use the initial query to rank documents

Obtain document labels

- Usually from the user

Use the labeled data to create a new query

- Use your favorite algorithm

Query: a b

5 Documents	Labels
a a e b c a	+
a b d e f	–
b a d d b e	+
a g h b	–
b i a j	

New Query: 2 a 1 b 1 e 2 d

Relevance Feedback

Key issue: How much training data?

- In the previous example, 2 positive examples, 2 negative examples

How much training data would you expect to be reasonable?

Relevance Feedback: State of the Art

Machine learning is effective if given enough training data

- 10-20 judged documents is good, 100-200 is great

But, people do not enjoy judging documents

- It is boring, and there is no immediate reward
- It is faster to reformulate the query

Typically, relevance feedback is only used in situations where it is practical to expect many judged documents

- E.g., review of legal documents (high Recall retrieval)
  - People will look at hundreds / thousands of documents
  - Judging a dozens or hundreds of documents is a small additional burden

Relevance Feedback: State of the Art

Relevance feedback works

- Improved queries can be learned from judged documents

But, many popular search engines don’t use relevance feedback

- Most people don’t like giving relevance judgments
- Search providers don’t like the risk of doing something stupid
  - If many documents are judged, results are very reliable
  - If few documents are judged, results are highly variable

Major open problems

- Stability and consistency (e.g., don’t ever be stupid)
- Inferring relevance from implicit feedback

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Pseudo-Relevance Feedback
(Automatic Relevance Feedback)

Relevance feedback is supervised machine learning

Pseudo relevance feedback is unsupervised machine learning

- Treat the initial query as a classifier
- Use it to label some data
- Use the labeled data to generate a better classifier
  - Noisy training data

Typically there is just one iteration of this cycle

- Additional iterations increase risk but do not increase reward

Pseudo-Relevance Feedback
(Automatic Relevance Feedback)

Typically…

- Use the original, unexpanded query to retrieve documents
- Assume that the top N documents are relevant, e.g., N=50
  - This is the positive training data
  - Some documents won't be relevant, but the goal is to learn vocabulary patterns
- If non-relevant documents are needed, select them from lower in the ranking
  - This is the negative training data
- Apply a relevance feedback algorithm
  - Term weighting and term selection
- Use the modified query to retrieve documents

Relevance Feedback in the Vector Space: The Rocchio Algorithm

Goal: Make the query more similar to relevant documents

New Query: A weighted average of original query vector, the relevant document vectors, and non-relevant document vectors

$$Q_{expanded} = Q_{original} + \alpha \underbrace{\frac{1}{|R|} \sum_{d \in R} \vec{d}}_{\text{Average of Rel docs}} - \beta \underbrace{\frac{1}{|NR|} \sum_{d \in NR} \vec{d}}_{\text{Average of Non-rel docs}}$$

Notation

- **R and NR:** Judged relevant and non-relevant documents
- **d:** A document vector, e.g., terms with lnc.ltc weights: $\dfrac{\log(tf)+1}{\sqrt{\sum(\log(tf)+1)^2}}$
- ***α* and *β*:** Weights on Relevant and Non-Relevant judgments

Relevance Feedback in the Vector Space: The Rocchio Algorithm

Goal: Make the query more similar to relevant documents

New Query: A weighted average of original query vector, the relevant document vectors, and non-relevant document vectors

$$Q_{expanded} = Q_{original} + \alpha \underbrace{\boxed{\frac{1}{|R|}\sum_{d \in R} \vec{d}}}_{\textbf{Average of Rel docs}} - \beta \underbrace{\boxed{\frac{1}{|NR|}\sum_{d \in NR} \vec{d}}}_{\textbf{Average of Non-rel docs}}$$

Variations:

- Different values of α and β
- Vector length (number of terms added to the query)
- Which documents are used for training (all, best, uncertain, etc)

Relevance Feedback in the Vector Space: The Rocchio Algorithm

Average of Rel docs:

$$\frac{1}{|R|}\sum_{d\in R}\vec{d}$$

$$\begin{array}{ccccccccccc}
\text{Vocabulary} & & d_1 & & d_2 & & d_3 & & d_4 & & \text{average} \\
\begin{array}{|c|}\hline t_1\\ t_2\\ t_3\\ t_4\\ t_5\\ t_6\\ \dots\\ t_n\\ \hline\end{array} & \dfrac{1}{4}\left[\vphantom{\begin{array}{c}1\\1\\1\\1\\1\\1\\1\\1\end{array}}\right. & \begin{array}{|c|}\hline 0.00\\ 0.60\\ 0.45\\ 0.10\\ 0.00\\ 0.07\\ \dots\\ 0.24\\ \hline\end{array} & + & \begin{array}{|c|}\hline 0.02\\ 0.50\\ 0.39\\ 0.12\\ 0.02\\ 0.00\\ \dots\\ 0.29\\ \hline\end{array} & + & \begin{array}{|c|}\hline 0.00\\ 0.70\\ 0.53\\ 0.03\\ 0.04\\ 0.03\\ \dots\\ 0.31\\ \hline\end{array} & + & \begin{array}{|c|}\hline 0.04\\ 0.60\\ 0.12\\ 0.19\\ 0.00\\ 0.13\\ \dots\\ 0.28\\ \hline\end{array} & \left.\vphantom{\begin{array}{c}1\\1\\1\\1\\1\\1\\1\\1\end{array}}\right] = & \begin{array}{|c|}\hline 0.0150\\ 0.6000\\ 0.3725\\ 0.1100\\ 0.0150\\ 0.0575\\ \dots\\ 0.2800\\ \hline\end{array}
\end{array}$$

Pseudo-Relevance Feedback
(Automatic Relevance Feedback)

Use the initial query to rank documents

Query: a b

Documents

a a e b c a

a b d e f

b a d d b e

a g h b

b i a j

Pseudo-Relevance Feedback
(Automatic Relevance Feedback)

Use the initial query to rank documents

Use the top *n* documents to identify candidate terms

- E.g., using Rocchio

Query: a b

Documents

a a e b c a

a b d e f

b a d d b e

a g h b

b i a j

Candidate Terms

a	0.6000
d	0.3725
b	0.2800
c	0.1100
e	0.0575
f	0.0150
g	0.0150
h	0.0132

Pseudo-Relevance Feedback (Automatic Relevance Feedback)

Use the initial query to rank documents

Use the top *n* documents to identify candidate terms

- E.g., using Rocchio

Use the top *m* terms to create an improved query

- α=1 and β=0 in this example
- *m*=4 in this example

Query: a b

Documents

a a e b c a

a b d e f

b a d d b e

a g h b

b i a j

Candidate Terms

a	0.6000
d	0.3725
b	0.2800
c	0.1100
e	0.0575
f	0.0150
g	0.0150
h	0.0132

New Query: 1.6 a 1.28 b 0.3725 d 0.11 c

Pseudo-Relevance Feedback (Automatic Relevance Feedback)

Use the initial query to rank documents

Use the top *n* documents to identify candidate terms

- E.g., using Rocchio

Use the top *m* terms to create an improved query

Use the improved query to retrieve documents

New Query: 0.6 a 0.3725 d
0.28 b 0.11 c

Documents

a a e b c a

b a d d b e

a d d a

a d d c

a b d e f

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Relevance Feedback in Okapi

Okapi uses a typical pseudo relevance feedback architecture

1. The initial query $Q_{original}$ retrieves the top-ranked n documents
2. Extract potential expansion terms from top n documents
3. Calculate a score for each potential expansion term
4. Use the top m terms to create a new query $Q_{learned}$
5. $Q_{learned}$ retrieves a new (better) set of documents

Relevance Feedback in Okapi

Candidate terms: Any term in any relevant document

Term weight

$$
\begin{aligned}
w_{\text{expansion}}(t) &= P(t \mid R)\, w_t && \textbf{w}_\textbf{t}\textbf{: RSJ weight} \\
&\approx \frac{rdf_t}{|R|} w_t && \textbf{MLE estimate of P(t|R)} \\
&\propto rdf_t\ w_t && \textbf{Drop the constant |R|} \\
&= rdf_t \left(\log \frac{N - df_t + 0.5}{df_t + 0.5} \right) && \textbf{Showing the RSJ weight}
\end{aligned}
$$

rdf_t: # of relevant docs containing t

R: Set of relevant docs

(Robertson and Zaragoza, 2009)

Relevance Feedback in Okapi

Select the top *n* documents

- E.g., n=10-30

Select the top *m* terms

- E.g., m=10-30

Use the top *m* terms to form an unweighted bag-of-words query

- Initial query: prostate cancer treatments
- Learned query: metastasize surgery erectile urologic impotence prognosis prostate antigen psa prostatectomy

(Walker et al, 1997; Robertson and Zaragoza, 2009)

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Indri's Pseudo-Relevance Feedback: Overview

Indri uses a typical pseudo relevance feedback architecture

1. The initial query $Q_{original}$ retrieves the top-ranked *n* documents
2. Extract potential expansion terms from top *n* documents
3. Calculate a score for each potential expansion term
4. Use the top *m* terms to create an expansion query $Q_{learned}$
5. Combine $Q_{original}$ and $Q_{learned}$ to create $\mathrm{Q}_{\mathrm{expanded}}$
6. $\mathrm{Q}_{\mathrm{expanded}}$ retrieves a new (better) set of documents

Indri Relevance Feedback: Potential Expansion Terms

Any document feature can be a potential expansion term

Unigrams (terms) are the most common choice

- Bigrams and trigrams ("phrases") are also possible
  - E.g., white_house, deep_learning, new_york_city, …
  - Higher computational cost, because there are more of them
  - Usually only a small additional value

Indri Relevance Feedback: Scoring Potential Expansion Terms

For each candidate expansion term t, calculate *p(t|I)*

$$p(t \mid I) = \textstyle\sum_d p(t \mid d)\, p(d \mid I)$$

A relevance model

$$= \frac{\sum_d p(t \mid d)\, p(I \mid d)\, p(d)}{p(I)}$$

Apply Bayes Rule to *p(d|I)*

$$\propto \textstyle\sum_d p(t \mid d)\, p(I \mid d)\, p(d)$$

p(I) is constant, so drop it

$$\propto \textstyle\sum_d \underbrace{p(t \mid d)}\, \underbrace{p(I \mid d)}$$

Assume *p(d)* is uniform

Indri score for d using $Q_{original}$

$$p(t \mid d) = \frac{tf_{t,d} + \mu\, p_{MLE}(t \mid C)}{length(d) + \mu}$$

The usual *p(t|d)* calculation

Indri default for PRF is $\mu = 0$

(Metzler, 2007)

Indri Relevance Feedback: Scoring Potential Expansion Terms

The original Indri score for expansion terms does not include a penalty for frequent terms

- Thus, it can select words that are ‘almost stopwords’
- E.g., ‘announced’ and ‘yesterday’ in the Wall Street Journal

A later version corrects this problem

$$\begin{aligned} p(t \mid I) &\propto \sum_d p(t \mid d) p(I \mid d) \log \frac{1}{p(t \mid C)} \\ &\propto \sum_d p(t \mid d) p(I \mid d) \boxed{\log \frac{length_{terms}(C)}{ctf_t}} \end{aligned}$$

A form of idf

Use this in HW3

(Metzler, 2007)

Indri Relevance Feedback: Create an Expanded Query

The original query is $Q_{original}$

The learned query is $Q_{learned}$

```
#wand ( p(t_1 | I)   t_1
        p(t_2 | I)   t_2
        p(t_3 | I)   t_3
          …)
```

The expanded query is

$Q_{expanded}$ = #wand (w $Q_{original}$ (1-w) $Q_{learned}$)

Indri Relevance Feedback: Create an Expanded Query

The expanded query is

$Q_{expanded}$ = #wand (w $Q_{original}$ (1-w) $Q_{learned}$)

$Q_{original}$: a b c

$Q_{expanded}$: #WAND (0.5 #AND (a b c)
    0.5 #WAND (0.72 g 0.63 a 0.59 c 0.43 e …))

$Q_{original}$: #AND (a #NEAR (b c) #OR (d e))

$Q_{expanded}$: #WAND (0.6 #AND (a #NEAR (b c) #OR (d e))
    0.4 #WAND (0.81 g 0.74 a 0.63 d 0.41 e …))

Indri Relevance Feedback: Parameters

Parameters are set heuristically

- **fbdocs:** The number of judged documents
- **fbterms:** The number of terms to add to the query
  - Indri's default is 10
  - It is not unusual to use many more if fbdocs is high
- ***μ*:** The smoothing weight to use for new terms [0 - ∞]
  - Indri's default is 0
- **w:** The amount of weight to place on the original query [0 - 1]
  - Indri's default is 0.5

Query Expansion in Indri

Topic 523: facts about the five main clouds

Original query: facts five main clouds

Expansion terms:

0.321 clouds	0.116 cloud	0.070 weather
0.050 main	0.049 earth	0.046 facts
0.045 space	0.038 water	0.033 atmosphere
0.032 ice	0.029 radiation	0.029 jupiter
0.029 rain	0.027 planet	0.023 jupiters
0.017 atmospheric	0.016 sky	0.011 wavelength
0.011 hydrogen	0.010 infrared	

Effect: MAP 0.06 → 0.29 **(+350%)**

Query Expansion in Indri

Topic 509: steroids; what does it do to your body

Original query: steroids your body

Expansion terms:

0.152 steroids	0.124 body	0.079 effects
0.078 drug	0.071 drugs	0.062 treatment
0.051 steroid	0.050 side	0.046 anabolic
0.037 muscle	0.037 skin	0.035 doctor
0.034 disease	0.030 blood	0.023 taking
0.022 hormones	0.021 hormone	0.019 symptoms
0.019 testosterone	0.012 inhaled	

Effect: MAP 0.20 → 0.46 **(+126%)**

Query Expansion in Indri

Topic 522: how is water supplied to the mojave desert region?

Original query: water supplied mojave desert region

Expansion terms:

0.197 desert	0.181 water	0.115 valley
0.077 california	0.063 san	0.056 mojave
0.042 619	0.033 area	0.032 park
0.030 hesperia	0.028 victorville	0.026 bernardino
0.024 basin	0.016 natural	0.016 adelanto
0.016 land	0.015 canyon	0.014 sierra
0.010 geological	0.009 nevada	

Effect: MAP 0.27 → 0.11 **(–61%)**

Query Expansion in Indri

Topic 538: fha

Original query: fha

Expansion terms:

0.163 mortgage	0.159 fha	0.120 hud
0.104 home	0.098 loan	0.058 loans
0.050 housing	0.048 mortgages	0.033 insurance
0.031 financing	0.027 payment	0.020 conventional
0.016 lender	0.016 purchase	0.015 insured
0.012 lenders	0.011 borrower	0.007 borrowers
0.007 refinance	0.006 adjustable	

Effect: MAP 0.31 → 0.22 **(–30%)**

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Parameters: How Many Documents?

There is no good theory about how many documents to use for pseudo relevance feedback

The most common solution is the top n documents

- n = 10, 50, 100, …
- n is based on a guess about the quality of the initial retrieval
  - The number of relevant documents
  - The quality of the documents

Treat it as a collection-dependent parameter to be tuned

Parameters:
How Many Terms is Enough?

The ‘right’ number of expansion terms is related to the number of documents used for query expansion

- More documents → More evidence for selecting terms
- More documents → A larger candidate vocabulary

5-50 terms is common

Parameters:
How Many Terms is Enough?

Most systems expand by a static number of terms

- i.e., use the same number of terms for all queries
- Different systems use different numbers of expansion terms

Research shows that these systems are well-tuned

- No other static number of terms would provide better results

The best number of terms varies by query

- Picking the right number yields significant improvements

An interesting research topic

(Billerbeck and Zobel, 2003; Ogilvie, et al., 2009)

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

The Corpus

Usually the initial and final query are run on the same corpus

- Learn the vocabulary patterns in this corpus

The initial and final query can be run on different corpora

- Done if the initial retrieval is likely to produce a noisy result
- E.g., web search, Twitter search, …

Example

- Run the initial query on wikipedia
- Generate high-quality expansion terms
- Run the expanded query on the web corpus

Does Query Expansion Improve Accuracy?
Wikipedia-Based Query Expansion

ClueWeb09 (500 million documents)

Method	MAP	P@10
Indri	0.0751	0.3120
Indri + wikipedia PRF (initial exact match)	0.1399	0.4520
Indri + wikipedia PRF (initial best match)	0.1169	0.3980

ClueWeb09 Category B (50 million document subset)

Method	MAP	P@5
Indri + spam filter + SDM	0.1135	0.3250
Indri + spam filter + SDM + wikipedia PRF	0.1482	0.4875

(Nguyen and Callan, 2011)
(Bendersky, Fisher, and Croft, 2011)

The Corpus

Wikipedia-based query expansion is widely done

…but you can use any high-quality corpus

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Does Query Expansion Improve Accuracy?

Two query expansion methods on four TREC datasets

Data	Method	MAP	P@10	Data	Method	MAP	P@10
T-1,2	Indri	0.1818	0.4443	T-7	Indri	0.1890	0.4200
	PRF$_1$	0.2419	0.4913		PRF$_1$	0.2175	0.4320
	PRF$_2$	0.2406	0.5363		PRF$_2$	0.2169	0.4480
T-8	Indri	0.2013	0.3960	wt10g	Indri	0.1741	0.2760
	PRF$_1$	0.2361	0.4160		PRF$_1$	0.1829	0.2630
	PRF$_2$	0.2268	0.4340		PRF$_2$	0.1946	0.2960

(Collins-Thompson and Callan, 2007)

Does Query Expansion Improve Accuracy?
Wikipedia-Based Query Expansion

ClueWeb09 (500 million documents)

Method	MAP	P@10
Indri	0.0751	0.3120
Indri + wikipedia PRF (initial exact match)	0.1399	0.4520
Indri + wikipedia PRF (initial best match)	0.1169	0.3980

ClueWeb09 Category B (50 million document subset)

Method	MAP	P@5
Indri + spam filter + SDM	0.1135	0.3250
Indri + spam filter + SDM + wikipedia PRF	0.1482	0.4875

(Nguyen and Callan, 2011)
(Bendersky, Fisher, and Croft, 2011)

Does Query Expansion Improve Accuracy?

PRF is viewed as 'recall enhancing'

- Adding more query terms allows more documents to match

PRF improves MAP consistently

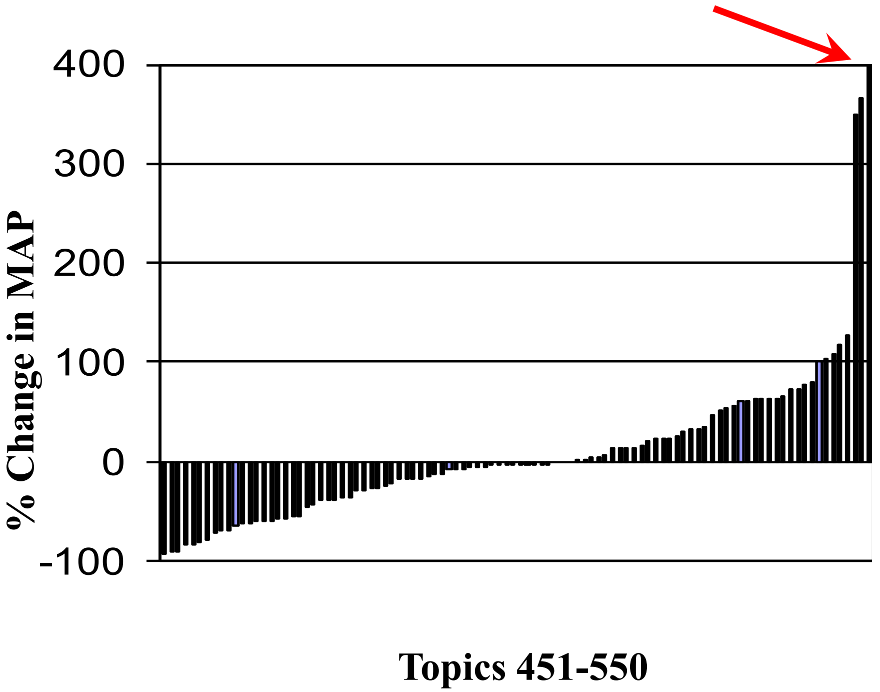
- 15-20% improvement is typical

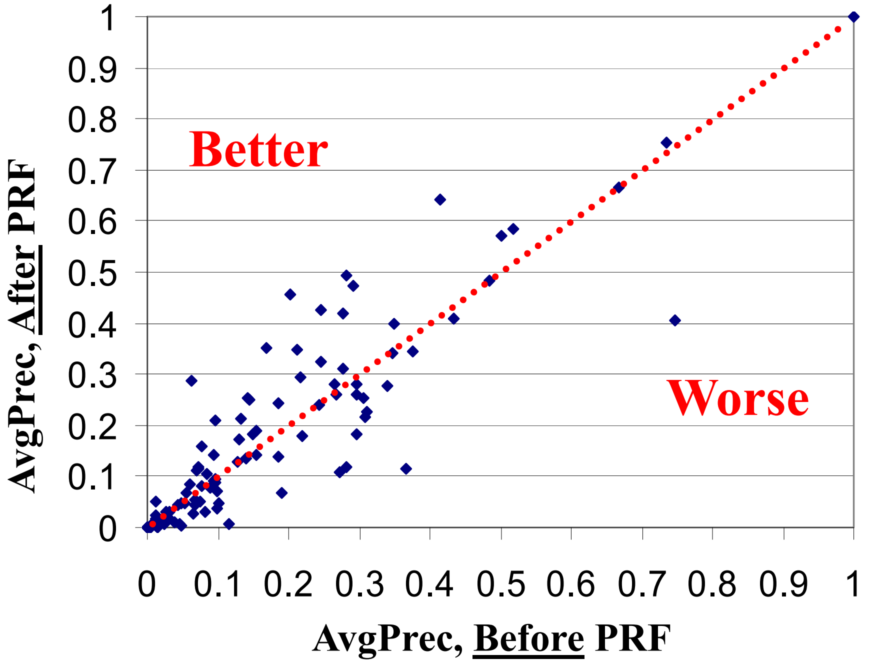
PRF may or may not improve P@10 or NDCG@10

- Very sensitive to the quality of the initial retrieval

Does Query Expansion Improve Accuracy?

+4,000%





TREC Topics 451-500

Does Query Expansion Improve Accuracy?

Average effectiveness improves

- But, many queries are harmed

Query expansion is used when Recall is important, or when average performance matters

- E.g., legal retrieval, TREC, research papers, …

Used less often or very carefully for interactive systems

- Too many queries are hurt
- People remember search engine mistakes more than successes
- Most needed on ‘hard’ queries, but most effective on ‘easy’ queries

Outline

Relevance feedback

Pseudo relevance feedback

- Vector space (Rocchio)
- Okapi BM25
- Inference networks (Indri)
- Parameter values
- Corpus
- Effect on retrieval accuracy

Pseudo Relevance Feedback: Summary

We covered three pseudo relevance feedback methods

- Vector space (Rocchio)
- Okapi BM25
- Indri

All of them look similar

- Select terms from the n top-ranked documents
- Weight terms based on something that looks like tf or tf.idf
- Select the best m terms
- Form an expansion query that uses terms and weights
- Use the expansion query to retrieve documents

Pseudo Relevance Feedback: Summary

Pseudo relevance feedback is unsupervised machine learning

- Treat the initial query as a classifier
- Use it to label some data
  - The top-ranked documents
- Use the labeled data to generate a better classifier

Typically there is just one iteration of this cycle

- Additional iterations increase risk but do not increase reward

Pseudo Relevance Feedback: Summary

Pseudo relevance feedback is widely used by researchers

- It is rare to see a TREC system that does not use PRF

Why?

- Typically researchers focus on average-case analysis
- PRF reliably improves MAP
  - MAP is the most common evaluation metric due to its reliability for evaluating which system is best

Why Isn't Pseudo-Relevance Feedback More Common?

The main unsolved problem is how to reduce variance

- Query expansion improves MAP by about 20% on average
- But, you may really annoy 1/3 of your users

Typical research directions

- Only expand if the initial retrieval results seem high quality
- Only expand if the top N documents appear homogeneous
  - Thus, they agree about which terms are important
- Generate several candidate expansion queries
  - Use only terms that appear in most of the candidate expansions
- PRF in neural retrieval models
- …

For Additional Information

- M. Bendersky, D. Fisher, and W. B. Croft. UMass at TREC 2010 Web Track: Term dependence, spam filtering, and quality bias. In *TREC 2010 Conference Proceedings (TREC 2010)*. 2011.
- K. Collins-Thompson and J. Callan. "Estimation and use of uncertainty in pseudo-relevance feedback." In *Proceedings of the Thirtieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Amsterdam. 2007.
- D. A. Metzler, Beyond bags of words: Effectively modeling dependence and features in information retrieval. PhD dissertation, University of Massachusetts. 2007.
- D. Nguyen and J. Callan. Combination of evidence for effective web search. In *TREC 2010 Conference Proceedings (TREC 2010)*. 2011.
- S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4). 2009.
- G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley. 1989.
- S. Walker, S.E. Robertson, M. Boughanem, G.J.F. Jones, K. Sparck Jones. “Okapi at TREC-6: Automatic ad hoc, VLC, routing, filtering, and QSDR. TREC-6 Proceedings. 1997.