

CIS 833 – Information Retrieval and Text Mining

Lecture 20

Relevance Feedback and Query Expansion

November 10, 2015

Credits for slides: Allan, Arms, Manning, Lund, Noble, Page.

Planning

- PageRank implementation: last assignment (due Dec 1st)
- Final exam: November 19th or December 3rd ?
- Project presentation: finals week (during the exam time)
- Project report: by the end of the finals week

Wikipedia Corpus

Pages:

```
<page><title> Page_Title </title><text> Page_body_goes_here </text></page>
```

Page_Titles are case-insensitive

Links:

```
[[Name of other article]]
[[Name of other article|Label]]
[[Name of other article#Section]]
```

Not a link:

```
[[Text:Text]] – ignore
```

(also ignore titles that contain “.”, e.g. <title>Text:Text</title>)

```
<page> <title>Scotland</title> <text>"For mair airticles
adae wi fowk, places an things in Scotland tak a keek at
[:category:Scotland]]." {{Infobox Kintra |native_name
= Scotland <small>([[Scots Inglis|Inglis]]&nbsp;&nbsp;&nbsp;[[Scots
leid|Scots]])</small><br /><!-- -->"Alba" <small>([[Scots Gaelic
leid|Gaelic]])</small> |conventional_long_name = |
common_name = Scotland |image_flag =
Flag of Scotland.svg |image_coat = Royal coat of
arms of Scotland.svg |naitional_motto = <small>"[[In My
Defens God Me Defend]]" ([[Scots language|Scots]]) (Often
shown abbreviated as ""IN DEFENS""</SMALL> |image_map
= Scotland Location UK.PNG |offeercial_leid = Nane;
[[Inglis leid|Inglis]] "de facto"; [[Gaelic leid|Gaelic]] haes some
legal status; [[Scots leid|Scots]] gien some recogneetion forby |
map_caption = {{map_caption|location_color=orange|
region=the [[United Kingdom]]|region_color=camel}} |caipital
= [[Edinburgh]] |lairgest_ceety = [[Glesga]] |government
```

Link Graph – Job1

Map:

```
<page><title> p </title><text>... [[q1]]... [[q2]] ...[[q3]]... </text></page>
```

Emit ((p,0), p) and ((q1,1), p), ((q2,1), p), ((q3,1),p)

```
<page><title> p </title><text>... </text></page>
```

Emit ((p,0), p)

Custom partitioner to ensure that all entries having p in their key (both (p,0) and (p,1)) will go to the same reducer

```
private static class MyPartitioner1 implements Partitioner<PairOfStringInt, Text> {  
    public int getPartition(PairOfStringInt key, Text value, int numReduceTasks)  
    {  
        return (key.getLeftElement().hashCode()) % numReduceTasks;  
    }  
}
```

Link Graph – Job1

Reduce: ((p,0), p), ((p,1), s1, s2, s3)

sCurrentArticle is a global variable per reducer.

If key.getRightElement is 0, as in ((p,0), p), then the reducer emits (p, p) and sCurrentArticle is set to p.

If the key.RightElement is 1, as in ((p,1), s1, s2, s3), and sCurrentArticle is p, then the reducer emits (s1, p), (s2,p),(s3,p).

If the key.RightElement is 1, as in ((p,1), s), but sCurrentArticle is different than p, those p pages are ignored (they don't have an entry in the original file).

Link Graph – Job2

The mapper emits (source_page, destination_page).

The reducer aggregates after source_node and constructs the adjacency list, by ignoring destination_pages that are identical with source_page.

This way, we can end up with pages with empty adjacency list, i.e. no outgoing links.

Where we are ...

- Improving IR results - recall
 - E.g., searching for *canine* doesn't match with *dog*.
- Options for improving results...
 - The complete landscape
 - Local methods
 - Relevance feedback
 - Pseudo relevance feedback
 - Global methods
 - Query expansion
 - Thesauri
 - Automatic thesaurus generation
 - Focus on relevance feedback first (Ch. 9 textbook)

Big Picture

- Relevance feedback
 - Adjust query with direct interaction
 - User looks at returned list of documents and provides feedback
 - System returns a revised ranked list
 - Adjust query with indirect interaction
 - By observing what the user looks at.
 - System returns a revised ranked list
 - Can a better query be created automatically by analyzing relevant and non-relevant documents?
- Pseudo-relevance feedback
 - Adjust query without interaction
 - Generate ranked list but do not present it
 - Use information to create a new ranked list that *is* presented
 - Can a better query be created automatically by assuming that some documents are relevant?

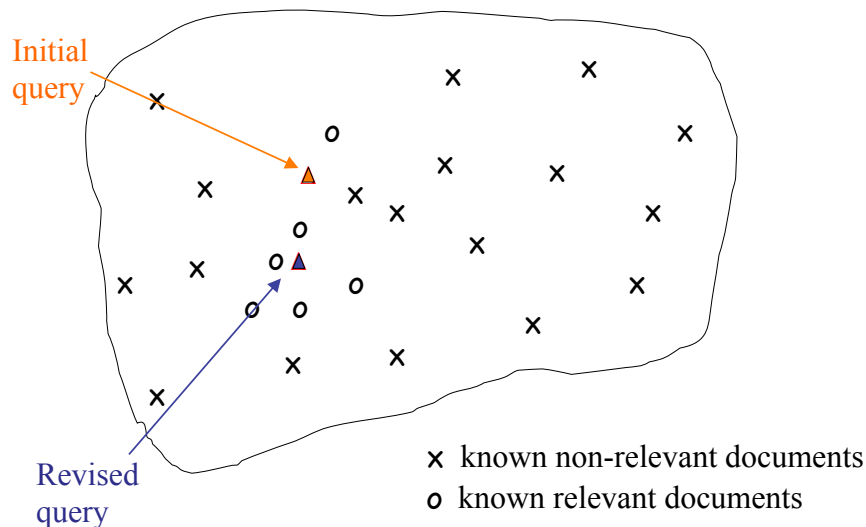
Rocchio 1971 Algorithm (SMART)

- Used in practice:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

- D_r = set of known relevant doc vectors
- D_{nr} = set of known irrelevant doc vectors
 - Different from C_r and C_{nr}
- q_m = modified query vector; q_0 = original query vector; α, β, γ : weights (hand-chosen or set empirically)
- **Add** the vectors for **relevant** docs to the query doc
- **Subtract** the vectors for **irrelevant** docs from the query doc
- New query moves toward relevant documents and away from irrelevant documents

Relevance Feedback on Initial Query







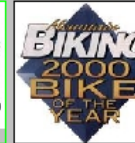







Relevance Feedback: Problems

- Long queries are inefficient for typical IR engine.
 - Long response times for user.
 - High cost for retrieval system.
 - Partial solution:
 - Only reweight certain prominent terms
 - Perhaps top 20 by term frequency
- **Users are often reluctant to provide explicit feedback**
- It's often harder to understand why a particular document was retrieved after applying relevance feedback












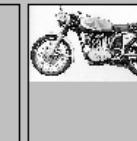


Relevance Feedback

Interface showing a grid of 12 images related to bicycles and motorcycles, with relevance scores below each image. The interface includes navigation buttons: Browse, Search, Prev, Next, Random.

					
(144473, 16458)	(144457, 252140)	(144456, 262857)	(144456, 262863)	(144457, 252134)	(144483, 265154)
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
					
(144483, 264644)	(144483, 265153)	(144518, 257752)	(144538, 525937)	(144456, 249611)	(144456, 250064)
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0

Results after Relevance Feedback

Interface showing a grid of 12 images related to bicycles and motorcycles, with relevance scores below each image. The interface includes navigation buttons: Browse, Search, Prev, Next, Random.

					
(144538, 523493)	(144538, 523835)	(144538, 523529)	(144456, 253569)	(144456, 253568)	(144538, 523799)
0.54182	0.56319296	0.584279	0.64501	0.650275	0.66709197
0.231944	0.267304	0.280881	0.351395	0.411745	0.358033
0.309876	0.295889	0.303398	0.293615	0.23853	0.309059
					
(144473, 16249)	(144456, 249634)	(144456, 253693)	(144473, 16328)	(144483, 265264)	(144478, 512410)
0.6721	0.675018	0.676901	0.700339	0.70170796	0.70297
0.393922	0.4639	0.47645	0.309002	0.36176	0.469111
0.278178	0.211118	0.200451	0.391337	0.339948	0.233859

Evaluating Relevance Feedback

- By construction, reformulated query will rank explicitly-marked relevant documents higher and explicitly-marked irrelevant documents lower.
- Method should not get credit for improvement on *these* documents, since it was told their relevance.
- In machine learning, this error is called “**testing on the training data.**”
- Evaluation should focus on generalizing to **other** un-rated documents.

Fair Evaluation of Relevance Feedback

- Remove from the corpus any documents for which feedback was provided.
- Measure recall/precision performance on the remaining documents - *residual collection*.
- Compared to complete corpus, specific recall/precision numbers may decrease since relevant documents were removed.
- However, **relative** performance on the residual collection can provide fair data on the effectiveness of relevance feedback.
- Train relevance feedback on one collection, test both original and modified query on a different collection.

Evaluation: Caveat

- True evaluation of usefulness must compare to other methods taking the same amount of time.
- Alternative to relevance feedback: User revises and resubmits query.
- Users may prefer revision/resubmission to having to judge relevance of documents.
- There is no clear evidence that relevance feedback is the “best use” of the user’s time.

Relevance Feedback on the Web

- Some search engines offer a similar/related pages feature (this is a trivial form of relevance feedback)
 - Google (more news on ...)
- But some don’t because it’s hard to explain to average user
- Excite initially had true relevance feedback, but abandoned it due to lack of use

Relevance Feedback: Summary

- Relevance feedback can be very effective.
- Effectiveness depends on number of judged documents.
- Significantly outperforms best human queries, given enough judged documents.
- Results can be unpredictable with less than five judged documents.
- Not used often in production systems, e.g., Web
 - consistent mediocre performance preferred to inconsistently good/great results
 - Stick with “documents like this one” variant
- An area of active research

Pseudo-Relevance Feedback

- True relevance feedback is supervised
 - Feedback is done based on *genuine* user annotations
- What happens if we try to guess what is relevant?
- Assume many top ranked documents are relevant
 - Optionally find a collection of probably non-relevant documents
- Modify query on that assumption
- Re-run that new query and show results to user
- What happens?

Pseudo-Relevance Feedback

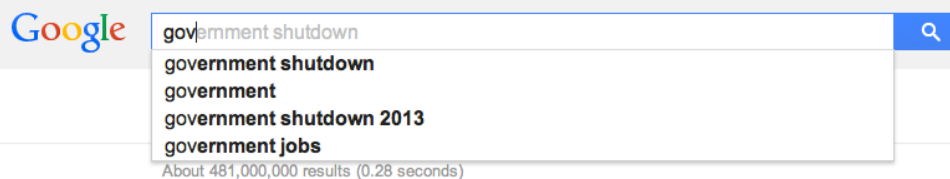
- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.
- Works very well on average
- But can go horribly wrong for some queries.
- Several iterations can cause query drift.
- Why?

Query Expansion

Query Expansion

- In relevance feedback, users give additional input (relevant/non-relevant) on **documents** – the input is used to reweight terms in the documents
- In query expansion, users give additional input (good/bad search term) on **words or phrases**

Query Assist



Would you expect such a feature to increase the query volume at a search engine?


Query Assist

- Generally done by query log mining
- Recommend frequent recent queries that contain partial string typed by user
- A ranking problem! View each prior query as a doc – Rank-order those matching partial string ...

government jobs
government shutdown 2013
government grants
government auctions
governor of poker
governors state university
government of canada
government liquidation
government furlough 2013
governor rick perry

Search Assist: On | Off

Google

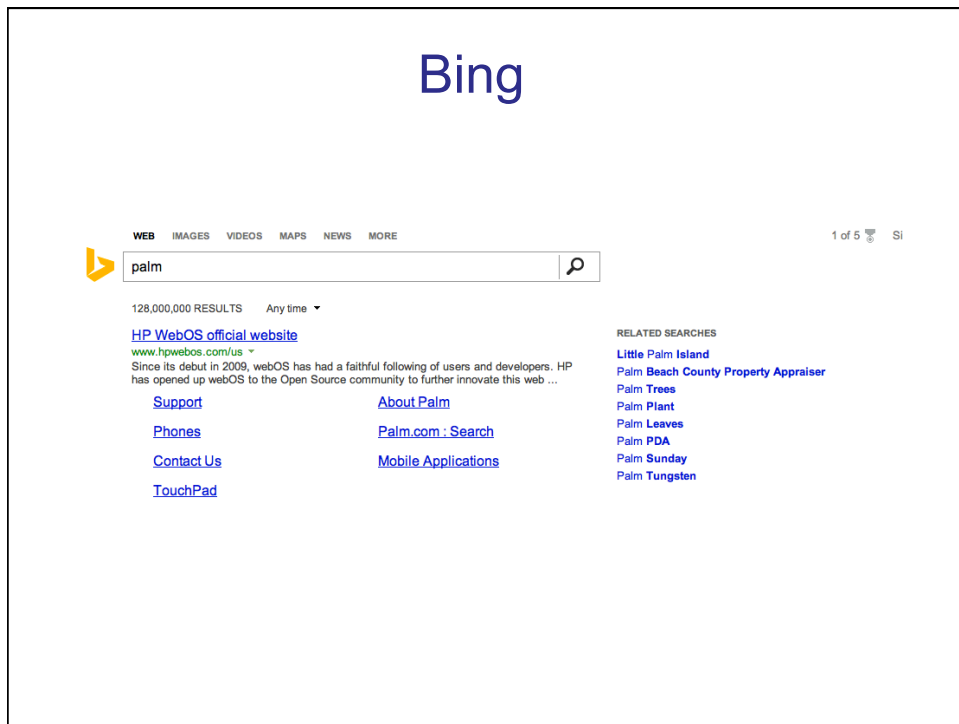


About 125,000,000 results (0.14 seconds)[Advanced search](#)

Searches related to palm

palm desktop	palm inc
palm beer	palm oil
palm wiki	palm microscopy
palm sunday	palm tree types

Bing



How Do We Augment the User Query?

- A thesaurus provides information on synonyms and semantically related words and phrases.
- Manual thesaurus
 - E.g. MedLine: physician, syn: doc, doctor, MD, medico
- Global Analysis: (static; of all documents in collection)
 - Automatically derived thesaurus
 - (co-occurrence statistics)
 - Refinements based on query log mining
 - Common on the web
- Local Analysis: (dynamic)
 - Analysis of documents in **result set**

Example of Manual Thesaurus

The screenshot shows the PubMed interface. At the top, there are logos for NCBI, PubMed, and the National Library of Medicine (NLM). Below the logos is a navigation bar with tabs for PubMed, Nucleotide, Protein, Genome, Structure, PopSet, and Taxonomy. The main search area has a search bar with the text 'cancer' and buttons for 'Go' and 'Clear'. Below the search bar are links for 'Limits', 'Preview/Index', 'History', 'Clipboard', and 'Details'. On the left side, there is a sidebar with links for 'About Entrez', 'Text Version', 'Entrez PubMed Overview', 'Help | FAQ', 'Tutorial', 'New/Noteworthy', 'E-Utilities', 'PubMed Services', 'Journals Database', 'MeSH Browser', 'Single Citation', and 'MeSH Browser'. The main content area shows the 'PubMed Query:' section with the query: `("neoplasms"[MeSH Terms] OR cancer[Text Word])`. Below the query is a 'Search' button and a 'URL' field.

Thesaurus-Based Query Expansion

- For each term, t , in a query, expand the query with synonyms and related words of t from the thesaurus
 - feline → feline cat
- May weight added terms less than original query terms.
- Generally increases recall
- Widely used in many science/engineering fields
- May significantly decrease precision, particularly with ambiguous terms.
 - “interest rate” → “interest rate fascinate evaluate”
- There is a high cost of manually producing a thesaurus
 - And for updating it for scientific changes

WordNet

- A detailed database of semantic relationships between English words.
- Developed by famous cognitive psychologist George Miller and a team at Princeton University.
- About 144,000 English words.
- Nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called *synsets*.

WordNet Synset Relationships

- **Antonym**: front → back
- **Attribute**: benevolence → good (noun to adjective)
- **Pertainym**: alphabetical → alphabet (adjective to noun)
- **Similar**: unquestioning → absolute
- **Cause**: kill → die
- **Entailment**: breathe → inhale
- **Holonym**: chapter → text (part-of)
- **Meronym**: computer → cpu (whole-of)
- **Hyponym**: tree → plant (specialization)
- **Hypernym**: fruit → apple (generalization)

WordNet Query Expansion

- Add synonyms in the same synset.
- Add hyponyms to add specialized terms.
- Add hypernyms to generalize a query.
- Add other related terms to expand query.

Statistical Thesaurus

- Existing human-developed thesauri are not easily available in all languages.
- Human thesauri are limited in the type and range of synonymy and semantic relations they represent.
- Semantically related terms can be discovered from statistical analysis of corpora.