Information retrieval

Relevance feedback & query expansion

Corso di Laurea Magistrale in Informatica

Università di Roma Tor Vergata

Prof. Giorgio Gambosi

a.a. 2021-2022

Derived from slides originally produced by C. Manning and by H. Schütze

Relevance

- Need to evaluate the quality of an information retrieval system and, in particular, its ranking algorithm with respect to relevance.
- A document is relevant if it gives the user the information she was looking for.
- To evaluate relevance, we need an evaluation benchmark with three elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - An assessment of the relevance of each query-document pair

Precision and recall

- Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

- Recall (R) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

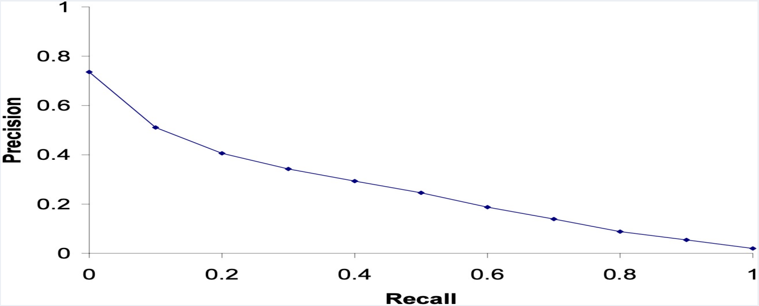
A combined measure: F

- F allows us to trade off precision against recall.
- Balanced F:

$$F_1 = \frac{2PR}{P + R}$$

- This is a kind of soft minimum of precision and recall.

Precision/recall graph



- Relates recall to precision (inversely related)
- 70% chance of getting the first document right (roughly)
- When we want to look at at least 50% of all relevant documents, then for each relevant document we find, we will have to look at about two nonrelevant documents.
- That's not very good.
- High-recall retrieval is an unsolved problem.

How can we improve recall in search?

- Two ways of improving recall: relevance feedback and query expansion
- As an example consider
  - query q: [aircraft]
  - document d containing "plane", but not containing "aircraft"
- A simple IR system will not return d for q.
- Even if d is the most relevant document for q
- In order to improve on this:
  - Return relevant documents even if there is no term match with the (original) query

Options for improving recall

- Local: Do a “local”, on-demand analysis for a user query
  - Main local method: relevance feedback
- Global: Do a global analysis once (e.g., of collection) to produce a thesaurus
  - Use thesaurus for query expansion

Relevance feedback: Basic idea

1. The user issues a (short, simple) query
2. The search engine returns a set of documents
3. The user marks some docs as relevant, some as nonrelevant
4. The search engine computes a new query which (hopefully) provides a better representation of the information need
5. The search engine runs new query and returns new results
6. New results have (hopefully) better recall

This process could be iterated: several rounds of relevance feedback.

The term ad hoc retrieval usually refers to regular retrieval without relevance feedback.

Example 2: A (non-image) example

Initial query: [new space satellite applications]

Results for initial query: (r = rank)

	r		
+	1	0.539	NASA Hasn't Scrapped Imaging Spectrometer
+	2	0.533	NASA Scratches Environment Gear From Satellite Plan
	3	0.528	Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
	4	0.526	A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
	5	0.525	Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
	6	0.524	Report Provides Support for the Critics Of Using Big Satellites to Study Climate
	7	0.516	Arianespace Receives Satellite Launch Pact From Telesat Canada
+	8	0.509	Telecommunications Tale of Two Companies

User then marks relevant documents with "+".

Expanded query after relevance feedback

From the selected documents content.

2.074	new	15.106	space
30.816	satellite	5.660	application
5.991	nasa	5.196	eos
4.196	launch	3.972	aster
3.516	instrument	3.446	arianespace
3.004	bundespost	2.806	ss
2.790	rocket	2.053	scientist
2.003	broadcast	1.172	earth
0.836	oil	0.646	measure

Compare to original query: [new space satellite applications]

Results for expanded query (old ranks in parentheses)

	r		
*	1 (2)	0.513	NASA Scratches Environment Gear From Satellite Plan
*	2 (1)	0.500	NASA Hasn't Scrapped Imaging Spectrometer
	3	0.493	When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
	4	0.493	NASA Uses 'Warm' Superconductors For Fast Circuit
*	5 (8)	0.492	Telecommunications Tale of Two Companies
	6	0.491	Soviets May Adapt Parts of SS-20 Missile For Commercial Use
	7	0.490	Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
	8	0.490	Rescue of Satellite By Space Agency To Cost $90 Million

Key concept for relevance feedback: centroid

- The centroid is the center of mass of a set of points.
- Recall that we represent documents as points in a high-dimensional space.
- Thus: we can compute centroids of documents.
- Definition:

$$\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

where D is a set of documents and $\vec{v}(d) = \vec{d}$ is the vector we use to represent document d.

Optimal query

- Assume the whole sets of relevant C_r and not relevant C_{nr} documents in the collection are known
- the optimal query $\vec{q}_{opt}$ is then the one that maximizes

$$S(\vec{q}, C_r, C_{nr}) = s(\vec{q}, \vec{\mu}(C_r)) - s(\vec{q}, \vec{\mu}(C_{nr}))$$

  where s is a similarity measure
- that is, $\vec{q}_{opt}$ is the vector that separates relevant and nonrelevant docs maximally.
- Under cosine similarity, this corresponds to maximizing with respect to:

$$\vec{q} \cdot \vec{\mu}(C_r) - \vec{q} \cdot \vec{\mu}(C_{nr}) = \vec{q} \cdot (\vec{\mu}(C_r) - \vec{\mu}(C_{nr}))$$

  which results into

$$\vec{q}_{opt} = \vec{\mu}(C_r) - \vec{\mu}(C_{nr}) = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{|C_{nr}|} \sum_{\vec{d}_j \in C_{nr}} \vec{d}_j$$

  that is, the optimal query is the vector difference between the centroids of relevant and not relevant documents

Optimal query

- Unfortunately, C_r and C_{nr} are not known: hints from relevance feedback can be used, if available
- The Rocchio algorithm implements relevance feedback in the vector space model by deriving a new query from a prvious one and hints from RF
- Given the results of a query $\vec{q}_0$, let D_r and D_{nr} the sets of relevant and not relevant documents identified in relevance feedback
- Rocchio derives a modified query $\vec{q}_m$

$$\begin{aligned}\vec{q}_m &= \alpha\vec{q}_0 + \beta\mu(D_r) - \gamma\mu(D_{nr}) \\ &= \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j\end{aligned}$$

  where α, β, and γ are predefined weights
- New query moves towards relevant documents and away from nonrelevant documents.

- When can relevance feedback enhance recall?
- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Assumption A2: Relevant documents contain similar terms

Violation of A1

- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Violation: Mismatch of searcher's vocabulary and collection vocabulary
- Example: cosmonaut / astronaut

Violation of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]
- Several unrelated "prototypes"
  - Subsidies for tobacco farmers vs. anti-smoking campaigns
  - Aid for developing countries vs. high tariffs on imports from developing countries
- Relevance feedback on tobacco docs will not help with finding docs on developing countries.

Relevance feedback: Problems

- Relevance feedback is expensive.
  - Relevance feedback creates long modified queries.
  - Long queries are expensive to process.
- Users are reluctant to provide explicit feedback.
- It's often hard to understand why a particular document was retrieved after applying relevance feedback.
- The search engine Excite had full relevance feedback at one point, but abandoned it later.

Pseudo-relevance feedback

- Pseudo-relevance feedback automates the "manual" part of true relevance feedback.
- Pseudo-relevance feedback algorithm:
  - Retrieve a ranked list of hits for the user's query
  - Assume that the top k documents are relevant.
  - Do relevance feedback (e.g., Rocchio)
- Works very well on average
- But can go horribly wrong for some queries.
  - Because of query drift
  - If you do several iterations of pseudo-relevance feedback, then you will get query drift for a large proportion of queries.

Query expansion

- Query expansion is another method for increasing recall.
- We use “global query expansion” to refer to “global methods for query reformulation”.
- In global query expansion, the query is modified based on some global resource, i.e. a resource that is not query-dependent.
- Main information we use: (near-)synonymy

“Global” resources used for query expansion

- A publication or database that collects (near-)synonyms is called a thesaurus.
- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)
- Query-equivalence based on query log mining

Thesaurus-based query expansion

- For each term t in the query, expand the query with words the thesaurus lists as semantically related with t.
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
- Widely used in specialized search engines for science and engineering
- It's very expensive to create a manual thesaurus and to maintain it over time.

Automatic thesaurus generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents
- Fundamental notion: similarity between two words
- Definition 1: Two words are similar if they co-occur with similar words.
  - “car” $\approx$ “motorcycle” because both occur with “road”, “gas” and “license”, so they must be similar.
- Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.
  - You can harvest, peel, eat, prepare, etc. apples and pears, so apples and pears must be similar.

Query expansion at search engines

- Main source of query expansion at search engines: query logs
- Example 1: After issuing the query [herbs], users frequently search for [herbal remedies].
  - → "herbal remedies" is potential expansion of "herb".
- Example 2: Users searching for [flower pix] frequently click on the URL photobucket.com/flower. Users searching for [flower clipart] frequently click on the same URL.
  - → "flower clipart" and "flower pix" are potential expansions of each other.