Query Expansion

Mandar Mitra

Indian Statistical Institute

Motivation

- Searching depends on matching keywords between user-query and document

$$Sim(Q, D) = \vec{Q}.\vec{D} = \sum_i q_i \times d_i$$

Motivation

- Searching depends on matching keywords between user-query and document

$$Sim(Q, D) = \vec{Q}.\vec{D} = \sum_i q_i \times d_i$$

- *Vocabulary mismatch:* searchers and document creators may use different keywords to denote same "concept"

Query

casualties in traffic accidents

Relevant document

Four people were injured when a truck . . .

Motivation

- Searching depends on matching keywords between user-query and document

$$Sim(Q, D) = \vec{Q}.\vec{D} = \sum_i q_i \times d_i$$

- *Vocabulary mismatch:* searchers and document creators may use different keywords to denote same "concept"

Query

casualties in traffic accidents

Relevant document

Four people were injured when a truck . . .

- Orthogonality of dimensions (words) / binary independence
  $\Rightarrow$ poor retrieval quality
- Problem aggravated by short queries + large, heterogeneous databases

Motivation

- Problem: vocabulary mismatch
- Solution: expand the query by adding related words/phrases.

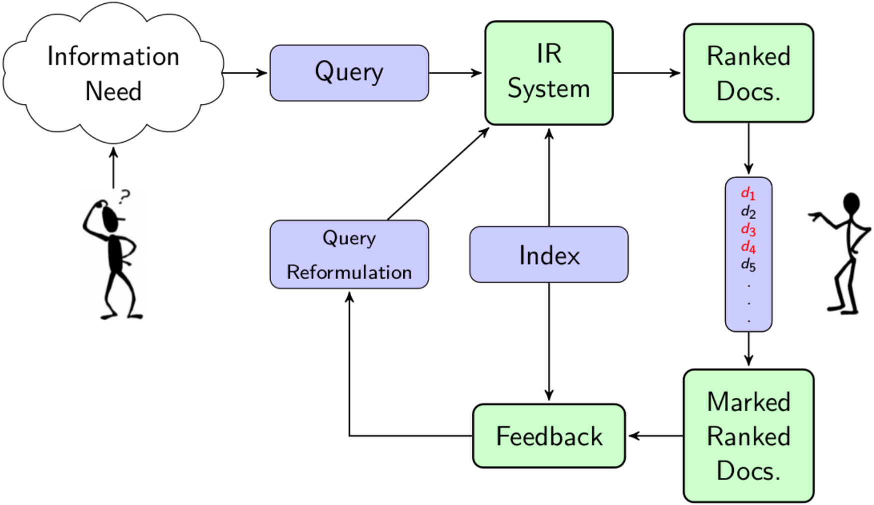
Motivation

- Problem: vocabulary mismatch
- Solution: expand the query by adding related words/phrases.
- Issues:
  - **select** which terms to add to query
  - **calculate weights** for added terms

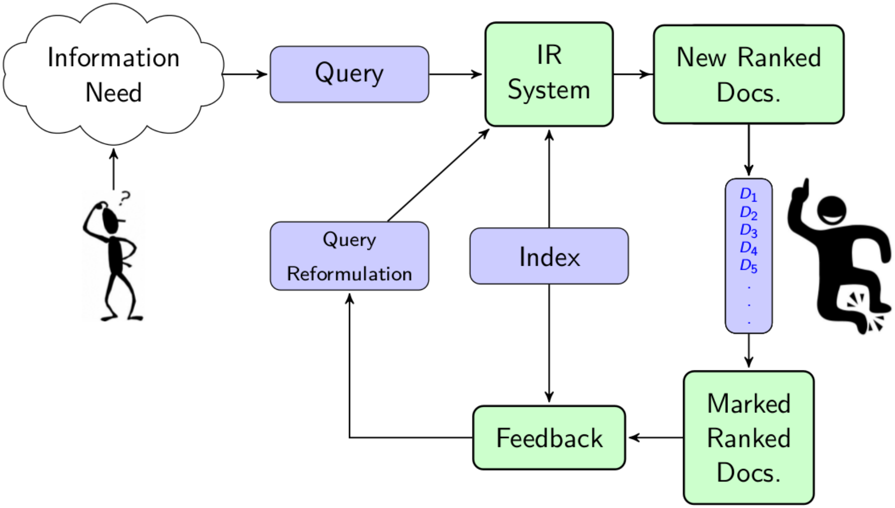
Motivation

- Problem: vocabulary mismatch
- Solution: expand the query by adding related words/phrases.
- Issues:
  - **select** which terms to add to query
  - **calculate weights** for added terms
- What kinds of queries are likely to be benefited by expansion?
  - fairly focused (as opposed to broad/exploratory queries)
  - long-standing / “serious” searches

Relevance feedback: a graphical representation



Relevance feedback: a graphical representation



Relevance feedback

- Original query is used to retrieve some number of documents.
- User examines some of the retrieved documents and provides feedback about which documents are relevant and which are non-relevant.
- System uses the feedback to “learn” a better query:
  - select/emphasize words that occur more frequently in relevant documents than non-relevant documents;
  - eliminate/de-emphasize words that occur more frequently in non-relevant than in relevant documents.
- Resulting query should bring in more relevant documents and fewer non-relevant documents.

Vector Space Model

Rocchio's algorithm

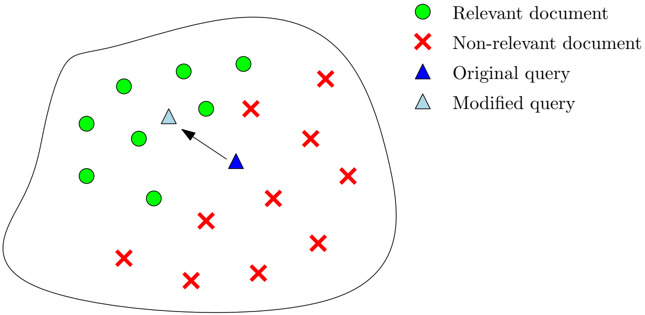
Principle

Maximize the difference between average similarities for relevant and non-relevant documents.

Rocchio's algorithm

Principle

Maximize the difference between average similarities for relevant and non-relevant documents.



Rocchio's algorithm

Principle

Maximize the difference between average similarities for relevant and non-relevant documents.

- Mathematically:

$$C = \frac{1}{N_{rel}} \sum_{D_i \in Rel} Sim(Q, D_i) - \frac{1}{N_{nonrel}} \sum_{D_i \in NRel} Sim(Q, D_i)$$

$$= \vec{Q}. \left[\frac{1}{N_{rel}} \sum_{D_i \in Rel} \vec{D}_i - \frac{1}{N_{nonrel}} \sum_{D_i \in NRel} \vec{D}_i \right]$$

- In practice:

$$\vec{Q}_{new} = \alpha\, \vec{Q}_{old} + \frac{\beta}{N'_{rel}} \sum_{D_i \in Rel} \vec{D}_i - \frac{\gamma}{N'_{nonrel}} \sum_{D_i \in NRel} \vec{D}_i$$

Rocchio's algorithm: example

Original query	0	4	0	8	0	0	$\alpha = 1.0$	0	4	0	8	0	0
Positive feedback	2	4	8	0	0	2	$\beta = 0.5$	1	2	4	0	0	1
Negative feedback	8	0	4	4	0	16	$\gamma = 0.25$	2	0	1	1	0	4

Rocchio's algorithm: example

Original query	0	4	0	8	0	0	$\alpha = 1.0$	0	4	0	8	0	0
Positive feedback	2	4	8	0	0	2	$\beta = 0.5$	1	2	4	0	0	1
Negative feedback	8	0	4	4	0	16	$\gamma = 0.25$	2	0	1	1	0	4
							New query	-1	6	3	7	0	-3

Rocchio's algorithm: issues

How to choose α, β, γ?

- Use a higher β, γ if there are a lot of judged documents

Rocchio's algorithm: issues

How to choose α, β, γ?

- Use a higher β, γ if there are a lot of judged documents

What should be regarded as *NRel*?

- Documents marked non-relevant by user?
  - too little information
- All documents not known to be relevant?
  - documents completely unrelated to the query may affect term-weights
    e.g. *"What disk drive should I buy for my Mac?"*
    Is *computer* really a good keyword?
- Use non-relevant documents within a *query zone* [Singhal et al., SIGIR 97].

Rocchio's algorithm: issues

How should terms be selected? How many?

- Rank terms by # relevant documents they occur in.
- Add 50-100 terms.

Blind/adhoc/pseudo relevance feedback

- In the absence of feedback, *assume* top-ranked documents are relevant.
- Optionally, use low-ranked documents to form the query zone.

Blind/adhoc/pseudo relevance feedback

- In the absence of feedback, *assume* top-ranked documents are relevant.
- Optionally, use low-ranked documents to form the query zone.
- Obvious danger: if initial retrieval is poor, adhoc feedback can aggravate the problem.
  Example: *What is the economic impact of recycling tires?*
  query is hijacked by *plastics*, *recycling* in general

Blind/adhoc/pseudo relevance feedback

- In the absence of feedback, *assume* top-ranked documents are relevant.
- Optionally, use low-ranked documents to form the query zone.
- Obvious danger: if initial retrieval is poor, adhoc feedback can aggravate the problem.
  Example: *What is the economic impact of recycling tires?*
  query is hijacked by *plastics*, *recycling* in general
- Most groups at TREC found adhoc feedback useful on average.
- Suggestion: find ways to improve the initial retrieval.

Blind feedback: improvements

- All *aspects* of query should be present in a highly-ranked document.
- Use Boolean filters to ensure this.
  - use proximity constraints
  - use soft matching
- Can use cooccurrence patterns of terms to estimate their relatedness

Language Modelling

Relevance based language model

Reference: Lavrenko and Croft, SIGIR 01

- (Unigram) language model = probability distribution over words
- *Relevance* model R = probability distribution over words followed by / observed in relevant documents

Relevance based language model

Reference: Lavrenko and Croft, SIGIR 01

- (Unigram) language model = probability distribution over words
- *Relevance* model R = probability distribution over words followed by / observed in relevant documents

Relevance based language model

Reference: Lavrenko and Croft, SIGIR 01

- (Unigram) language model = probability distribution over words
- *Relevance* model R = probability distribution over words followed by / observed in relevant documents ✗

  OR

  Relevance model = probability distribution of words *cooccurring with query words* / observing w along with query terms *in relevant documents* ✓

Relevance based language model

$\boxed{\text{Objective: estimate } P(w, Q)}$

Given:

- Query $Q = \{q_1, q_2, \ldots, q_k\}$
- Top-ranked (pseudo-relevant) documents $\mathcal{M} = \{d_1, d_2, \ldots, d_M\}$
- **Assumption:** $q_1, q_2, \ldots, q_k$ and w are picked *independently*

Estimating $P(w, Q)$

$$P(w, Q) = \sum_{D \in M} \color{green}{P(D)}\color{red}{P(w, Q|D)}$$

Estimating $P(w, Q)$

$$P(w, Q) = \sum_{D \in M} P(D) P(w, Q|D)$$

$$P(w, Q|D) = P(w|D) \prod_{q \in Q} P(q|D)$$

Estimating $P(w, Q)$

$$P(w, Q) = \sum_{D \in M} P(D) P(w, Q|D)$$

$$P(w, Q|D) = P(w|D) \prod_{q \in Q} P(q|D)$$

$$P(w, Q) = \sum_{D \in M} P(D) P(w|D) \prod_{q \in Q} P(q|D)$$

Estimating $P(w, Q)$

$$P(w, Q) = \sum_{D \in M} P(D) P(w, Q | D)$$

$$P(w, Q | D) = P(w | D) \prod_{q \in Q} P(q | D)$$

$$P(w, Q) = \sum_{D \in M} P(D) P(w | D) \prod_{q \in Q} P(q | D)$$

- $\prod_{q \in Q} P(q | D)$: LM based retrieval score of D
- $P(w | D)$: maximum likelihood estimate of w in D
- $P(D)$: prior probability of selection of the document

RM3

Reference: Abdul Jaleel et al., TREC 04

Mix the relevance model with query likelihood model

$$P'(w|R) = \mu P(w|R) + (1 - \mu) P(w|Q)$$

RM3

Reference: Abdul Jaleel et al., TREC 04

Mix the relevance model with query likelihood model

$$P'(w|R) = \mu P(w|R) + (1-\mu) P(w|Q)$$

$$P(w|Q) = \frac{tf(w,Q)}{|Q|}$$

Thesaurus-based expansion

Thesaurus-based expansion

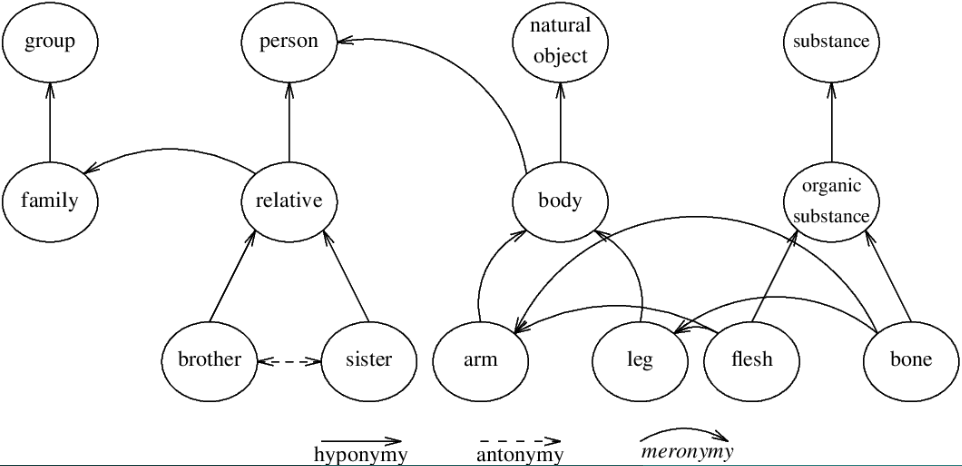
- Manual thesauri:
  - general purpose (Roget's Thesaurus, WordNet) – difficult to use for document retrieval
  - retrieval-oriented (INSPEC, MeSH) – expensive to build and maintain
- Automatic thesauri: based on information about co-occurrence of words in a collection

WordNet

- Electronic dictionary + thesaurus
- Divided into sections: nouns, verbs, adjectives, adverbs

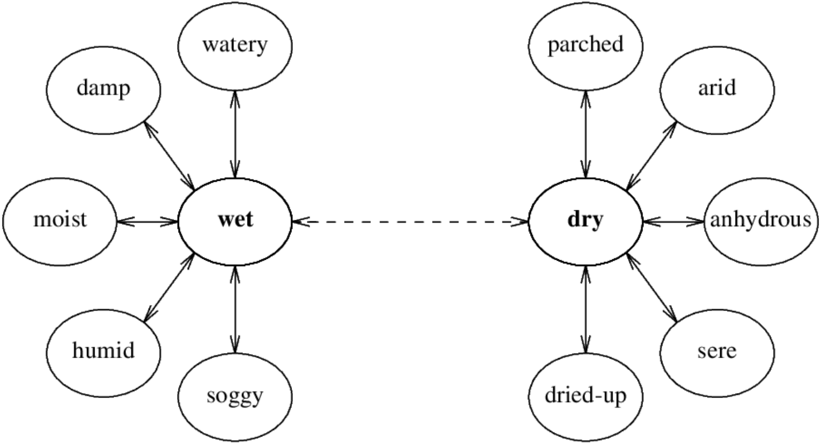
WordNet

- Electronic dictionary + thesaurus
- Divided into sections: nouns, verbs, adjectives, adverbs



WordNet

- Electronic dictionary + thesaurus
- Divided into sections: nouns, verbs, adjectives, adverbs



Using WordNet

```
>>> from nltk.corpus import wordnet as wn

>>> wn.synsets('dog')
[Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'), ...

>>> print(wn.synset('dog.n.01').definition())
a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds

>>> print(wn.synset('dog.n.01').examples()[0])
the dog barked all night
```

Expansion using WordNet

Reference: Fang, ACL 2008

$$\{q_1, q_2, \ldots, q_k\} \longrightarrow \{q_1, q_2, \ldots, q_k,\ q'_1, q'_2, \ldots, q'_{k'}\}$$

Expansion using WordNet

Reference: Fang, ACL 2008

$$\{q_1, q_2, \ldots, q_k\} \longrightarrow \{q_1, q_2, \ldots, q_k, \; q'_1, q'_2, \ldots, q'_{k'}\}$$

Term-term similarity

- Based on WordNet relations R (synonymy, hypernymy, holonymy)

$$Sim(t_1, t_2) = \begin{cases} \alpha_R & \text{if } t_1 \sim_R t_2, \\ 0 & \text{otherwise.} \end{cases}$$

Expansion using WordNet

Reference: Fang, ACL 2008

$$\{q_1, q_2, \ldots, q_k\} \longrightarrow \{q_1, q_2, \ldots, q_k,\ q'_1, q'_2, \ldots, q'_{k'}\}$$

Term-term similarity

- Based on WordNet relations R (synonymy, hypernymy, holonymy)

$$Sim(t_1, t_2) = \begin{cases} \alpha_R & \text{if } t_1 \sim_R t_2, \\ 0 & \text{otherwise.} \end{cases}$$

- Based on definitions $D(t_1), D(t_2)$

$$Sim(t_1 t_2) = \frac{|D(t_1) \cap D(t_2)|}{|D(t_1) \cup D(t_2)|}$$

Expansion using WordNet

Reference: Fang, ACL 2008

$$\{q_1, q_2, \ldots, q_k\} \longrightarrow \{q_1, q_2, \ldots, q_k,\ q'_1, q'_2, \ldots, q'_{k'}\}$$

Term-term similarity

- Based on WordNet relations R (synonymy, hypernymy, holonymy)

$$Sim(t_1, t_2) = \begin{cases} \alpha_R & \text{if } t_1 \sim_R t_2, \\ 0 & \text{otherwise.} \end{cases}$$

- Based on definitions $D(t_1), D(t_2)$

$$Sim(t_1 t_2) = \frac{|D(t_1) \cap D(t_2)|}{|D(t_1) \cup D(t_2)|}$$

Automatic thesaurus based expansion

Reference: Jing and Croft, 1994

Approach:

- Association: if two terms co-occur within the same para, they constitute an association

$$\langle \text{ term1, term2, assoc. frequency } \rangle$$

- Gather data about associations over a large amount of text
- Refine associations
  - discard associations with frequency 1
  - discard terms associated with too many other terms
    e.g. *people*, *state*, *company*
- Each term $\equiv$ pseudo document ($T = (\langle t_1, w_1 \rangle, \ \ldots, \ \langle t_n, w_n \rangle)$)
- Compare query to the term vectors; add most similar terms to the query
  Example: *1986 US Immigration Law*
  Similar terms: *illegal immigration*, *amnesty program*, *simpson-mazzoli*

Automatic thesaurus based expansion

Results:

- Data: 500,000 documents (news, computer abstracts, govt. documents); 50 queries
- Baseline average precision: 37%
- Improves 6 - 30% by using thesaurus
- 2 weeks to generate association data!
- Processing time can be reduced without major loss in performance by using a subset of the document collection

References

- *A Survey of Automatic Query Expansion in Information Retrieval.* Claudio Carpineto, Giovanni Romano. ACM Computing Surveys, 44(1), January 2012.
  `http://doi.acm.org/10.1145/2071389.2071390`
- *A Re-examination of Query Expansion Using Lexical Resources.* Hui Fang. Proceedings of ACL-08: HLT, pages 139147, 2008.