Information Retrieval Methods

Helena Ahonen-Myka
Spring 2006, part 7
Matching methods (relevance feedback) (2/2)
Translation from Finnish: Greger Lindén

In this part

- Relevance feedback for modifying the query

Relevance feedback

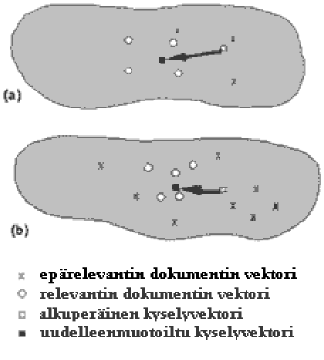
- It is hard for the user to specify good search terms
  - Especially if the user does not know the contents of the documents in the collection
- It is typical to modify the search during the retrieval process
  - First the user writes the search terms that come into mind
  - After seeing the result, the user can specify the query more precisely
- The system may also modify the query automatically based on relevance feedback

Relevance feedback

- Basic assumption: relevant documents for a retrieval task are similar (= “cluster hypothesis”)
- Idea:
  - The user has studied k documents in the answer list and recognised some of them as relevant and some of them as non-relevant
  - The query vector is modified so that it becomes more similar to the known relevant documents and more dissimilar to the known non-relevant documents
  - We assume that the new query finds more relevant documents or moves the relevant documents up in the answer list (before the non-relevant)

Relevance feedback

- In the following figure (a) there are some terms in the document vectors that discriminate relevant documents from non-relevant ones
  - With relevance feedback we can easily modify the query vector to become more similar to the centroid vector of the relevant documents
- In figure (b) the relevant documents are clustered very closely together and non-relevant documents are scattered
  - Relevance feedback also works in this case


(a)
(b)
x epärelevantin dokumentin vektori
○ relevantin dokumentin vektori
□ alkuperäinen kyselyvektori
■ uudelleenmuotoiltu kyselyvektori

X vector of non-relevant document
O vector of relevant document
□ original query vector
■ modified query vector

Relevance feedback

- If the vectors of relevant and non-relevant documents are scattered evenly in the query vector environment, relevance feedback does not work
- The relevant documents will probably not form a continuous cluster
  - Known relevant documents can be clustered and we see if the above problem arises
  - If there are several clusters, we can divide the query into several parts

Relevance feedback

- A query can be iterated several times
  - The user always evaluates new documents
  - → the sets of known relevant and known non-relevant documents grow at each iteration
- The process ends, e.g.,
  - when the user is satisfied with the result (or is too bored to give further feedback …), or
  - after a certain number of iterations, or
  - when modifying the query does not greatly affect the size of the contents of the answer any more

Relevance feedback

- Relevance feedback is an effective method
  - Even after just one iteration the average precision may increase with 40-60%
- The effectiveness is probably based on the fact that we find out more about the user’s information needs through the relevance feedback
  - And we use this information to add terms that describe the information need and to change the relative significance of the terms
  - (Original) user queries are usually short

Relevance feedback

- Which principle will help us to modify the query vector towards the vectors of the relevant documents (and away from the vectors of the non-relevant documents)?
- Let us assume that we know which documents in the collection are relevant and which are not
- We can form an optimal query that
  - Maximises the similarity between the query and the relevant documents
  - Minimises the similarity between the query and the non-relevant documents

Relevance feedback

- The optimal query could be described with the formula

$$Q_{opt} = \frac{1}{|R|}\sum_{D_i \in R} D_i - \frac{1}{|N|}\sum_{D_i \in N} D_i$$

- where R is the set of relevant documents, N the set of non-relevant documents

Relevance feedback

- In practice, we cannot form the optimal query
  - If we knew which documents are relevant, we would not need to process the query
- But the original query may be modified so that at each iteration it comes closer and closer to the optimal query
  - When the user gives relevance feedback, each time we find a larger subset R’ of all the relevant documents R and also a larger subset N’ of all the non-relevant documents N

Relevance feedback

- The query may be modified by adding terms from the vectors of relevant documents and removing terms from the vectors of non-relevant documents

$$Q^{i+1} = Q^{i} + \frac{1}{|R'|}\sum_{D_i \in R'} D_i - \frac{1}{|N'|}\sum_{D_i \in N'} D_i$$

Relevance feedback

- Alternatively, we can define coefficients α and β to denote the desired relationship between the relevant and non-relevant documents

$$Q^{i+1} = Q^{i} + \alpha \sum_{D_i \in R'} D_i - \beta \sum_{D_i \in N'} D_i$$

- $\alpha = \beta = 0.5$ → equally important
- $\alpha = 1$ and $\beta = 0$ → only relevant documents are included

Relevance feedback

- We can also take all recognised relevant documents into account but only the non-relevant documents that were listed the highest in the answer set

Relevance feedback, alternations

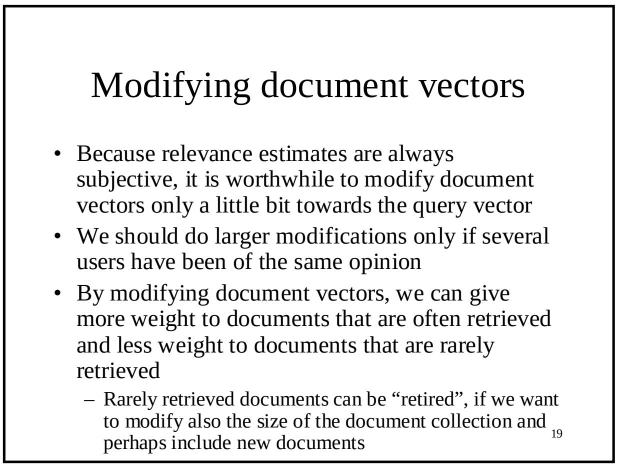
- When the user estimates the relevance of whole documents, problems can arise
  - Relevant documents may contain many terms that do not concern the information need
    - and they still affect the new query
  - The relevance feedback cannot change the original query very much: if the query did not return any relevant documents in the first place, relevance feedback would not change the situation
- We could also ask the user which terms or phrases seem to be significant and only use these
  - but this requires the user to be more active

Relevance feedback, alternations

- The idea of relevance feedback can also be used when we cannot use the user’s feedback (or do not want to “disturb” the user) -> “pseudo-feedback”
- We choose the 10 best documents in the answer set and use the terms in them to modify the query
- We order the terms according to their frequency and remove stopwords
- We extend the query with, e.g., the ten most frequent terms

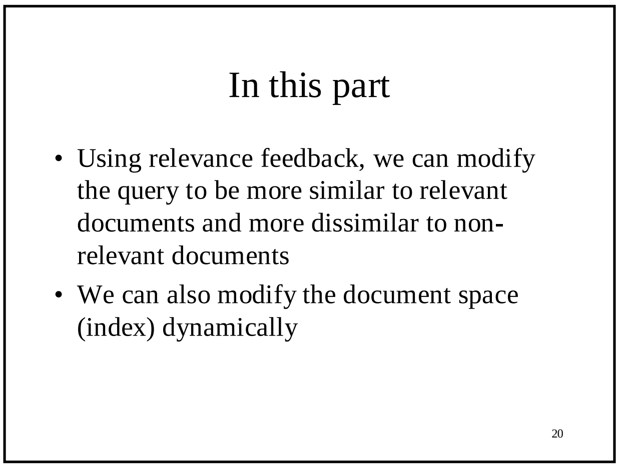
Modifying document vectors

- When we use relevance feedback,we modify the query vector
- We could also try to influence the document space → we modify the index dynamically
- If a set of documents has been recognised to be relevant, we modify the vectors of these documents to be more similar to the query vector
- At the same time, the document vectors will become more similar to each other and also more easily returned together



Modifying document vectors

- Because relevance estimates are always subjective, it is worthwhile to modify document vectors only a little bit towards the query vector
- We should do larger modifications only if several users have been of the same opinion
- By modifying document vectors, we can give more weight to documents that are often retrieved and less weight to documents that are rarely retrieved
  - Rarely retrieved documents can be “retired”, if we want to modify also the size of the document collection and perhaps include new documents



In this part

- Using relevance feedback, we can modify the query to be more similar to relevant documents and more dissimilar to non-relevant documents
- We can also modify the document space (index) dynamically