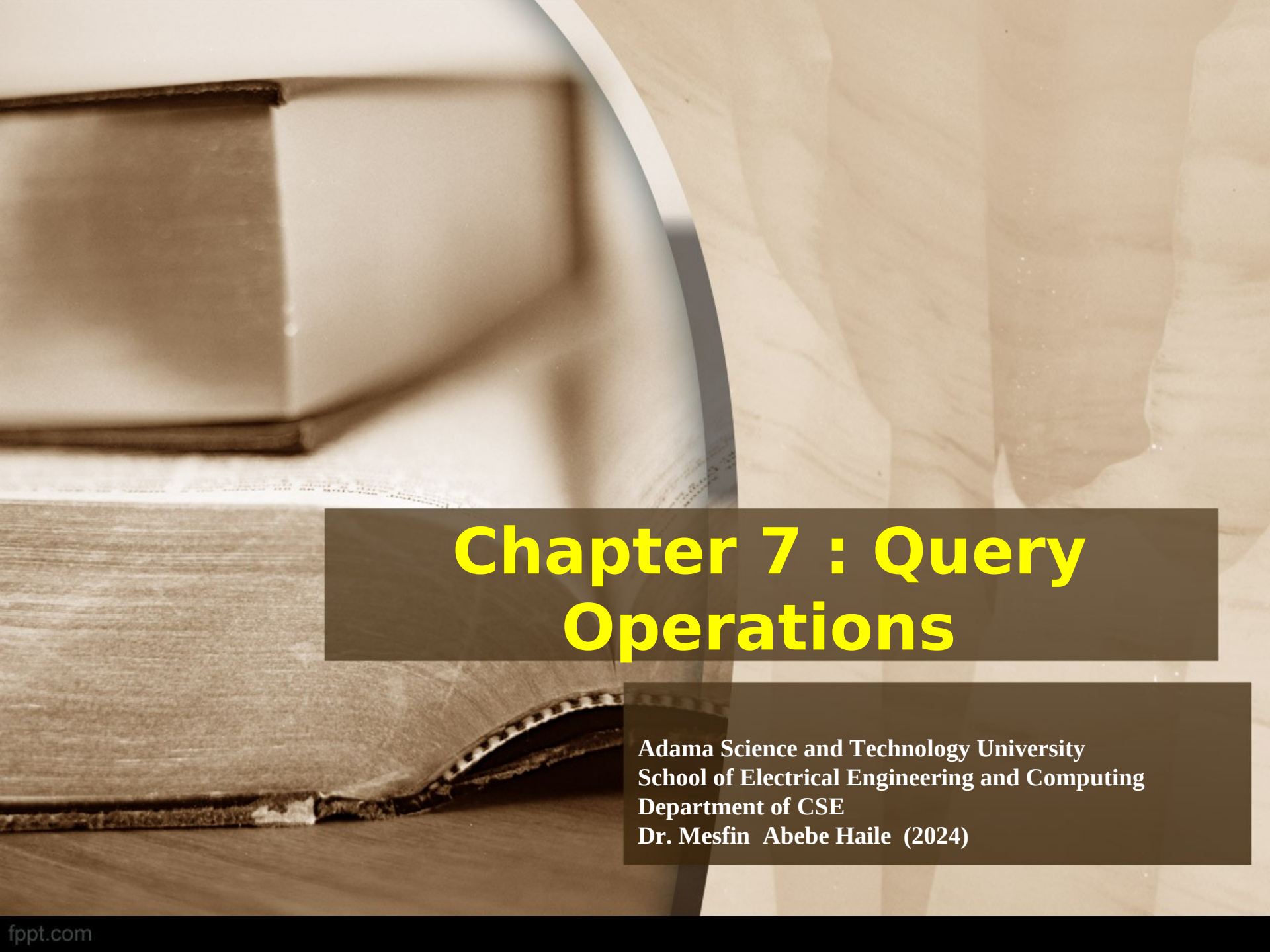




Chapter 7 : Query Operations

Adama Science and Technology University
School of Electrical Engineering and Computing
Department of CSE
Dr. Mesfin Abebe Haile (2024)

Introduction

- ❖ No *detailed knowledge* of collection and searching environment.
 - ❖ Difficult to *formulate queries* well designed for searching
 - ❖ Need *many formulations of queries* for effective searching
- ❖ ***First formulation***: often naïve attempt to *retrieve relevant* information.
- ❖ Documents initially retrieved:
 - ❖ Can be examined for *relevance information* (by the user or automatically by the system) to *provide relevance feedback*.
 - ❖ Improve *query formulations* for retrieving *additional relevant documents* (using query reformulation techniques)

Query Reformulation

- ❖ Identify terms related to *query terms*.
- ❖ Revise query to account for feedback using *two basic techniques*:
 - ❖ *Query Expansion*: Add *new terms* related to query terms from relevant documents.
 - ❖ *Term Reweighting*: *modify term weights* based on documents relevance for the users query.
 - ❖ Increase weight of terms in *relevant documents and decrease* weight of terms in *irrelevant documents*.
- ❖ Several *algorithms* for *query reformulation*.

Term Reweighting for Query Reformulation

- ❖ Term weight vectors of documents *assessed relevant*.
 - ❖ *Similarities* among themselves.
- ❖ Term weight vectors of documents *assessed non-relevant*.
 - ❖ *Dissimilar* for those of relevant documents.
- ❖ Reformulated query:
 - ❖ Closer to term weight vectors of *relevant documents*.

Term Reweighting for Query Reformulation: Rochio Formula

For query q :

- ❖ D_r : set of *relevant* documents among *retrieved documents*.
- ❖ D_n : set of *non-relevant* documents among *retrieved documents*.
- ❖ C_r : set of *relevant documents* among all documents in collection.

$$q_{i+1} = \alpha q_i + \frac{\beta}{|D_r|} \sum_{d_j \in D_r} d_j - \frac{\gamma}{|D_n|} \sum_{d_j \in D_n} d_j$$

- ❖ α, β, γ : tuning constants.
 - ❖ Initial formulation $\alpha=1$
 - ❖ Usually information *in relevant documents* is more important than in *non-relevant documents* ($\gamma \ll \beta$).

Term Reweighting for Query Reformulation: Ide Formula

❖ *Initial formulation* $\alpha = \beta = \gamma = 1$

❖ Same comments as for the *Rocchio formula*.

$$q_{i+1} = \alpha q_i + \beta \sum_{d_j \in Dr} d_j - \gamma \sum_{d_j \in Dn} d_j$$

❖ Both Ide and Rocchio: no optimal criterion.

❖ E.g. you are given a *query vector* $q_i = (2, 3, 1, 2, 5)$; and *documents*; $d_1(3, 3, 2, 0, 9)$; $d_2(2, 2, 1, 0, 12)$; $d_3(3, 2, 1, 0, 9)$; $d_4(2, 2, 1, 3, 1)$; $d_5(1, 2, 1, 3, 3)$;

❖ 3 documents identified as *relevant by a user*, (i.e. d_1 - d_3); and 2 documents as *irrelevant* (i.e. d_4 - d_5).

❖ Compute the modified query *using standard Rocchio equation* with $\alpha = \beta = \gamma = 1$

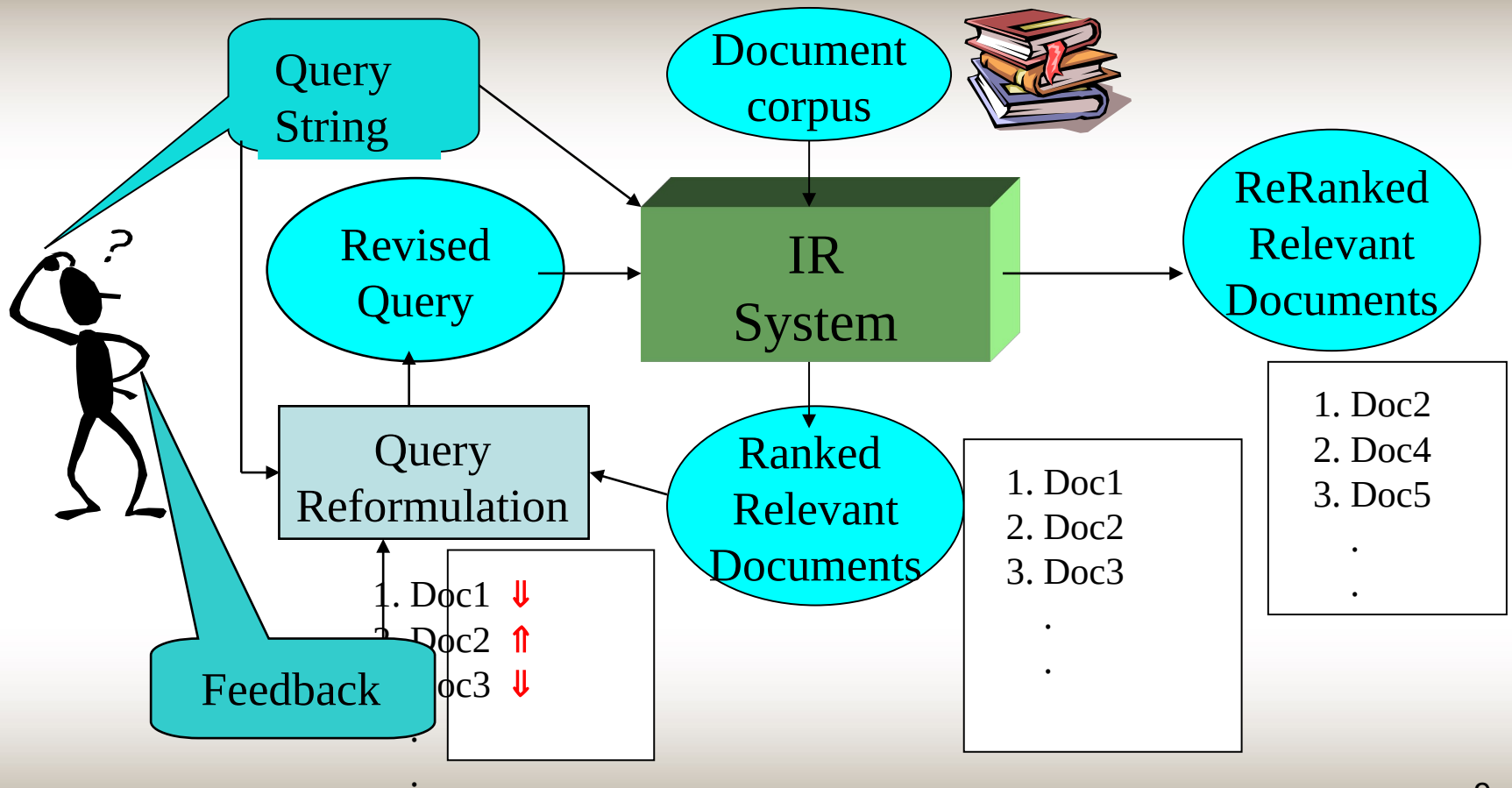
Approaches for Query Operations

- ❖ Users relevance feedback:
- ❖ Approaches based on feedback from users *about relevance* of documents retrieved.
- ❖ Pseudo-relevance feedback:
- ❖ Approaches based on information derived from set of *initially retrieved* documents (local set of documents), which is called *Local Analysis*.
- ❖ Approaches based on *global information* derived from document collection, which is called *Global Analysis*.

Users Relevance Feedback

- ❖ Most popular *query reformulation strategy*.
- ❖ **Cycle:**
 - ❖ User presented with list of *retrieved documents*.
 - ❖ User marks those which are *relevant*.
 - ❖ In practice: *top 10-20 ranked documents* are examined.
 - ❖ Select *important terms from documents* assessed relevant by users.
 - ❖ Enhance importance of these *terms in a new query*.
- ❖ **Expected:**
 - ❖ New *query moves towards relevant documents* and away from *non-relevant documents*.

Relevance Feedback Architecture



Relevance Feedback

- ❖ After initial searching results are presented, allow the user to *provide feedback* on the *relevance of one or more* of the retrieved documents.
- ❖ Use this *feedback information* to reformulate the query.
- ❖ Produce new results based on *reformulated query*.
- ❖ Allows more *interactive, multi-pass* process.

Pseudo Relevance Feedback

- ❖ Use *relevance feedback* methods without explicit user input.
 - ❖ Obtain *relevance feedback* automatically;
 - ❖ Identify terms related to *query terms* (e.g. *synonyms*, *stemming variations*, terms close to query terms in text)
- ❖ Just assume the top *m retrieved documents are relevant*, and use them to *reformulate the query*.
- ❖ Allows for *query expansion* that includes terms that are *correlated* with the *query terms*.
- ❖ Two *strategies*:
 - ❖ Local strategies
 - ❖ Global strategies

Local Analysis

- ❖ Examine *only documents retrieved* automatically for query to determine query expansion.
- ❖ At query time, *dynamically determine similar terms* based on analysis of *top-ranked retrieved documents*.
- ❖ Base correlation analysis on only the “*local*” set of *retrieved documents* for a specific query.
- ❖ Avoids ambiguity by *determining similar* (correlated) terms only within *relevant documents*.
 - ❖ “Apple computer” → “Apple computer Power book laptop”

Global Analysis

- ❖ *Expand query using information* from whole set of documents in collection.
 - ❖ Determine term *similarity* through a *pre-computed statistical analysis* of the complete corpus.
- ❖ Thesaurus-like structure using all documents:
 - ❖ Approach to *automatically built thesaurus*.
 - ❖ (e.g. similarity thesaurus based on co-occurrence frequency)
 - ❖ Approach to *select terms for query expansion*.
- ❖ A thesaurus provides information on *synonyms* and *semantically related words* and phrases.
- ❖ Example: physician
 - similar/synonymous: doctor, medical, MD
 - related: general practitioner, surgeon

Thesaurus-based Query Expansion

- ❖ For *each term*, t, in a query, expand the query with synonyms and related words of t from the *thesaurus*.
- ❖ May *weight added terms* less than original query terms.
 - ❖ Generally increases *recall*.
- ❖ May significantly *decrease precision*, particularly with ambiguous terms.
 - ❖ “interest rate” → “interest rate fascinate evaluate”

Global vs. Local Analysis

- ❖ Global analysis requires *intensive term correlation computation* only *once at system development time*.
 - ❖ Local analysis requires *intensive term correlation computation* for every query at run time (although number of terms and documents is less than in global analysis).
- ❖ But *local analysis* gives better results.
 - ❖ Term ambiguity may introduce *irrelevant statistically* correlated terms during *global analysis*.
 - ❖ “Apple computer” → “Apple red fruit computer”

Global Analysis Refinements

- ❖ Only expand query with terms that are *similar to all terms* in the *query*.

$$\text{sim}(k_i, Q) = \sum_{k_j \in Q} c_{ij}$$

- ❖ “fruit” not added to “Apple computer” since it is far from “computer.”
- ❖ “fruit” added to “apple pie” since “fruit” close to both “apple” and “pie.”
- ❖ Use *more sophisticated term weights* (instead of just frequency) when computing *term correlations*.

Query Expansion Conclusions

- ❖ Expansion of queries with related terms can *improve performance*, particularly *recall*.
- ❖ However, must select similar terms very carefully to *avoid problems*, such as loss of *precision*.

Question & Answer



Thank You !!!