Assignment 1

Tutorial Week 4

By now …

- Have installed Maven and Lucene
- Tried editing the main java class examples, compiling and running Lucene to build an index

Today

- Use the cranfield collection
- Download the documents
- Index the 1400 document collection using Lucene -
  - Preprocessing
  - Choice of analyzers
- Query the index (with 225 test queries)
- Evaluate the search engine you just created using trec eval - software for evaluation of IR systems

Download the documents

- http://ir.dcs.gla.ac.uk/resources/test_collections/cran/
- tar -xzf cran.tar.gz
  - cran.all.1400
  - cranqrel
  - cran.qry

Cranfield Collection

- 3 Files in download

1) Queries File - to search
2) Documents File - to find
3) Relevance Judgements File – to evaluate the search

1) Queries file – cran.qry

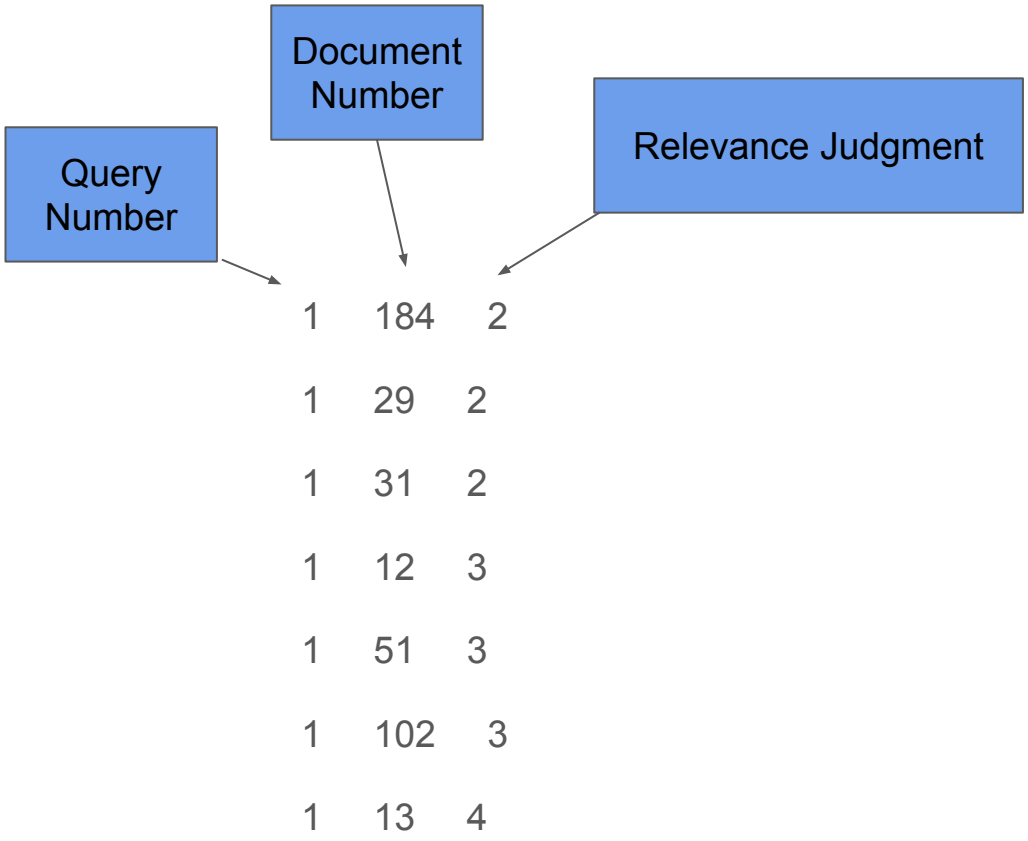
- 225 Example queries
- Example: *What similarity laws must be obeyed when constructing aeroelastic models of heated high speed aircraft*
- Lines beginning with .I indicate query id number to follow
- Lines beginning with .W indicate textual query to follow
- Queries have a (non sequential) id number – ids do not correspond to qrels file but the order does, i.e. qrel id=1 corresponds to the first query in crn.qry file, qrel id=3 is the 3rd query in qry file (although labelled with id 004 there)
- Note also, they are all questions – not question marks needed at end

2) Documents File – cran.all.1400

- Contains a set of 1,400 documents containing answers to the questions in the query file (1)
- .I id number of document
- .T title of the document
- .A author of the document
- .W textual document
- Note: other attributes of documents can also be present using other identifiers (can ignore these)

3) Relevance Judgments File - cran.qrel

- Note: no header
- 1836 relevance judgments
- Each query can have a different number of relevant documents associated with it e.g doc 1 has 29 relevant documents (rel judgments 1-4; doc 2 has 25, … min 2 documents, max = 40)
- Queries numbered here 1 to 225

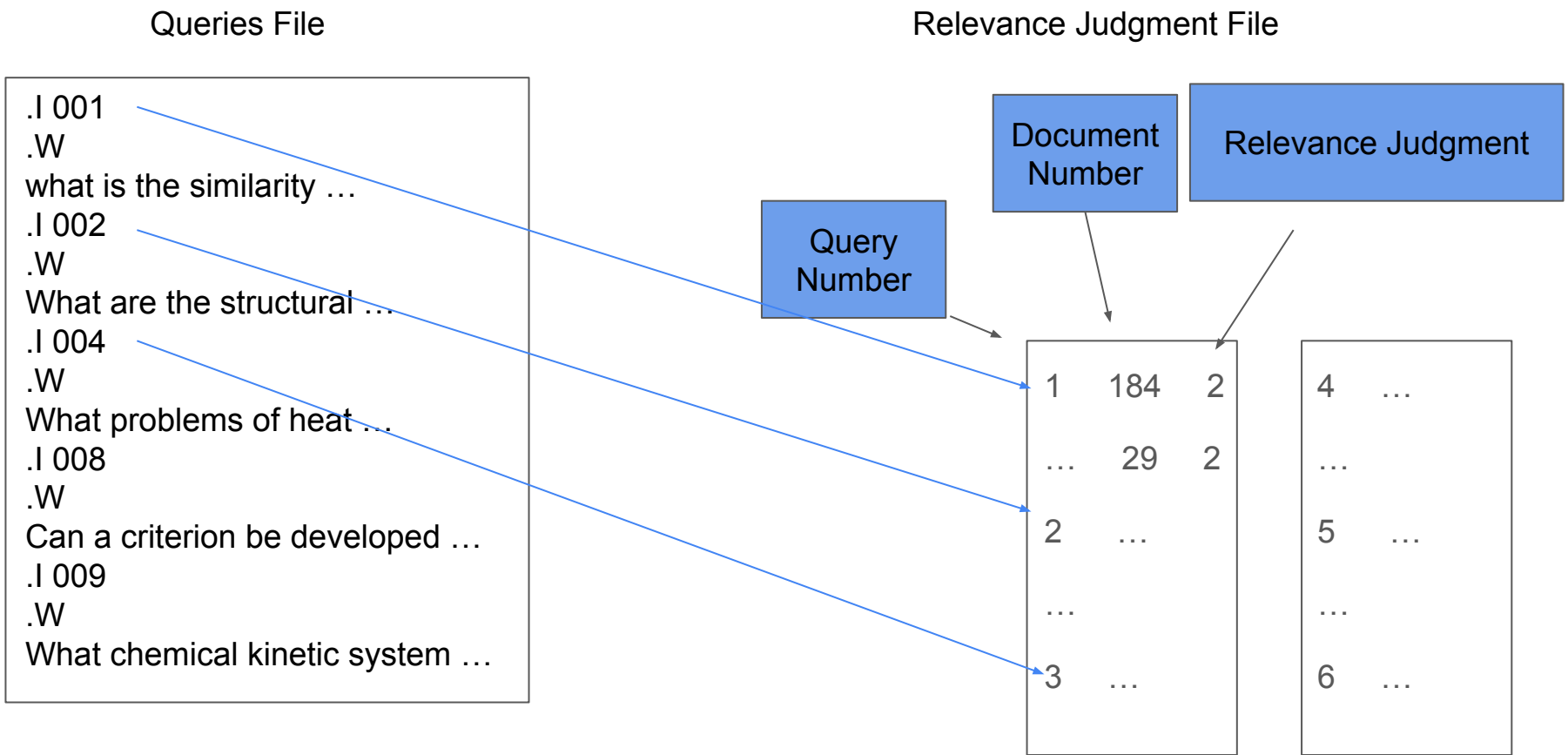


Relevance Judgments

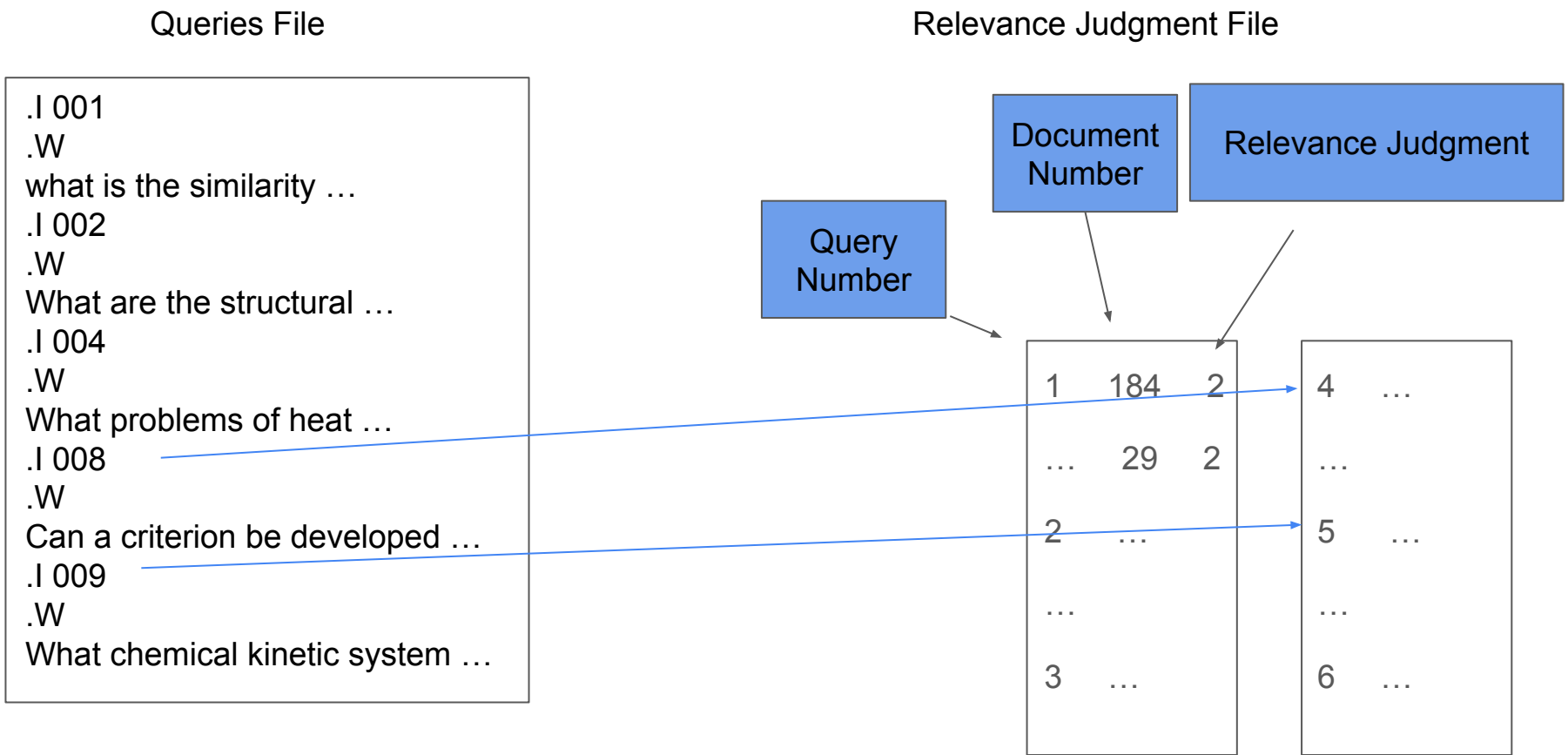
- 1 = Complete answer
- 2 = high degree of relevance
- 3 = useful
- 4 = minimum interest
- 5 = no interest

Note: no 5’s included as only documents of some relevance for the query were annotated

Combining Query and Relevance Judgments Files



Combining Query and Relevance Judgments Files



Next Step: Index the documents

- Index the 1400 document collection using Lucene -
  - Preprocessing (see Lucene documentation)
  - Choice of analyzers (see Lucene documentation)

Next Step: Query the index

- For each of the 225 cranfield textual queries
  - Apply the same preprocessing as documents
  - Apply a similarity method and scoring method to each 225 x 1400 query document pairs
  - Record the top ranked 50 documents for each query and importantly their scores
- Take for example, query 1
  - You will now have a record of the top 50 scoring documents for query 1 out of the overall 1400
- Then do this for other 224 queries
  - You will now have 50*225 scored documents
  - All of this can be stored in a single file
  - To know the format required – need to look at evaluation code – trec_eval

Evaluate the method

- To evaluate how well your search algorithm works we will use: trec eval software
- wget https://trec.nist.gov/trec_eval/trec_eval-9.0.7.tar.gz
- Extract the files: tar -xzf trec_eval-9.0.9.tar.gz
- Look for usage: in README for installation – very straightforward – make
  - Inside trec_eval-9.0.7 enter “make”
  - Then test the installation using “make quicktest”
- Run trec_eval on your results file
  - ./trec_eval <qrels file corrected for trec_eval> <results file>
  - P_5 in the output shows your precision@5 map score
  - Use all precision points in report