Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

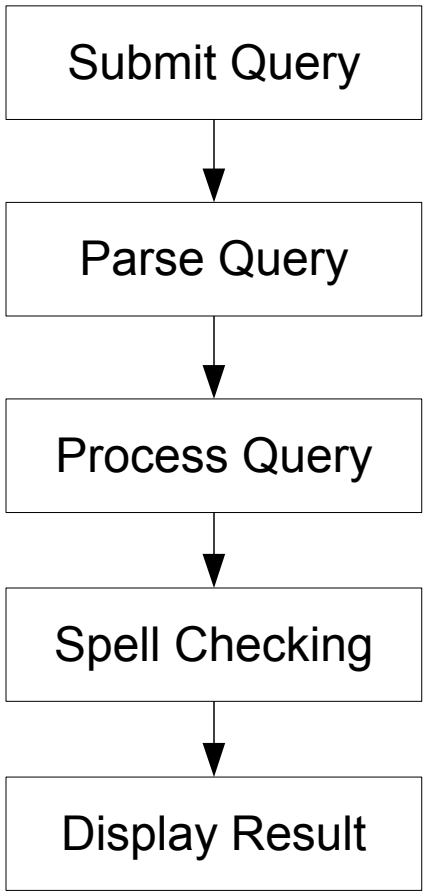
Search Engines 09

Query Processing

July 21, 2009

Alexander Lüders | Kai Schlichting

Query Processing Overview



"albert einstein" Go

Page: **1**

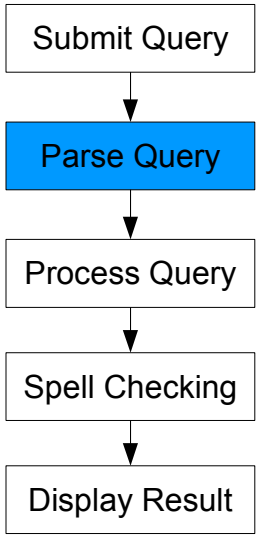
Albert Einstein
Snippet for article #736
Score: 25.77

Arthur Stanley Eddington
Snippet for article #2274
Score: 18.45

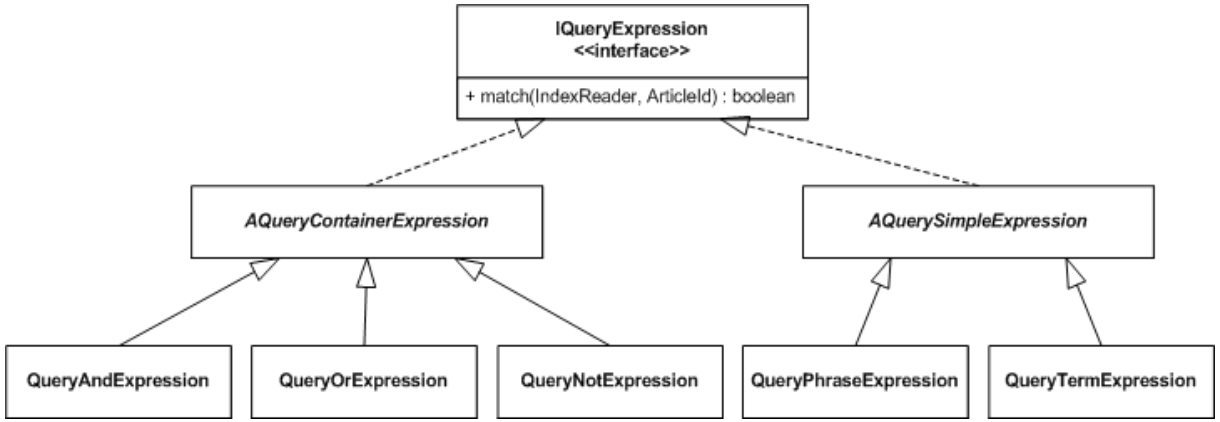
Albert
Snippet for article #1504
Score: 12.21

Atomic theory
Snippet for article #2844
Score: 11.62

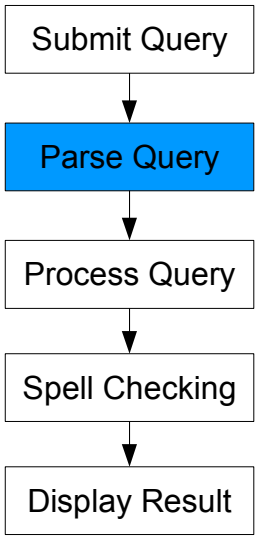
Query Parsing (1/2)



- Boolean query language
  - AND (implicit), OR, NOT, phrases
  - “albert einstein” OR schweitzer
- Extensible query language
  - Specification of a grammar (jjtree parser generator)
  - Class model

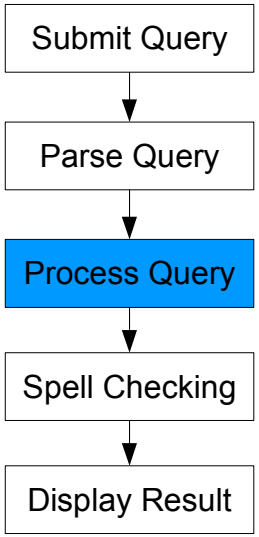


Query Parsing (2/2)

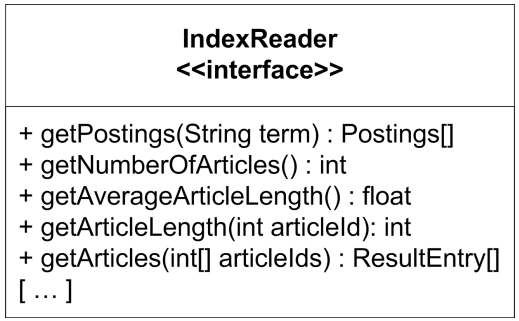


- Query preprocessing
  - Remove invalid chars (valid: a-z-A-Z0-9-_")
  - Remove " if odd count
  - Stemming (Porter)
  - Lower case

Process Query

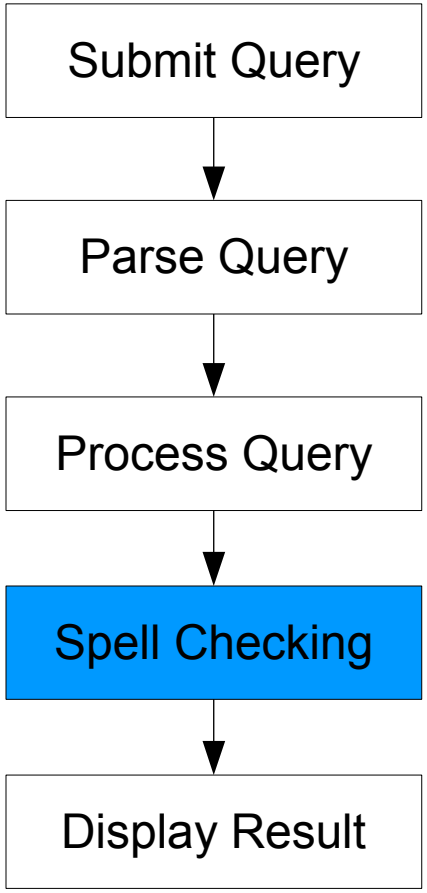


- Term-at-a-time algorithm
  - Faster due to sequential disk access
  - Increased memory consumption
- IndexReader: Interface to the index

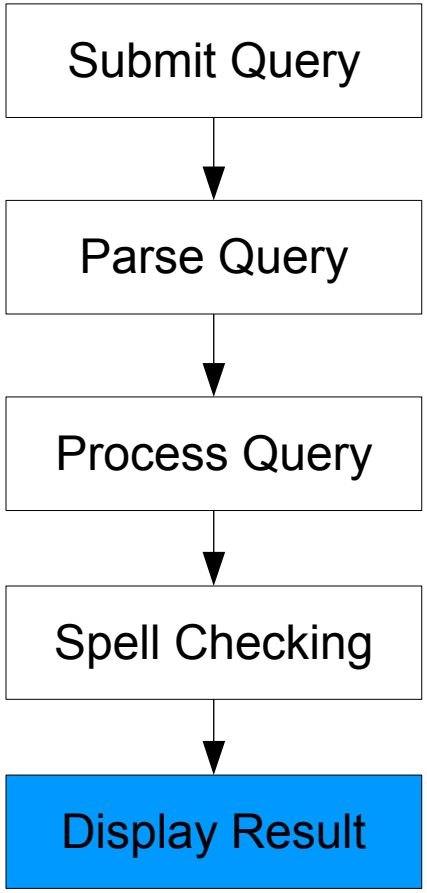


- Ranking: BM25 algorithm

Spell Checking



Demo


Submit Query
Parse Query
Process Query
Spell Checking
Display Result

Performance Analysis

- 30 – 500 ms per query
- Connection pooling
  - Establishing and closing connection is very slow
  - Connection pool opens and holds limited number of connections
- Caching of inverted lists
  - Fast results of queries with cached query terms
  - Fast “next page” access

Conclusion

- Features
  - Ranking
  - Spell checking
  - Extensible boolean query language
  - Replaceable index layer
- Further performance optimizations possible, but not absolutely necessary

Thank You For Your Attention!

Questions?