ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN

Serminar môn học: Hệ thống tìm kiếm thông tin

APACHE LUCENE

Giảng viên hướng dẫn: PGS.TS. Hồ Bảo Quốc
Trình bày:
- 20C12030 - Huỳnh Lâm Phú Sĩ
- 20C12007 - Trần Đình Lâm

Ngày 10 tháng 07 năm 2021

LUCENE

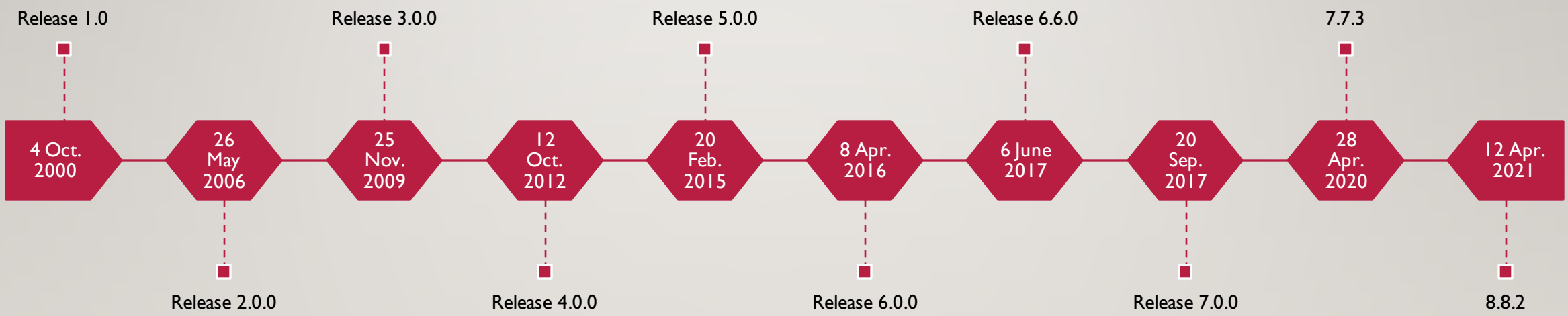
- Tổng quan
- Mô hình quan niệm
- Cài đặt vật lý
- Demo

TỔNG QUAN

- Thư viện Java mã nguồn mở
- Lập chỉ mục (indexing) và tìm kiếm (searching) mạnh mẽ
- Tác giả gốc: Doug Cutting (1999), Co-founder Apache Hadoop
- Core của các dự án lớn nổi tiếng:
  - Apache Solr
  - Elastic Search (2010)
  - Apache Nutch
  - MongoDB Atlas Search

elasticsearch

TỔNG QUAN - CÁC PHIÊN BẢN LUCENE

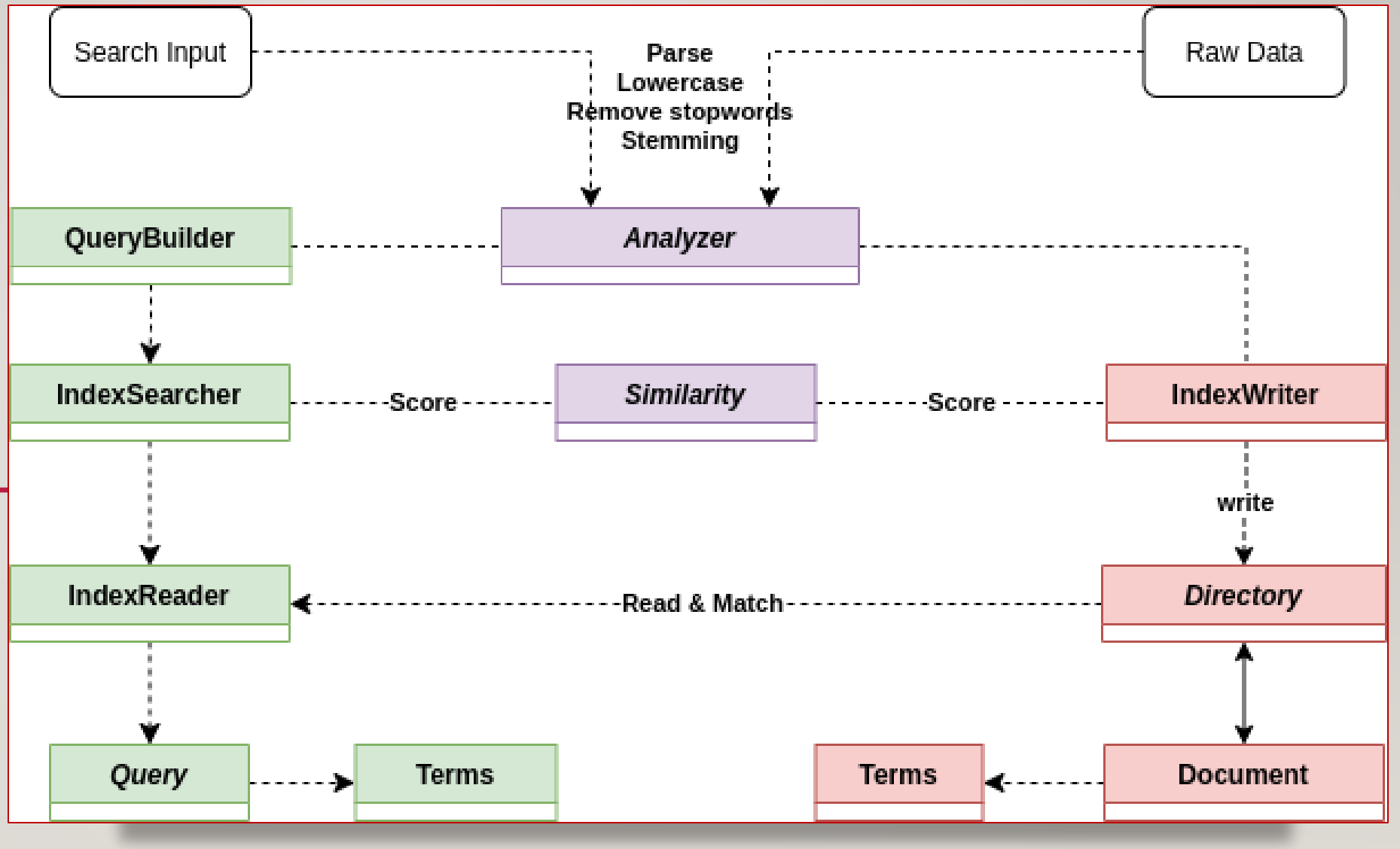


TỔNG QUAN - ĐẶC TRƯNG

- 100% Java
- Hiệu năng cao
- Gọn nhẹ, ít tốn bộ nhớ
- Search chính xác, nhanh, đa dạng loại query
- Thiết kế hướng đối tượng, linh hoạt và dễ thay thế, mở rộng
- Đa nền tảng, cộng đồng lớn mạnh
- Mở rộng nhiều ngôn ngữ (Pascal, Perl, C#, C++, Python, Rupy, PHP)

MÔ HÌNH QUAN NIỆM CHUNG

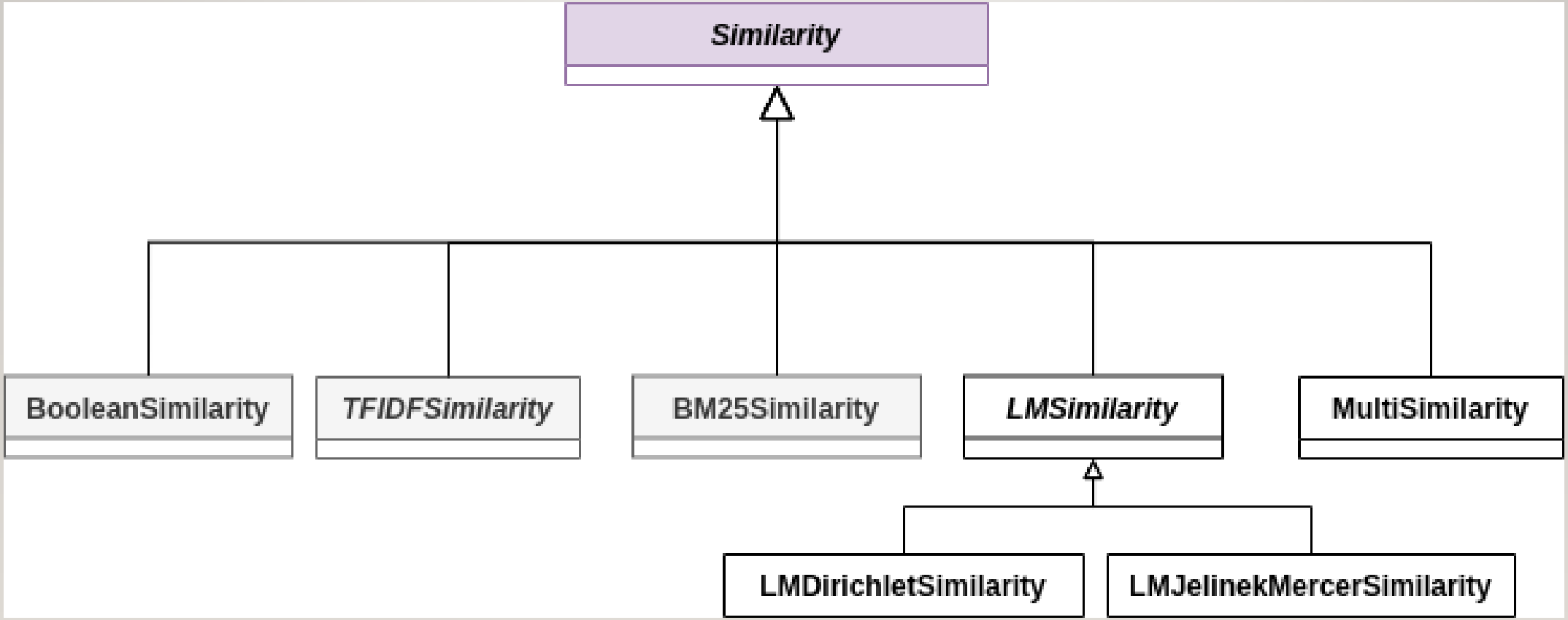
MÔ HÌNH LUCENE



ANALYZERS

WhitespaceAnalyzer	• [The] [JHUG] [meeting] [is] [on] [this] [Saturday]
SimpleAnalyzer	• [the] [jhug] [meeting] [is] [on] [this] [Saturday]
StopAnalyzer	• [jhug] [meeting] [saturday]
KeywordAnalyzer	
StandardAnalyzer	• [jhug] [meeting] [Saturday]

SIMILARITIES



BM25 SIMILARITY

- Mô hình BM25 là một trong những mô hình nổi tiếng nhất của các mô hình tìm kiếm được phát triển từ mô hình xác suất
- Đây không phải là một phương pháp đơn lẻ mà là một tập hợp các phương pháp tính điểm với các thành phần và thông số khác nhau
- Công thức cài đặt:

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

TFIDF SIMILARITY

- Mô hình Vector sử dung độ tương đồng TFIDF là một trong những mô hình so khớp được cài đặt sẵn trong Lucene.
- Là sự kết hợp của Boolean model (chọn document) và Vector Space Model (tính điểm tương đồng)

$$\text{cosine-similarity}(q,d) = \frac{V(q) \cdot V(d)}{|V(q)|\,|V(d)|}$$

VSM Score

CÔNG THỨC QUAN NIỆM

$$score(q,d) = coord\text{-}factor(q,d) \cdot query\text{-}boost(q) \cdot \frac{V(q) \cdot V(d)}{|V(q)|} \cdot doc\text{-}len\text{-}norm(d) \cdot doc\text{-}boost(d)$$

Lucene Conceptual Scoring Formula

- coord-factor: số term trong query nằm trong document đang xét
- Query-boost: chỉ số được xác định vào thời điểm truy vấn
- doc-boost: độ quan trọng của tài liệu, được thiết lập bởi người dùng
- doc-len-norm: chuẩn hoá Euclidean để chuẩn hoá document về vector đơn vị

CÔNG THỨC CÀI ĐẶT

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} \left(\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d) \right)$$

Lucene Practical Scoring Function

tf(t in d): term frequency được tính bằng công thức

$$\text{tf}(t \text{ in } d) = \text{frequency}^{1/2}$$

Idf(t): inverse document frequency được tính bằng công thức

$$\text{idf}(t) = 1 + \log\left(\frac{\text{numDocs}}{\text{docFreq}+1} \right)$$

CÔNG THỨC CÀI ĐẶT

Coord(d,q): được tính bằng số từ của query được tìm thấy trong document chia cho số từ của query

QueryNorm(q): nhân tố chuẩn hoá để tính điểm mà có thể so sánh giữa các truy vấn.
Nó được tính bởi lớp Similarity và chỉ ảnh hưởng và thời gian tìm kiếm.

$$\text{queryNorm(q)} = \text{queryNorm(sumOfSquaredWeights)} = \frac{1}{\text{sumOfSquaredWeights}^{1/2}}$$

$$\text{sumOfSquaredWeights} = \text{q.getBoost()}^2 \cdot \sum_{\text{t in q}} \left(\text{idf(t)} \cdot \text{t.getBoost()} \right)^2$$

CÔNG THỨC CÀI ĐẶT

Norm(t,d): là đặc trưng cho các nhân tố boost và length (gồm document boost và field boost). Khi document được đánh chỉ mục, tất cả các thành phần đều phải được nhân lên

$$\text{norm}(t,d) = \texttt{doc.getBoost()} \cdot \texttt{lengthNorm(field)} \cdot \prod_{\text{field } f \text{ in } d \text{ named as } t} \texttt{f.getBoost()}$$

MÔ HÌNH CÀI ĐẶT

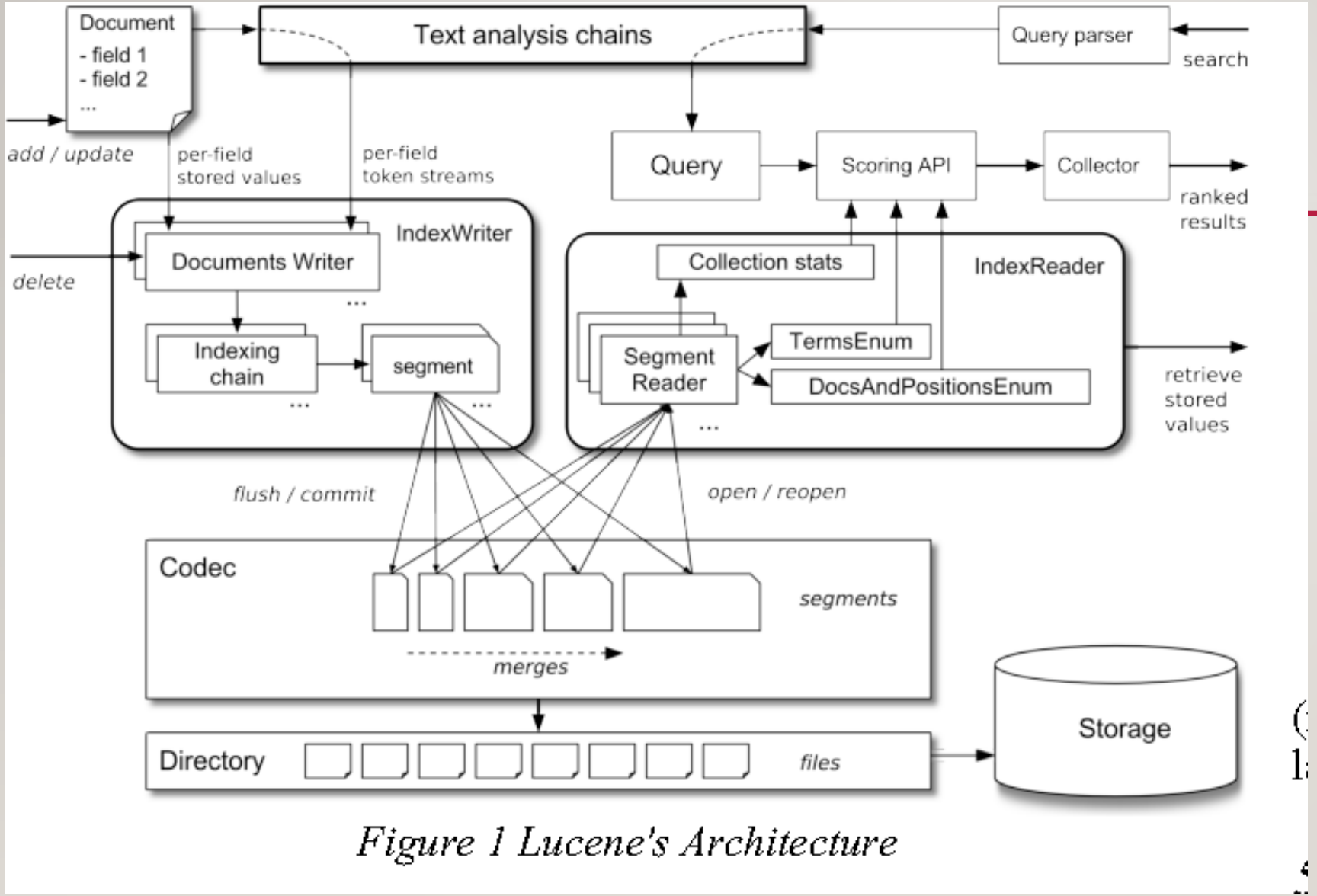
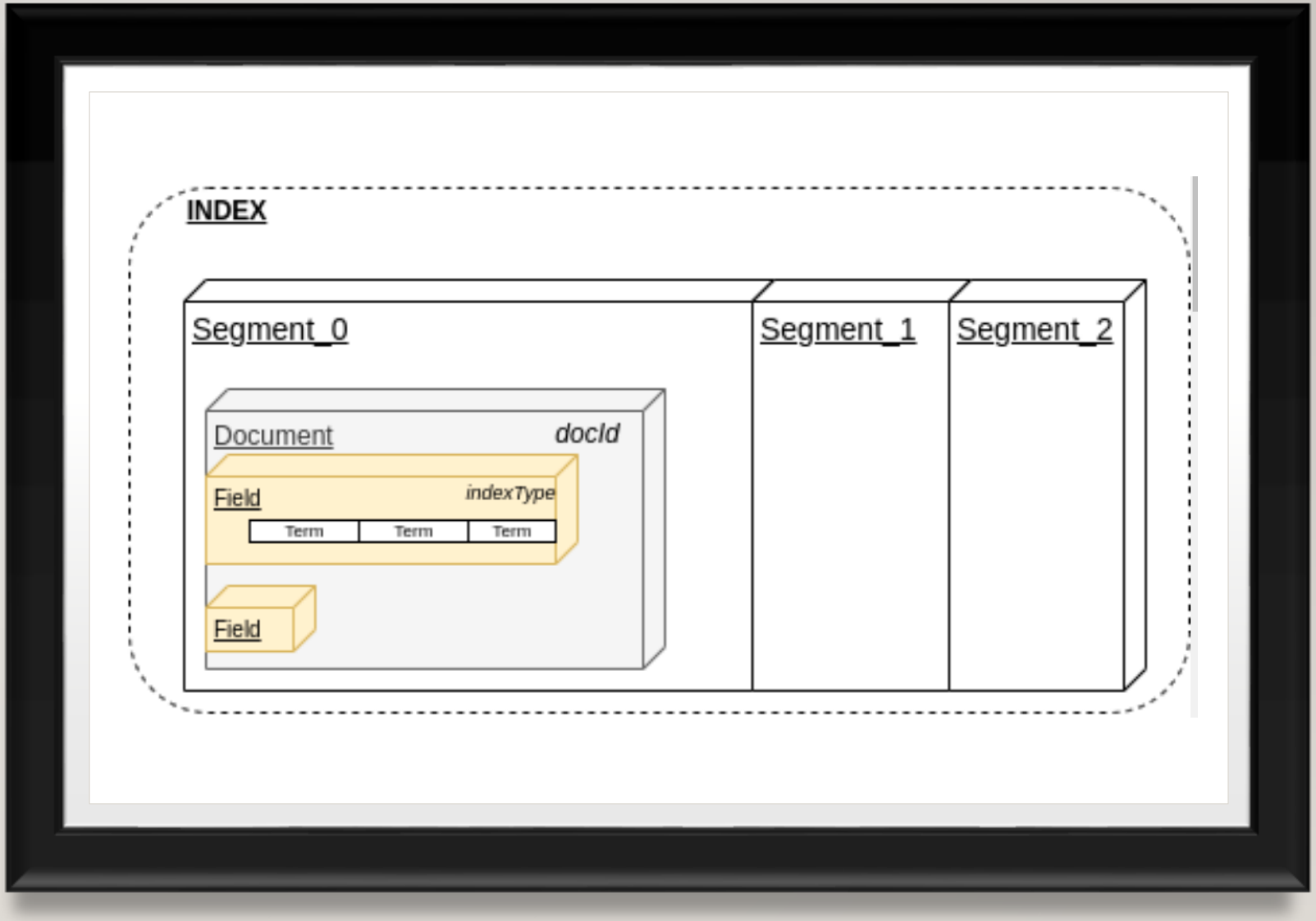
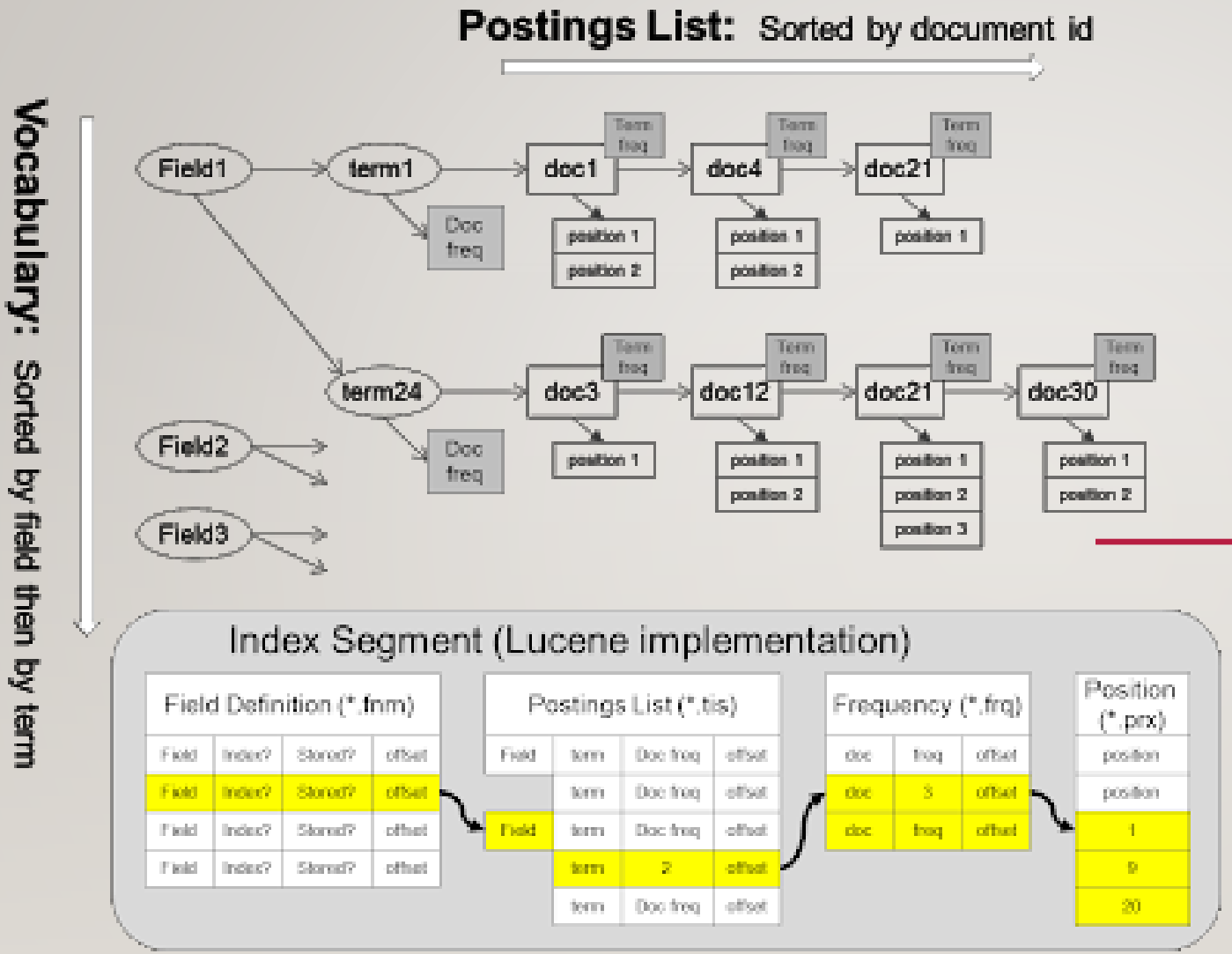


Figure 1 Lucene's Architecture

INDEX

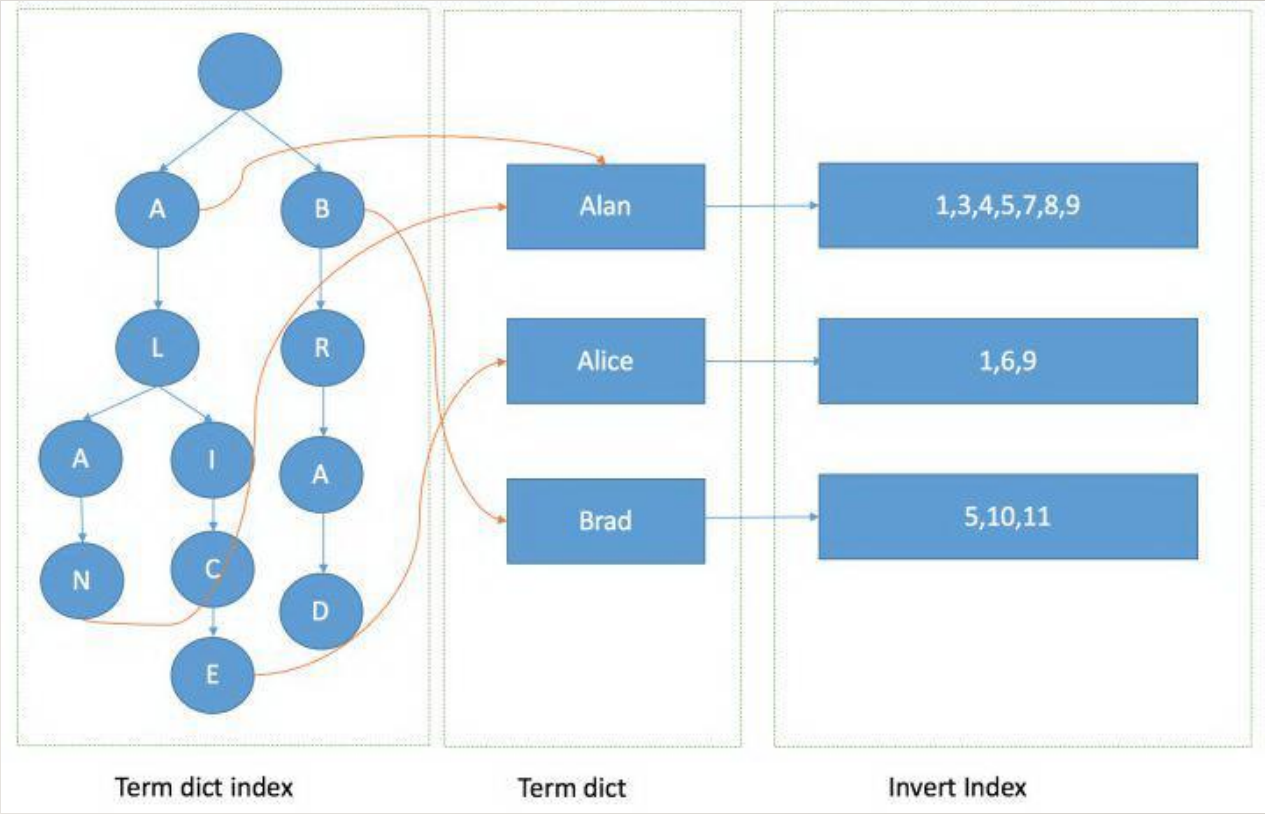

INDEX
Segment_0
Segment_1
Segment_2
Document
docId
Field
indexType
Term
Term
Term
Field

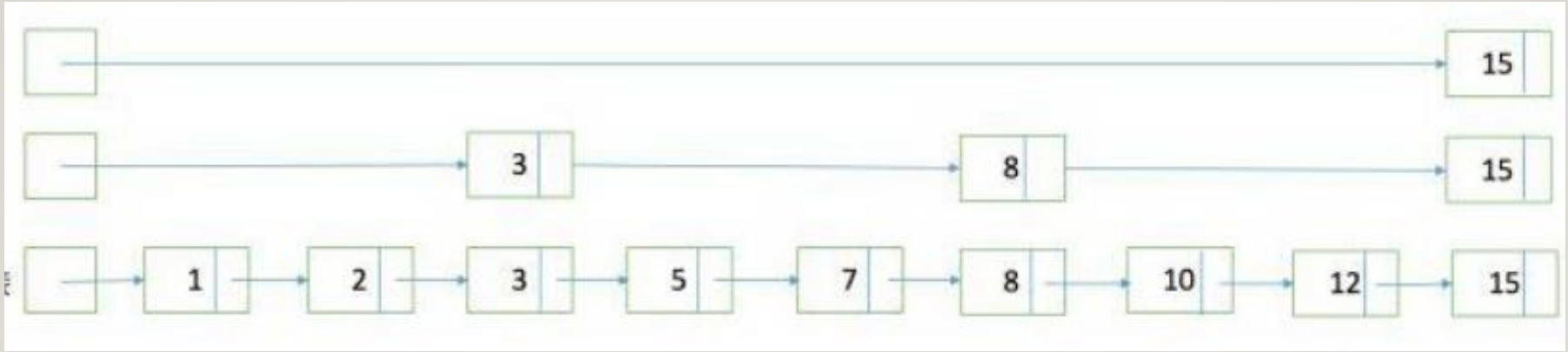
FIELDS


Postings List: Sorted by document id
Vocabulary: Sorted by field then by term
Field1
term1
Doc freq
doc1
doc4
doc21
Term freq
position 1
position 2
term24
doc3
doc12
doc21
doc30
position 3
Field2
Field3
Index Segment (Lucene implementation)
Field Definition (*.fnm)
Field
Index?
Stored?
offset
Postings List (*.tis)
term
Doc freq
Frequency (*.frq)
doc
freq
Position (*.prx)
position

INDEX FILE

Term dict index





Skip list

DEMO

- Triển khai một ví dụ suggester & spell checker

TÀI LIỆU THAM KHẢO

- Erik Hatcher - Lucene in Action (2004)
- Andrzej Białecki - Apache Lucene 4 (2012)
- Apache Lucene - lucene.apache.org