DIS17 – Search Engine Technologies

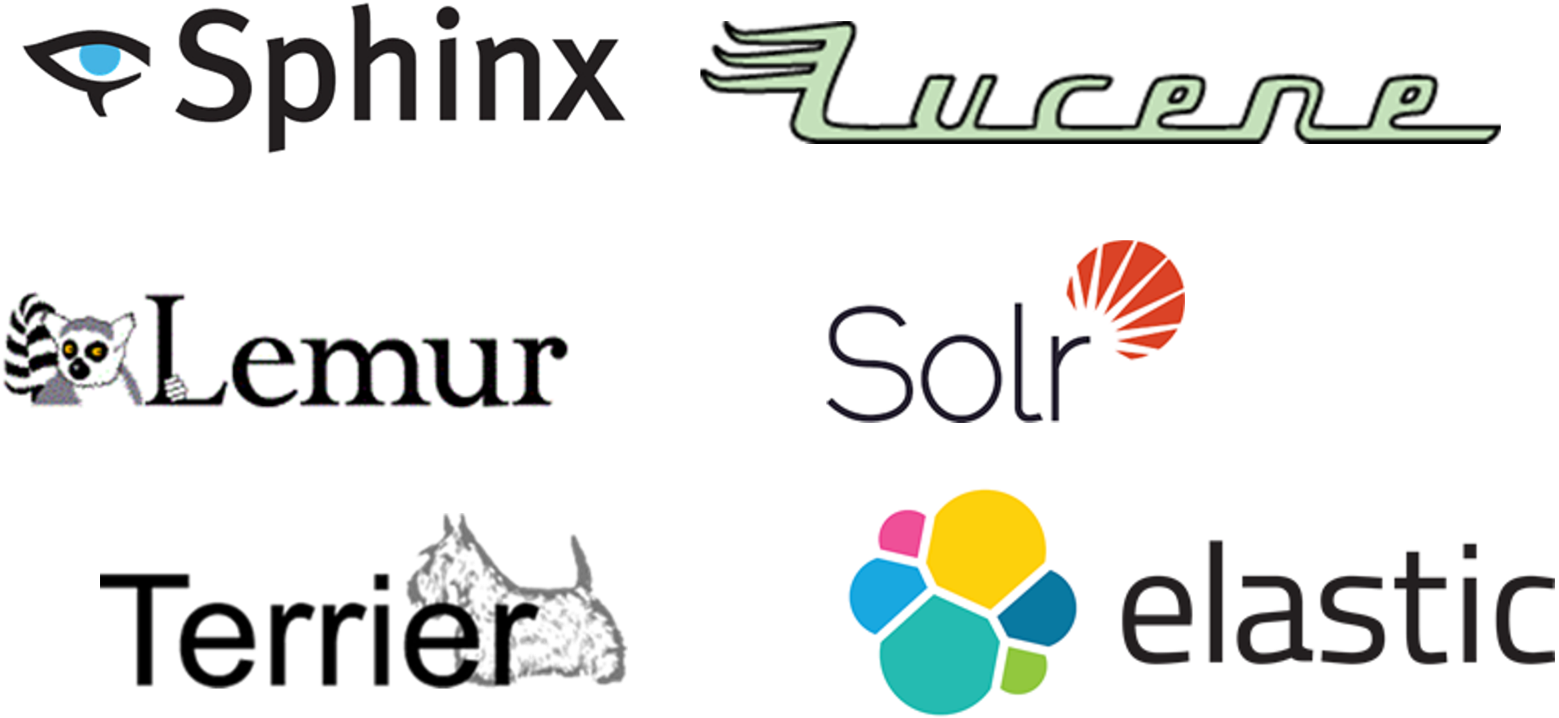
03 – Introduction to Solr

Philipp Schaer, Technische Hochschule Köln, Cologne, Germany

Version: WS 2021

Technology
Arts Sciences
TH Köln

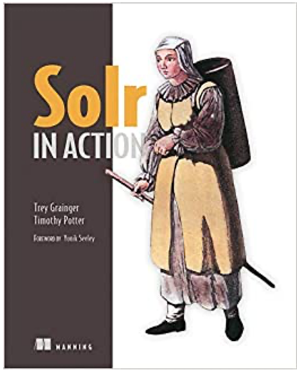
Overview of prominent search engines



Good reads on Solr and Elasticsearch

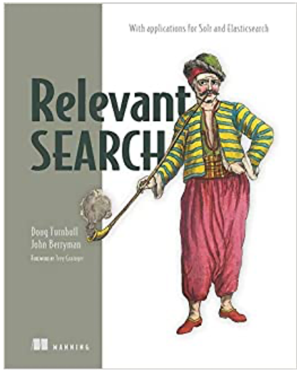
First Chapter of “Solr in Action”

- great overview of the basics of search
- great overview of the and functionality of Solr
- https://www.manning.com/books/solr-in-action



Third Chapter of “Relevant Search”

- detailed description on how to use ElasticSearch
- default ranking and its problem
- how to make it better
- https://www.manning.com/books/relevant-search


With applications for Solr and Elasticsearch
Relevant
SEARCH
Doug Turnbull
John Berryman
Foreword by Trey Grainger
MANNING

Solr

- is a **search engine** (no database!)
- based on **Lucene**
- provides **out-of-the-box** indexing functionalities
- works over **HTTP** and according to the REST principle
  → use CURL, Python API etc.
- has also a **GUI**, if you prefer clicking over coding
- is **open source** (Apache license)
- focuses on **text data**
- and much more. …

In Solr it‘s all about text!

Solr is a search engine that **focuses on text**, which means it‘s

- **Text-centric** (handles unstructured text well, as opposed to a database, which tends to focus on structured data)
- **Read-centric** (more content is read out of a search engine than written into it)
- **Document-centric** (the indexed items are flat units of information, mostly documents - no multimedia, no network data, etc.)
- **Flexible schema** (the data to be indexed does not have to follow the same format and structure - there may also be different distributions of content)

Where Solr struggles...

There are things for which Solr is not well suited, e.g.

- when **more than the usual 10-100** documents are expected as a result, e.g. 1 million result documents;
- when **large subsets** of the index are to be analyzed;
- when **relationships** between documents are important;
- when **access rights** and **security** are important;

Again, in all clarity:

- Solr is not a web search engine like Google or Bing!
- Solr has nothing to do with Search Engine Optimization (SEO)!

Solr – an awesome community

- Solr provides a lot of information about itself, and the Solr community is very active in helping each other with issues.
- There is also a very extensive and extraordinary well-explained Tutorial about various aspects of solr:
  - https://solr.apache.org/guide/8_10/solr-tutorial.html
  - very illustrative from scratch on
  - all you need to start with Solr

This will be your **first assignment!**
Work on the tutorial, to get a feeling for the software…

ElasticSearch

- is a full-text, distributed **NoSQL database**
- also based on **Lucene**
- also **Open Source**
- also uses **RESTful** for communication
- can index many different types of content, not only on text
  - Geodata, Business data
- is a store, a search and an **analytics engine**

Where ElasticSearch **struggles**:

- Has a slight latency in indexing.
- Does not support permission management (security)

Solr vs. ElasticSearch

Querying

- Solr: Uses simple URL parameters
- ElasticSearch: Uses JSON

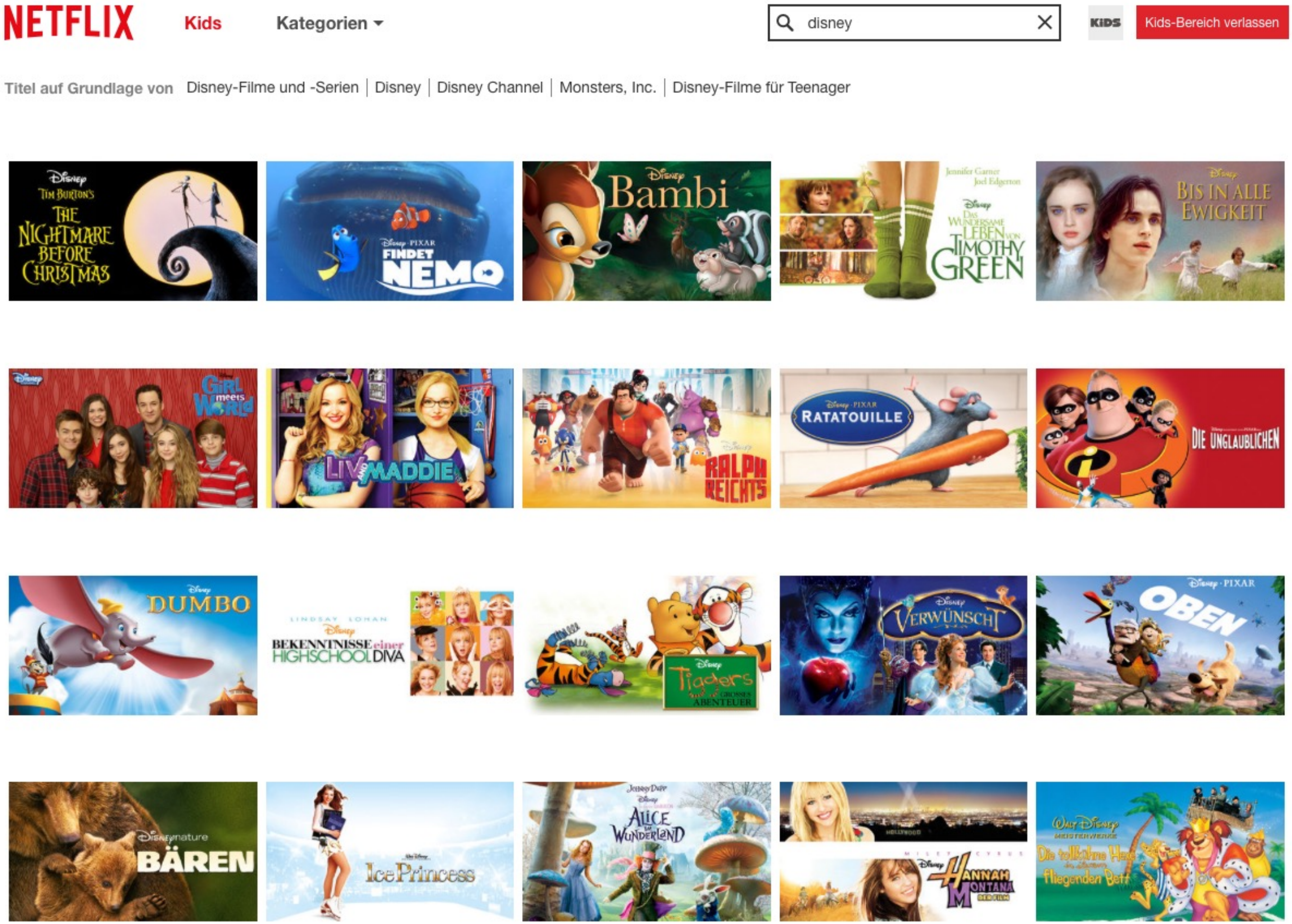
Advantage Solr

- working with static data (e-commerce), as it uses an uninverted reader for faceting and sorting

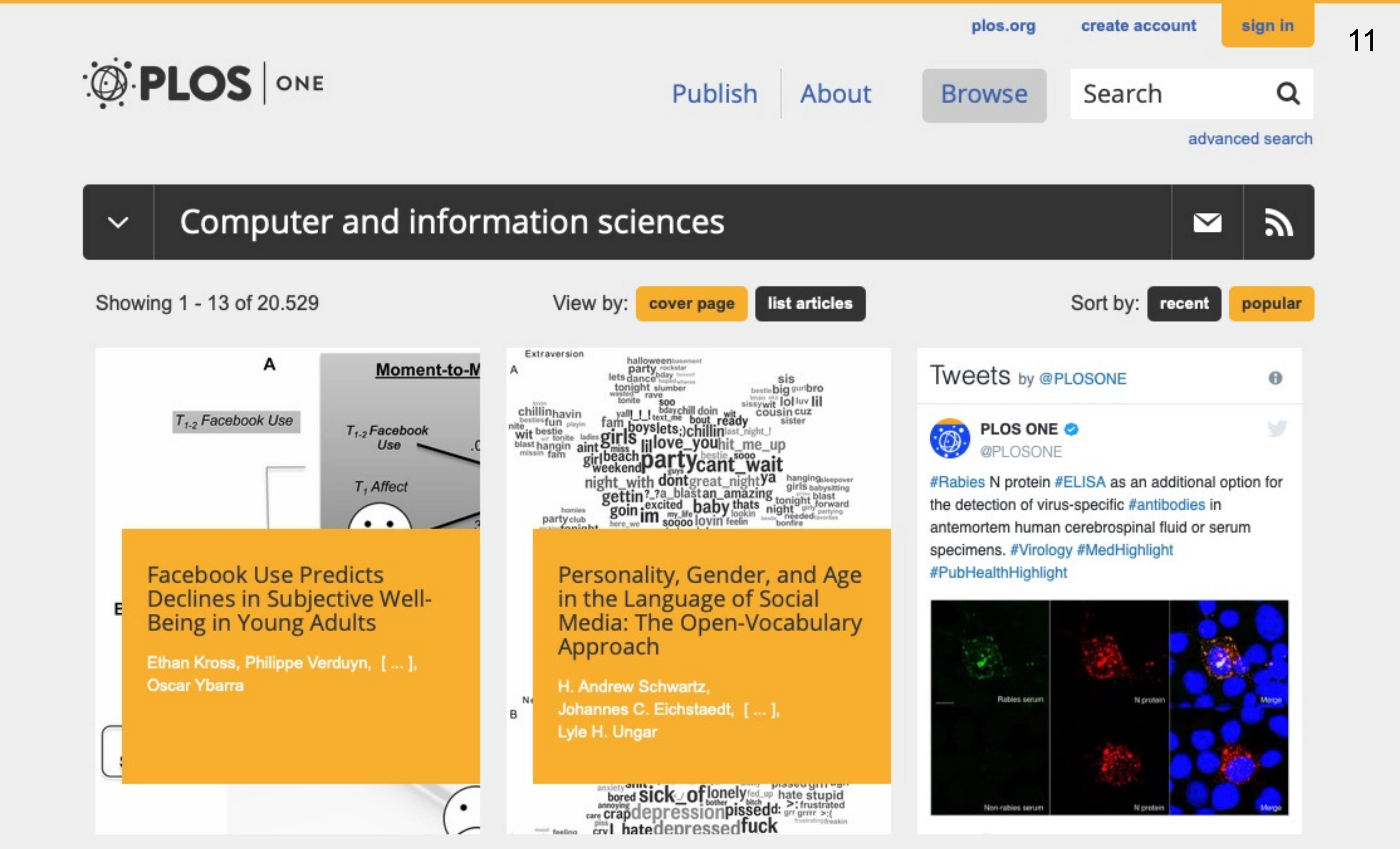
Advantage ElasticSearch

- working with time series, like log analysis, business analytics, etc.

There are a lot of differences, too many to cover, but if you're interested, here is a good review: https://sematext.com/blog/solr-vs-elasticsearch-differences/


NETFLIX
Kids
Kategorien
disney
KIDS
Kids-Bereich verlassen
Disney Tim Burton's The Nightmare Before Christmas
Disney · Pixar Findet Nemo
Disney Bambi
Jennifer Garner Joel Edgerton Disney Das wundersame Leben von Timothy Green
Disney Bis in alle Ewigkeit
Disney Girl meets World
Liv and Maddie
Disney Ralph reichts
Disney · Pixar Ratatouille
Die Unglaublichen
Disney Dumbo
Lindsay Lohan Disney Bekenntnisse einer Highschool Diva
Disney Tiggers grosses Abenteuer
Disney Verwünscht
Disney · Pixar Oben
Disneynature Bären
Walt Disney Ice Princess
Johnny Depp Disney Alice im Wunderland
Miley Cyrus Disney Hannah Montana der Film
Walt Disney Meisterwerke Die tollkühne Hexe in ihrem fliegenden Bett

Titel auf Grundlage von Disney-Filme und -Serien | Disney | Disney Channel | Monsters, Inc. | Disney-Filme für Teenager



Computer and information sciences

Showing 1 - 13 of 20.529

View by: cover page list articles

Sort by: recent popular

Facebook Use Predicts Declines in Subjective Well-Being in Young Adults

Ethan Kross, Philippe Verduyn, […], Oscar Ybarra

Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach

H. Andrew Schwartz, Johannes C. Eichstaedt, […], Lyle H. Ungar

Tweets by @PLOSONE

PLOS ONE @PLOSONE

#Rabies N protein #ELISA as an additional option for the detection of virus-specific #antibodies in antemortem human cerebrospinal fluid or serum specimens. #Virology #MedHighlight #PubHealthHighlight

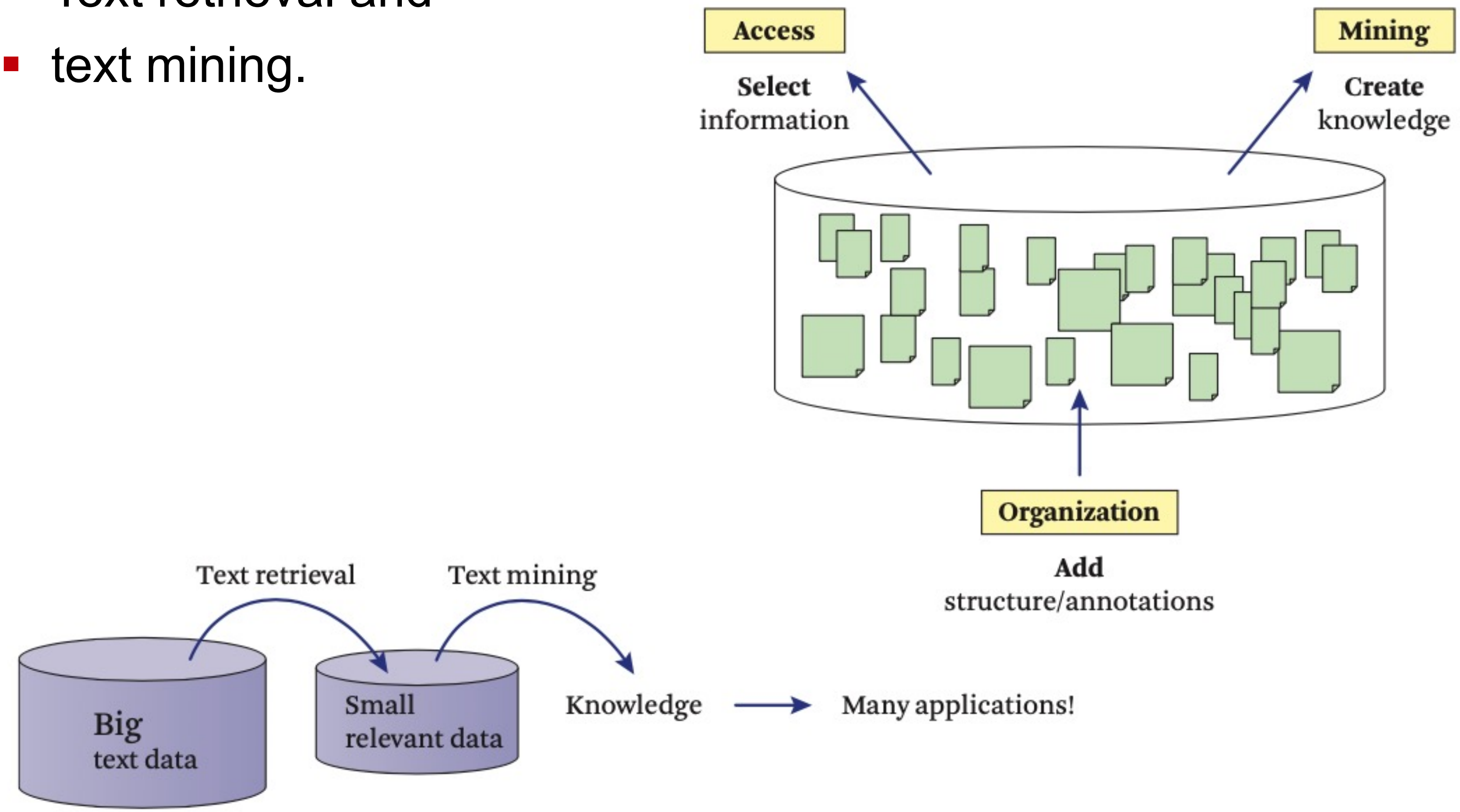
http://api.plos.org/search?q=title:DNA

http://api.plos.org/solr/search-fields/

Text Information Processing (TIP)

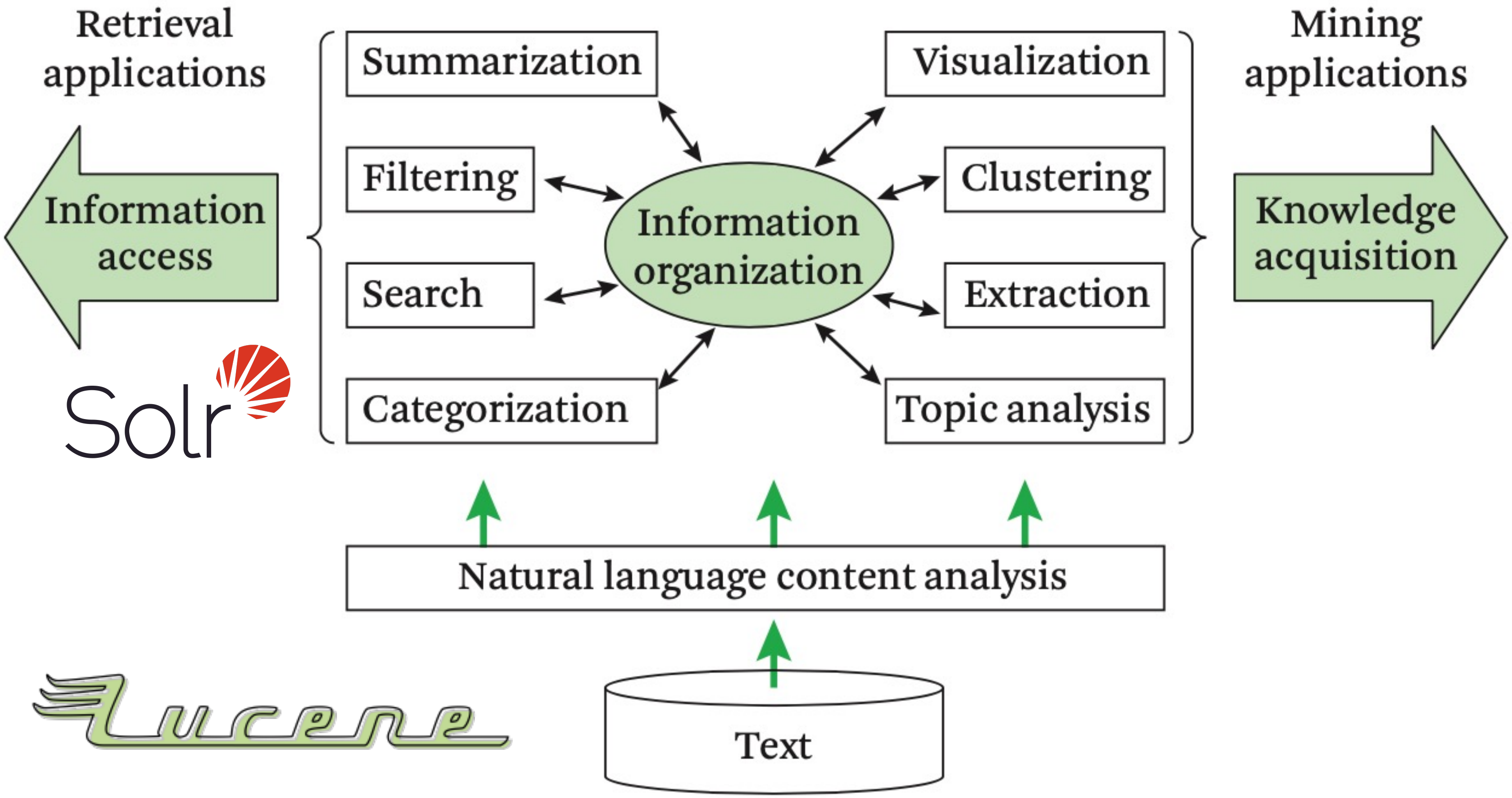
Two main techniques for analyzing big text data:

- Text retrieval and
- text mining.



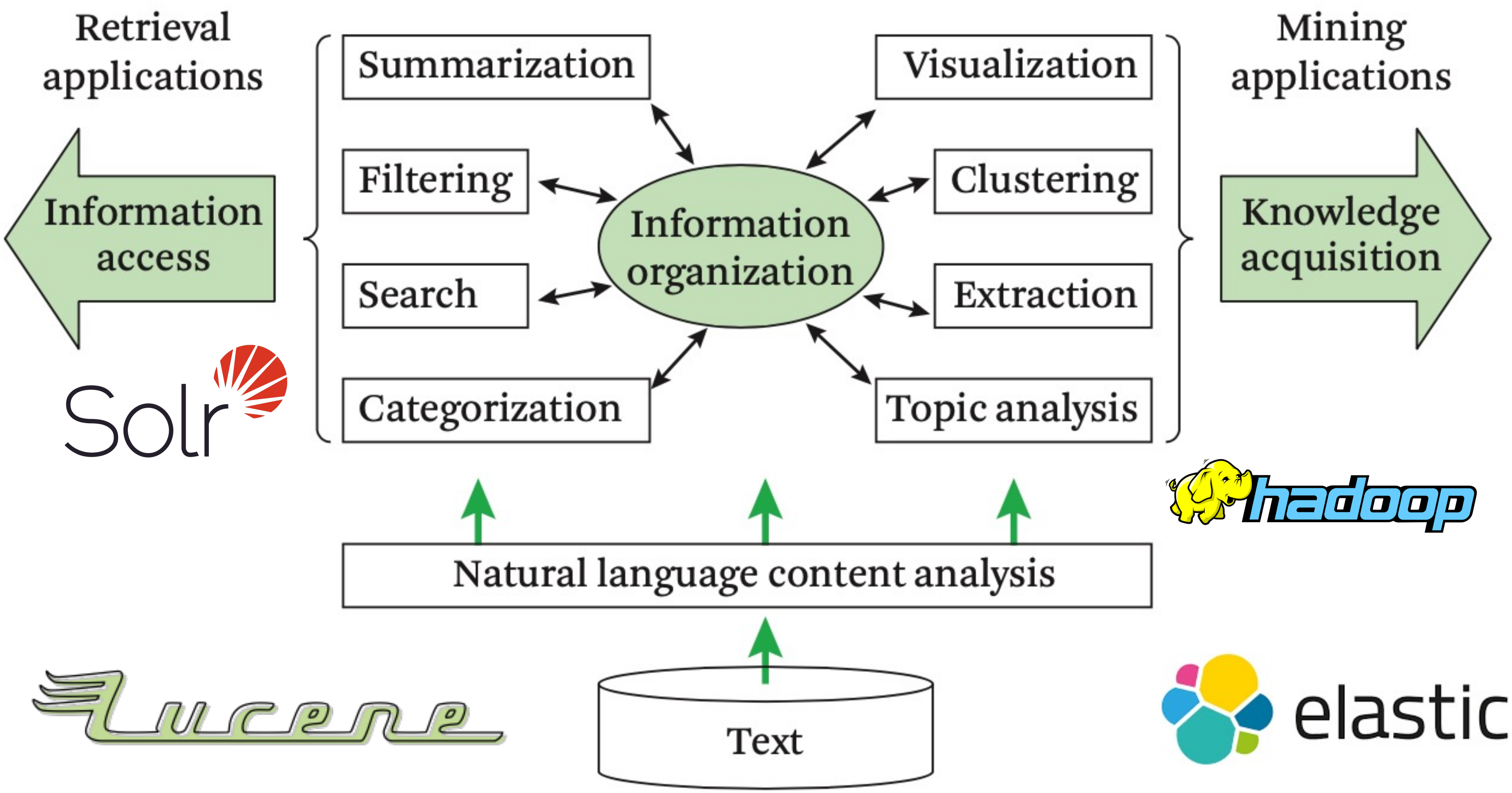
Zhai & Massung (2016), p. 5 and 7

Conceptional Framework TIP



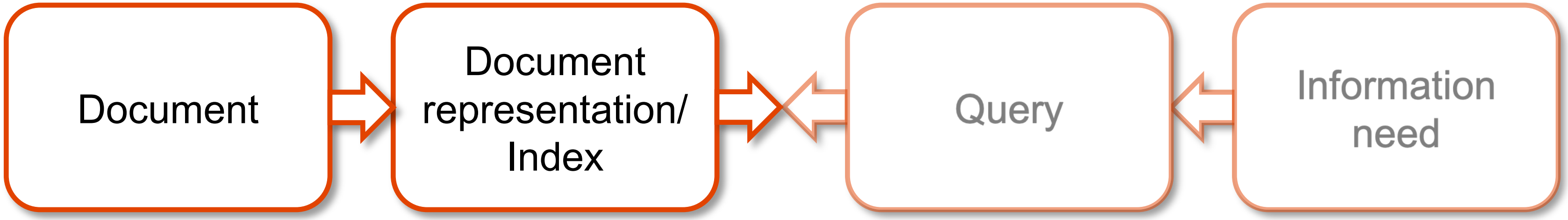
Zhai & Massung (2016), p. 11 – Logos not included in original representation

Conceptional Framework TIP



Zhai & Massung (2016), p. 11 – Logos not included in original representation

From theory to practice: The Index

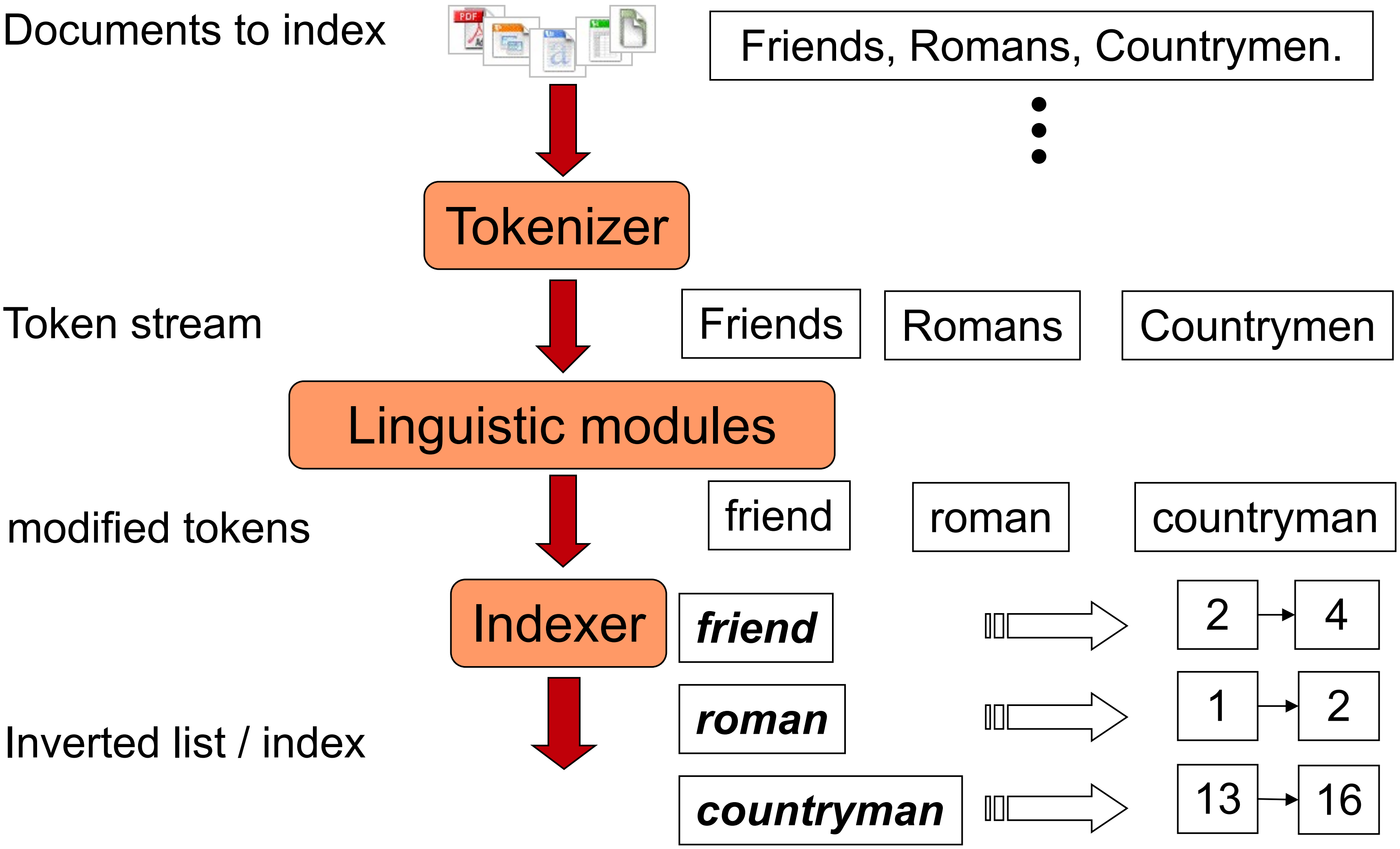


Basic task to solve before you begin to index:

- What preprocessing steps are necessary to build up the index and a good **vocabulary**?
- **Which terms** to include in the index?
- **How** and **in which form** to include the terms in the index?

- In Solr we define these in the so-called **schema** …

Basic index pipeline



Documents to index

Friends, Romans, Countrymen.

Tokenizer

Token stream

Friends

Romans

Countrymen

Linguistic modules

modified tokens

friend

roman

countryman

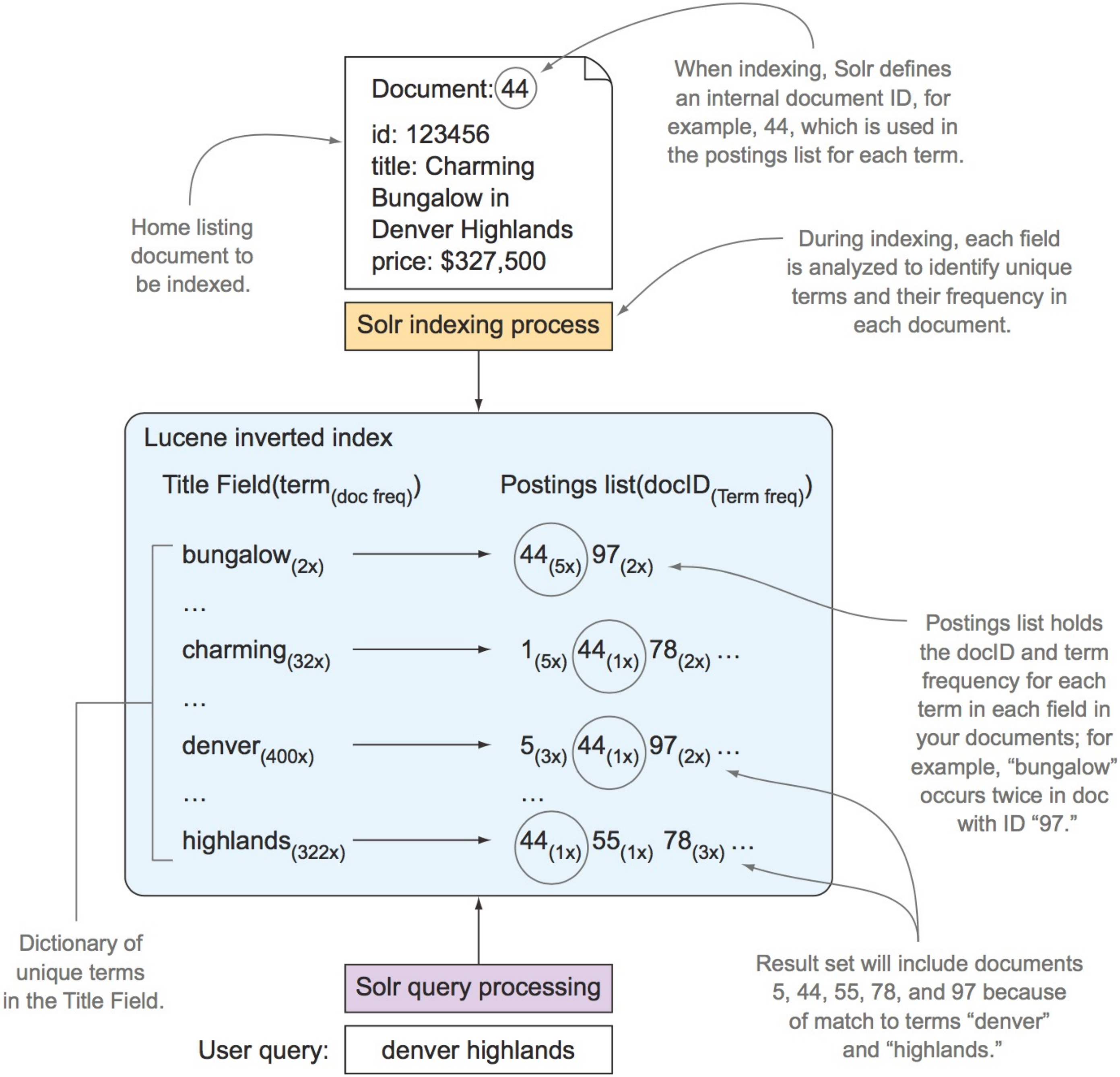
Indexer

Inverted list / index

friend 2 → 4

roman 1 → 2

countryman 13 → 16



Document: 44

id: 123456
title: Charming Bungalow in Denver Highlands
price: $327,500

When indexing, Solr defines an internal document ID, for example, 44, which is used in the postings list for each term.

Home listing document to be indexed.

Solr indexing process

During indexing, each field is analyzed to identify unique terms and their frequency in each document.

Lucene inverted index

Title Field(term$_{\text{(doc freq)}}$)	Postings list(docID$_{\text{(Term freq)}}$)
bungalow$_{(2x)}$	44$_{(5x)}$ 97$_{(2x)}$
…	
charming$_{(32x)}$	1$_{(5x)}$ 44$_{(1x)}$ 78$_{(2x)}$ …
…	
denver$_{(400x)}$	5$_{(3x)}$ 44$_{(1x)}$ 97$_{(2x)}$ …
…	…
highlands$_{(322x)}$	44$_{(1x)}$ 55$_{(1x)}$ 78$_{(3x)}$ …

Postings list holds the docID and term frequency for each term in each field in your documents; for example, "bungalow" occurs twice in doc with ID "97."

Dictionary of unique terms in the Title Field.

Solr query processing

User query: denver highlands

Result set will include documents 5, 44, 55, 78, and 97 because of match to terms "denver" and "highlands."

Solr in Action, p. 12 - https://www.manning.com/books/solr-in-action

Well, Solr and Elastic do all. We're done, right?

Remember the **diversity of search**?

- web search, e-commerce, expert search, location search, etc.

Solr or Elasticsearch **can do a lot**, but

- don't work well for your problem out of the box
  - if it did, there is nothing unique about your product!
- Solr and ElasticSearch are **"search programming frameworks"**
  - it lets you program **your understanding** of what's relevant
  - you can focus on the art and science of delivering relevant results
  - ….and of course meet the business goals!! ;-)

Live-Demo Solr