Basic Text Search Functionalities

- Text search
  find matching documents
- Document ranking
  calculate importance of a document according to the query
- Did you mean
  calculate more important query terms than the specified one
- Highlighting
  Highlight the relevant query terms in the document/summary
- Automatic summarizations
  Create a summary of a document for getting an overview
- Feature extraction:
  extract characteristic keywords from a document
- Fuzzy Search
  search similar terms or phrases
- Natural Language Search
  answer natural language queries

AdvancedText Search Functionalities

- "See Also" search:
  get more documents like this
- Feature extraction:
  find characteristic keywords
- Entity extraction
  extract entities like proper names/company names
- Document classification:
  assign a document to predefined categories
- Term search:
  find better search terms; discover interesting relationships
- Document clustering:
  discover sets of related documents
- Sentiment Analysis
  discover the sentiments of a document about a topic

Document Analysis I

- Crawling / document input

  get documents from web or any other source
- Document filtering

  convert a document from any format to plain text/html/xml
- Tokenization

  determine word, clause and sentence boundaries
- Normalization

  upper/lowercase; spelling variants; umlauts
- Tagging

  determine word category (noun, verb, adjective, adverb, etc.)
- Stemming

  singular/plural; case inflections

Document Analysis II

- Entity extraction

  extract entities like company names, proper names and facts based on dictionaries and rules

- Term Identification

  based on word category and stopword list

- Phrase Generation (noun phrases)

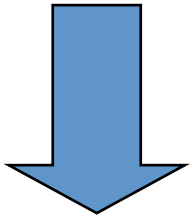
  combine adjacent words of particular categories (NN, AN, NPN, ...)

- Term Selection after processing a document collection

  for text mining : delete low - and high-frequency terms; delete redundant phrases

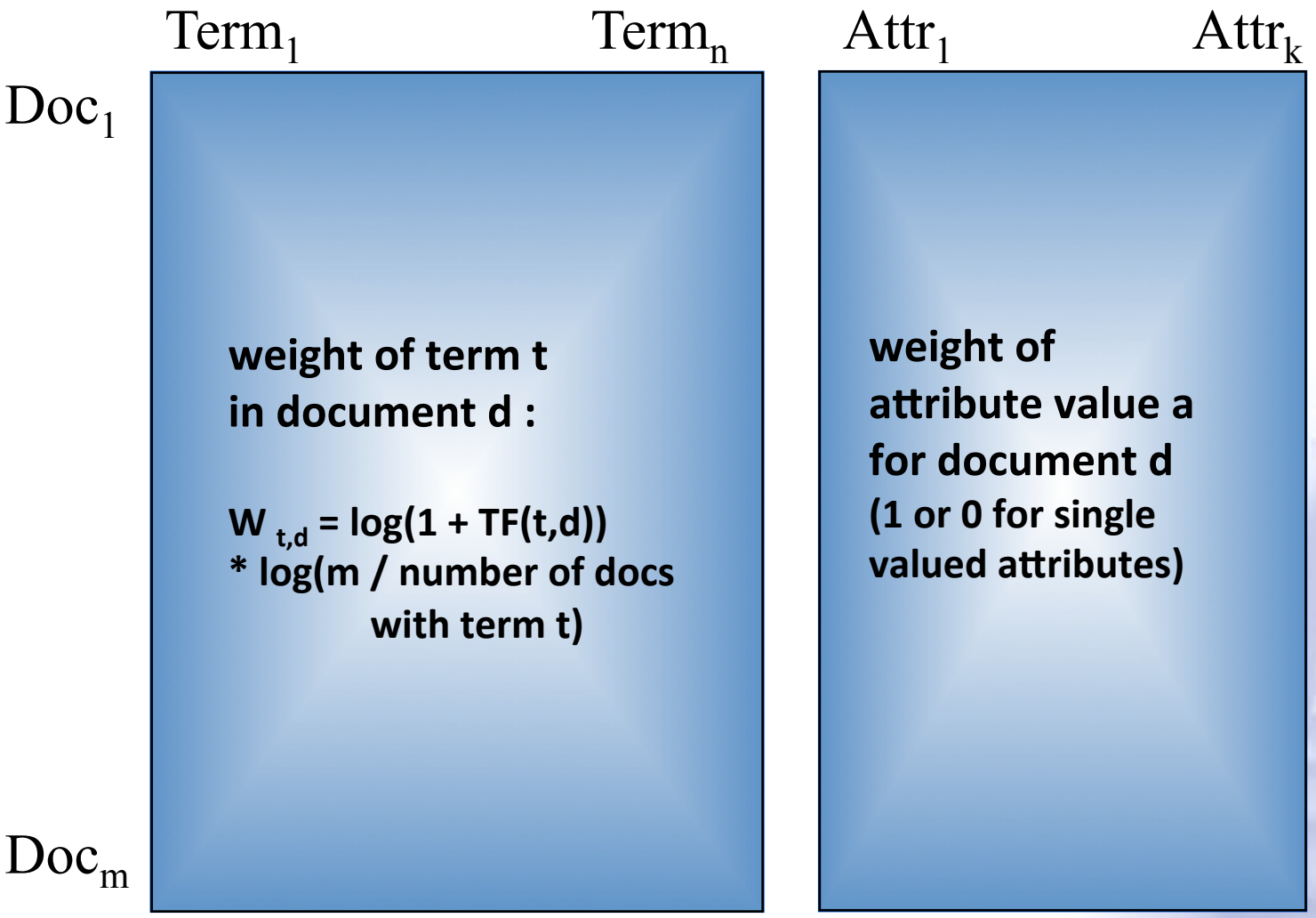
Term Generation Example

"But Lieberman's criticism of Clinton's behavior may have been more of a personal move than a political one.“



Lieberman; Lieberman's criticism; Lieberman's criticism of Clinton;
Lieberman's criticism of Clinton's behavior
criticism; criticism of Clinton; criticism of Clinton's behavior
Clinton; Clinton‘s behavior
behavior
personal move
move

Vector space retrieval model



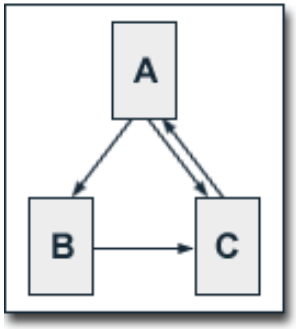
Ranking : Page Rank

- Developed by Lawrence Page and Sergey Brin
- Based on relations between websites not on the content itself
- General concept: Random surfer model
- Weight of a page : PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C (Tn))
  - PR(A) is the PageRank of page A
  - PR(Ti) is the PageRank of Ti, which links to page A
  - C(Ti) is the number of links of page Ti
  - d is a damping factor with 0 <= d <= 1

Example :

PR(A) = 0.5 + 0.5 PR(C)
PR(B) = 0.5 + 0.5 (PR(A) / 2)
PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))

PR(A) = 14/13 = 1.07692308
PR(B) = 10/13 = 0.76923077
PR(C) = 15/13 = 1.15384615



Ranking : PAGE Rank

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

Ranking : BM25 / OKAPI distance

- Concept
  - Document weight is based on f * IDF
- Formulas

$$bm25(q,d) = \sum_{t \in q} IDF \times \frac{(k_1+1) f(d,t)}{K + f(d,t)}$$

$$IDF = \log\left(\frac{N - f(t) + 0.5}{f(t) + 0.5}\right)$$

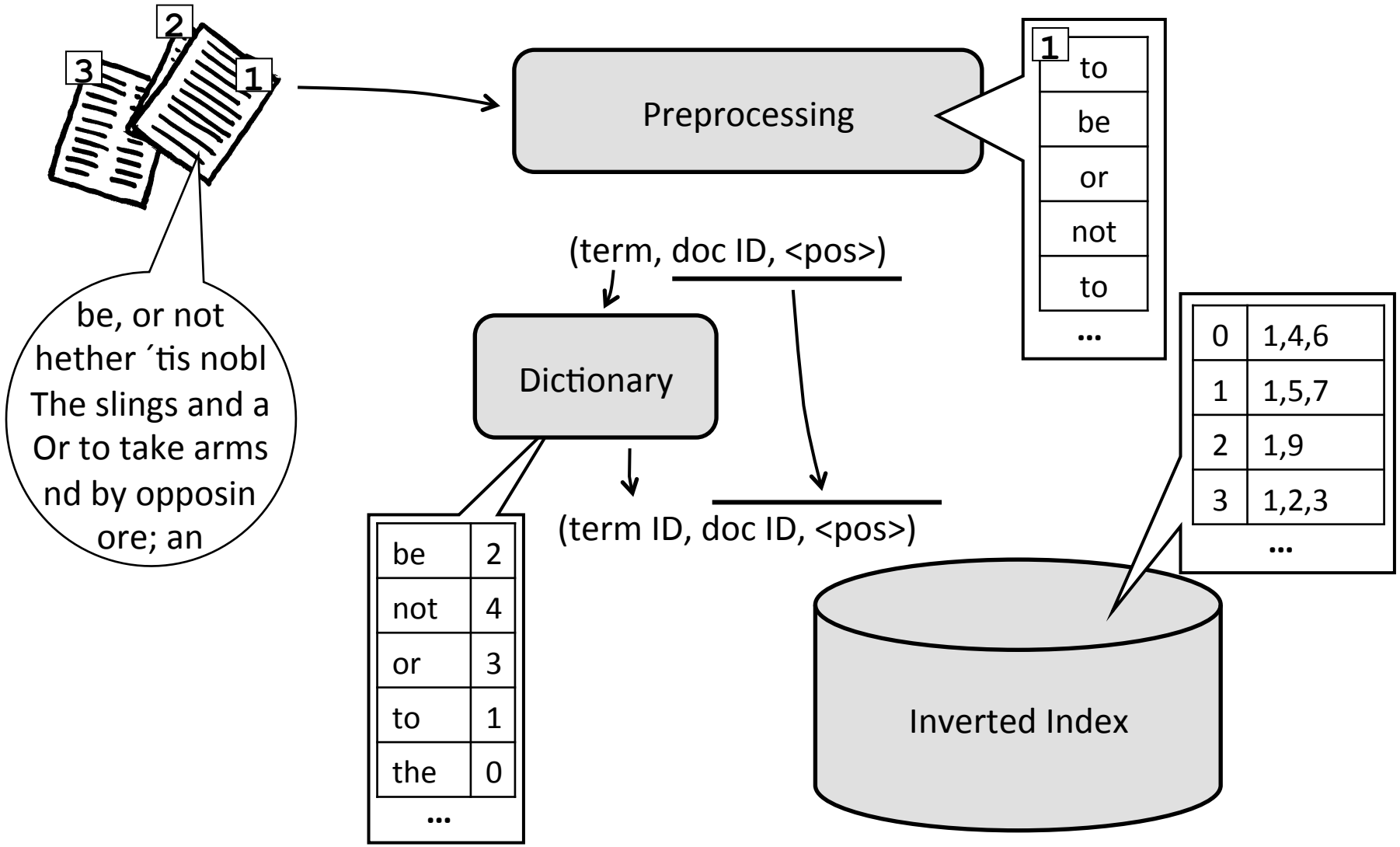
f(d,t) is the term frequency of term t in document d,

N is the number of documents

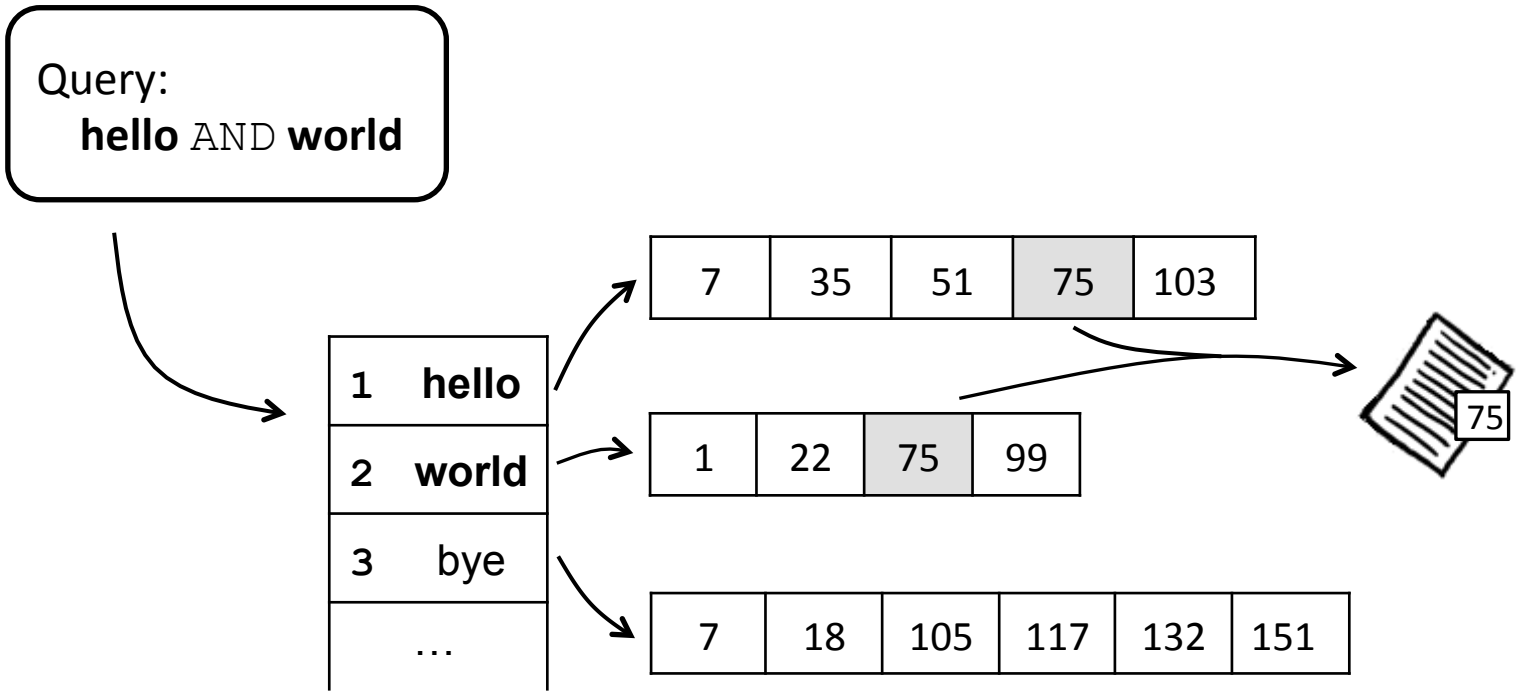
f(t) is the document frequency of term t

K, k1 are constants

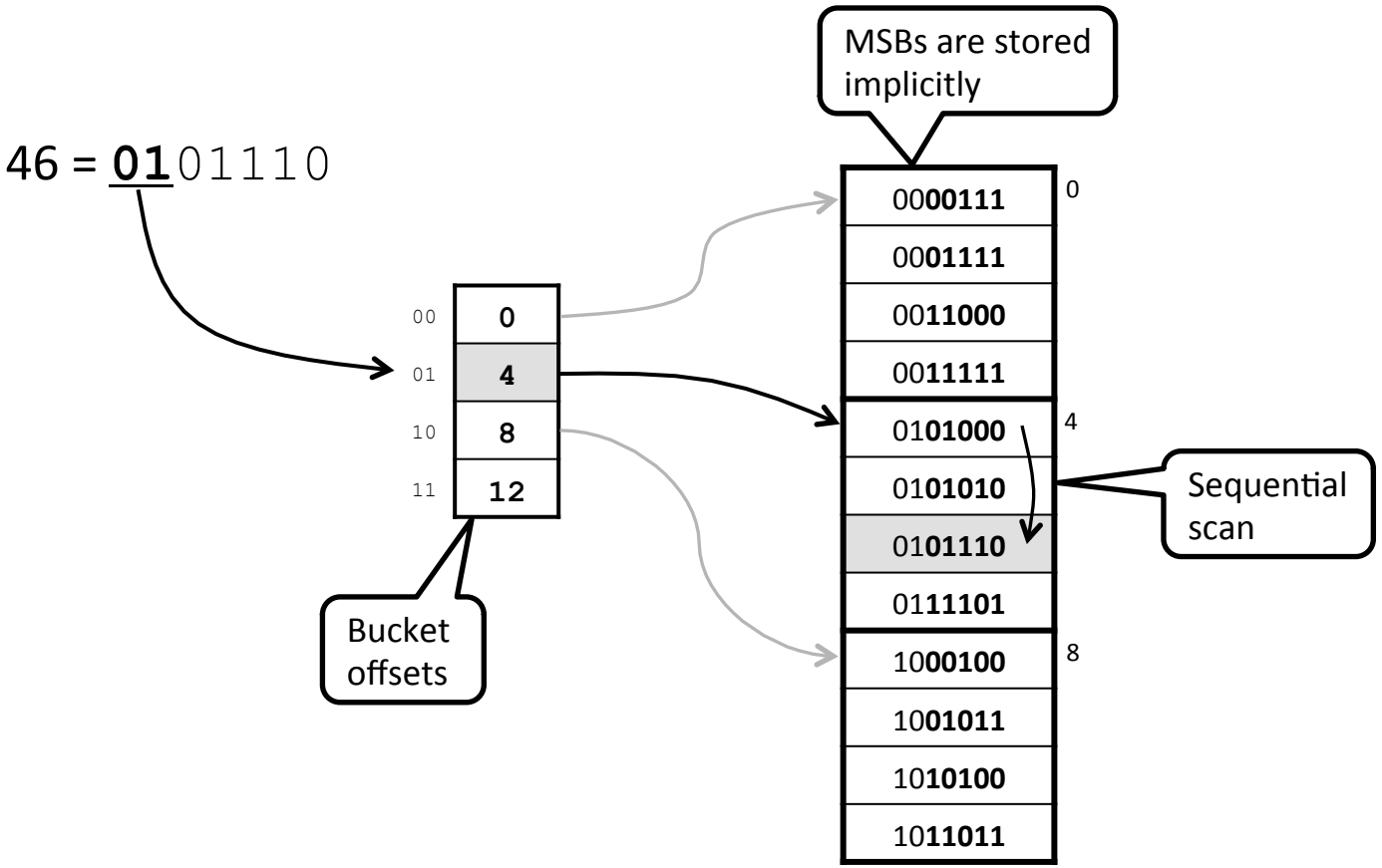
Indexing Process



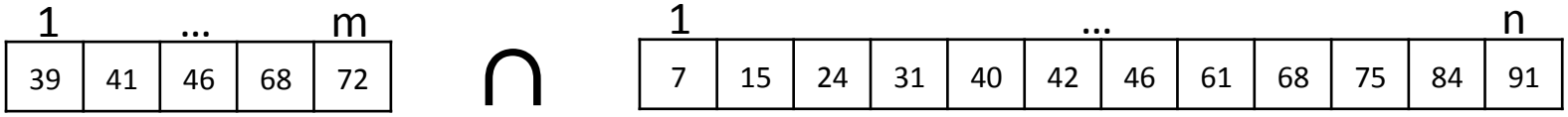
Inverted Index

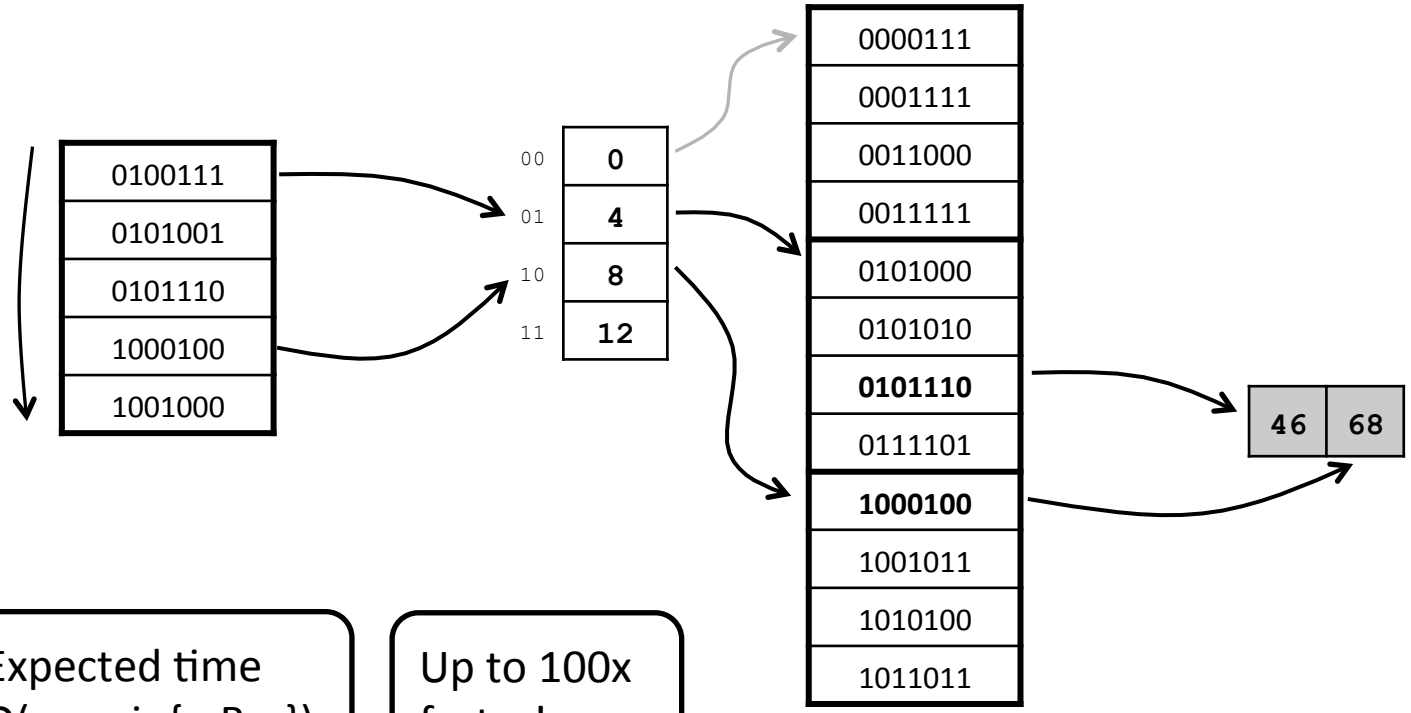


Two-Level Inverted List



Inverted List Intersection

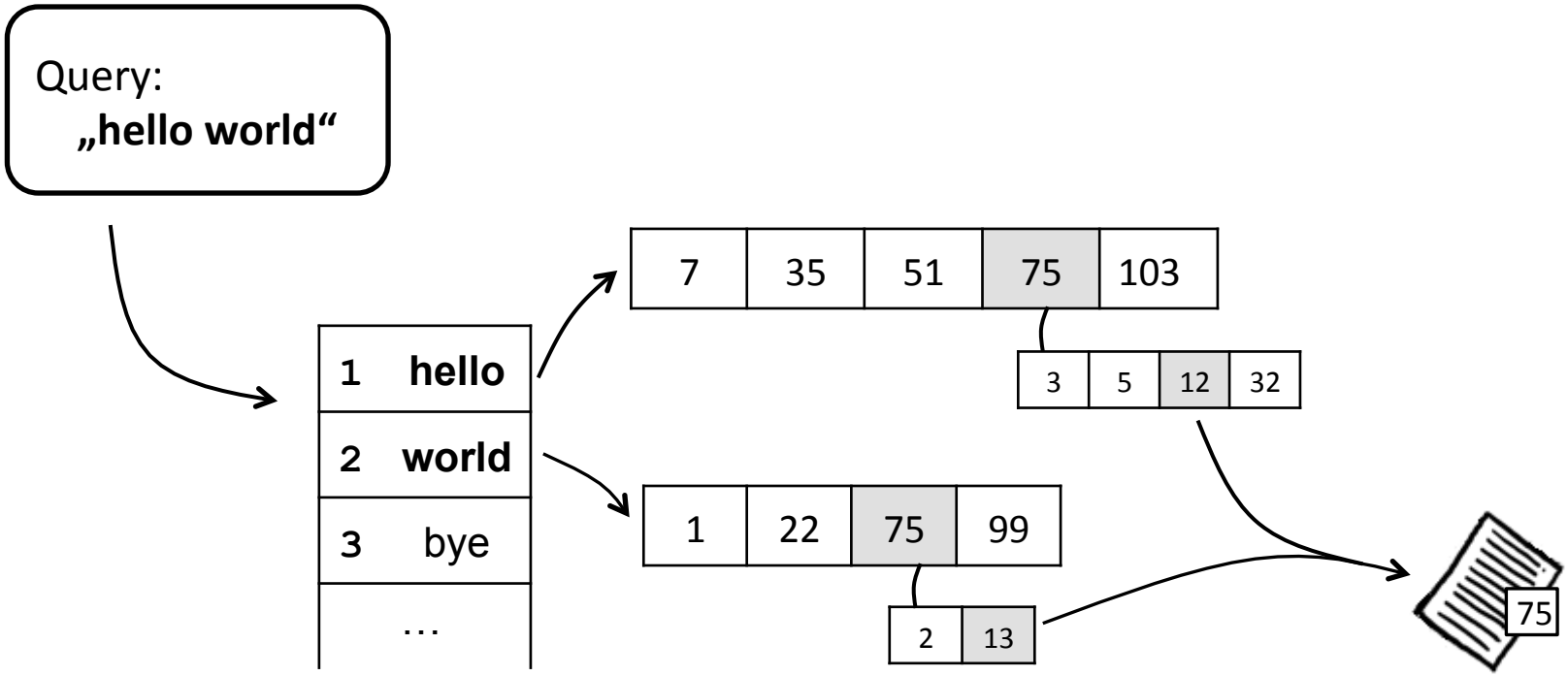




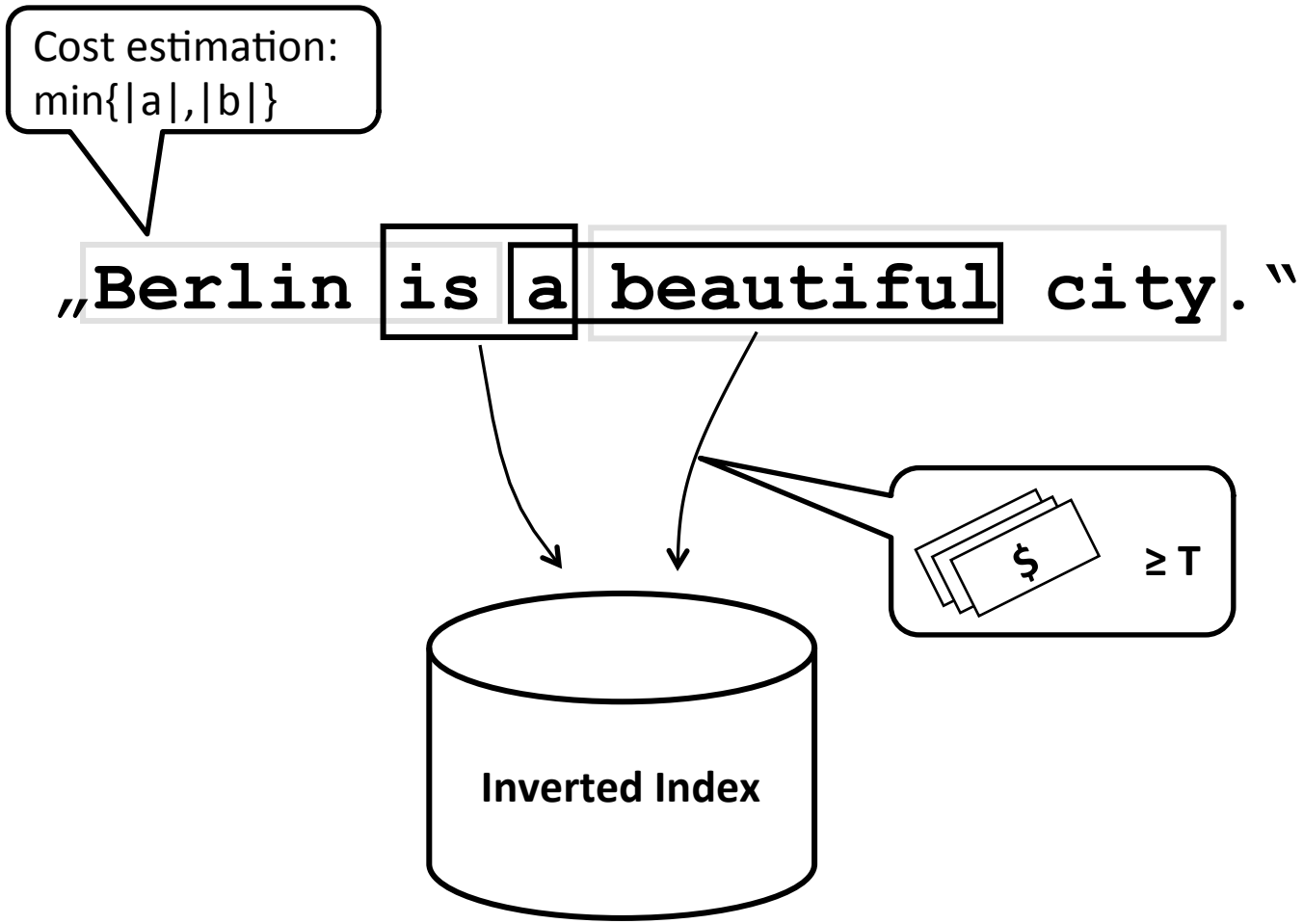
Expected time
O(**m**+min{n,B**m**})

Up to 100x faster!

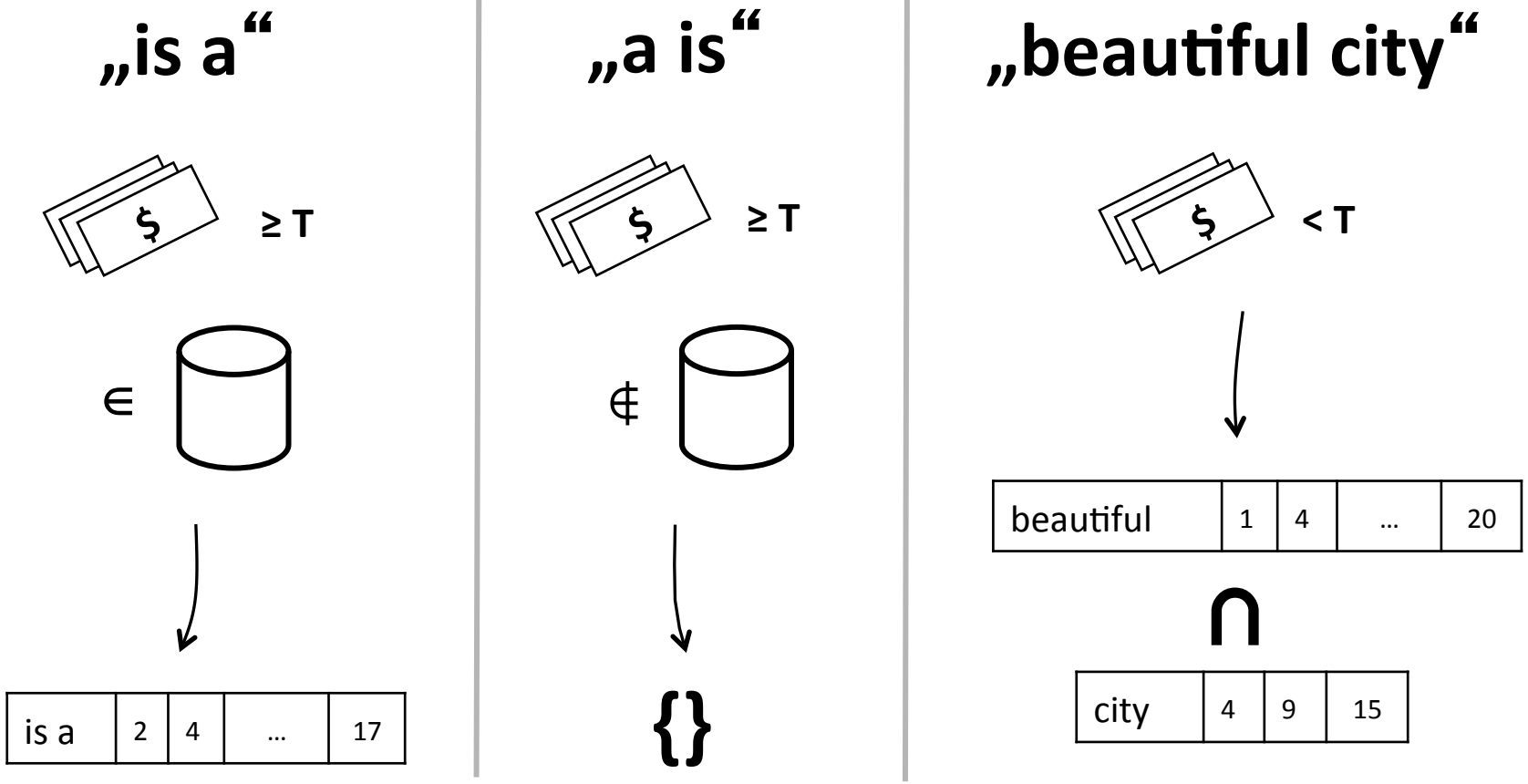
Phrase Search



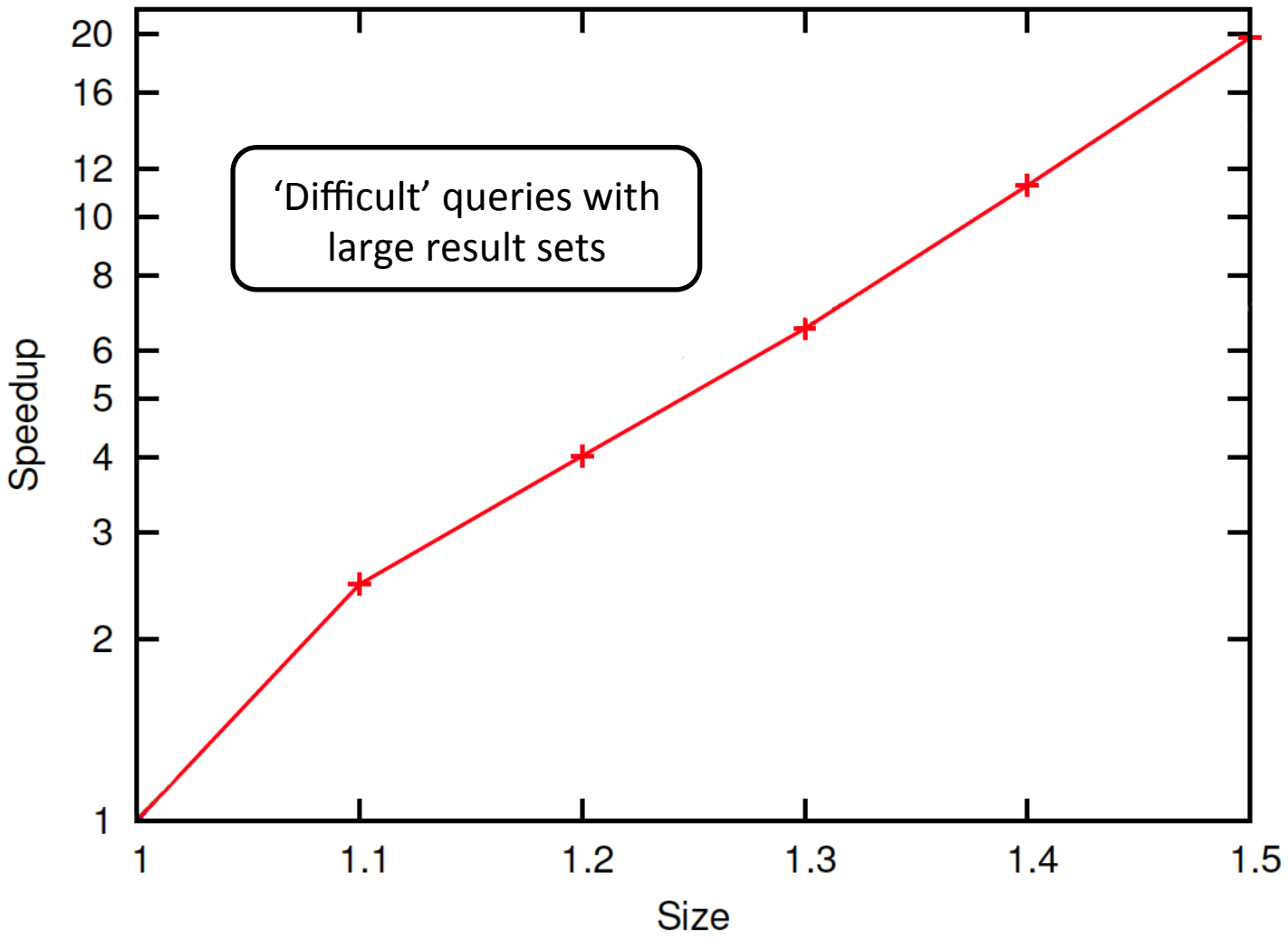
Two-Word Phrase Index



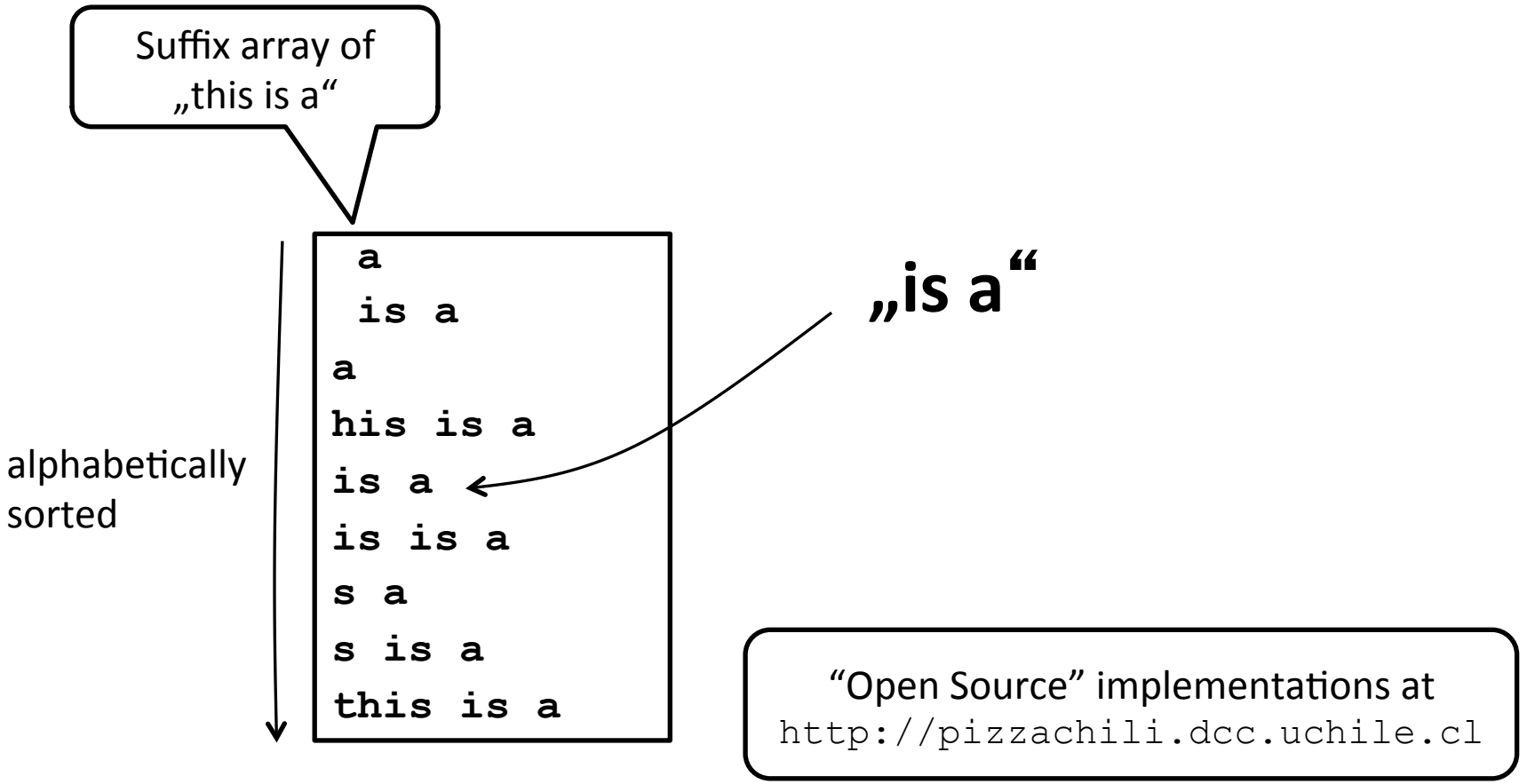
Querying the Phrase Index



Phrase Index: Speed-Up

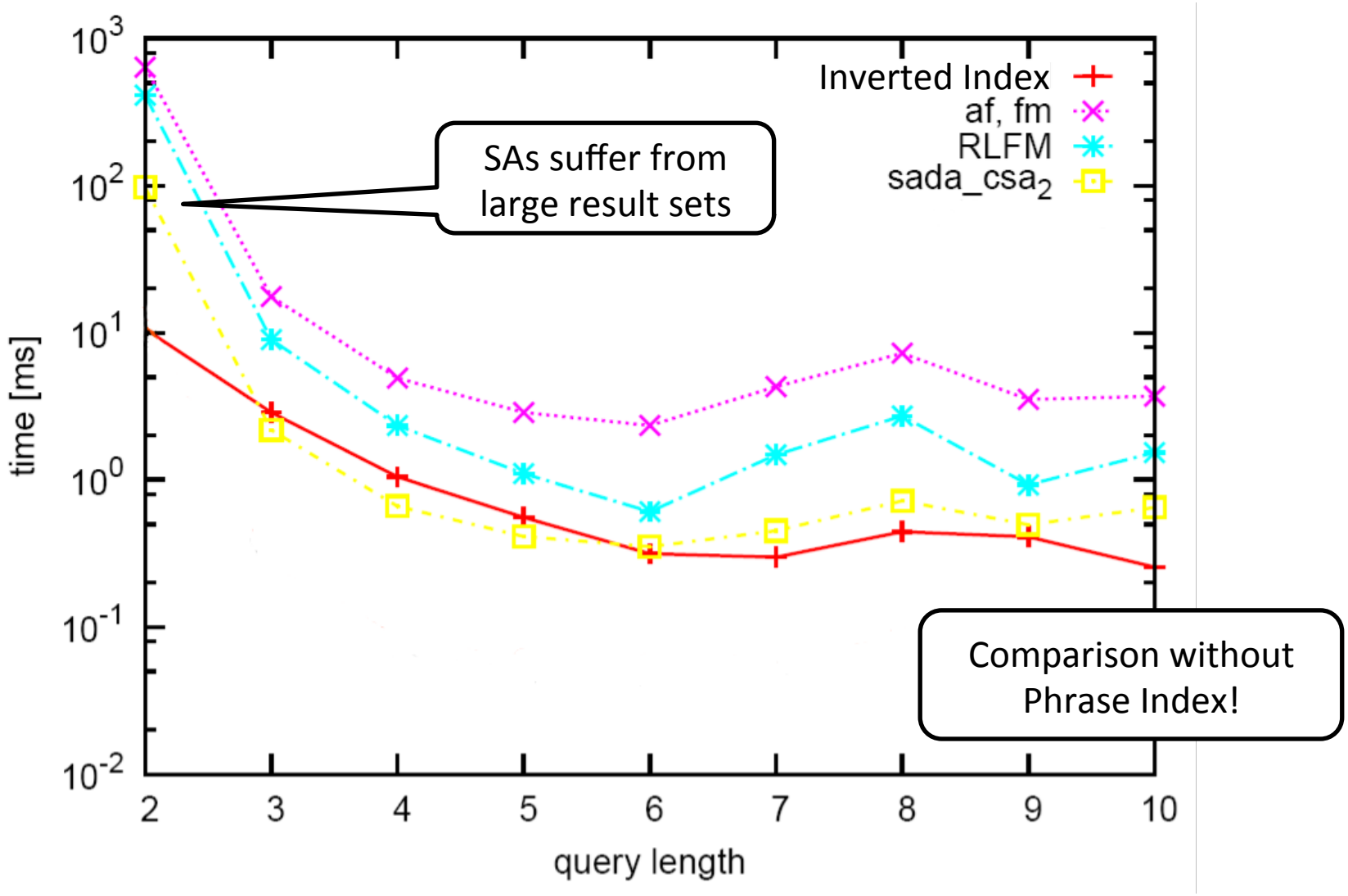


Suffix Arrays



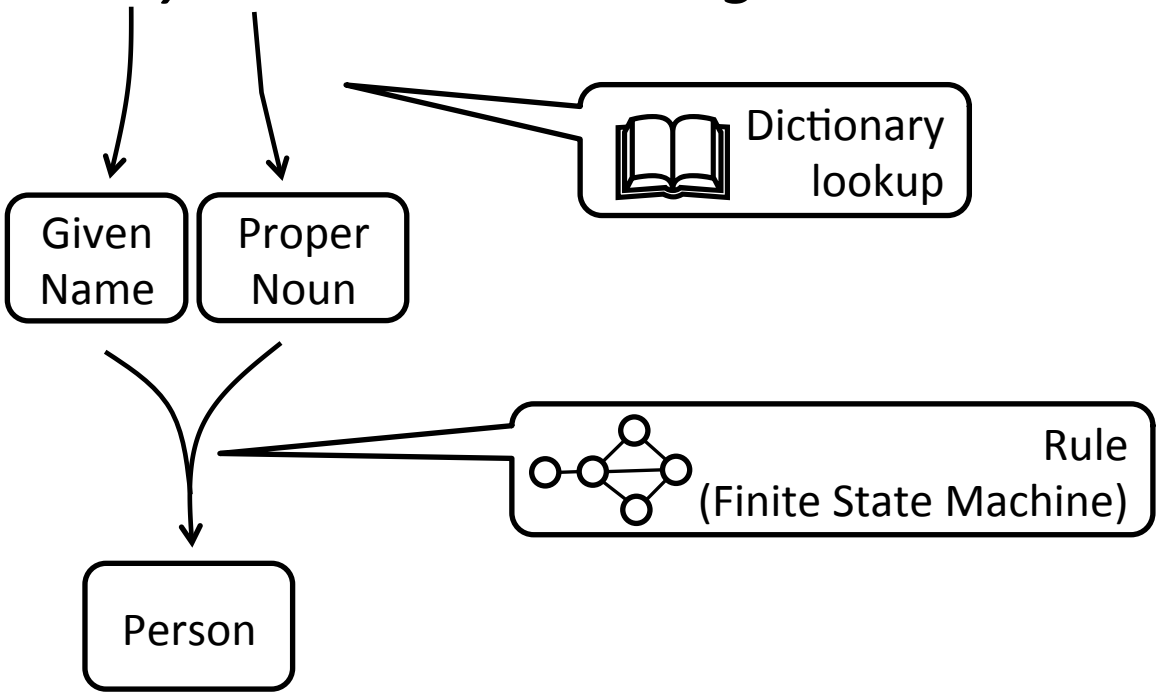
Phrase Search

Inverted Index vs. Suffix Arrays

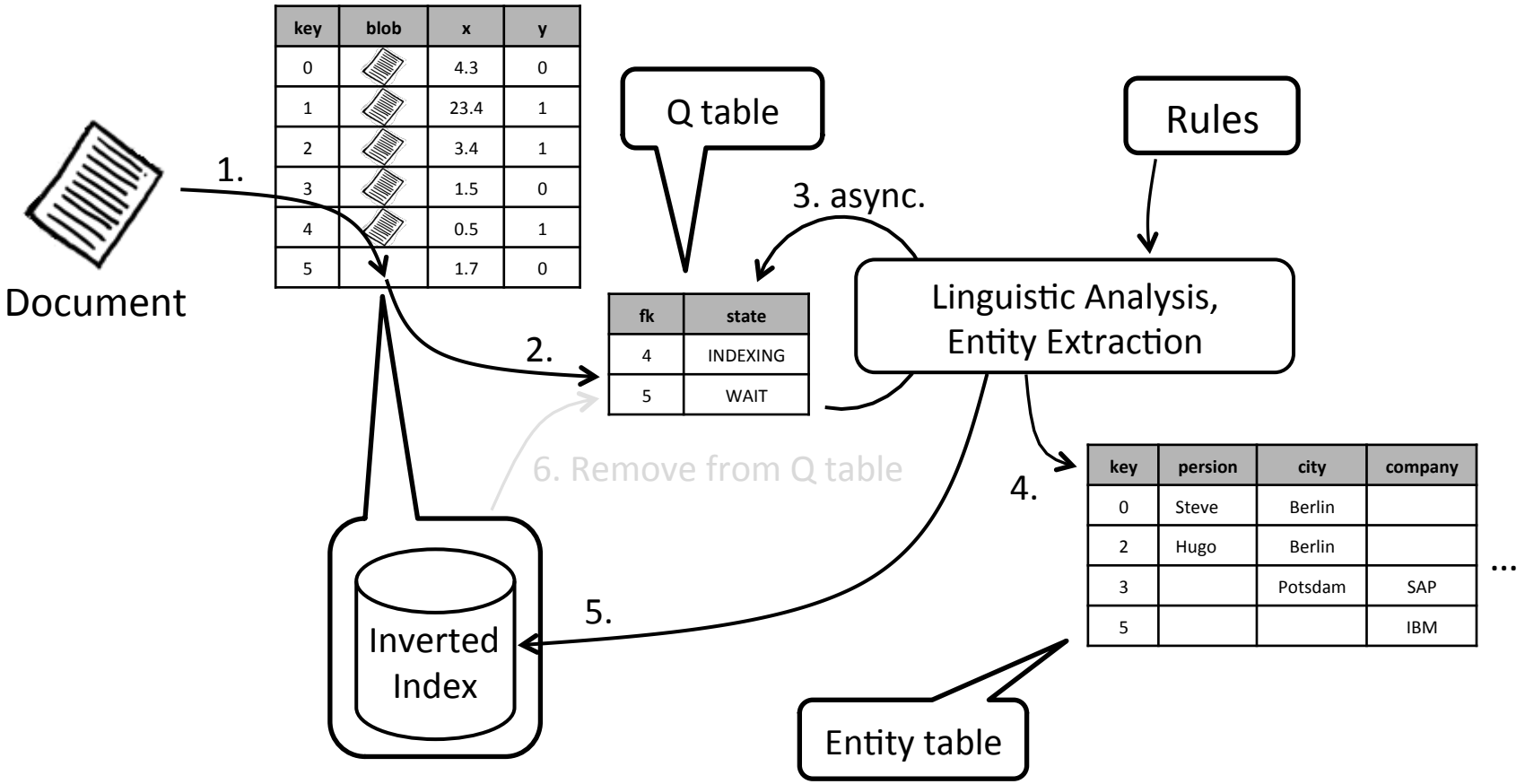


Text Analysis: Entity Extraction

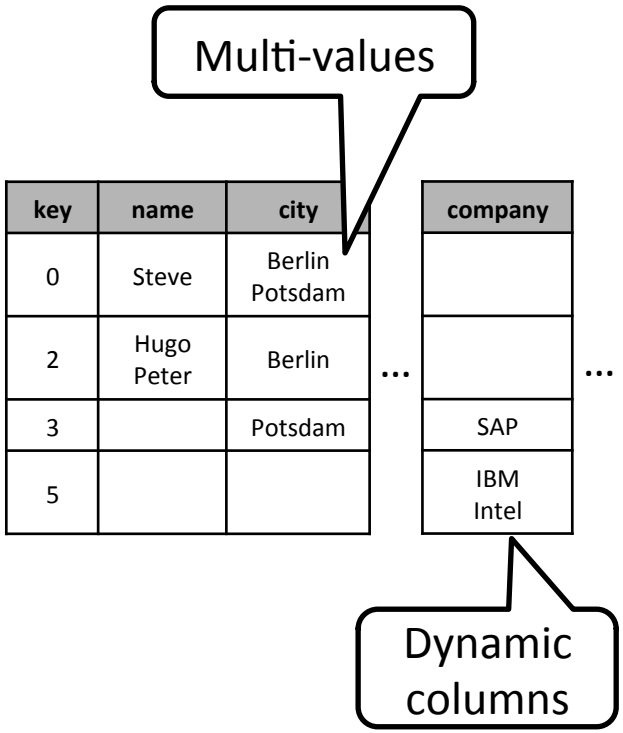
"I saw Ricky Lake while visiting New York."



Indexing Unstructured Data



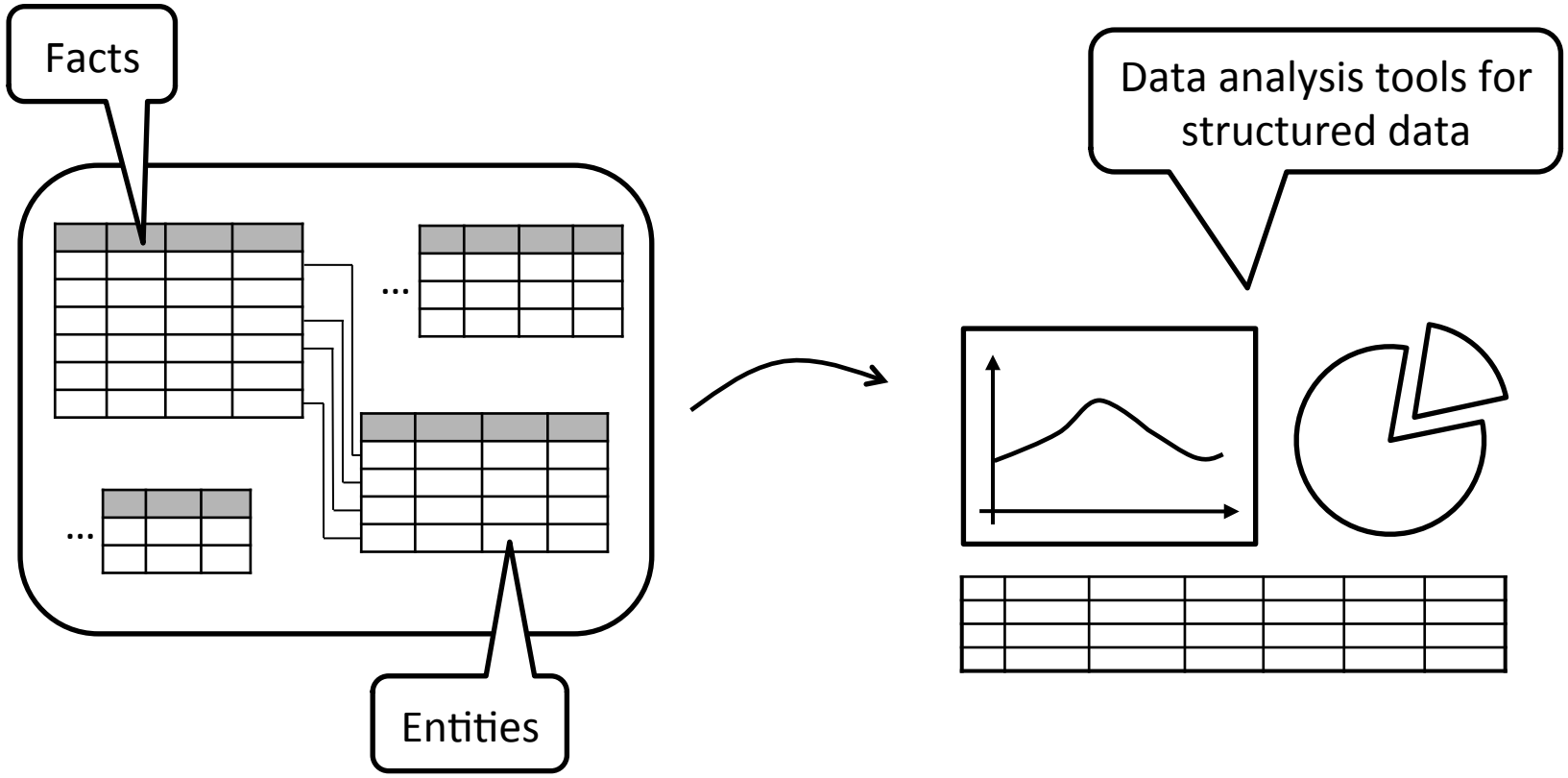
Entity tables



Entity/value pairs

key	entity	value
0	name	Steve
0	city	Berlin
0	city	Potsdam
2	name	Hugo
2	name	Peter
2	city	Berlin
3	city	Potsdam
3	company	SAP
5	company	IBM
5	company	Intel

Analytical View combining Structured and Unstructured Data



Property Table

- Infinite number of fields
- Multi value support
- Included into relational model
- Field creation is a DML operation
- Internal optimization for efficient processing
- Support of range fields
- Validity support

Document Classification

- KNN Classifier
  - compute the k most similar documents to query document
  - compute class weights according to classes assigned to those k nearest neighbours
- Simple Centroid Classifier
  - compute centroid (average) vector for each class
  - compute similarity between query vector and centroids
- Weighted Centroid Classifier
  - Compute a weight for each document based on how well it distinguishes between classes
  - compute the weighted centroids
  - compute similarity between query vector and centroids

- Kernel method
  - support vector machine