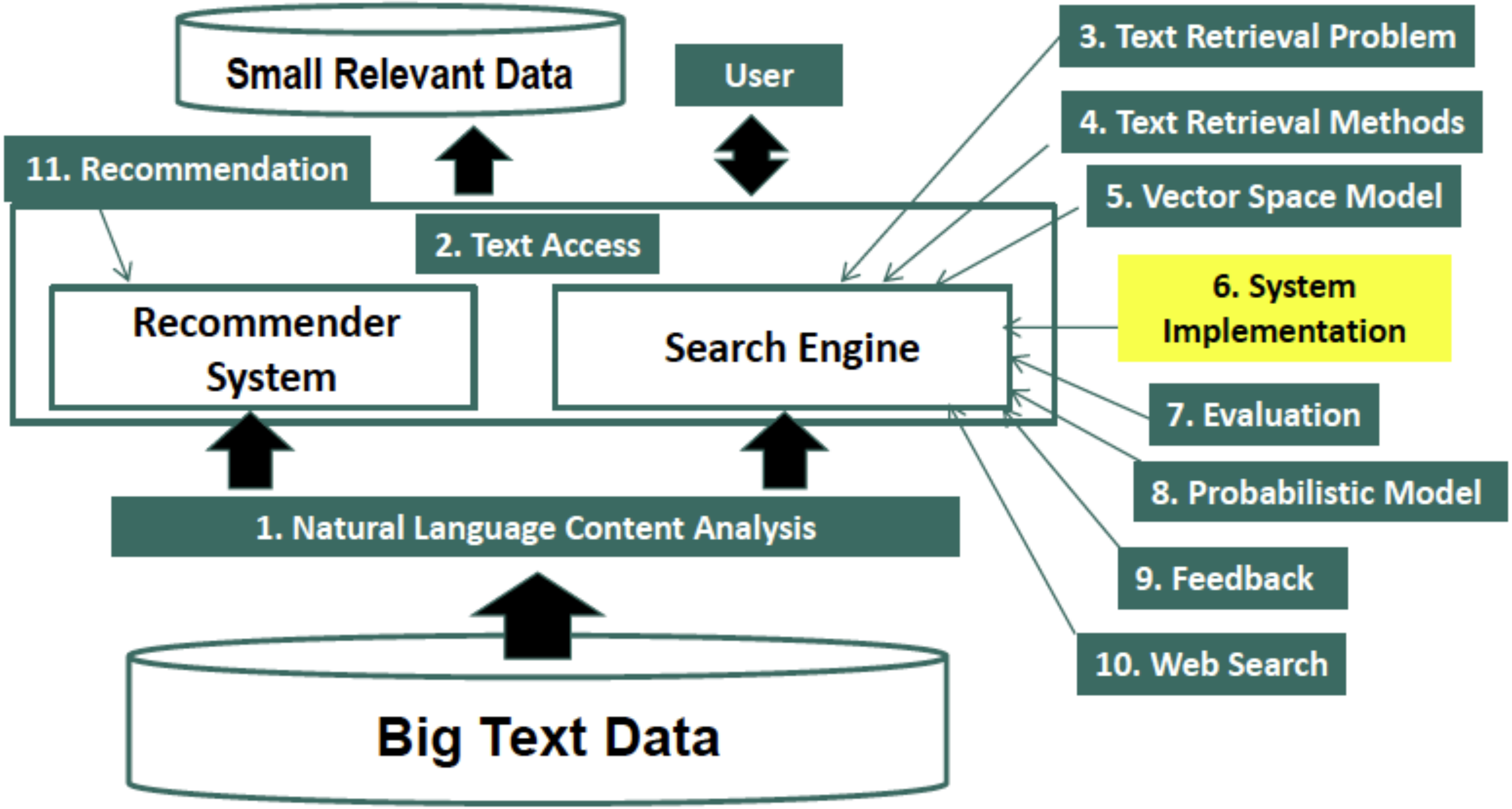
Information Retrieval & Text Mining

Implementation

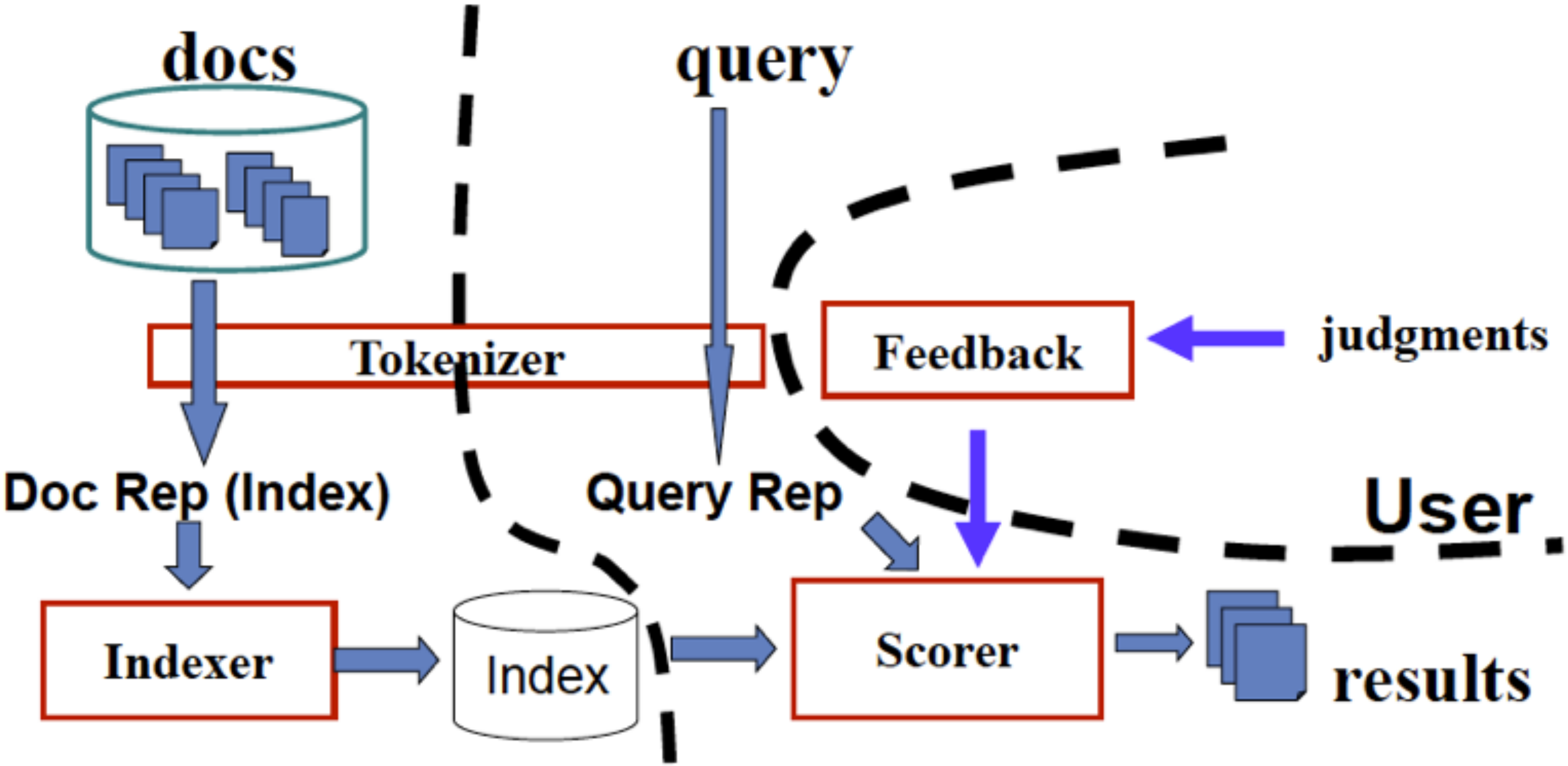
Text Retrieval System

Dr. Saeed Ul Hassan
Information Technology University

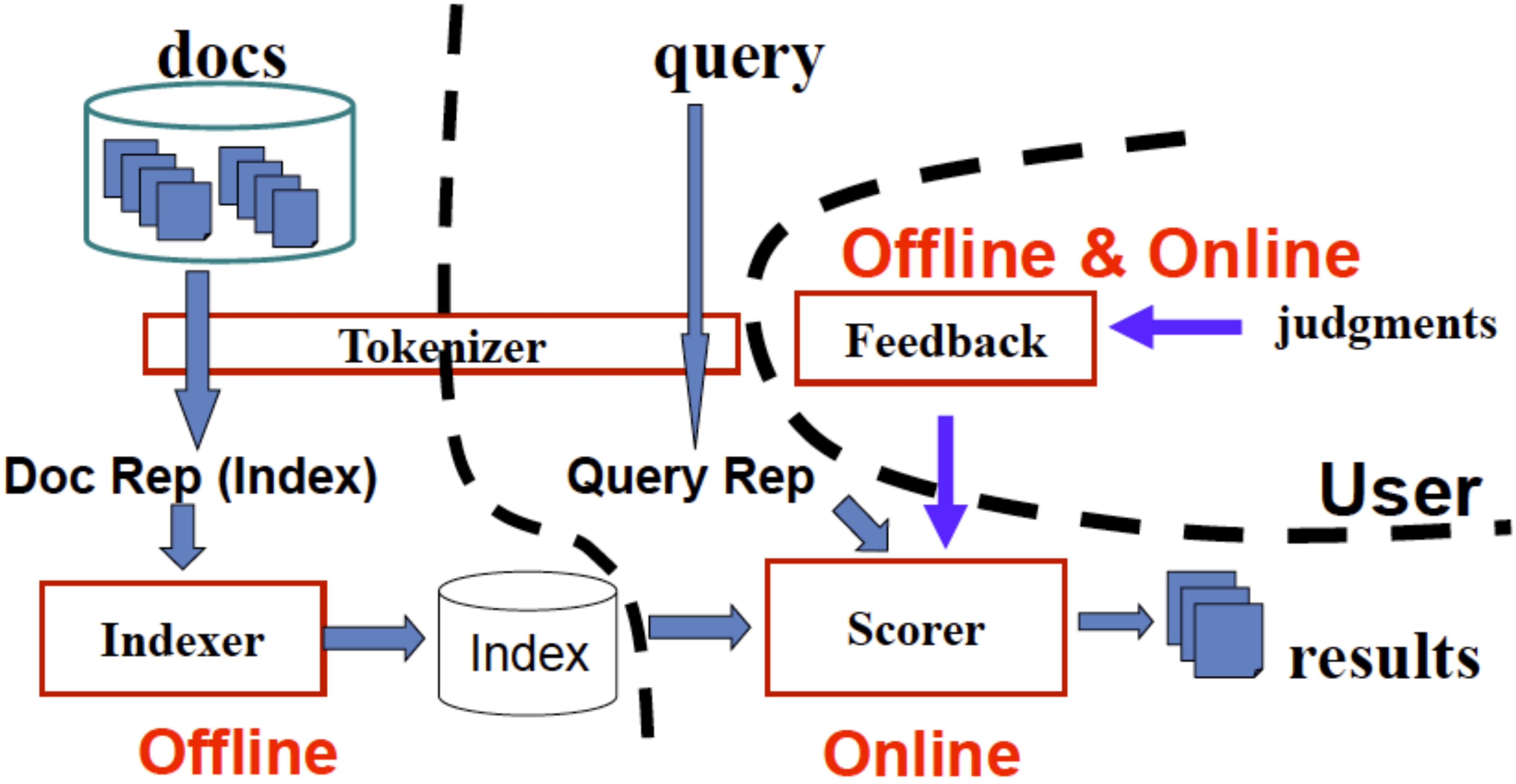
Implementation of Text Retrieval Systems



Typical TR System Architecture



Typical TR System Architecture



Tokenization

- Normalize lexical units: Words with similar meanings should be mapped to the same indexing term
- Stemming: Mapping all inflectional forms of words to the same root form, e.g.
  - computer -> compute
  - computation -> compute
  - computing -> compute
- Some languages (e.g., Chinese) pose challenges in word segmentation

How would you do it for Urdu?
How about Roman Urdu?

Indexing

- Indexing = Convert documents to data structures that enable fast search (precomputing as much as we can)
- Inverted index is the dominating indexing method for supporting basic search algorithms
- Other indices (e.g., document index) may be needed for feedback

How would you respond to single word query?

Think about it!!!

Inverted Index Example

doc 1

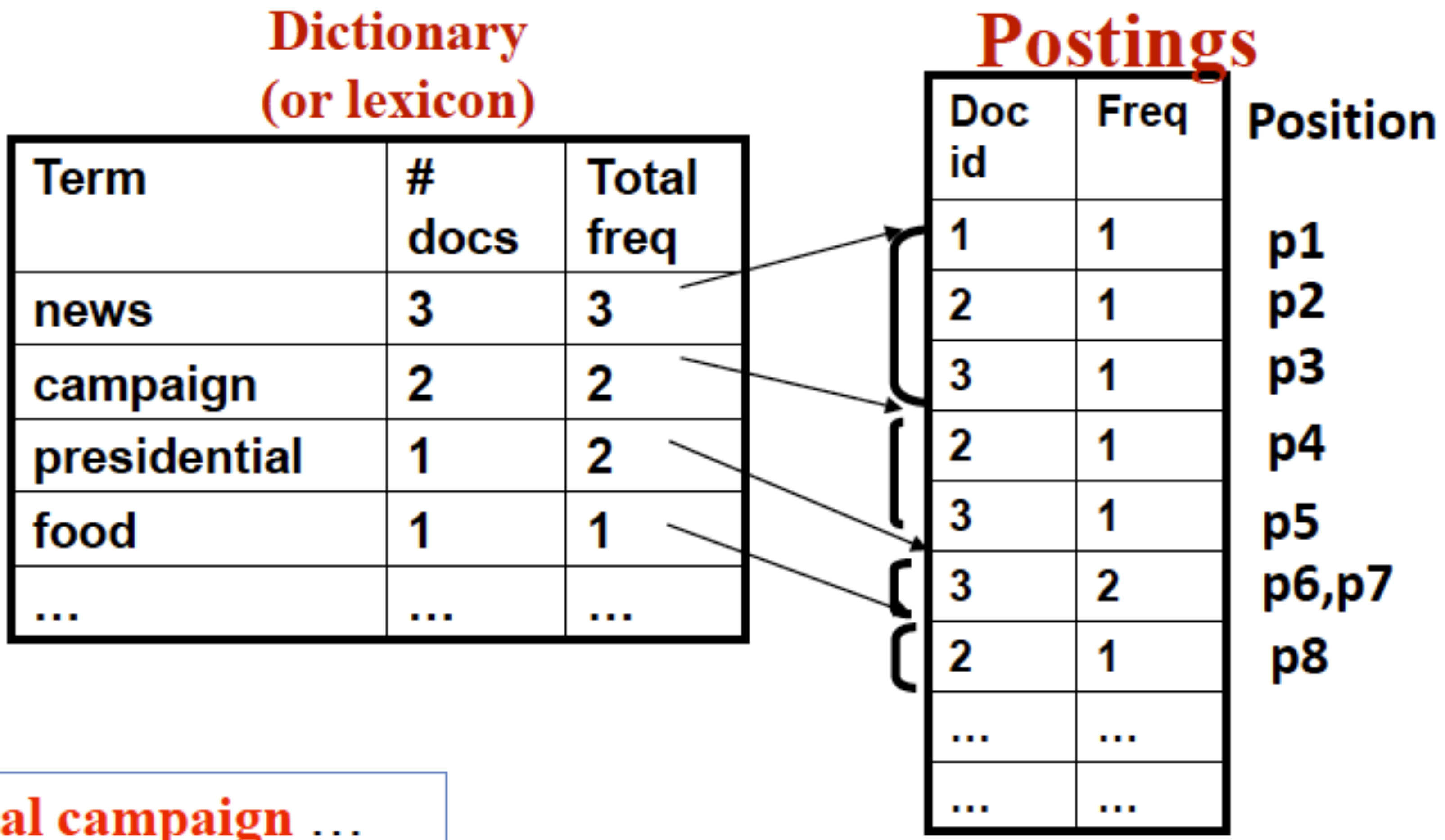
… **news about**

doc 2

… **news about**
organic food
campaign…

doc 3

… **news** of **presidential campaign** …
… **presidential** candidate …



Dictionary (or lexicon)

Term	# docs	Total freq
news	3	3
campaign	2	2
presidential	1	2
food	1	1
…	…	…

Postings

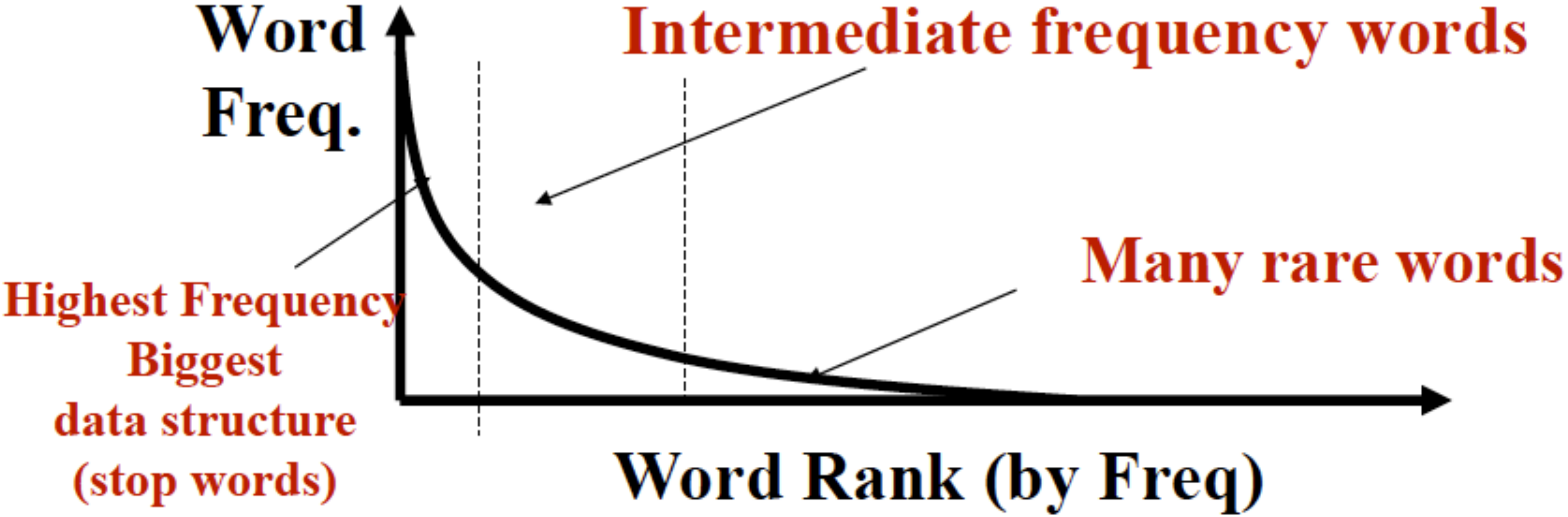
Doc id	Freq	Position
1	1	p1
2	1	p2
3	1	p3
2	1	p4
3	1	p5
3	2	p6,p7
2	1	p8
…	…	
…	…	

Empirical Distribution of Words

- There are stable language-independent patterns in how people use natural languages
- A few words occur very frequently; most occur rarely. E.g., in news articles,
  - Top 4 words: 10~15% word occurrences
  - Top 50 words: 35~40% word occurrences
- The most frequent word in one corpus may be rare in another

Zipf’s Law

- rank * frequency ≈ constant $F(w) = \frac{C}{r(w)^{\alpha}} \quad \alpha \approx 1, C \approx 0.1$



Data Structures for Inverted Index

- Dictionary: modest size
  - Needs fast random access
  - Preferred to be in memory
  - Hash table, B-tree, …
- Postings: huge
  - Sequential access is expected
  - Can stay on disk
  - May contain docID, term freq., term pos, etc
  - Compression is desirable

Compression vs. Processing: who is the winner?

Think about it!!!