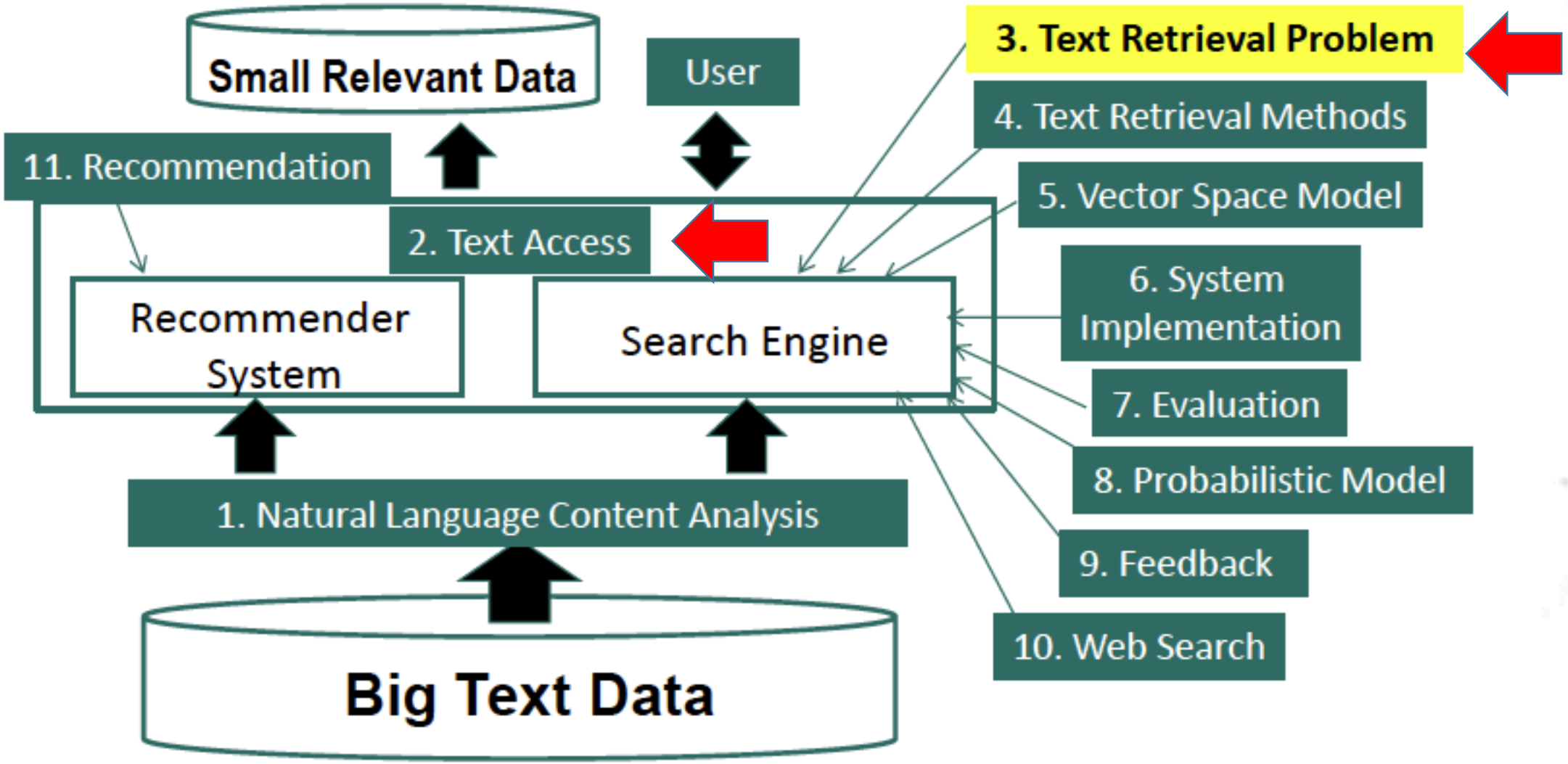
Information Retrieval & Text Mining

Overview of Text Retrieval Methods

Dr. Iqra Safder
FAST NUCES, Lahore

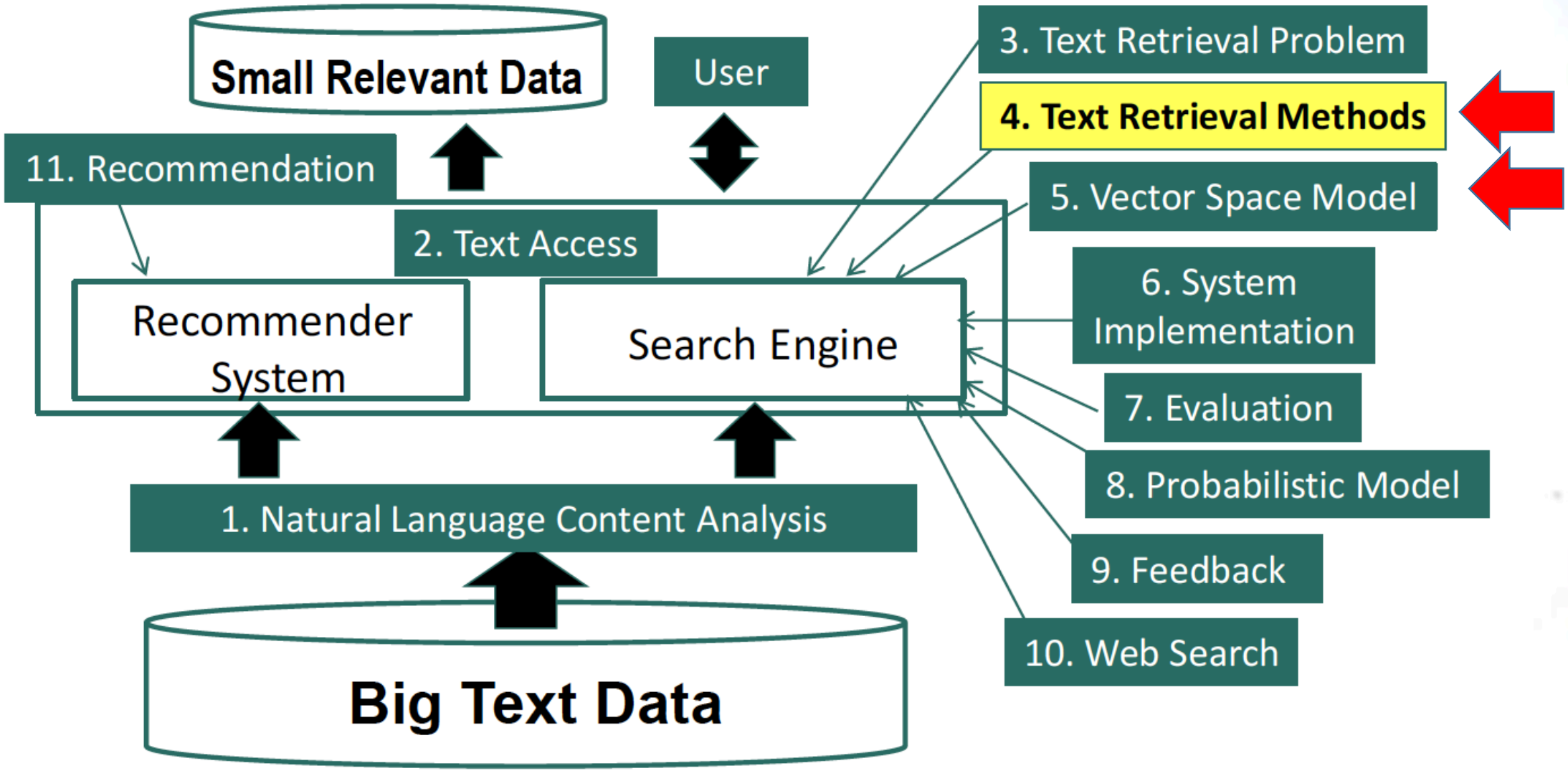
Recap


Small Relevant Data
User
3. Text Retrieval Problem
4. Text Retrieval Methods
11. Recommendation
5. Vector Space Model
2. Text Access
6. System Implementation
Recommender System
Search Engine
7. Evaluation
8. Probabilistic Model
1. Natural Language Content Analysis
9. Feedback
10. Web Search
Big Text Data

Recap

- Text retrieval is an empirically defined problem
  - Which algorithm is better must be judged by users
- Document ranking is generally preferred to
  - Help users prioritize examination of search results
  - Bypass the difficulty in determining absolute relevance (users help decide the cutoff on the ranked list)
- Main challenge: design an effective ranking function **f(q,d) =?**

Course Schedule



How to Design a Ranking Function

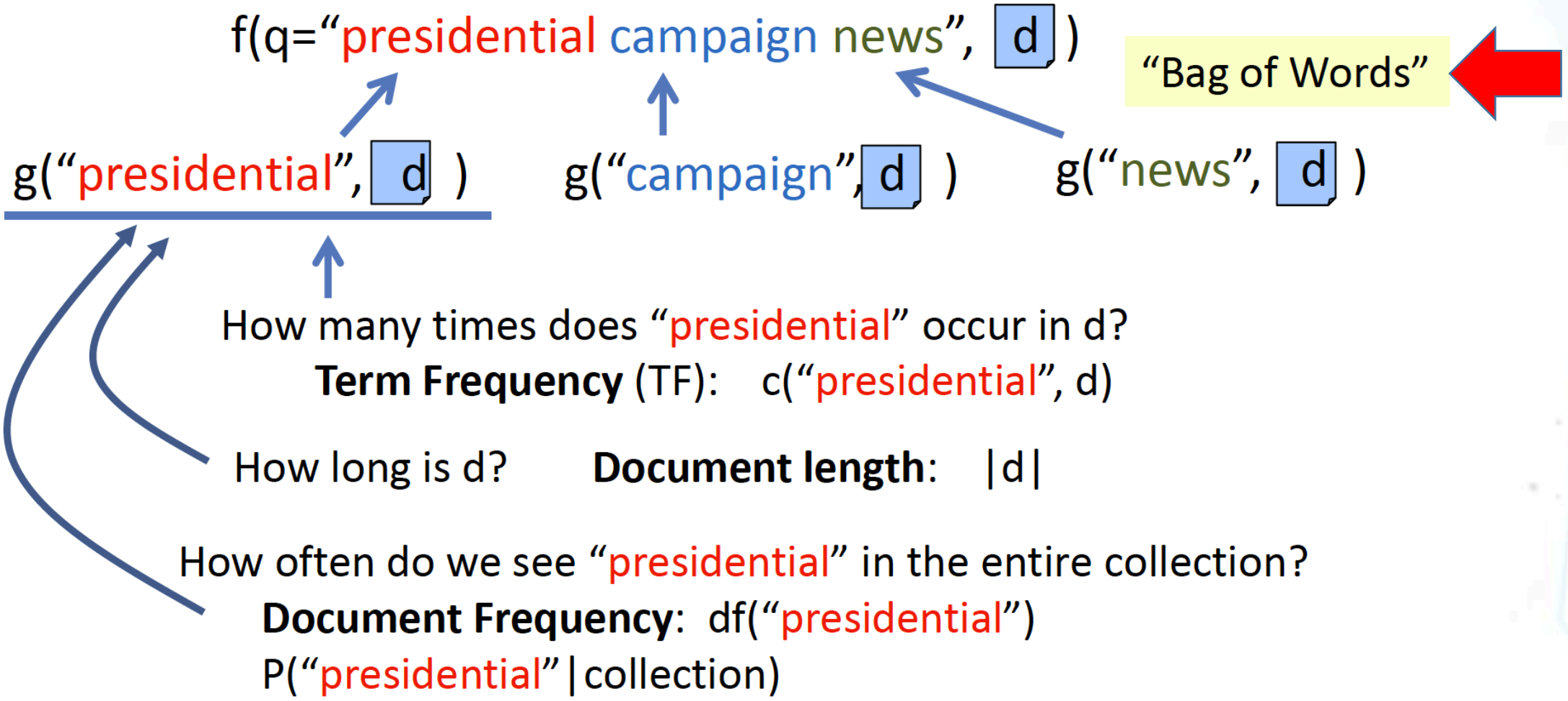
- **Query:** $q = q_1,\ldots,q_{m,}$ where $q_i \in V$
- **Document:** $d = d_1,\ldots,d_{n,}$ where $d_i \in V$
- **Ranking function:** $f(q, d) \in \Re$
- A good ranking function should rank relevant documents on top of non-relevant ones
- Key challenge: how to measure the likelihood that document d is relevant to query q
- **Retrieval model** = formalization of relevance (give a computational definition of relevance)

We must have a computational defination of the relevance.

Many Different Retrieval Models

- **Similarity-based models**: f(q,d) = similarity(q,d)
  - Vector space model
- **Probabilistic models**: f(d,q) = p(R=1|d,q), where R $\in$ {0,1}
  - Classic probabilistic model
  - Language model
  - Divergence-from-randomness model
- **Probabilistic inference model**: f(q,d) = p(d→q)
- **Axiomatic model**: f(q,d) must satisfy a set of constraints
- These different models tend to result in similar ranking functions involving similar variables

Common Ideas in State of the Art Retrieval Models



Matching a rare term in the collection is contributing more to the overall score than matching up common term.

Which Model Works the Best?

- When optimized, the following models tend to perform equally well [Fang et al. 11]:
  - **Pivoted length normalization**
  - **BM25**
  - **Query likelihood**

- BM25 is most popular

Summary

- Design of ranking function f(q,d) pre-requires a computational definition of relevance (retrieval model)
- Many models are equally effective with no single winner
- State of the art ranking functions tend to rely on
  - Bag of words representation
  - Term Frequency (TF) and Document Frequency (DF) of words
  - Document length

There is no single winner yet. Researchers are still active and working on this problem, trying to find a truly optimal retrieval model

Additional Readings

- Detailed discussion and comparison of state of the art models
  - Hui Fang, Tao Tao, and Chengxiang Zhai. 2011. Diagnostic Evaluation of Information Retrieval Models. *ACM Trans. Inf. Syst.* 29, 2, Article 7 (April 2011)

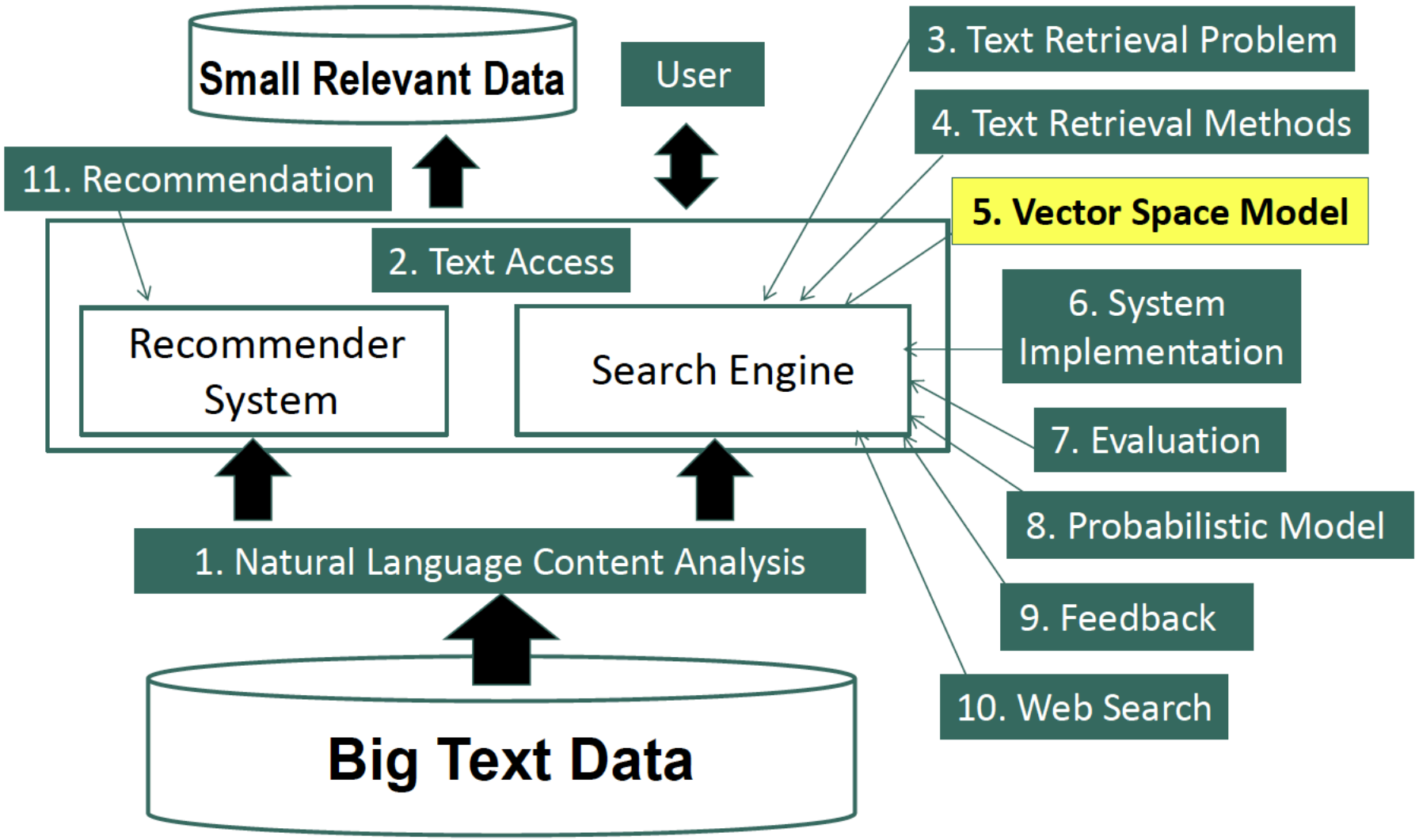
- Broad review of different retrieval models
  - ChengXiang Zhai, *Statistical Language Models for Information Retrieval* , Morgan & Claypool Publishers, 2008. (Chapter 2)

Information Retrieval & Text Mining

Vector Space Retrieval Model
Basic Idea

Dr. Iqra Safder
FAST NUCES, Lahore

Course Schedule



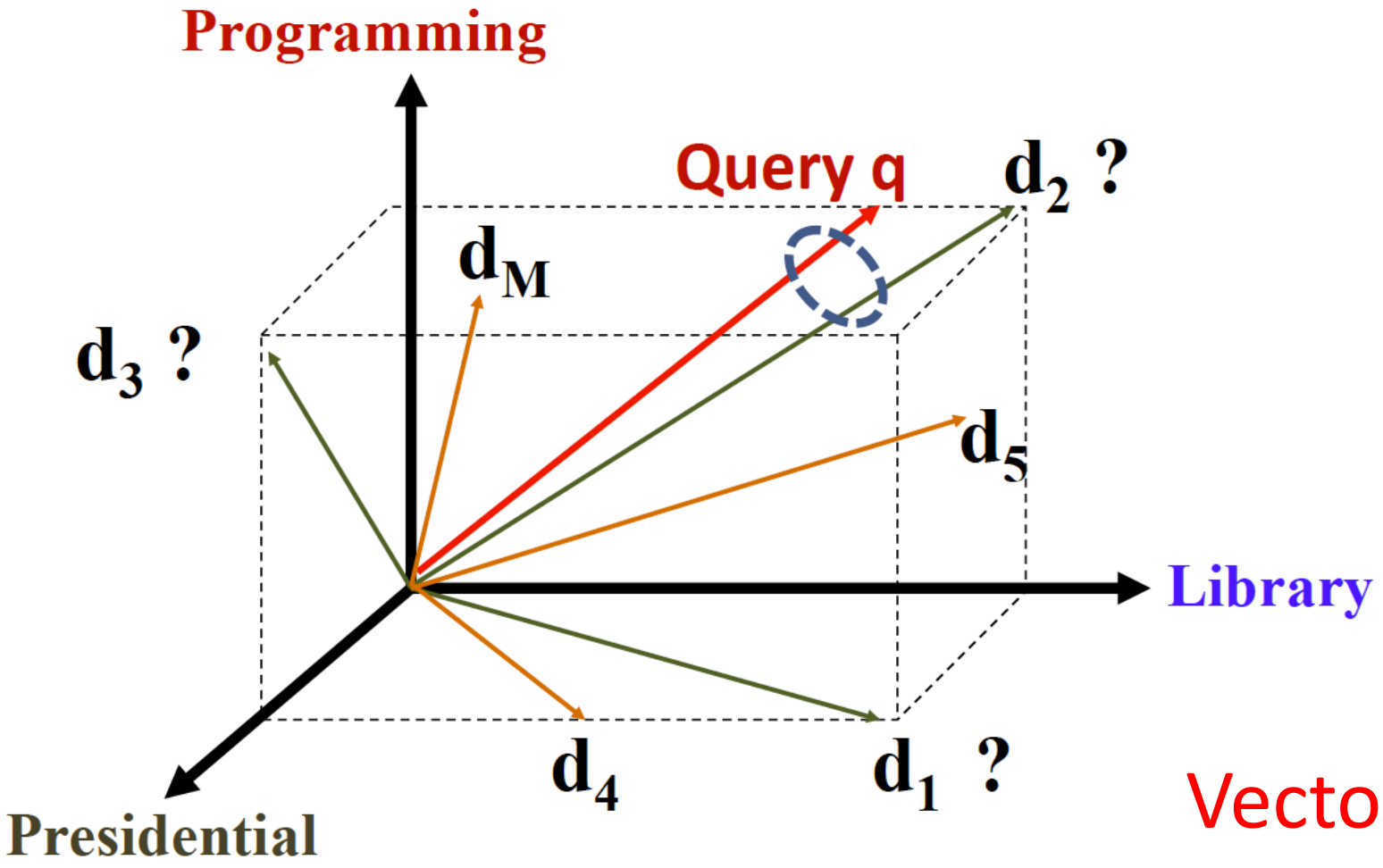
Many Different Retrieval Models

- Similarity-based models: f(q,d) = similarity(q,d)
  - Vector space model

Relevance based method.

Assumptions:
If a first document is more similar to the query then it is assumed as more relavent document than others.

Vector Space Model (VSM): Illustration



Vector Space

VSM Is a Framework

- Represent a doc/query by a term vector
  - **Term**: basic concept, e.g., word or phrase
  - Each term defines one dimension
  - N terms define an **N-dimensional space**
  - **Query** vector: **q**=(x_1, …x_N), $x_i \in \Re$ is query term weight
  - **Doc** vector: **d**=(y_1, …y_N), $y_j \in \Re$ is doc term weight

- relevance(q,d) $\propto$ similarity(**q**,**d**) =f(q,d)

What VSM Doesn't Say

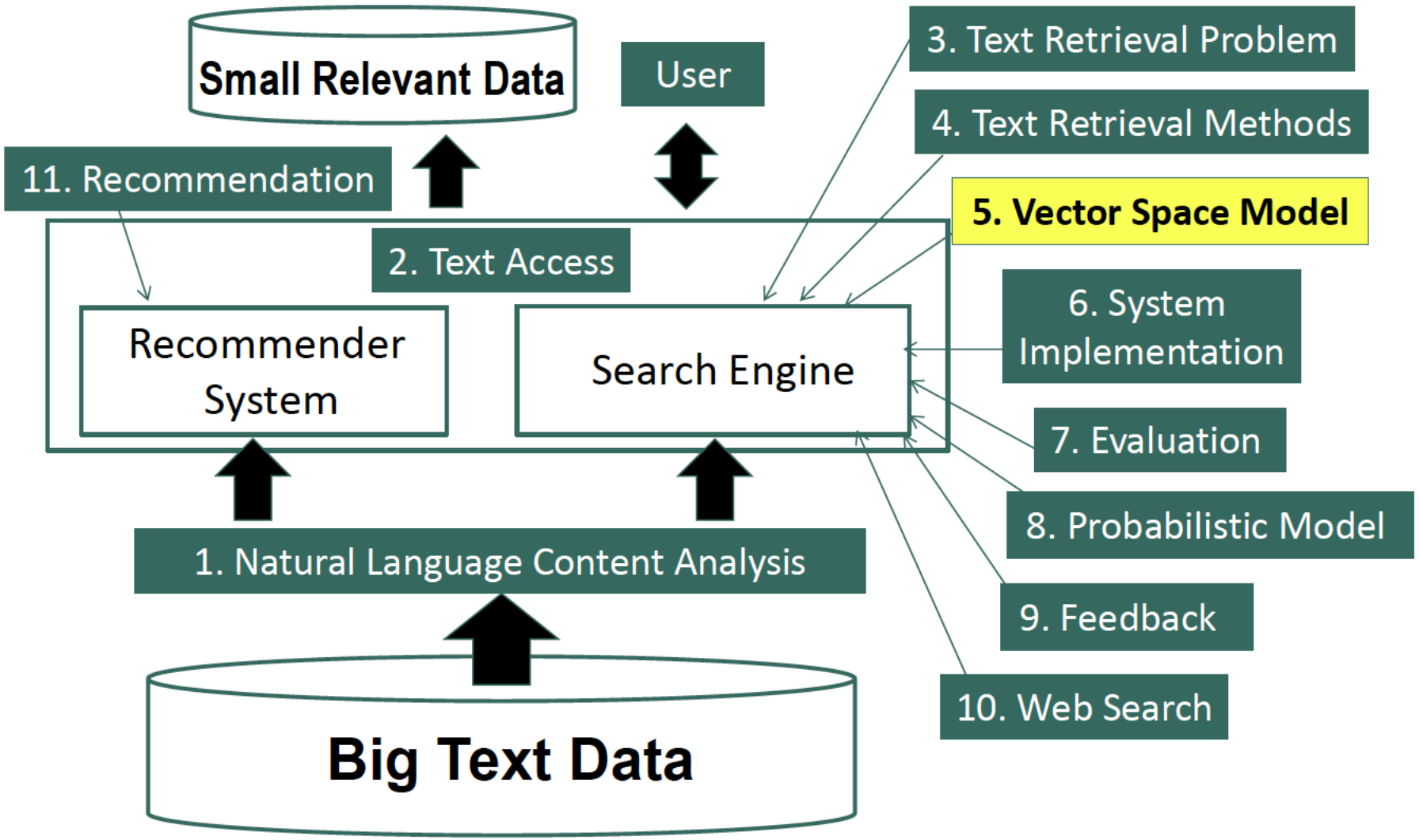
- How to define/select the "basic concept"
  - Concepts are assumed to be orthogonal → Statistically Independent
- How to place docs and query in the space (= how to assign term weights)
  - Term weight in query indicates importance of term
  - Term weight in doc indicates how well the term characterizes the doc
- How to define the similarity measure

Information Retrieval & Text Mining

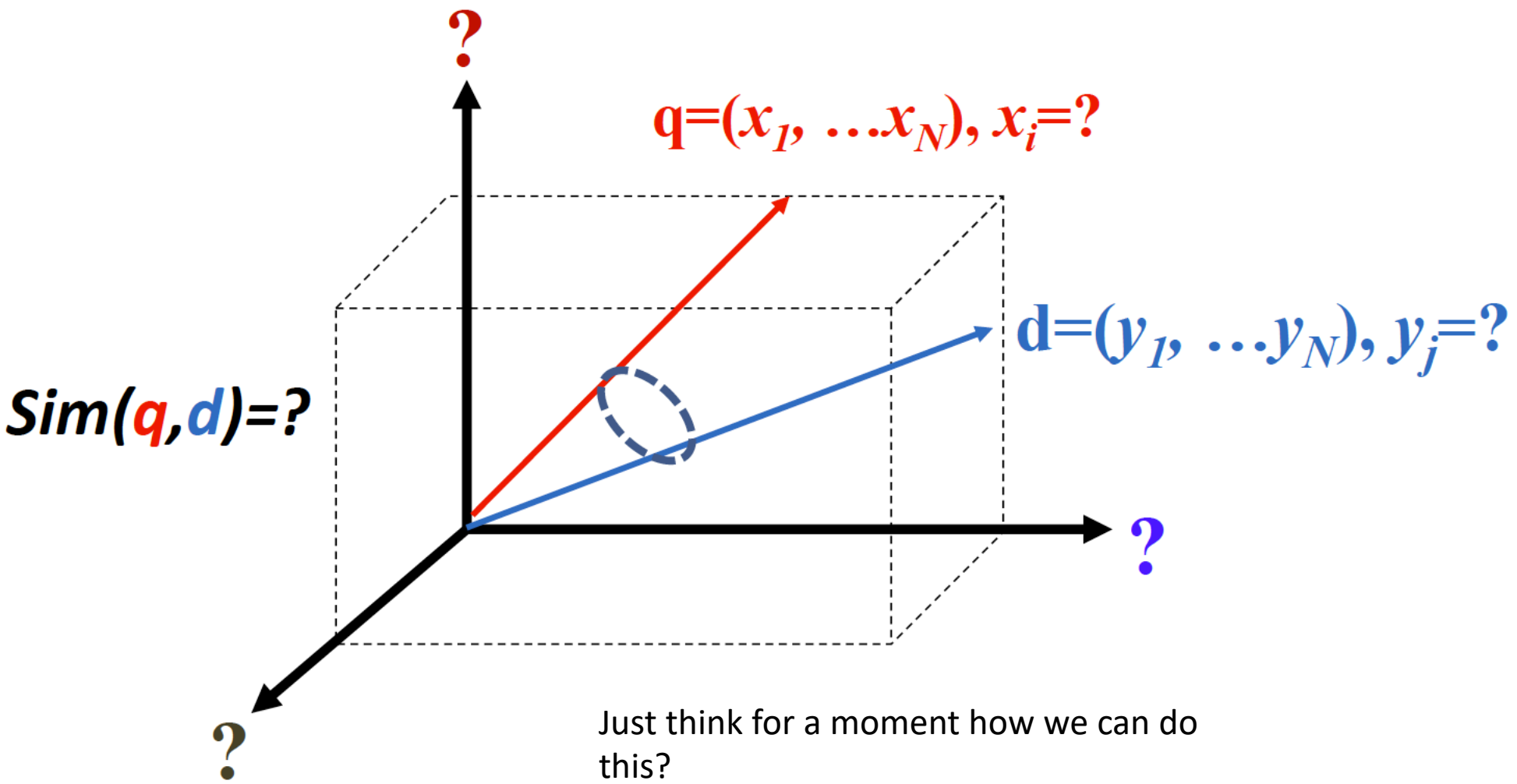
Vector Space Model
Simplest Instantiation

Dr. Iqra Safder
FAST NUCES, Lahore

Course Schedule

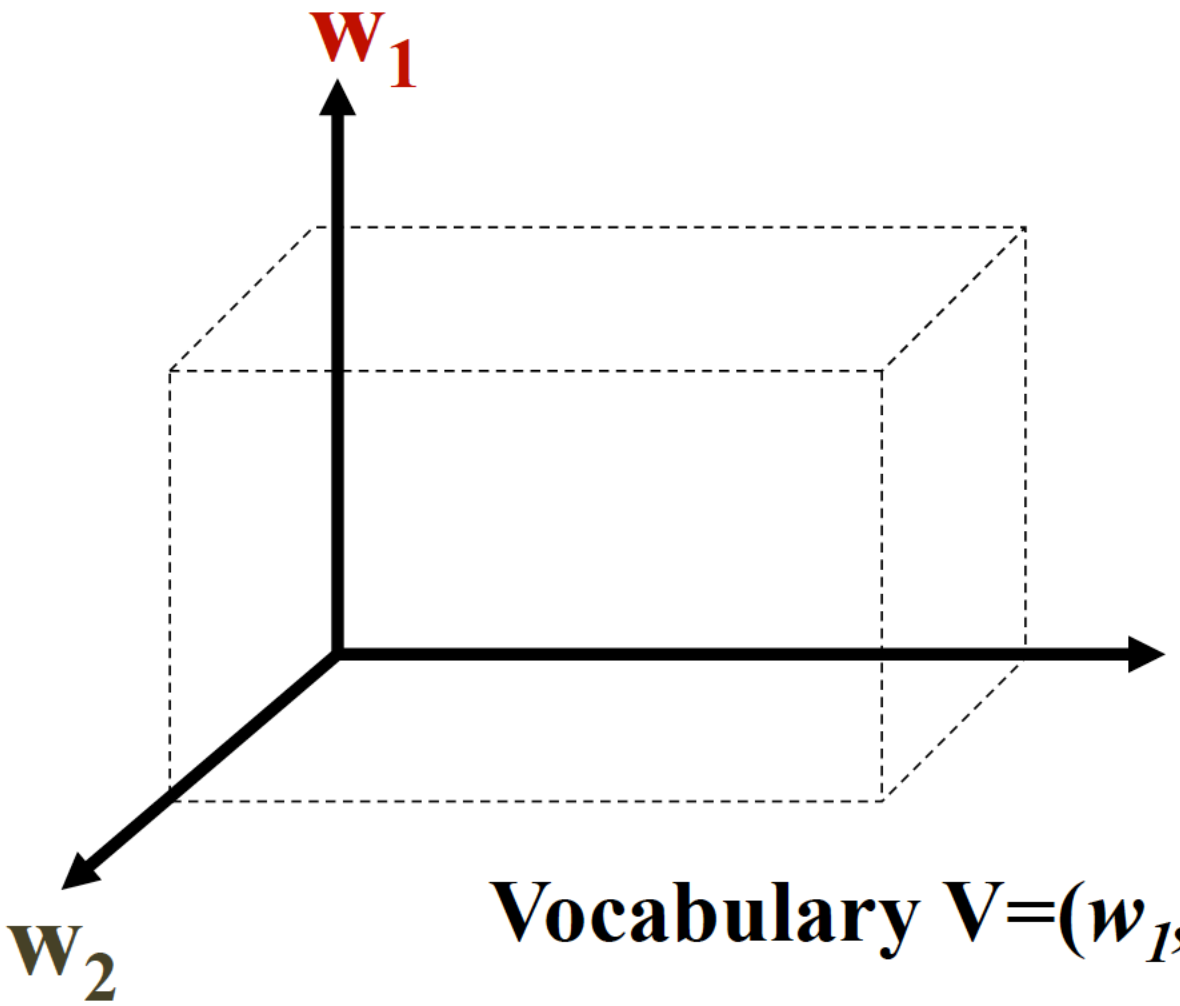


What VSM Doesn't Say



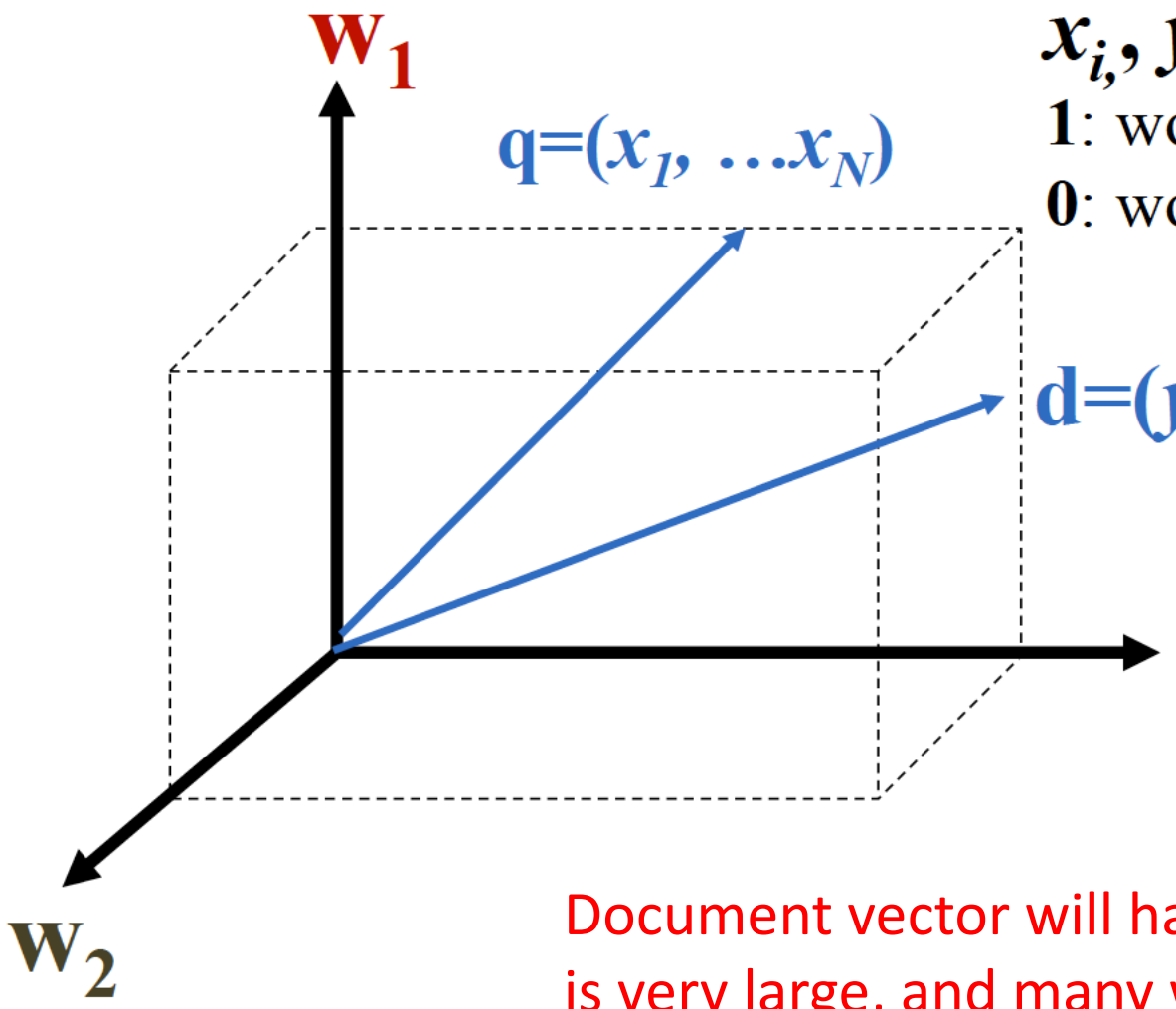
Just think for a moment how we can do this?

Dimension Instantiation: Bag of Words (BOW)



Vocabulary V=(w_1, …w_N)

Vector Placement: Bit Vector



$x_i, y_i \in \{0,1\}$

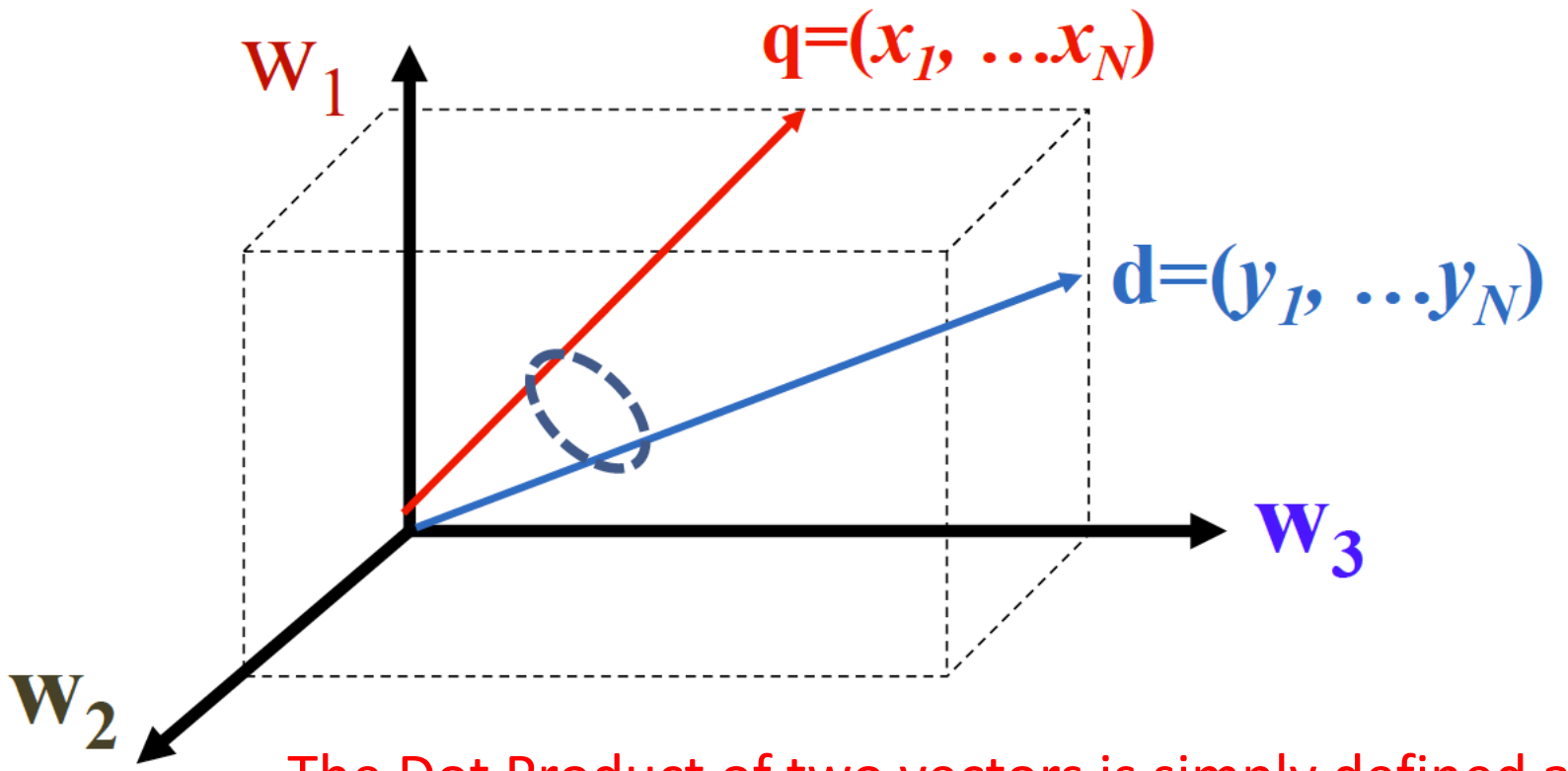
1: word W_i is **present**

0: word W_i is **absent**

Document vector will have Zeros because vacabulary is very large, and many words only occasionally occurrs in the document.

Similarity Instantiation: Dot Product

$$Sim(q,d)=q.d= x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$



The Dot Product of two vectors is simply defined as the sum of the products of the corresponding elements of the two vectors.

Simplest VSM= Bit-Vector + Dot-Product + BOW

$q=(x_1, \ldots x_N)$

$d=(y_1, \ldots y_N)$

$x_i, y_i \in \{0,1\}$

1: word W_i is **present**

0: word W_i is **absent**

$$Sim(q,d)=q.d= x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

What does this ranking function intuitively capture?
Is this a good ranking function?

An Example: How Would You Rank These Documents?

Query = "news about presidential campaign"

Ideal Ranking?

d1 … news about …

d2 … news about organic food campaign…

d3 … news of presidential campaign …

d4 … news of presidential campaign …
… presidential candidate …

d5 … news of organic food campaign…
campaign…campaign…campaign…

An Example: How Would You Rank These Documents?

Query = "news about presidential campaign"

		Ideal Ranking?
d1	… **news about** …	d4 +
d2	… **news about** organic food **campaign**…	d3 +
d3	… **news** of **presidential campaign** …	
d4	… **news** of **presidential campaign** … … **presidential** candidate …	d1 -
d5	… **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…	d2 -
		d5 -

Ranking Using the Simplest VSM

Query = "news about presidential campaign"

d1 … **news about** …

d3 … **news** of **presidential campaign** …

V= {news, about, presidential, campaign, food …. }

q= (1, 1, 1, 1, 0, …)

d1= (1, 1, 0, 0, 0, …)

f(q,d1)= 1*1+1*1 +1* 0+1*0+0*0 + …= 2

d3= (1, 0, 1, 1, 0, …)

f(q,d3)= 1*1+1*0 +1*1 +1*1+0*0+ … = 3

Is the Simplest VSM Effective?

Query = "news about presidential campaign"

d1	… **news about** …	f(q,d1)=2
d2	… **news about** organic food **campaign**…	f(q,d2)=3
d3	… **news** of **presidential campaign** …	f(q,d3)=3
d4	… **news** of **presidential campaign** … … **presidential** candidate …	f(q,d4)=3
d5	… **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…	f(q,d5)=2

Summary

- VSM instantiation: dimension, vector placement, similarity

- Simplest VSM
  - Dimension = word
  - Vector = 0-1 bit vector (word presence/absence)
  - Similarity = dot product
  - f(q,d) = number of **distinct** query words matched in d

Simple vector space model still doesn't work well, and we need to improve it. And this is a topic that we're going to cover in the next lecture.