https://vvtesh.sarahah.com/

Information Retrieval

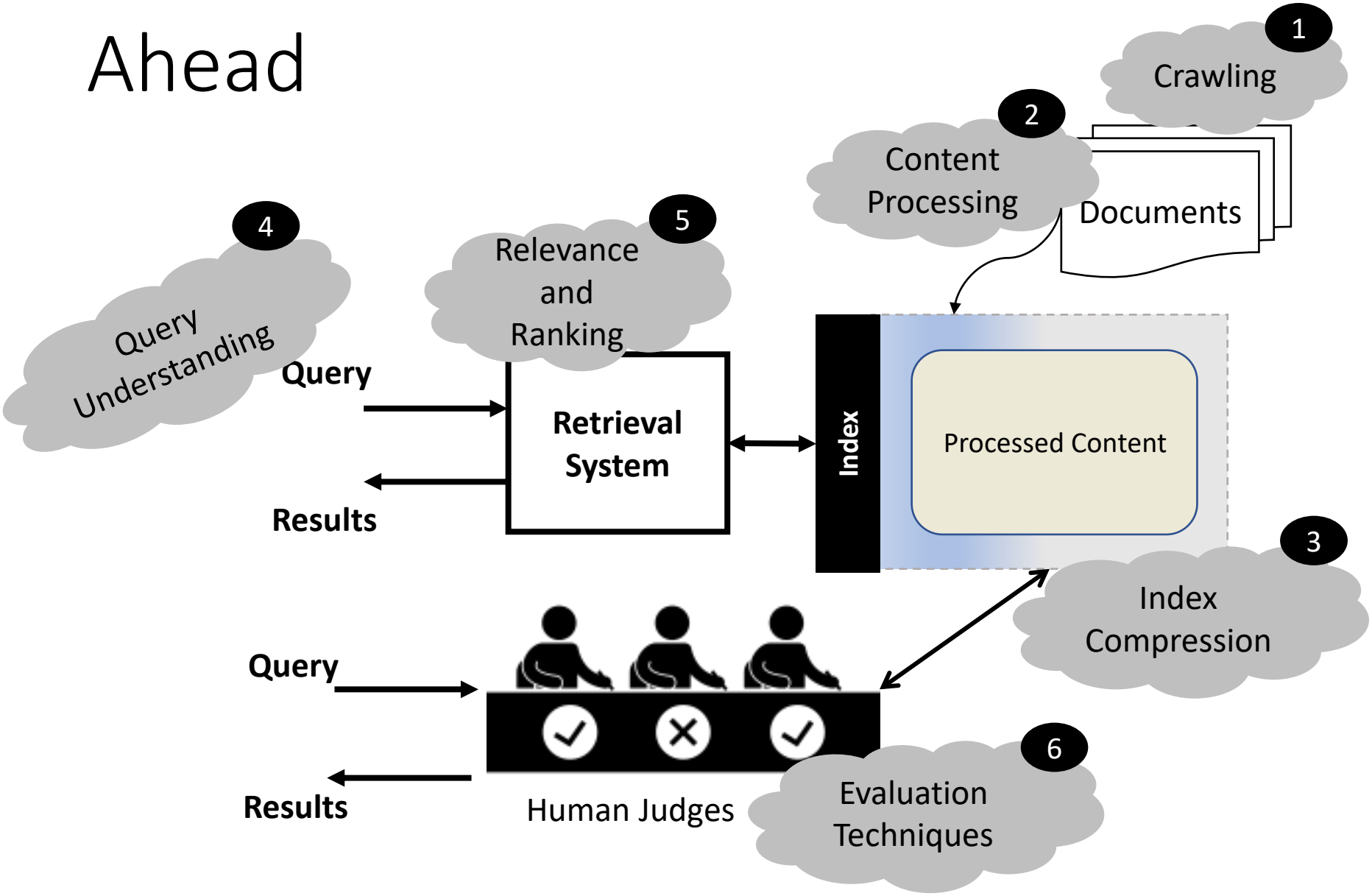
Venkatesh Vinayakarao

Term: Aug – Sep, 2019
Chennai Mathematical Institute

Computers understand very little of the meaning of human language. This profoundly limits our ability to give instructions to computers, the ability of computers to explain their actions to us, and the ability of computers to analyse and process text. Vector space models (VSMs) of semantics are beginning to address these limits.

– Turney and Pantel, JAIR 2010.

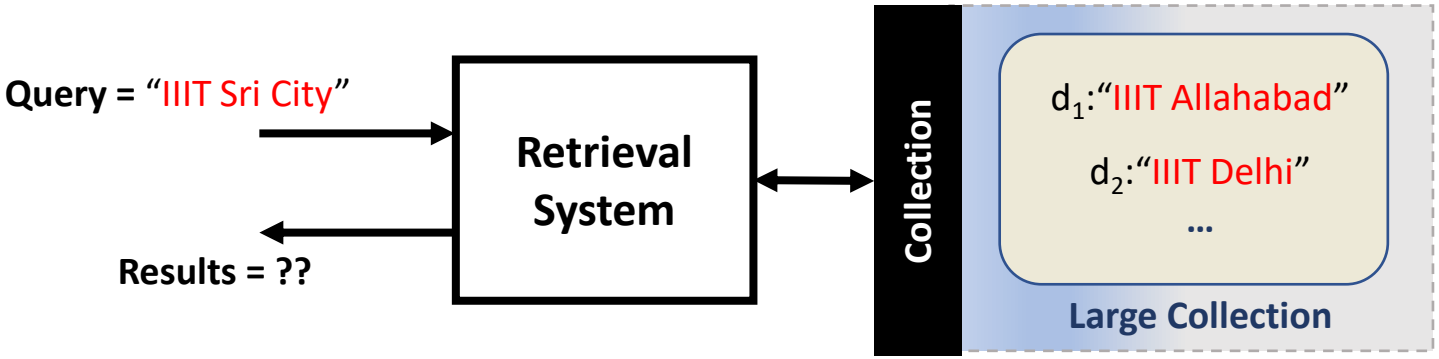
Information Retrieval – Road Ahead



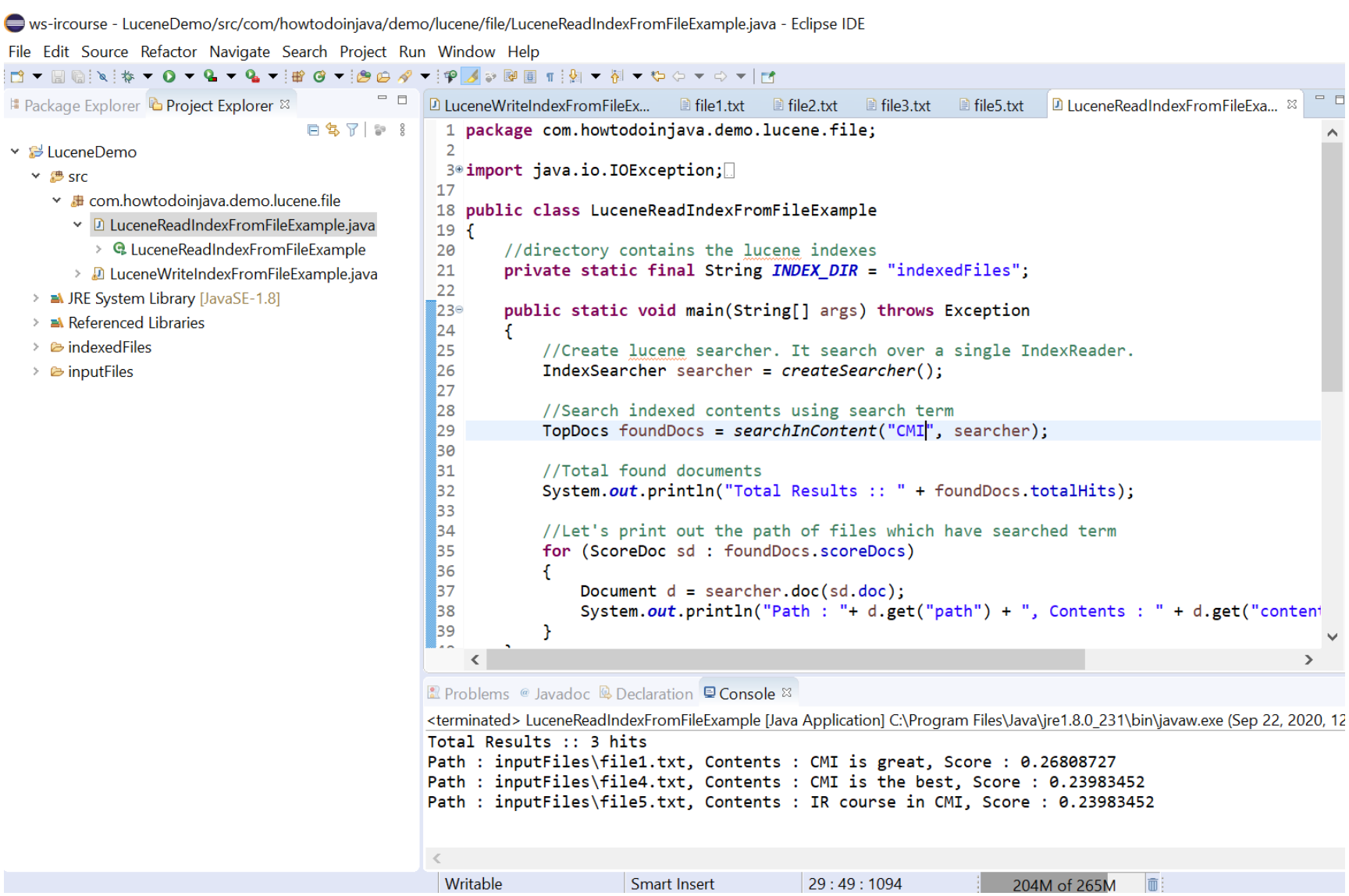
Simple Retrieval Problem

- A **collection** with 5 **documents** having the following contents
  - d1: IIIT ALLAHABAD
  - d2: IIIT DELHI
  - d3: IIIT GUWAHATI
  - d4: IIIT KANCHIPURAM
  - d5: IIIT SRI CITY
- **Query** is
  - IIIT SRI CITY
- Which **document** will you match and why?

The Problem – How to Rank?



Compare each document with the query and assign a score of relevance?

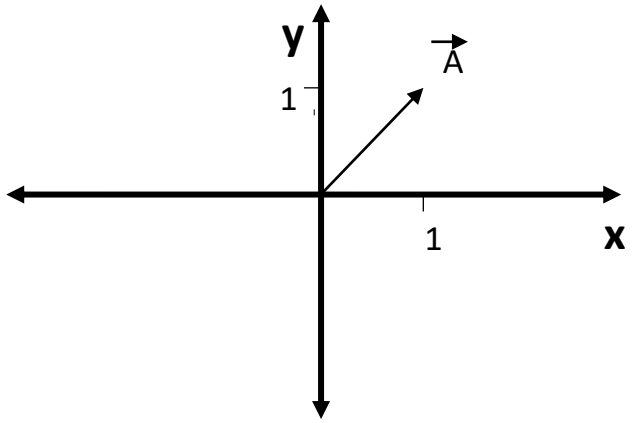

ws-ircourse - LuceneDemo/src/com/howtodoinjava/demo/lucene/file/LuceneReadIndexFromFileExample.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
LuceneDemo
src
com.howtodoinjava.demo.lucene.file
LuceneReadIndexFromFileExample.java
LuceneReadIndexFromFileExample
LuceneWriteIndexFromFileExample.java
JRE System Library [JavaSE-1.8]
Referenced Libraries
indexedFiles
inputFiles
LuceneWriteIndexFromFileEx...
file1.txt
file2.txt
file3.txt
file5.txt
LuceneReadIndexFromFileExa...
1 package com.howtodoinjava.demo.lucene.file;
2
3 import java.io.IOException;
17
18 public class LuceneReadIndexFromFileExample
19 {
20 //directory contains the lucene indexes
21 private static final String INDEX_DIR = "indexedFiles";
22
23 public static void main(String[] args) throws Exception
24 {
25 //Create lucene searcher. It search over a single IndexReader.
26 IndexSearcher searcher = createSearcher();
27
28 //Search indexed contents using search term
29 TopDocs foundDocs = searchInContent("CMI", searcher);
30
31 //Total found documents
32 System.out.println("Total Results :: " + foundDocs.totalHits);
33
34 //Let's print out the path of files which have searched term
35 for (ScoreDoc sd : foundDocs.scoreDocs)
36 {
37 Document d = searcher.doc(sd.doc);
38 System.out.println("Path : "+ d.get("path") + ", Contents : " + d.get("conten
39 }
Problems Javadoc Declaration Console
<terminated> LuceneReadIndexFromFileExample [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Sep 22, 2020, 12
Total Results :: 3 hits
Path : inputFiles\file1.txt, Contents : CMI is great, Score : 0.26808727
Path : inputFiles\file4.txt, Contents : CMI is the best, Score : 0.23983452
Path : inputFiles\file5.txt, Contents : IR course in CMI, Score : 0.23983452
Writable
Smart Insert
29 : 49 : 1094
204M of 265M

An Algebraic Approach

Revisiting
Linear Algebra

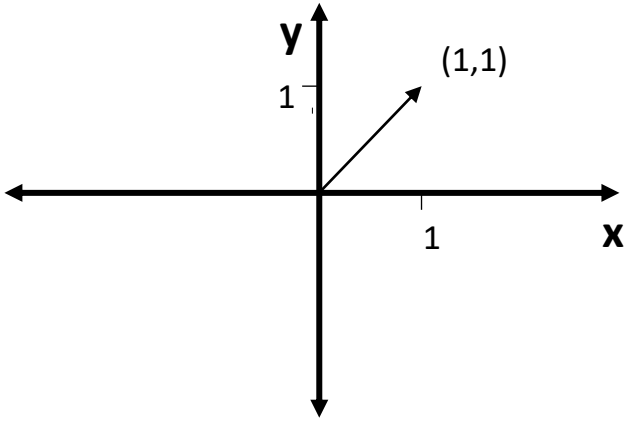
Vectors

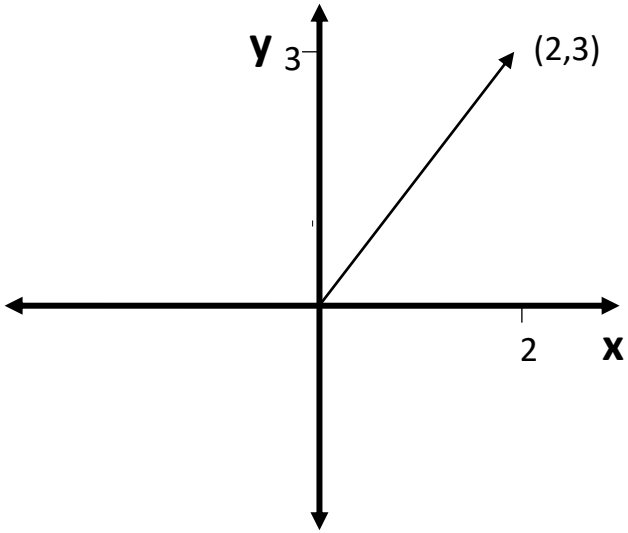
- Geometric entity which has magnitude and direction



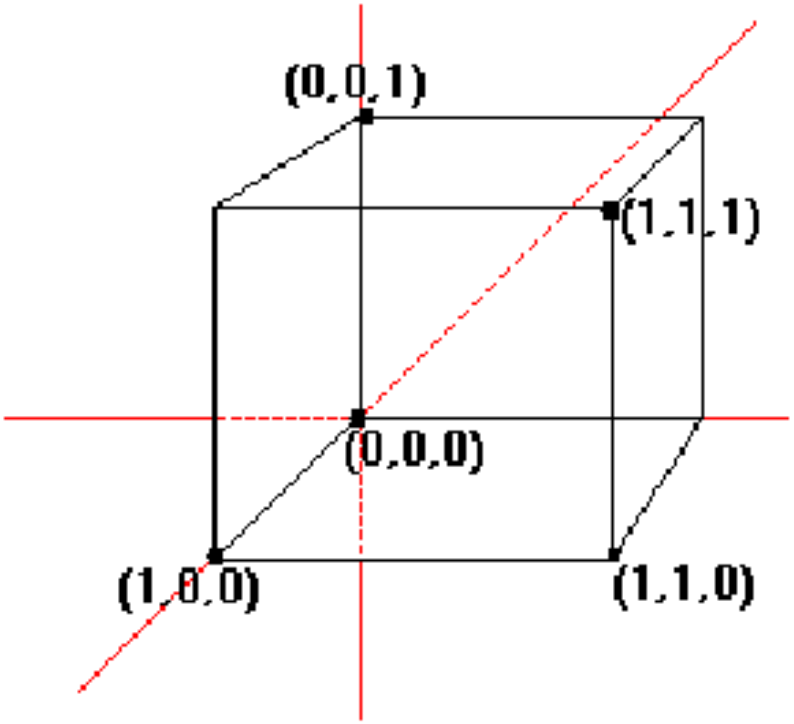
- If (x,y) is our vector of interest, this figure shows $\vec{A}$ vector = (1,1).

How is (2,3) Different?





What is (1,1,1) ?

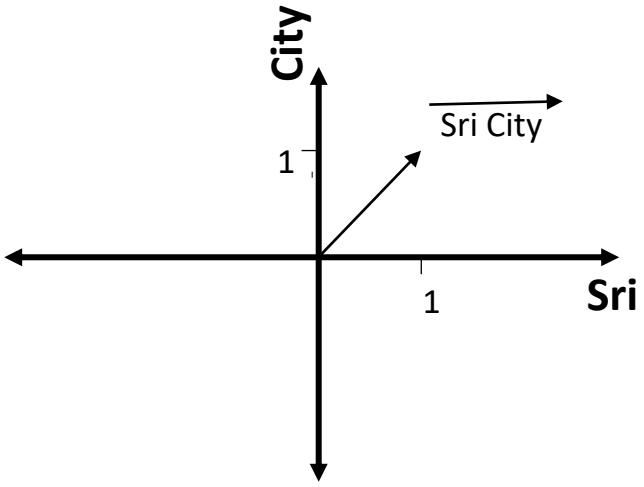


Remember!

A number is just a mathematical object. We give meaning to it!

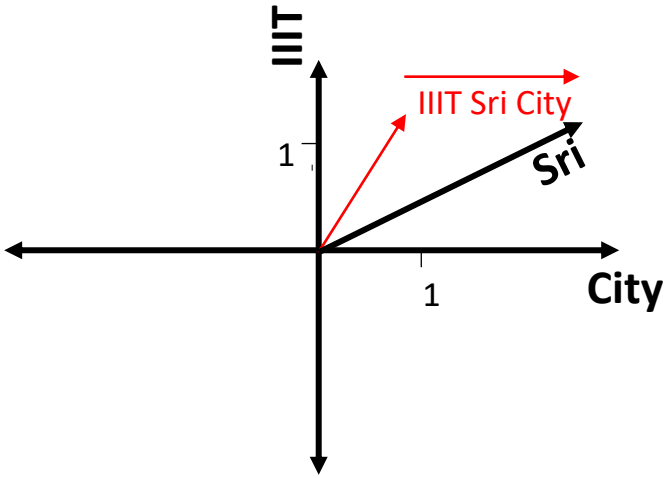
Sentences are Vectors

- “Sri City” as a vector



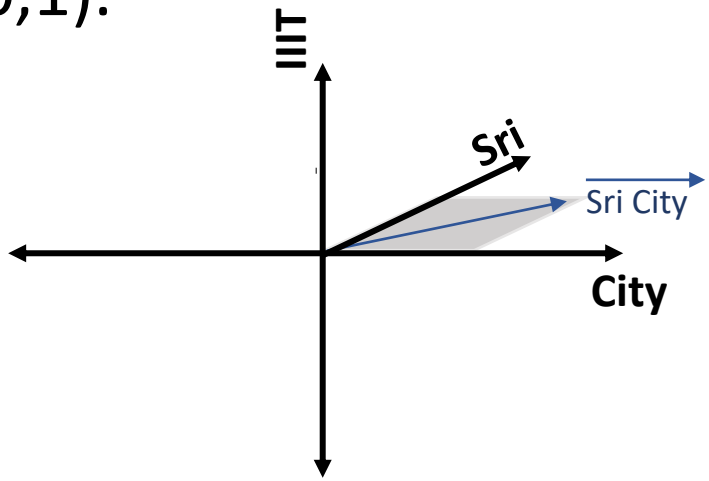
Sentences are Vectors

- "IIIT Sri City" is a 3-dimensional vector



Sentences are Vectors

- On this 3D space, “Sri City” vector will lie on the x (City) and z (Sri) plane. If (x,y,z) denotes the vector, “Sri City” is (1,0,1).



More Linear Algebra…

- So, we learned to represent English phrases on the vector space.
- We need something more!

Revisiting Matrices

Natural Language Phrases as Vectors

Let query q = "IIIT Sri City".

Let document, d_1 = "IIIT Sri City" and d_2 = "IIIT Delhi".

	IIIT	Sri	City	Delhi
q	1	1	1	0
d_1	1	1	1	0
d_2	1	0	0	1

q = (1,1,1,0), d_1= (1,1,1,0) and d_2 = (1,0,0,1)

Quiz

- Considering the following vectors:

	IIIT	Sri	City	Delhi
q	1	1	1	0
d_1	1	1	1	0
d_2	1	0	0	1

- What is the Natural Language (NL) equivalent of (0,1,1,0) ?
- What is the NL equivalent of (1,0,0,1) ?
- What is the vector for Delhi?
- What is the NL equivalent of q?

Which of the Following are Sets?

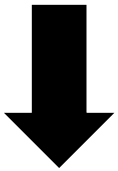
- {1, 2, 3, 4, 5, 6, 5, 7, 8, 9, 10, 11, 12, 13}
- {A, B, C, D, E, F, G, H, I, I, J, K, L, M, N, O}
- {apple, banana, orange, apple, banana, orange}

Bag

- {1, 2, 3, 4, 5, 6, 5, 7, 8, 9, 10, 11, 12, 13}
- {A, B, C, D, E, F, G, H, I, I, J, K, L, M, N, O}
- {apple, banana, orange, apple, banana, orange}

Set of Words Representation

- "IIIT Sri City" → {IIIT, Sri, City}
- "IIIT Sri City, Sri City" → {IIIT, Sri, City}



	IIIT	Sri	City
q	1	1	1

Leads to same term-document matrix

Set Similarity

- Similarity between two sets is easy

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Jaccard Similarity

Quiz

- What is the Jaccard similarity between
  - {1,2,3} and {4,5,6}?
  - {1,2,3} and {1,2,4}?
  - {1,2,3} and {1,2,3}?

Quiz

- What is the Jaccard similarity between
  - {1,2,3} and {4,5,6} = 0
  - {1,2,3} and {1,2,4} = $\frac{|\{1,2\}|}{|\{1,2,3,4\}|}$ = 0.5
  - {1,2,3} and {1,2,3} = 1

Quiz

- What is the Jaccard similarity between
  - “IIIT is Great” and “IIITD is Great”?

Quiz

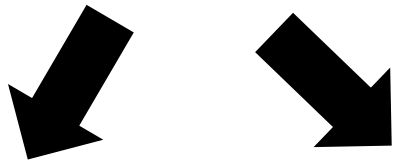
- What is the Jaccard similarity between
  - “IIIT is Great” and “IIITD is Great”?
- Same as Jaccard Similarity between {IIIT, is, Great} and {IIITD, is, Great}. Equals 0.5.

Ranked Retrieval

In the **Ranked Retrieval** model, we may want to model documents as **bag** of words.

Bag of Words Representation

- "IIIT Sri City" → {IIIT, Sri, City}
- "IIIT Sri City, Sri City" → [IIIT, Sri, Sri, City, City]



IIIT Sri City

	IIIT	Sri	City
q	1	1	1

IIIT Sri City, Sri City

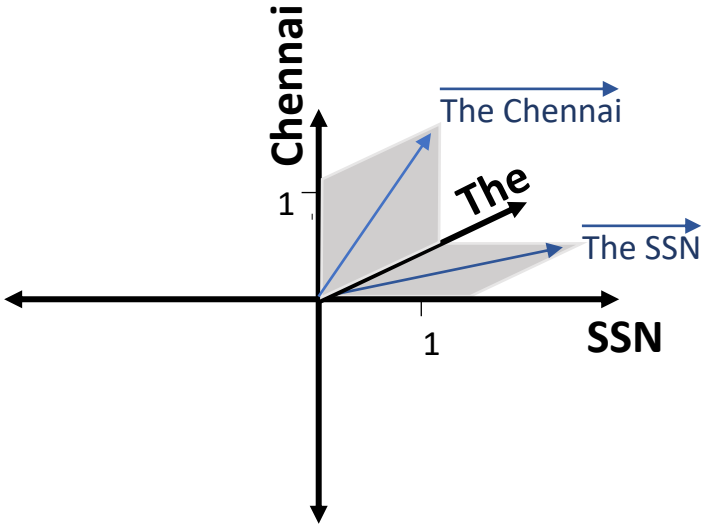
	IIIT	Sri	City
q	1	2	2

Leads to different term-document matrix

Set of Words → Bag of Words

Comparing Sentences

- We can compare sentences using the angle between vectors

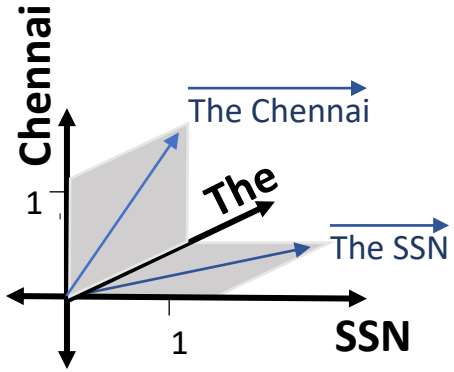


Angle between two vectors

- What is the angle between $\overrightarrow{\text{The}}$ and $\overrightarrow{\text{SSN}}$ vectors?
- What is the angle between $\overrightarrow{\text{SSN}}$ and $\overrightarrow{\text{Chennai}}$ vectors?
- What is the angle between $\overrightarrow{\text{The SSN}}$ and $\overrightarrow{\text{The SSN}}$ vectors?

Mathematical Notation

- We represent vectors as follows:
  - Vector = (dimension1, dimension2, dimension3, …)
    - First, define the dimensions
    - Next, put “1” if the word is present in the sentence, else “0”
- Example



Vector = (dimension1, d2, d3, …)

In our case,
vector = (The, SSN, Chennai)

So,

$\overrightarrow{\text{The Chennai}}$ = (1,0,1)

$\overrightarrow{\text{The SSN}}$ = (1,1,0)

Similarity Score

- D1 = “Chennai”
- D2 = “Chennai”

- Quiz
  - On a scale of 0 – 1, how similar are D1 and D2?
    - 0 ➔ Dissimilar
    - 1 ➔ Identical

Similarity Score

- D1 = “Chennai”
- D2 = “IIIT”

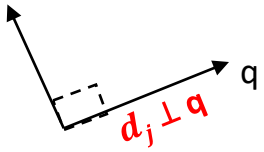
- Quiz
  - On a scale of 0 – 1, how similar are D1 and D2?

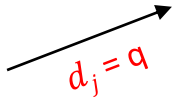
How to Convert 0 to 90 → 1 to 0

Revisiting Trigonometry

Converting from “0 – 90” to “1 – 0”

- For convenience, We convert the angles 0 – 90 to values 1 - 0
  - When vectors are same, we want to output 1.
  - When vectors are perpendicular, we want to output 0.





0 – 90 to 1 – 0: How?

	0°	30°	45°	60°	90°
sin θ	0	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{\sqrt{3}}{2}$	1
cos θ	1	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	0
tan θ	0	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$	Not defined

Back to Trigonometry

- If x and y are non-unit vectors, what is the cosine of angle between them (cos Θ)?

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos(\theta)$$

Matching Documents to Queries

- Document as a vector of term-weights

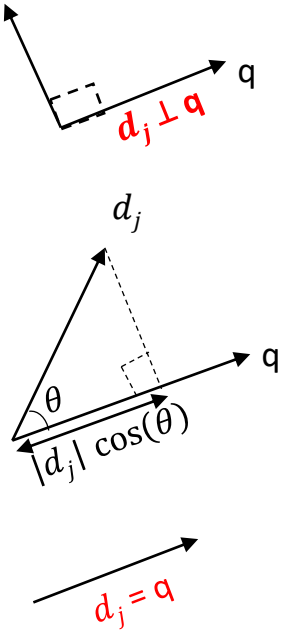
$$d_j = (w_{1j}, w_{2j}, \ldots w_{nj})$$

- Query as a vector of term-weights

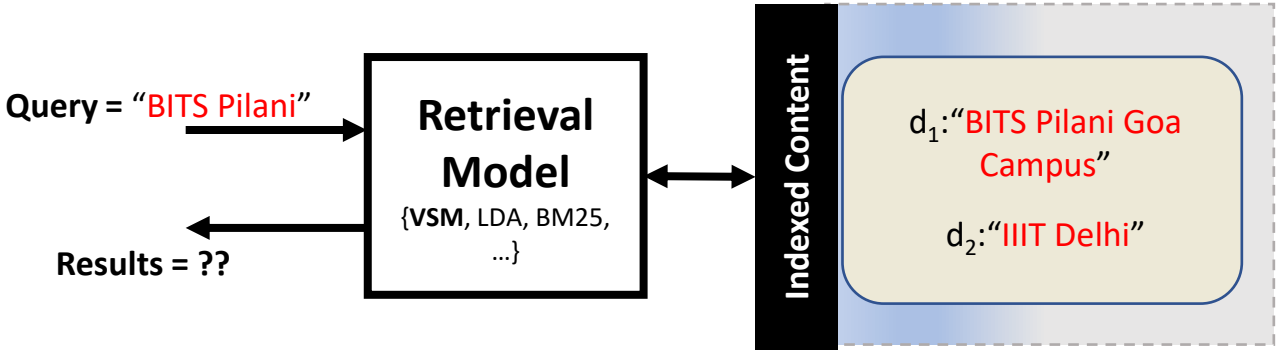
$$q = (w_1, w_2, \ldots wm)$$

- Similarity between these vectors can be represented as

$$Cosine\ Similarity = \cos(\theta) = \frac{d_j . q}{||d_j||\ ||q||}$$



Which Document to Retrieve?



A Boolean Term Document Matrix

	BITS	Pilani	Goa	Campus	IIIT	Delhi
q	1	1	0	0	0	0
d_1	1	1	1	1	0	0
d_2	0	0	0	0	1	1

Example

Let query q = "BITS Pilani".

Let document, d_1 = "BITS Pilani Goa Campus" and d_2 = "IIIT Delhi".

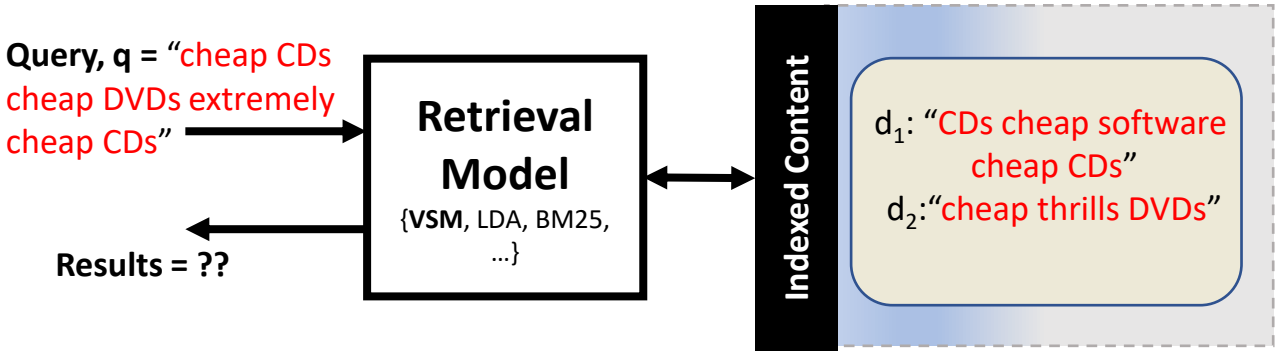
	BITS	Pilani	Goa	Campus	IIIT	Delhi
q	1	1	0	0	0	0
d_1	1	1	1	1	0	0
d_2	0	0	0	0	1	1

In our VSM, q = (1,1,0,0,0,0), d_1= (1,1,1,1,0,0) and d_2 = (0,0,0,0,1,1)

similarity(d_1, q) = $\frac{d_1.q}{||d_1||\,||q||} = \frac{1.1 + 1.1}{\sqrt{1^2+1^2+1^2+1^2}\,\sqrt{1^2+1^2}} = 0.71.$

similarity(d_2, q) = $\frac{d_2.q}{||d_2||\,||q||} = 0.$

Using the Bag of Words Model

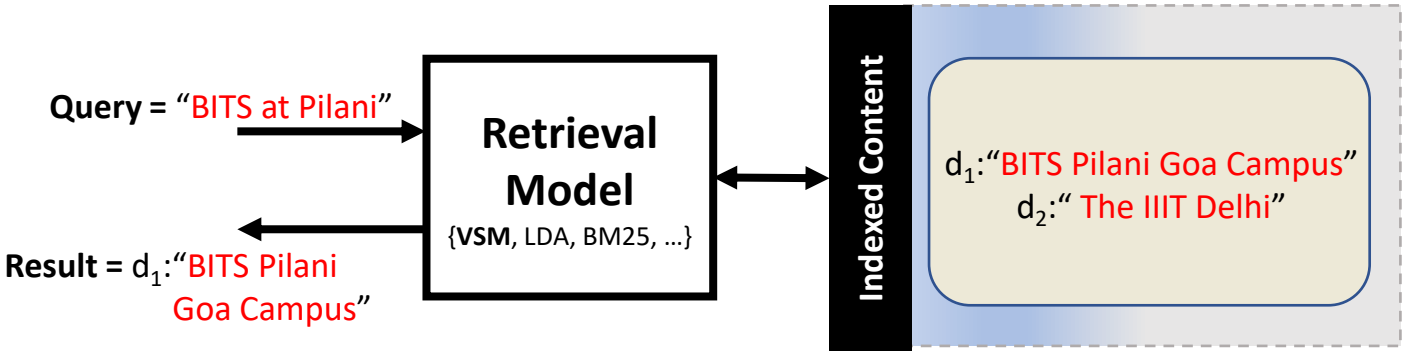


	cheap	CDs	DVDs	extremely	software	thrills
q	3	2	1	1	0	0
d_1	2	2	0	0	1	0
d_2	1	0	1	0	0	1

$sim(q,d_1) = 0.86$

$sim(q,d_2) = 0.59$

Not Every Word is Important!



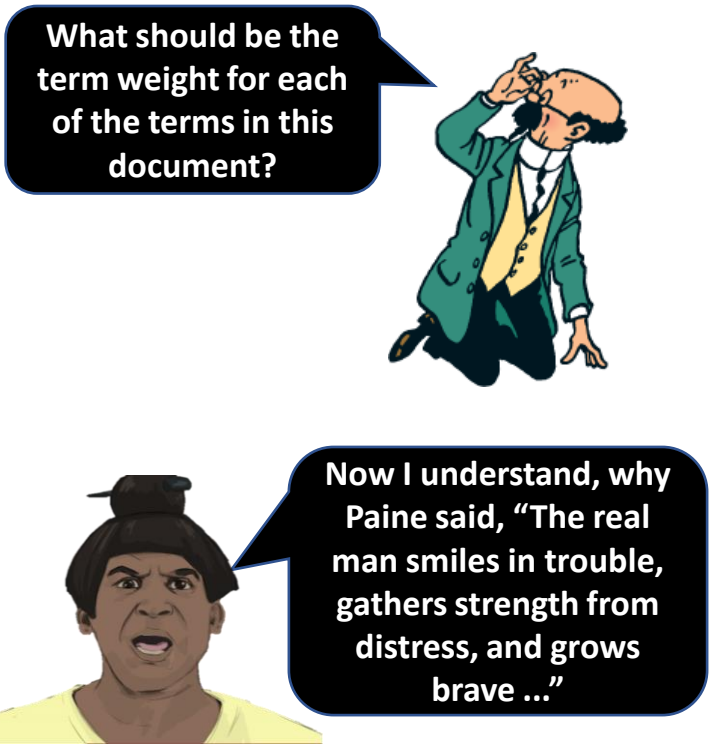
Let us add Term Weights

	BITS	the (* 0)	Pilani	Goa	Campus	IIIT	Delhi
q	1	1 * 0 = 0	1	0	0	0	0
d_1	1	0 * 0 = 0	1	1	1	0	0
d_2	0	1 * 0 = 0	0	0	0	1	1

$sim(q,d_1) = 0.71$

$sim(q,d_2) = 0$

Term Weighting with Inverse Document Frequency



Indexed Content

d_1: *”An inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.”*

d_2: *”A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms ”*

Inverse Document Frequency

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

where N = |D| = Total no. of documents.

$$idf(\text{"}the\text{"}, \{\mathbf{d_1}, \mathbf{d_2}\}) = \log \frac{2}{2} = 0$$

$$idf(\text{"}batsmen\text{"}, \{\mathbf{d_1}, \mathbf{d_2}\}) = \log \frac{2}{1} = 0.3$$

But, do you see any problem? Clue… divide by zero.

Indexed Content

d_1: *"An inverse document frequency factor is incorporated which diminishes **the** weight of terms that occur very frequently in **the** document set and increases **the** weight of terms that occur rarely."*

d_2: *"Sachin Ramesh Tendulkar is a former Indian international cricketer and a former captain of **the** Indian national team, regarded as one of **the** greatest batsmen of all time."*

Variants

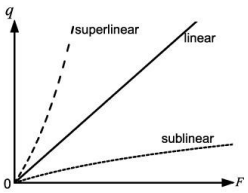
Wikipedia lists the following IDF and TF-IDF variants.

IDF Variants*

weighting scheme	IDF weight ($n_t = \lvert\{d \in D : t \in d\}\rvert$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log\left(1 + \frac{N}{n_t}\right)$
inverse document frequency max	$\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

TF-IDF Variants*

document term weight	query term weight
$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5\frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
$1 + \log f_{t,d}$	$\log(1 + \frac{N}{n_t})$
$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$



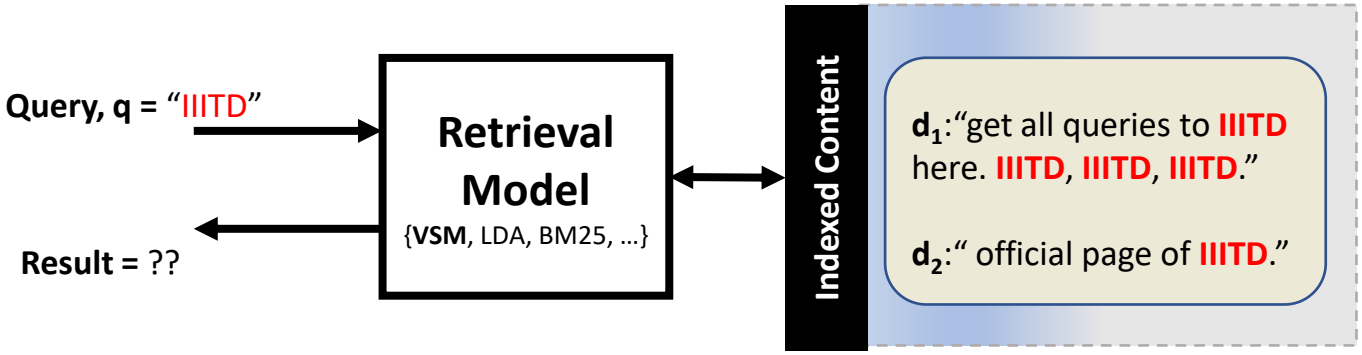
Sub-linear Damping

Add-One Smoothing

Normalization

*See Wikipedia page on TF-IDF (https://en.wikipedia.org/wiki/Tf%E2%80%93idf) for more information.

We are Biased Towards Long Documents



	IIITD	get	all	queries	to	here	official	page	of
q	1	0	0	0	0	0	0	0	0
d_1	4	1	1	1	1	1	0	0	0
d_2	1	0	0	0	0	0	1	1	1

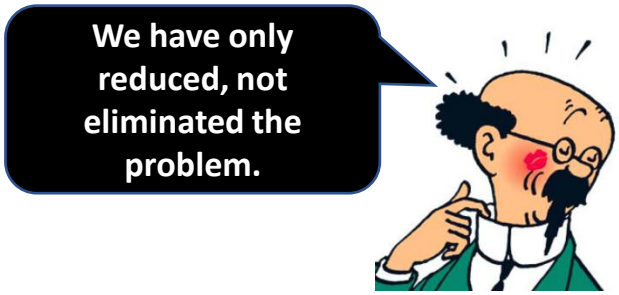
sim(q,d_1) = $\frac{d_1 . q}{||d_1|| \, ||q||} = \frac{4.1}{\sqrt{4^2+1^2+1^2+1^2+1^2+1^2}\sqrt{1^2}}$ **= 0.87**

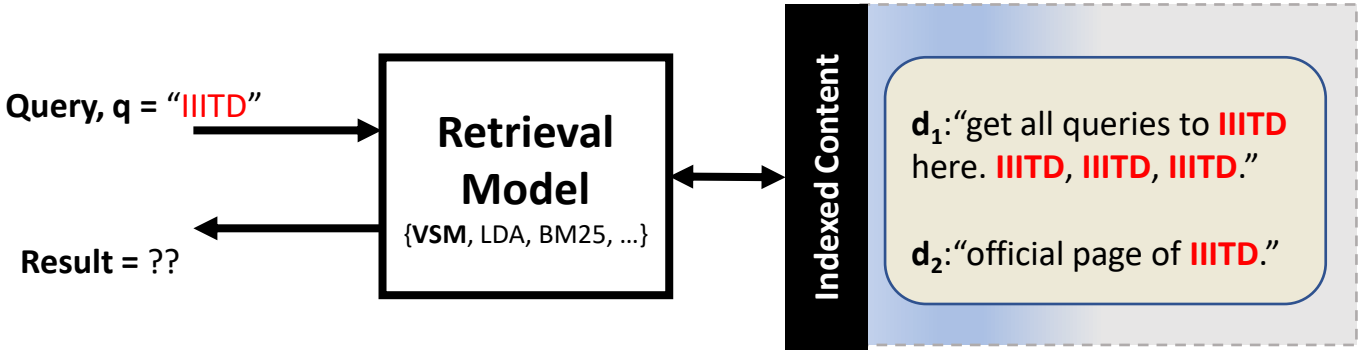
sim(q,d_2) = $\frac{d_2 . q}{||d_2|| \, ||q||} = \frac{1.1}{\sqrt{1^2+1^2+1^2+1^2}\sqrt{1^2}}$ **= 0.5**

Length Normalization

$$tf_L(t, dj) = tf(t, dj) * \log(1 + \frac{Avg.DL}{len(dj)})$$

$$Avg.DL = \frac{9+4}{2} = 6.5$$





$$tf_L("\text{IIITD}", d_1) = 4 * \log(1 + \frac{6.5}{9}) = 0.94$$

$$tf_L("\text{IIITD}", d_2) = 1 * \log(1 + \frac{6.5}{4}) = 0.42$$

Thank You.