S1-20_DSECLZC415 : Data Mining

(Lecture #14 - Mining Unstructured Data)

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Data Mining

Mining Unstructured Data

Structured data vs. Unstructured data

Structured data

comprised of clearly defined data types whose pattern makes them easily searchable

Unstructured data – "everything else"

comprised of data that is usually not as easily searchable, including formats like audio, video, and social media postings

Unstructured Data

Can be

- Text
- www
- Multimedia
- Graph
- Spatial data

……

Mining Text Data – NLP Challenges

Mining Text Data: An Introduction

Data Mining / Knowledge Discovery

Structured Data

HomeLoan (
Loanee: *Frank Rizzo*
Lender: *MWF*
Agency: *Lake View*
Amount: *$200,000*
Term: *15 years*
)

Multimedia

Loans($200K,[map],...)

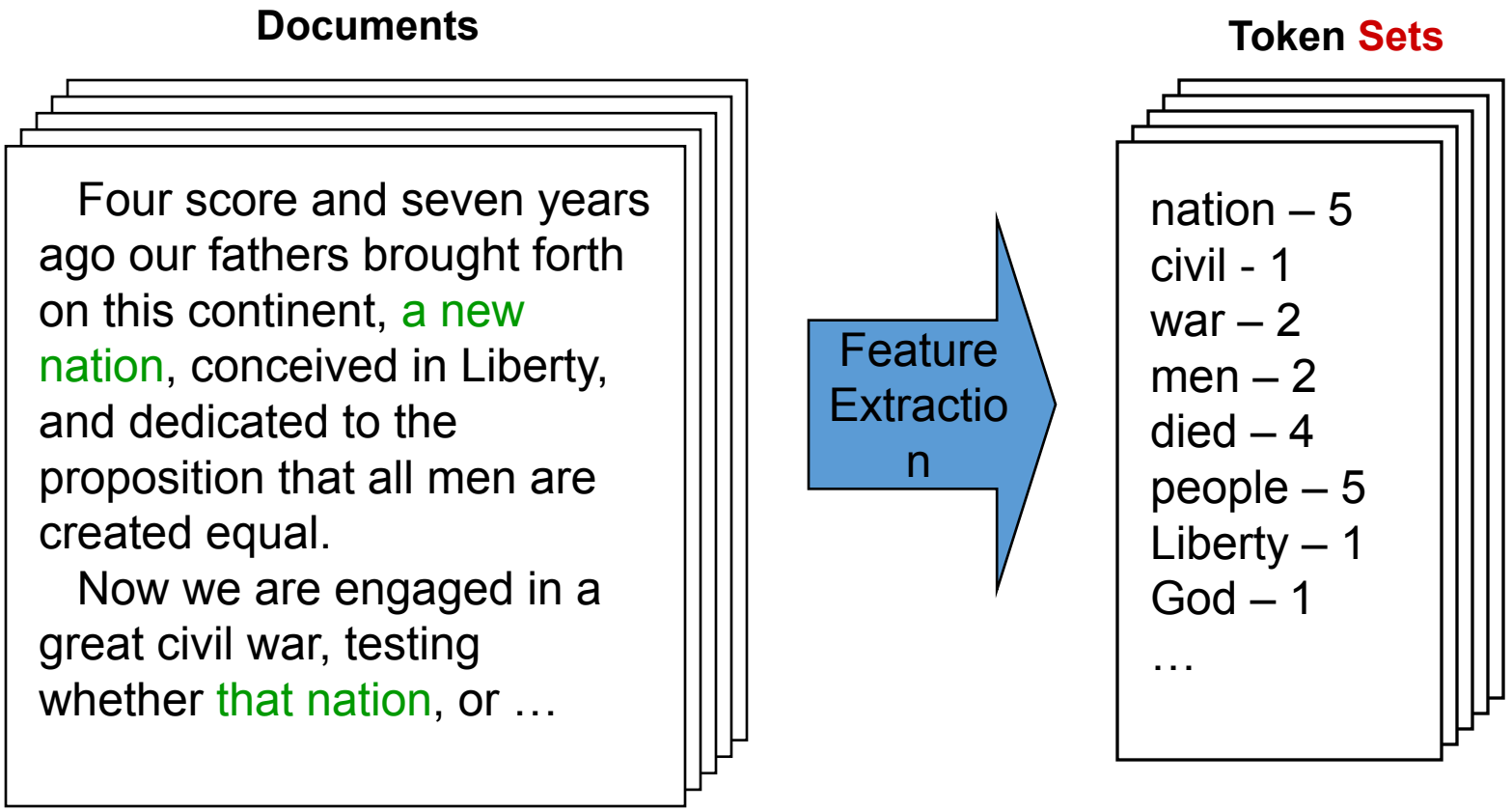
Free Text

Frank Rizzo bought his home from Lake View Real Estate in 1992.
He paid $200,000 under a15-year loan from MW Financial.

Hypertext

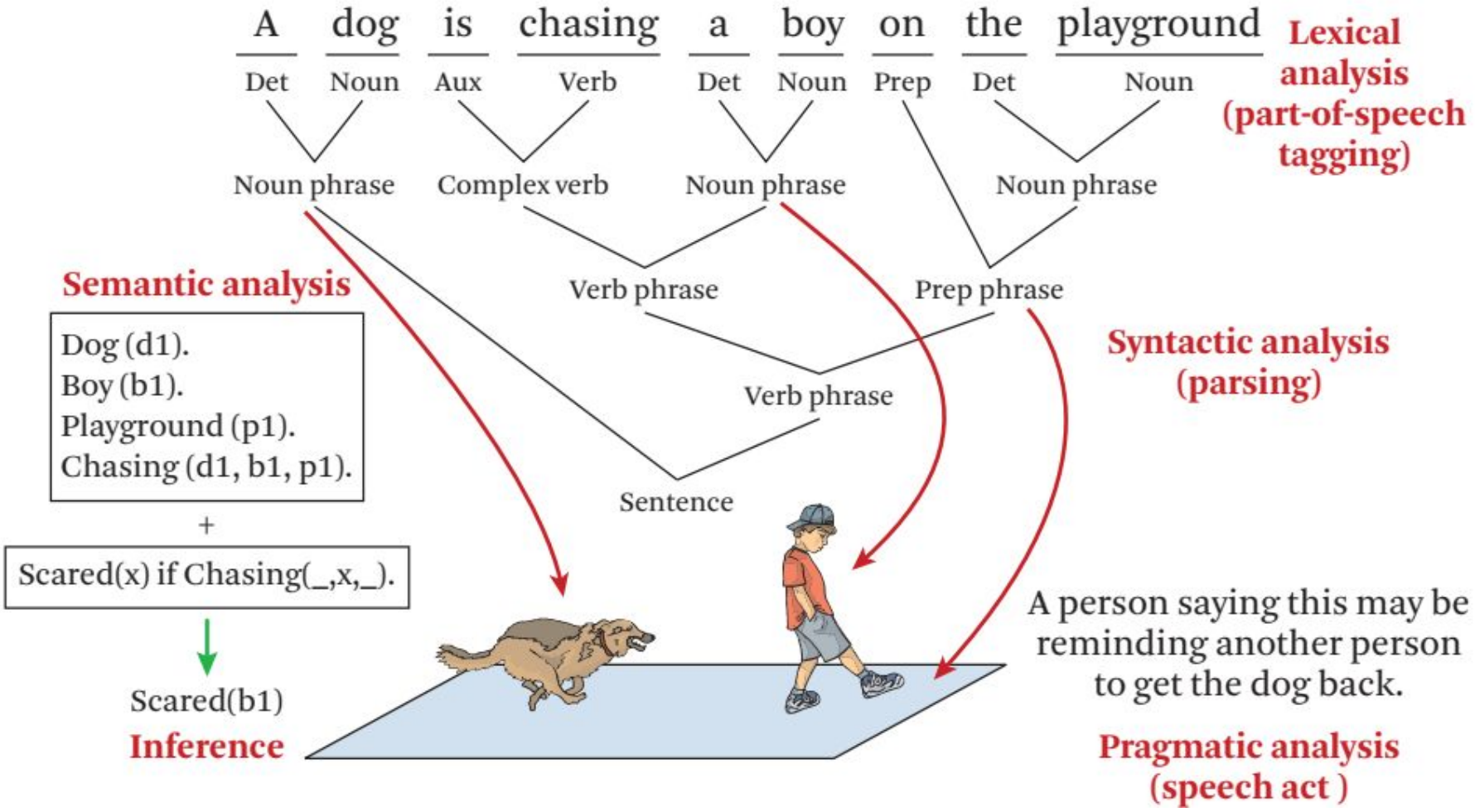
<a href>Frank Rizzo Bought <a hef>this home from <a href>Lake View Real Estate In 1992. <p>...

Bag-of-Tokens Approaches



Loses all order-specific information!
Severely limits context!

Natural Language Processing



Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining ChengXiang Zhai, Sean Massung, ACM Books 2016

General NLP—Too Difficult!

Word-level ambiguity

- **"design" can be a noun or a verb** (Ambiguous POS)
- **"root" has multiple meanings** (Ambiguous sense)

Syntactic ambiguity

- **"natural language processing"** (Modification)
- **"A man saw a boy *with a telescope*."** (PP Attachment)

Anaphora resolution

- **"John persuaded Bill to buy a TV for *himself*."**

  (*himself* = John or Bill?)

Presupposition

- **"He has quit smoking." implies that he smoked before.**

Humans rely on context to interpret (when possible).
This context may extend beyond a given document!

innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Information Retrieval

Text Databases and IR

Text databases (document databases)

- Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
- Data stored is usually *semi-structured*
- Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data

Information retrieval

- A field developed in parallel with database systems
- Information is organized into (a large number of) documents
- Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

Typical IR systems

- Online library catalogs
- Online document management systems

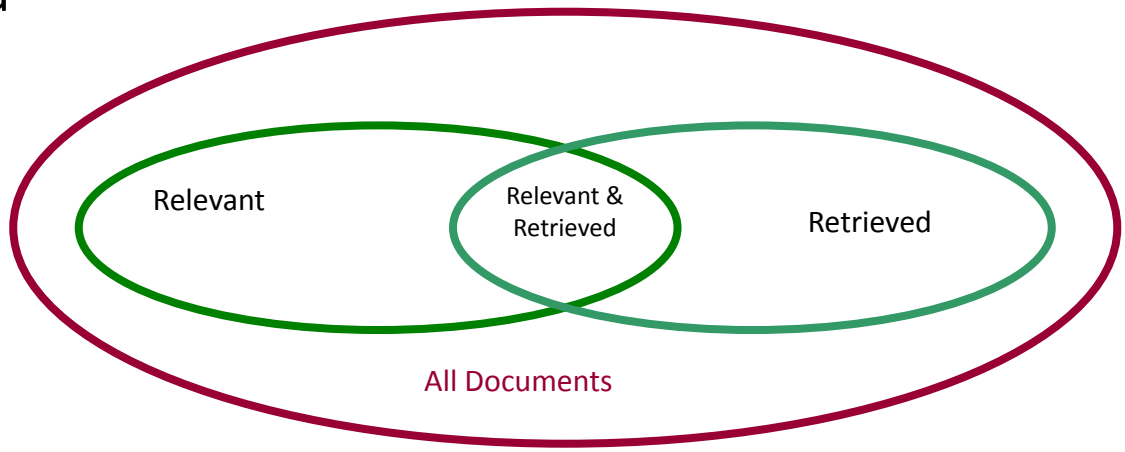
Information retrieval vs. database systems

- Some DB problems are not present in IR, e.g., update, transaction management, complex objects
- Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

Basic Measures for Text Retrieval

Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses)

Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved



$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|} \quad recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

Information Retrieval Techniques

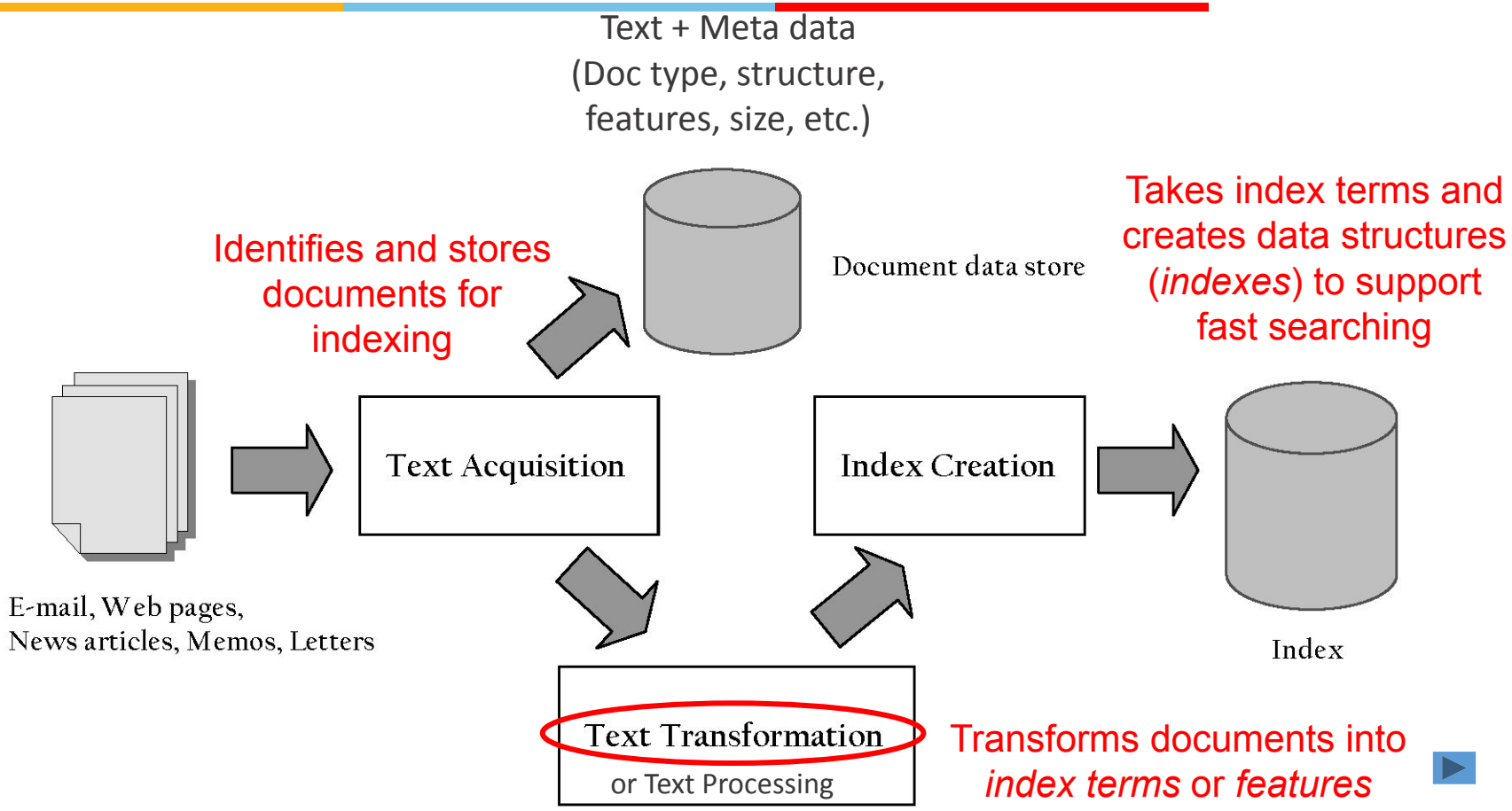
Basic Concepts

- A document can be described by a set of representative keywords called index terms.
- Different index terms have varying relevance when used to describe document contents.
- This effect is captured through the assignment of numerical weights to each index term of a document. (e.g.: frequency, tf-idf)

DBMS Analogy

- Index Terms Attributes
- Weights Attribute Values

Indexing Process



Zipf's Law

Distribution of word frequencies is very *skewed*

- Few words occur very often, many hardly ever occur
- e.g., "the" and "of", two common words, make up about 10% of all word occurrences in text documents

Zipf's law:

- The frequency *f* of a word in a corpus is inversely proportional to its rank *r* (assuming words are ranked in order of *decreasing* frequency)

$$f = \frac{k}{r} \Rightarrow f \times r = k$$

where *k* is a constant for the corpus

Top 50 Words from AP89

Word	*Freq.*	*r*	P_r(%)	$r.P_r$	*Word*	*Freq*	*r*	P_r(%)	$r.P_r$
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

Associated Press collection of news stories from 1989 (called AP89)

Vocabulary Growth

Heaps' Law, another prediction of *word occurrence*

As *corpus* grows, so does *vocabulary size.* However, fewer new words when corpus is already large

Observed relationship (Heaps' Law):

$$v = k \times n^{\beta}$$

where

v is the *vocabulary size* (number of *unique words*)

n is the *total number* of *words* in corpus

k, β are parameters that vary for each corpus

(typical values given are $10 \leq k \leq 100$ and $\beta \approx 0.5$)

- Predicting that the number of new words increases very rapidly when the corpus is small

Heaps' Law Predictions

Number of new words *increases* very rapidly when the corpus is small, and continue to increase indefinitely

Predictions for TREC collections are accurate for large numbers of words, e.g.,

- First 10,879,522 *words* of the AP89 collection scanned
- Prediction is 100,151 *unique words*
- Actual number is 100,024

Predictions for *small* numbers of words (i.e., < 1000) are much worse

Heaps' Law on the Web

Heaps' Law works with very *large* corpora

- New words occurring even after seeing 30 million!
- Parameter values different than typical TREC values

New words come from a variety of sources

- *Spelling errors*, *invented words* (e.g., product, company names), *code*, *other languages*, *email addresses*, etc.

Search engines must deal with these *large* and *growing vocabularies*

Text REtrieval Conference - TREC

innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Information Retrieval Techniques

Information Retrieval Techniques

Index Terms (Attribute) Selection:

- Stop list
- Word stem
- Index terms weighting methods

Terms □ Documents Frequency Matrices

Information Retrieval Models:

- Boolean Model (simplistic)
- Vector Space Model (Document Ranking considered)
- Probabilistic Retrieval Models (Ranked as per probability of relevance)

Boolean Model

Consider that index terms are either present or absent in a document

As a result, the index term weights are assumed to be all binaries

A query is composed of index terms linked by three connectives: not, and, and or

- e.g.: car *and* repair, plane *or* airplane

The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

Keyword-Based Retrieval

A document is represented by a string, which can be identified by a set of keywords

Queries may use expressions of keywords

- E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle
- Queries and retrieval should consider synonyms, e.g., repair and maintenance

Major difficulties of the model

- Synonymy: A keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
- Polysemy: The same keyword may mean different things in different contexts, e.g., mining

Similarity-Based Retrieval in Text Data

Finds similar documents based on a set of common keywords

- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.

Basic techniques

-Stop list

- Set of words that are deemed "irrelevant", even though they may appear frequently
- E.g., *a, the, of, for, to, with*, etc.
- Stop lists may vary when document set varies

Similarity-Based Retrieval in Text Data (contd)

- Word stem
  - Several words are small syntactic variants of each other since they share a common word stem
  - E.g., *drug, drugs, drugged*
- A term frequency table
  - Each entry *frequent_table(i, j)* = # of occurrences of the word t_i in document d_i
  - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
  - Relative term occurrences
  - Cosine distance:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{| v_1 || v_2 |}$$

Vector Space Model

Vector Space Model

Represent a doc by a term vector

- Term: basic concept, e.g., word or phrase
- Each term defines one dimension
- N terms define a N-dimensional space
- Element of vector corresponds to term weight
- E.g., d = (x_1,…,x_N), x_i is “importance” of term i

New document is assigned to the most likely category based on vector similarity.

What VS Model Does Not Specify

How to select terms to capture "basic concepts"

- Word stopping
  - e.g. "a", "the", "always", "along"
- Word stemming
  - e.g. "computer", "computing", "computerize" => "compute"
- Latent semantic indexing

How to assign weights

- Not all words are equally important: Some are more indicative than others
  - e.g. "algebra" vs. "science"

How to measure the similarity

How to Assign Weights

Two-fold heuristics based on frequency

- TF (Term frequency)
  - More frequent ***within*** a document more relevant to semantics
- IDF (Inverse document frequency)
  - Less frequent ***among*** documents more discriminative

Cornell SMART TF-IDF Model

TF is computed using the equation:

$$TF(d,t) = \begin{cases} 0 & if\, freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases}$$

IDF is computed using the equation:

$$IDF(t) = \log \frac{1 + |d|}{|d_t|}$$

where d is the document collection, and d_t is the set of documents containing term t. If $|d_t| << |d|$, the term t will have a large IDF scaling factor

Alternate TF-IDF Weighting

TF Weighting:

- More frequent => more relevant to topic
  - Raw TF= f(*t,d*): how many times term *t* appears in doc *d*

Normalization:

- Document length varies => relative frequency preferred
  - e.g., Maximum frequency normalization

$$TF(t,d) = 0.5 + \frac{0.5 * f(t,d)}{MaxFreq(d)}$$

IDF Idea:

- Less frequent ***among*** documents □ more discriminative

Formula:

$$IDF(t) = 1 + log(\frac{n}{k})$$

n — total number of docs
k — # docs with term t appearing

(IDF – inverse document frequency)

TF-IDF Weighting

TF-IDF weighting : **weight(t, d) = TF(t, d) * IDF(t)**

- Freqent within doc □ high tf □ high weight
- Selective among docs □ high idf □ high weight

Recall VS model

- Each selected term represents one dimension
- Each doc is represented by a feature vector
- Its *t*-term coordinate of document *d* is the TF-IDF weight
- This is more reasonable

Just for illustration …

- Many complex and more effective weighting variants exist in practice

How to Measure Similarity?

Given two document

$$D_i = (w_{i1}, w_{i2}, \cdots, w_{iN}) \quad D_j = (w_{j1}, w_{j2}, \cdots, w_{jN})$$

Similarity definition

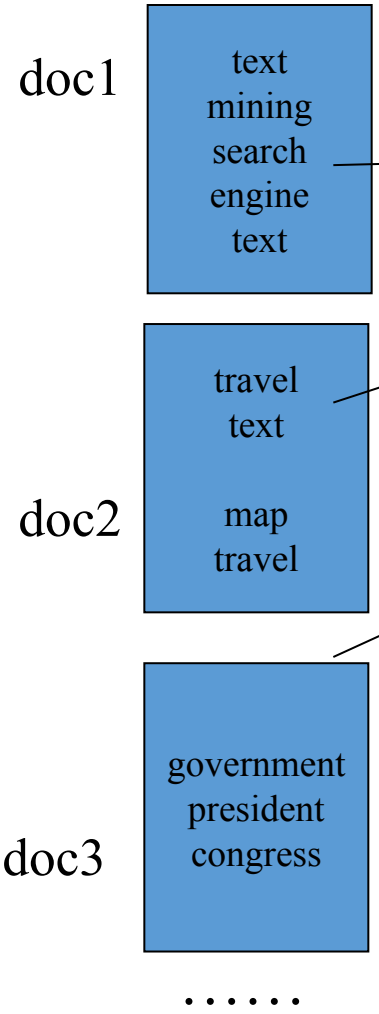
- dot product

$$Sim(D_i, D_j) = \sum_{t=i}^{N} w_{it} * w_{jt}$$

- normalized dot product (or cosine)

$$Sim(D_i, D_j) = \frac{\sum_{t=i}^{N} w_{it} * w_{jt}}{\sqrt{\sum_{t=1}^{N} (w_{it})^2 * \sum_{t=1}^{N} (w_{jt})^2}}$$

Illustrative Example



Sim(newdoc,doc1)=(4.8*2.4+4.5*4.5)/$||v_1||*||v_n||$

Sim(newdoc,doc2)=2.4*2.4/$||v_n||*||v_2||$

Sim(newdoc,doc3)=0

To whom is newdoc more similar?

	text	mining	travel	map	search	engine	govern	president	congress
IDF(faked)	2.4	4.5	2.8	3.3	2.1	5.4	2.2	3.2	4.3
doc1	2(4.8)	1(4.5)			1(2.1)	1(5.4)			
doc2	1(2.4)		2 (5.6)	1(3.3)					
doc3				1 (2.2)	1(3.2)	1(4.3)			
newdoc	1(2.4)	1(4.5)							

VS Model-Based Classifiers

What do we have so far?

- A feature space with similarity measure
- This is a classic supervised learning problem
  - Search for an approximation to classification hyper plane

VS model based classifiers

- K-NN
- Decision tree based
- Neural networks
- Support vector machine

innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Probabilistic Retrieval Models

Probabilistic Retrieval Models

The ranking function based on the probability that a given document d is relevant to a query q,

- or $p(R = 1 | d, q)$ where $R \in \{0, 1\}$ is a binary random variable denoting relevance.

In query likelihood retrieval model (one of the probabilistic models), we assume that this probability of relevance can be approximated by the probability of a query given a document and relevance,

- $p(q \mid d, R = 1)$.

Intuitively, if a user likes document d, how likely would the user enter query q in order to retrieve document d?

Probabilistic Retrieval Models

Clearly, $p(R = 1 \mid d, q) + p(R = 0 \mid d, q) = 1$

Query q	Document d	Relevant? R
q1	d1	1
q1	d2	1
q1	d3	0
d1	d4	0
q1	d5	1
⋮		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$$f(q,d) = p(R=1 \mid d,q) = \frac{\text{count}(q,d,R=1)}{\text{count}(q,d)}$$

$$P(R=1 \mid \text{q1}, \text{d1}) = 1/2$$

$$P(R=1 \mid \text{q1}, \text{d2}) = 2/2$$

$$P(R=1 \mid \text{q1}, \text{d3}) = 0/2$$

$$p(R=1 \mid d,q) = \frac{\text{count}(R=1,d,q)}{\text{count}(d,q)}.$$

Text Data Mining

Types of Text Data Mining

- Keyword-based association analysis
- Automatic document classification
- Similarity detection
  - Cluster documents by a common author
  - Cluster documents containing information from a common source
- Link analysis: unusual correlation between entities
- Sequence analysis: predicting a recurring event
- Anomaly detection: find information that violates usual patterns
- Hypertext analysis
  - Patterns in anchors/links
    - Anchor text correlations with linked objects

Keyword-Based Association Analysis

- Motivation
  - Collect sets of keywords or terms that occur frequently together and then find the association or correlation relationships among them
- Association Analysis Process
  - Preprocess the text data by parsing, stemming, removing stop words, etc.
  - Evoke association mining algorithms
    - Consider each document as a transaction
    - View a set of keywords in the document as a set of items in the transaction
  - Term level association mining
    - No need for human effort in tagging documents
    - The number of meaningless results and the execution time is greatly reduced

Text Classification

- Motivation
  - Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranets, etc.)
- Classification Process
  - Data preprocessing
  - Definition of training set and test sets
  - Creation of the classification model using the selected classification algorithm
  - Classification model validation
  - Classification of new/unknown text documents
- Text document classification differs from the classification of relational data
  - Document databases are not structured according to attribute-value pairs

Document Clustering

- Motivation
  - Automatically group related documents based on their contents
  - No predetermined training sets or taxonomies
  - Generate a taxonomy at runtime
- Clustering Process
  - Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
  - Hierarchical clustering: compute similarities applying clustering algorithms.
  - Model-Based clustering: clusters are represented by “exemplars”. (e.g.: Self-Organizing Maps)

VS Model-Based Classifiers

- What does VS model offer?
  - A feature space with similarity measure
  - This is a classic supervised learning problem
    - Search for an approximation to classification hyper plane
- VS model based classifiers
  - K-NN
  - Decision tree based (dimensionality/interpretability challenges)
  - Neural networks
  - Support vector machine

innovate achieve lead

BITS Pilani

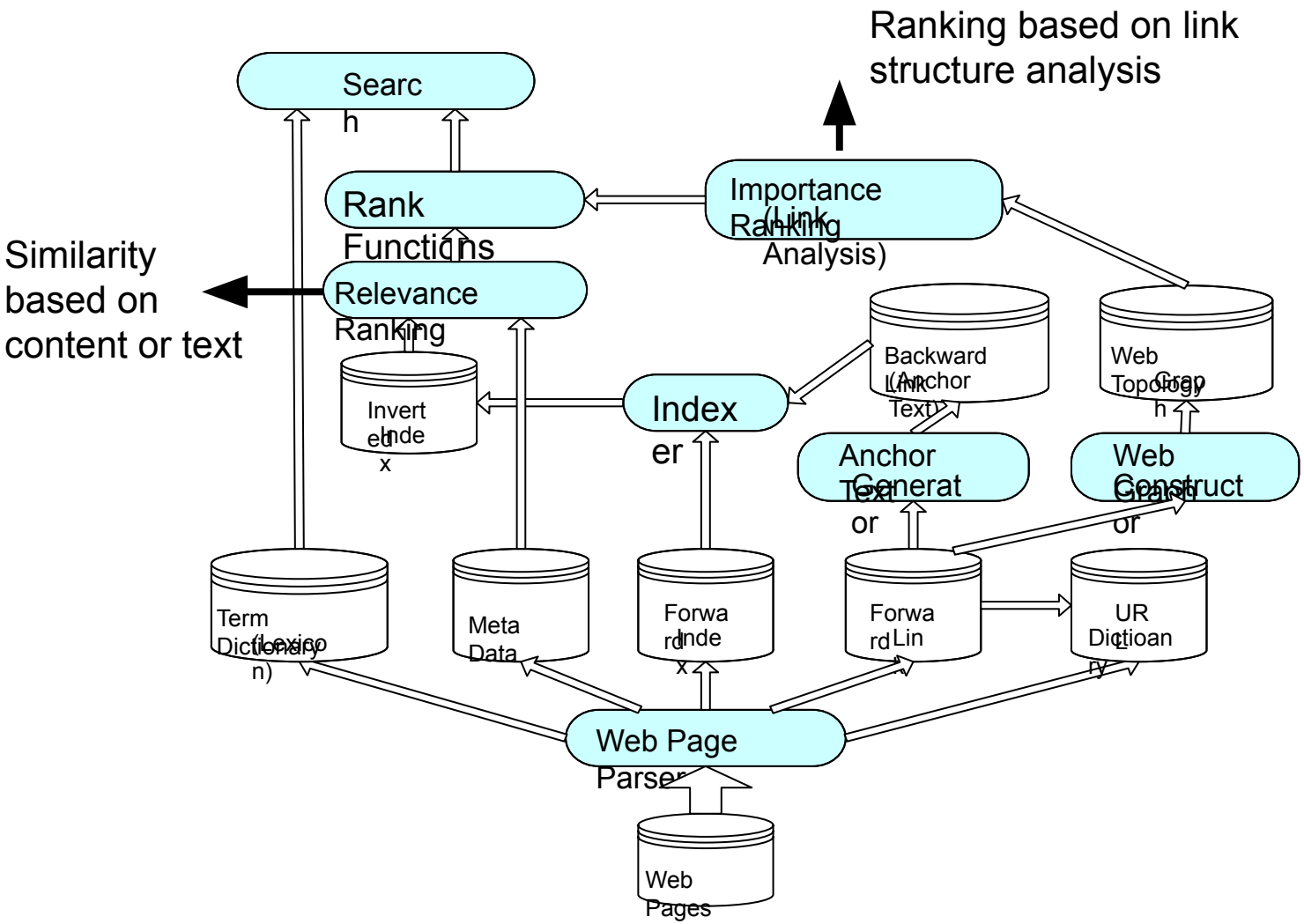
Pilani | Dubai | Goa | Hyderabad

Mining WWW

Challenges in Mining WWW

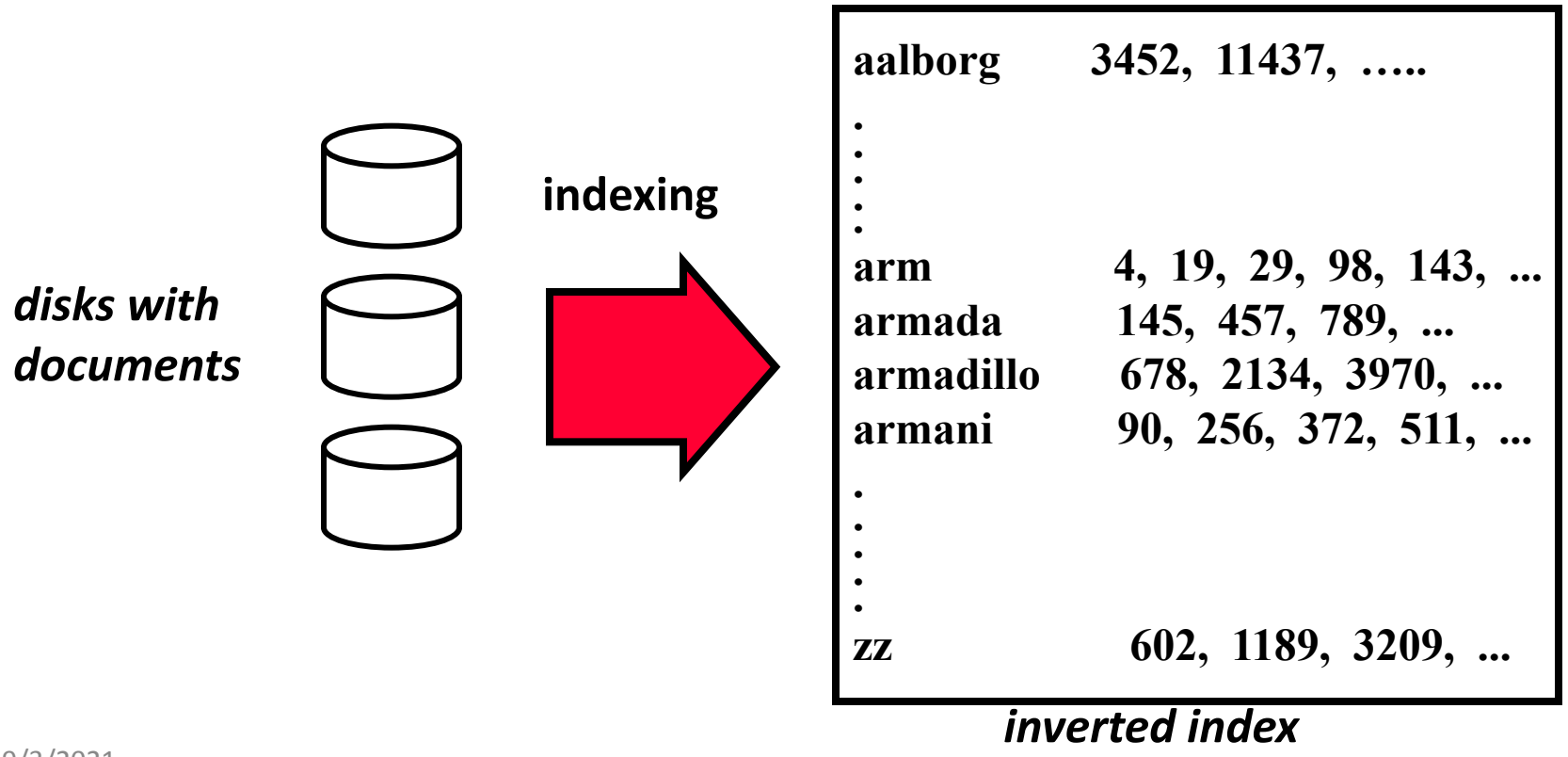
- The Web is too huge for effective data warehousing and data mining. It is barely possible to set up a data warehouse to replicate, store, or integrate all of the data on the Web
- The complexity of Web pages is far greater than that of any traditional text document collection. There is no index by category, nor by title, author, cover page, table of contents, etc.
- The Web is a highly dynamic information source. Not only does the Web grow rapidly, but is also constantly updated. Linkage information and access records are also updated frequently
- The Web serves a broad diversity of user communities. Users may have very different backgrounds, interests, and usage purposes.
- Only a small portion of the information on the Web is truly relevant or useful. It is said that 99% of the Web information is useless to 99% of Web users. How can the portion of the Web that is truly relevant to your interest be determined? How can we find high quality Web pages on a specified topic?

Search Engine – Two Rank Functions



Relevance Ranking

- Inverted index
  - A data structure for supporting text queries
  - like index in a book



inverted index

innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

PageRank

Introduction on PageRank

PageRank is a link analysis algorithm … with the purpose of "measuring" its (Webpage) relative importance within the set.

– **From Wikipedia, the free encyclopedia**

Developed by Larry Page as his PhD research topic

3 years later, he quit Stanford and founded Google with Brin

Apparently Larry Page had lost his PhD qualification.

PageRank: the intuitive idea

PageRank relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page's value or quality.

PageRank interprets a hyperlink from page *x* to page *y* as a vote, by page *x*, for page *y*.

However, PageRank looks at more than the sheer number of votes; it also analyzes the page that casts the vote.

- Votes casted by "important" pages weigh more heavily and help to make other pages more "important."

This is exactly the idea of **rank prestige** in social network.

More specifically …

A hyperlink from a page to another page is an implicit conveyance of authority to the target page.

- The more in-links that a page *i* receives, the more prestige the page *i* has.

Pages that point to page *i* also have their own prestige scores.

- A page of a higher prestige pointing to *i* is more important than a page of a lower prestige pointing to *i*.
- In other words, a page is important if it is pointed to by other important pages.

PageRank algorithm

According to rank prestige, the importance of page *i* (*i*'s PageRank score) is the sum of the PageRank scores of all pages that point to *i*.

Since a page may point to many other pages, its prestige score should be shared.

The Web as a directed graph $G = (V, E)$. Let the total number of pages be n. The PageRank score of the page i (denoted by $P(i)$) is defined by:

$$P(i) = \sum_{(j,i) \in E} \frac{P(j)}{O_j},$$

O_j is the number of out-link of j

Matrix notation

We have a system of *n* linear equations with *n* unknowns. We can use a matrix to represent them.

Let ***P*** be a *n*-dimensional column vector of PageRank values, i.e., $\boldsymbol{P} = (P(1), P(2), \ldots, P(n))^T$.

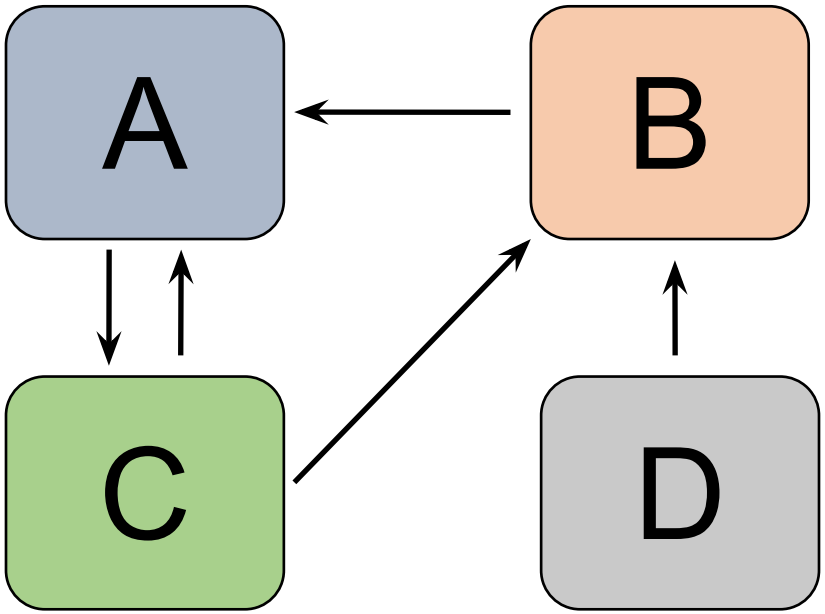
Let ***A*** be the adjacency matrix of our graph with

$$A_{ij} = \begin{cases} \frac{1}{O_i} & if\ (i,j) \in E \\ 0 & otherwise \end{cases}$$

We can write the *n* equations with (PageRank)

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P}$$

Calculation of PageRank on Webpage



- R(.) = PageRank of a Webpage

1. $R(A) = 100\%R(B) + 50\%R(C)$
2. $R(B) = 50\%R(C) + 100\%R(D)$
3. $R(C) = 100\%R(A)$

Signals for Google Search Engine

- Google apparently deals with the 15 percent of queries a day it gets which its systems have never seen before.
- For last 5 years, Google has been using RankBrain, which uses AI/ML to make a guess as to what words or phrases might have a similar meaning
- Google Executives state that three major search ranking factors are
    - Content (of query)
    - Links
    - RankBrain

https://searchengineland.com/now-know-googles-top-three-search-ranking-factors-245882

Clark, Jack. "Google Turning Its Lucrative Web Search Over to AI Machines". Bloomberg Business. 2015

Thank You

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
	Data Mining: Concepts and Techniques, Second Edition by Jiawei Han, Micheline Kamber Morgan Kaufmann Publishers
	Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining ChengXiang Zhai, Sean Massung, ACM Books 2016
	Web data Mining - Exploring Hyperlinks, Contents and Usage Data, By Bing Liu, Second Edition, Springer, July 2011
	Lucene in Action, 2nd Ed by Michael McCandless, Erik Hatcher, Otis Gospodnetic, Manning Publications
	Search Engines: Information Retrieval in Practice, Croft, Metzler, and Strohman, 2010