Data Mining & Machine Learning

Yong Zheng
Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

Schedule

- Text Mining
- Information Retrieval: Vector Space Model
- LDA
- Other Methods

Schedule

- Text Mining
- Information Retrieval: Vector Space Model
- LDA
- Other Methods

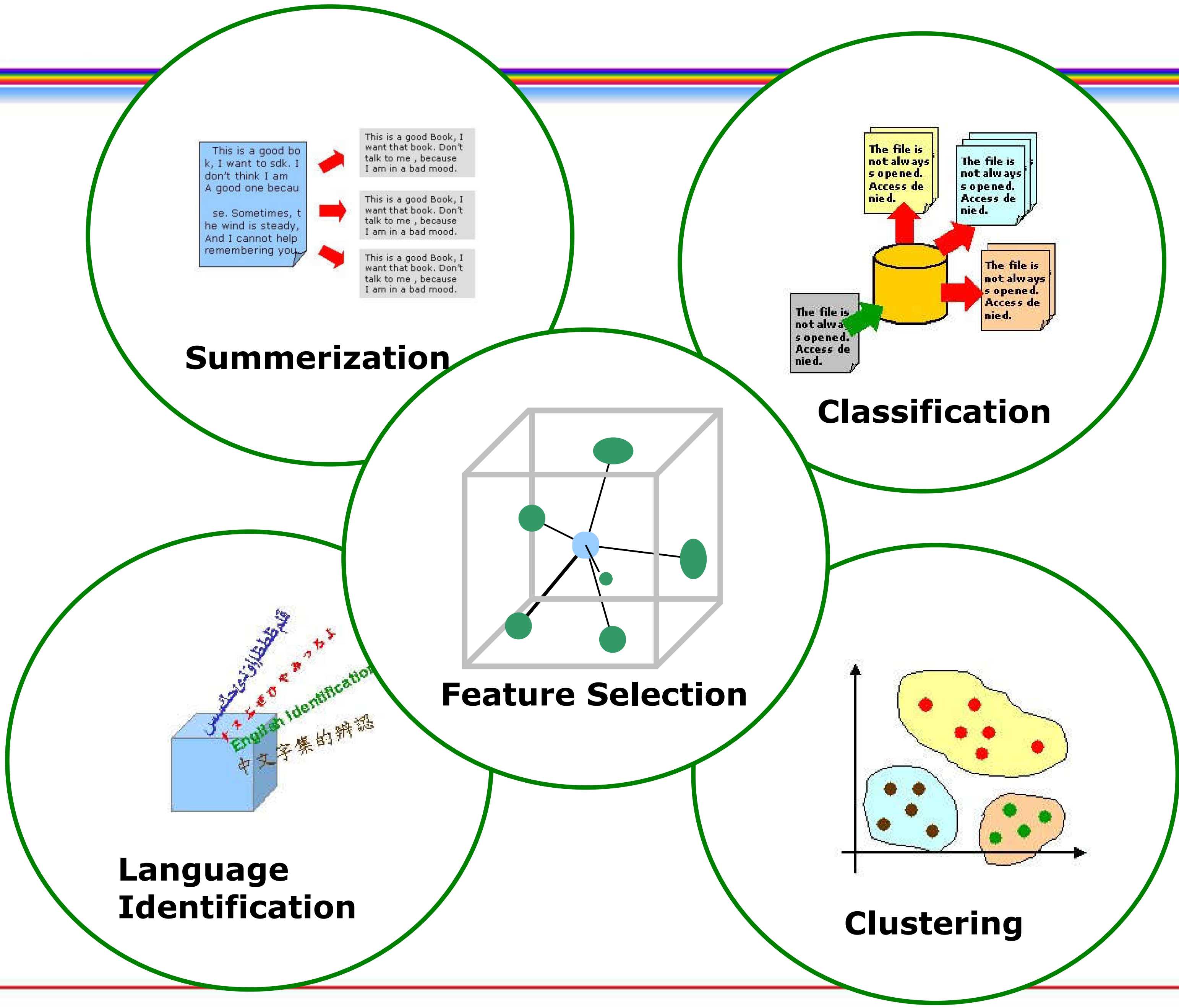
Text Processing

BookTitle	Author	Country	Publisher	Description
Learning Python, 5th Edition	Mark Lutz	US	O'Reilly	Get a comprehensive, in-depth introduction to the core Python language with this hands-on book. Based on author Mark Lutz's popular training course, this updated fifth edition will help you quickly write efficient, high-quality code with Python…..
Fluent Python: Clear, Concise, and Effective Programming 2nd Edition	Luciano Ramalho	BR	O'Reilly	Don't waste time bending Python to fit patterns you've learned in other languages. Python's simplicity lets you become productive quickly, but often this means you aren't using everything the language has to offer. With the updated edition of this hands-on guide, you'll learn how to write effective, modern Python 3 code by leveraging its best ideas.

Text Processing

- For variables with short terms, we can treat them as nominal variable, and convert them to binary variables, if necessary
- How about variables with long texts? Apparently, we cannot simply treat them as regular nominal variable and convert them to binary ones
- This is related to text processing/representations => how to represent long texts as numerical vectors

Text Analysis



Text Mining

- Text mining is about looking for patterns in natural language text
  - Natural Language Processing (NLP)
- May be defined as the process of analyzing text to extract information from it for particular purposes.
  - Information Extraction
  - Information Retrieval

Text Mining and Knowledge Management

- a recent study indicated that 80% of a company's information is contained in text documents
  - emails, memos, customer correspondence, and reports
- The ability to distil this untapped source of information provides substantial competitive advantages for a company to succeed in the era of a knowledge-based economy.

Text Categorization

- Text categorization is the problem of automatically assigned predefined categories to free text documents
  - Document classification
  - Web page classification
  - News classification

Information Retrieval

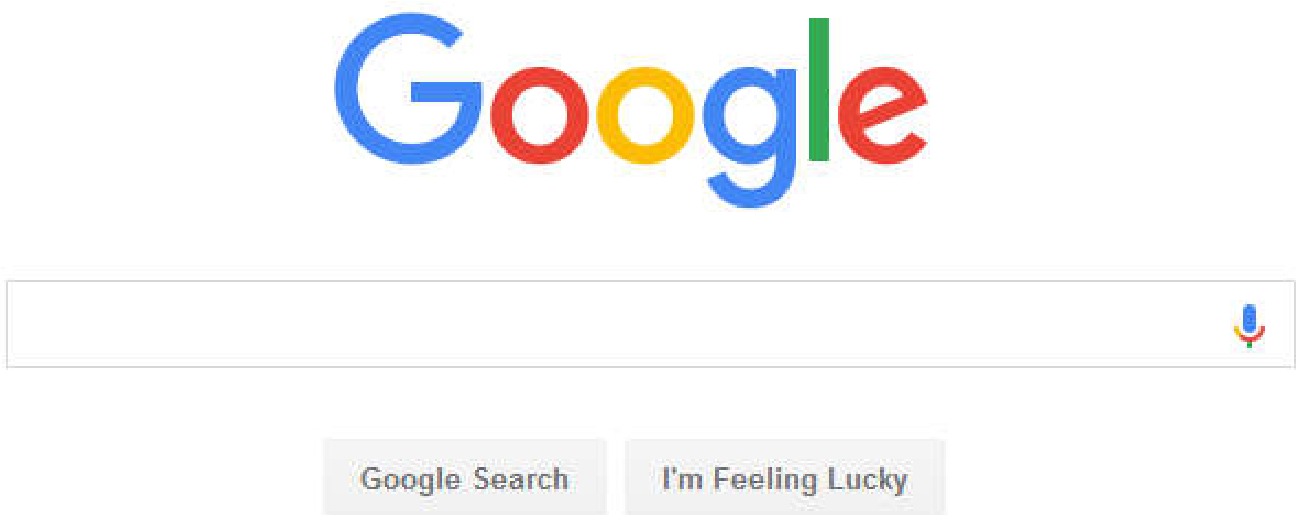
- Full text is hard to process, but is a complete representation to document
- Logical view of documents
- Models
  - Boolean Model
  - Vector Model, e.g., VSM
  - Probabilistic Model, e.g., LDA
- Modern Approaches: word2vec

Schedule

- Text Mining
- Information Retrieval: Vector Space Model
- LDA
- Other Methods

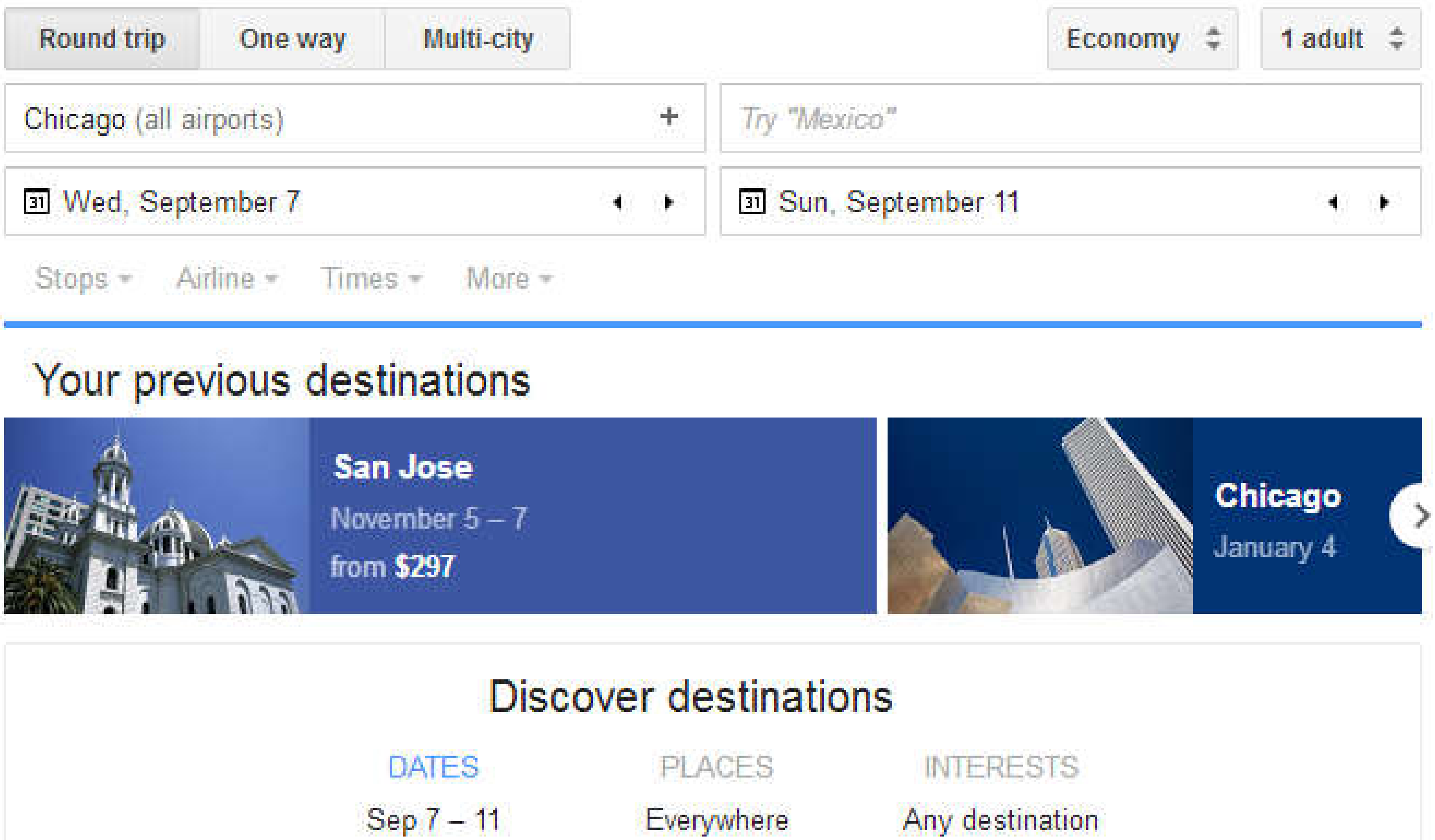
Information Retrieval

Information Retrieval, one example: Web Search



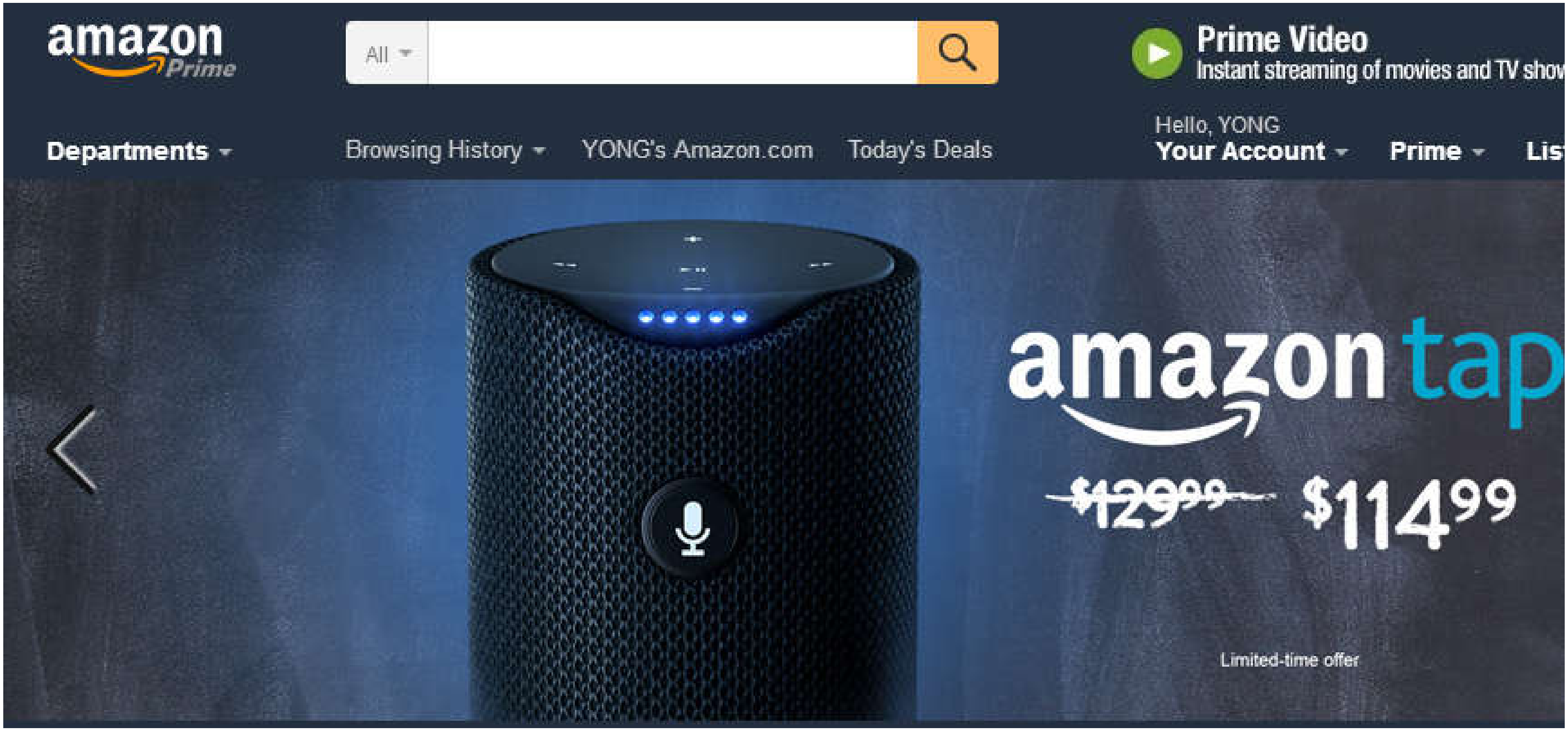
Information Retrieval

Information Retrieval, one example: Web Search



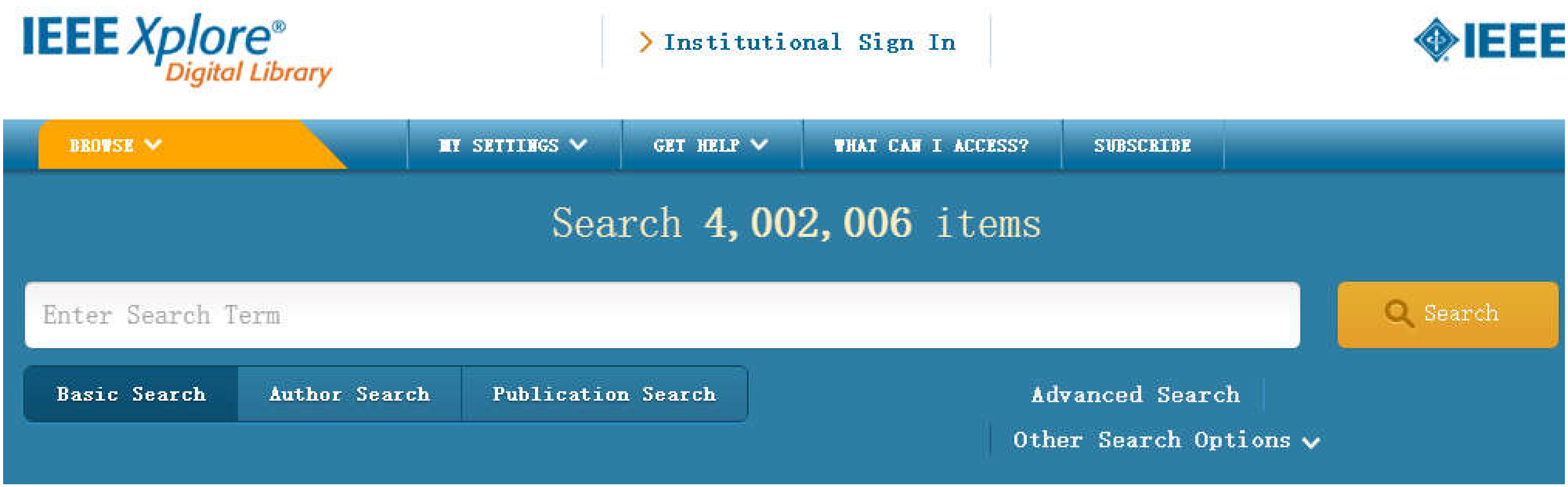
Information Retrieval

Information Retrieval, one example: Web Search



Information Retrieval

Information Retrieval, one example: Web Search

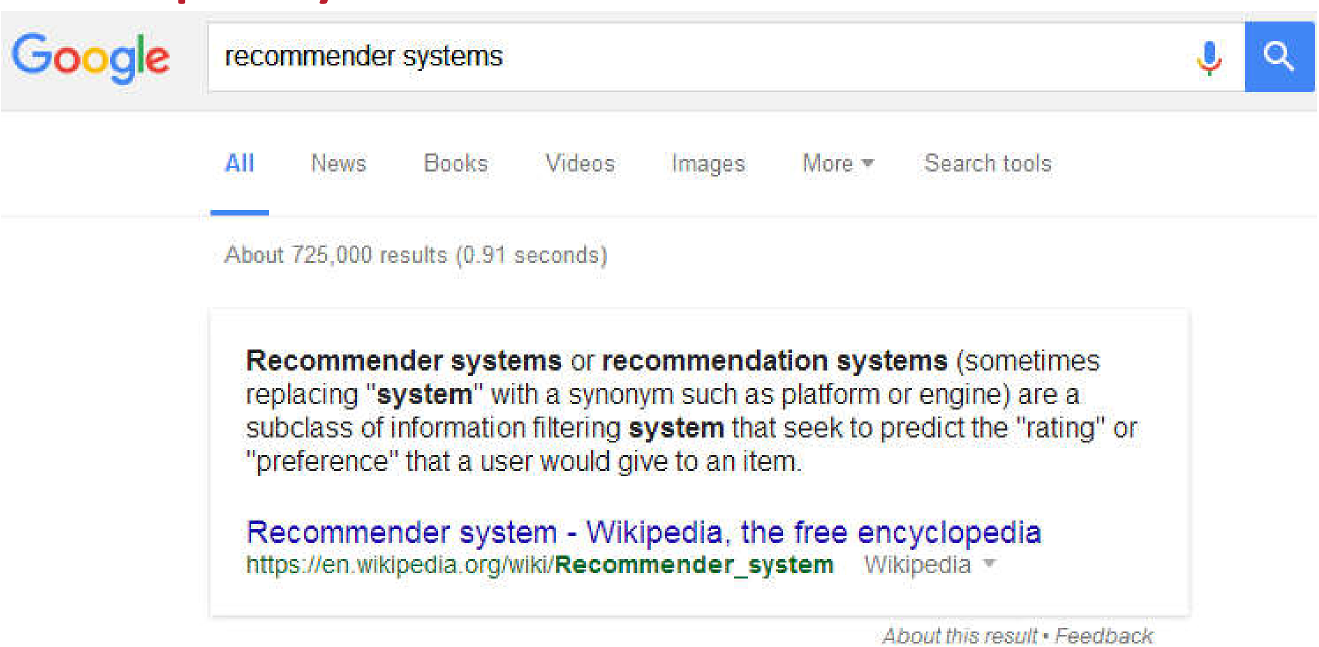


Information Retrieval

Task In Information Retrieval:

- Given a query
- Retrieve a list of documents related to the query

The query could be a term:

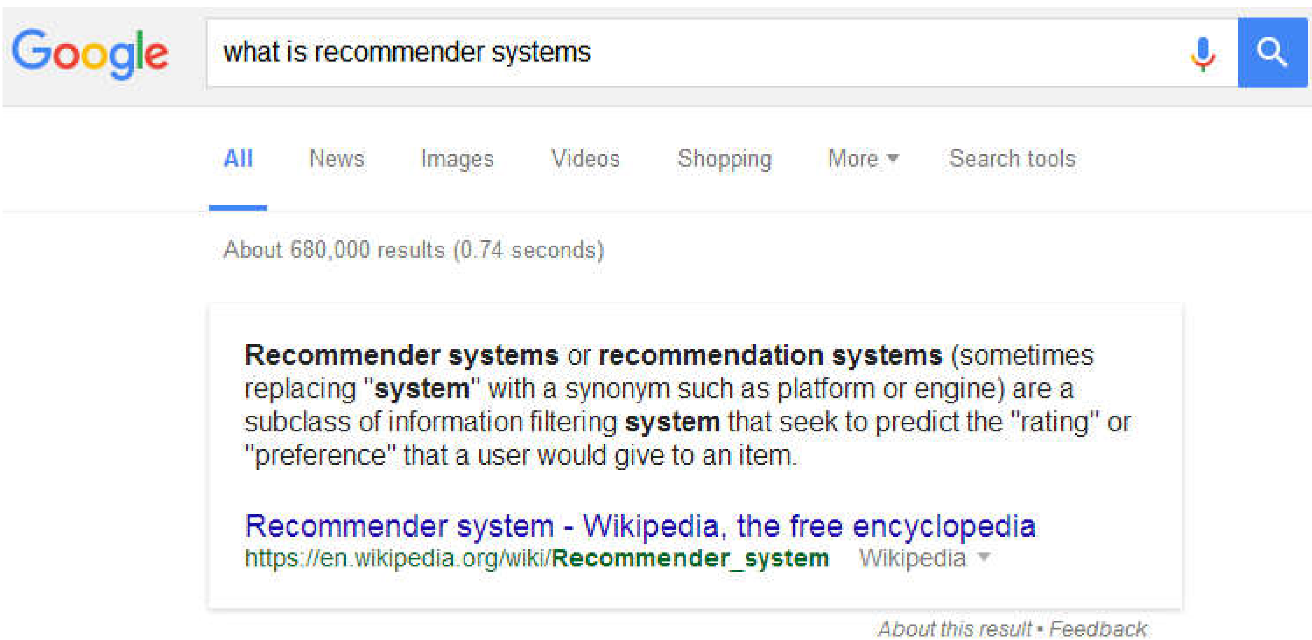


Information Retrieval

Task In Information Retrieval:

- Given a query
- Retrieve a list of documents related to the query

The query could be a sentence:



Information Retrieval

Task In Information Retrieval:

- Given a query
- Retrieve a list of documents related to the query

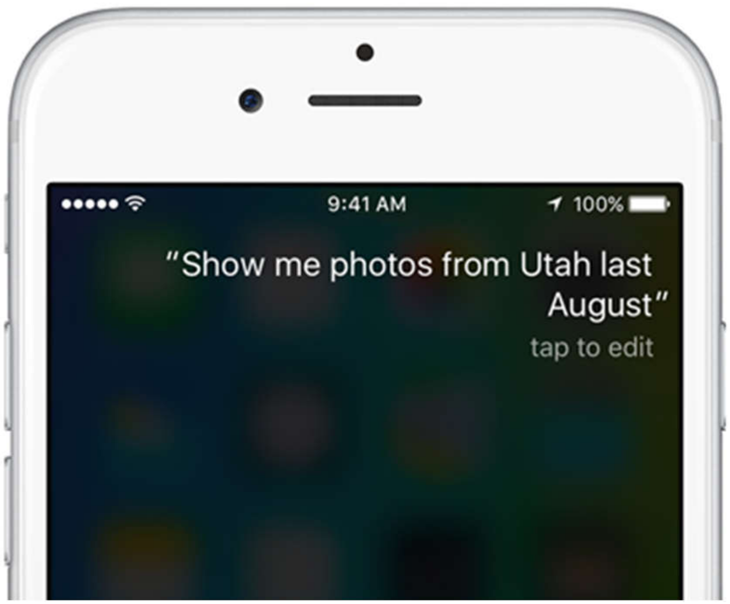
The query could be a picture:

Information Retrieval

Task In Information Retrieval:

- Given a query
- Retrieve a list of documents related to the query

The query could be an audio:



Information Retrieval

Task In Information Retrieval:

- Given a query
- Retrieve a list of documents related to the query

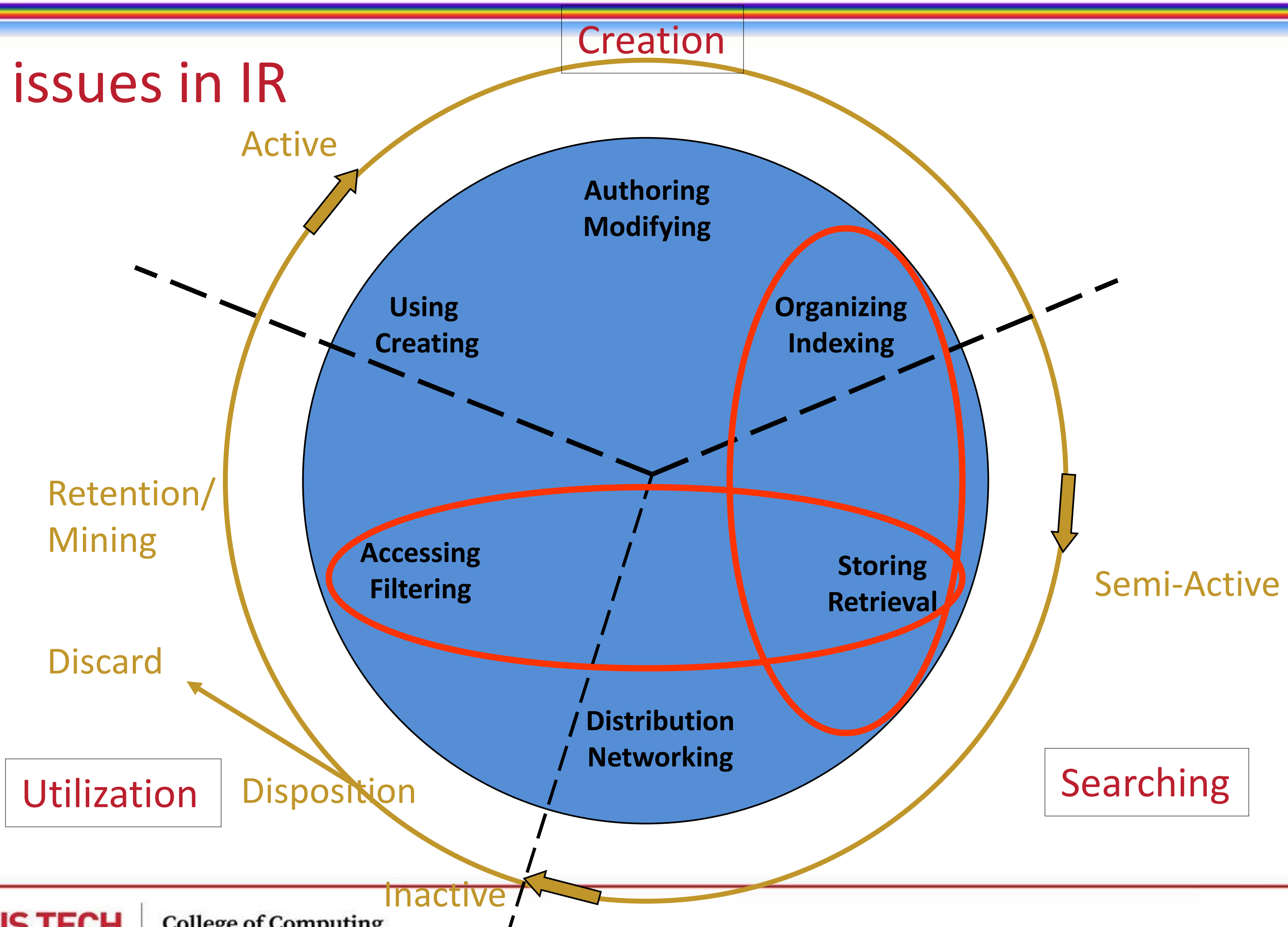
The query could be even anything!!!!

Thanks to the contributions by:

- Multimedia
- Natural language processing (NLP)

Information Retrieval

Key issues in IR



Information Retrieval

IR v.s. Database

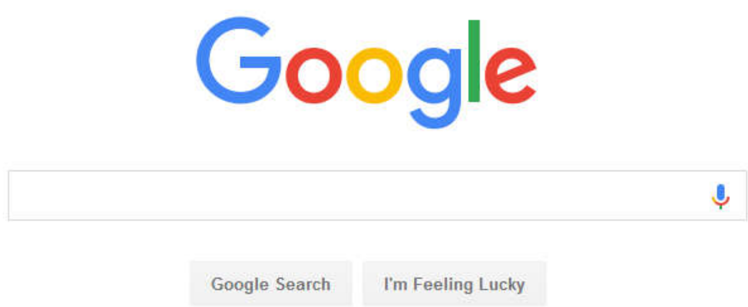
- Emphasis on effective, efficient retrieval of unstructured (or semi-structured) data
- IR systems typically have very simple schemas
- Query languages emphasize free text and Boolean combinations of keywords
- Matching is more complex than with structured data
  - easy to retrieve the wrong objects
  - need to measure the accuracy of retrieval
- Less focus on concurrency control and recovery (although update is very important).

Information Retrieval

The goal is to retrieve RELEVANT documents.

Popular techniques:

- Vector Space Model (VSM)
- Google PageRank



Vector Space Models (VSM)

- Information Retrieval by Vector Space Models
- The basic ideas in VSM:
  - Each Web page is viewed as a document
  - Each document is represented as a term vector
  - Each query can be represented as a term vector too
  - The RELEVANT documents related to each query therefore can be identified by the similarity between the query vector and other document vectors

Vector Space Models (VSM)

- Information Retrieval by Vector Space Models
- Techniques we need in VSM:
  - Stop Word Removal
  - Stemming
  - Term Weighting*
  - Vector Similarity Measures*
  - Evaluations*

Vector Space Models (VSM)

- Document Representation by Term Vectors

Document Ids

Term Weights (in this case normalized)

a document vector

	nova	galaxy	heat	hollywood	film	role	diet	fur
A	1.0	0.5	0.3					
B	0.5	1.0						
C		1.0	0.8	0.7				
D		0.9	1.0	0.5				
E				1.0	1.0			
F					0.9		1.0	
G	0.5		0.7				0.9	
H		0.6		1.0		0.3	0.2	0.8
I			0.7	0.5	0.1			0.3

VSM: PreProcessing

- To extract the terms, there are two important steps
  - Stop words removal
  - Word stemming

VSM: Stop Word Removal

- Stop words usually refer to the most common words in a language which do not take many meaningful information, such as a, an, the, is, this, that, as, at, do, does, etc, e.g., i.e., and so forth
- The simplest way to remove stop words is to filter out these words based on a predefined list of stop words
- For example: http://www.ranks.nl/stopwords

VSM: Word Stemming

- Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form
- For example, they may mean the same thing:

```
connection
connections
connective              --->    connect
connected
connecting
```

- More info: https://xapian.org/docs/stemming.html

VSM: Word Stemming

Stemming Algorithms:

- **Porter's Stemmer** (use a collection of rules)
  https://tartarus.org/martin/PorterStemmer/
  Online tool: http://9ol.es/porter_js_demo.html
- **N-grams** (based on the structural similarity)
  http://text-analytics101.rxnlp.com/2014/11/what-are-n-grams.html
  Online tool: http://guidetodatamining.com/ngramAnalyzer/

VSM: Term Weighting

Term weights must be incorporated into VSM

- Binary weights
  - Terms either appear or they don't; no frequency information used.
- Simple term frequency
  - Means either raw term counts or (more often) term counts normalized by the length of the document
- TF.IDF (inverse document frequency model)
- Term discrimination model
- Signal-to-noise ratio (based on information theory)

VSM: Term Weighting

- **Binary Weight:** If term appears in a document, mark it as 1; Otherwise, as 0

This representation can be particularly useful, since the documents (and the query) can be viewed as simple bit strings. This allows for query operations be performed using logical bit operations.

docs	*t1*	*t2*	*t3*
D1	1	0	1
D2	1	0	0
D3	0	1	1
D4	1	0	0
D5	1	1	1
D6	1	1	0
D7	0	1	0
D8	0	1	0
D9	0	0	1
D10	0	1	1
D11	1	0	1

VSM: Term Weighting

➢ Simply term frequency

The term weight is the raw term frequency (i.e., how many times a term appears in one document)

Terms	D1	D2	D3	D4	D5	D6	D7	…
t1	10	1	0	6	1	9	0	
t2	0	0	4	0	3	2	1	
t3	5	0	3	0	2	0	0	

VSM: Term Weighting

- TF.IDF (Term Frequency × Inversed Document Frequency)
  - Weight terms higher if they are frequent in relevant documents but infrequent in the collections as a whole (function by TF)
  - Weight more for rare words, less for common words (function by IDF)
  - Provide normalization function

VSM: Term Weighting

➢ TF.IDF weight and normalization

$$w_{ik} = tf_{ik} \cdot \log_2(N / n_k)$$

T_k = term k in document D_i

tf_{ik} = frequency of term T_k in document D_i

idf_k = inverse document frequency of term T_k in C

N = total number of documents in the collection C

n_k = the number of documents in C that contain T_k

$$idf_k = \log\left(\frac{N}{n_k}\right)$$

normalize usually means force all values to fall within a certain range, usually between 0 and 1, inclusive.

$$w_{ik} = \frac{tf_{ik} \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / n_k)]^2}}$$

VSM: Term Weighting

- TF.IDF weight and normalization

normalize usually means force all values to fall within a certain range, usually between 0 and 1, inclusive.

$$w_{ik} = \frac{tf_{ik} \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / n_k)]^2}}$$

t = the number of terms in document Di

The divisor comes from the vector norms:

Formally the l_p-norm of x is defined as:

$\|x\|_p = \sqrt[p]{\sum_i |x_i|^p}$ where $p \epsilon \mathbb{R}$

We use l2 norm and actually it is widely used in different occasions.

VSM: Term Weighting

- **TF:** measures how frequently a term occurs in a document.
- **IDF:** measures how important a term is.

For example, we have a collection of 10 million documents, one of these documents, D, contains 100 words and the word *cat* appears 3 times. The word *cat* appears in 1000 documents from the collection.

TF(cat in D) = 3

IDF(cat) = log (10,000,000/1,000) = 4

Weight = 12; You'd better use normalized weights

VSM: Vector Similarity Measures

- After the basic steps below, we are able to construct the vector space – each document is represented as term vectors, and the values in the vectors are the normalized TF.IDF weights
  - Stop word removal
  - Word stemming
  - TF.IDF weighting

VSM: Vector Similarity Measures

- Both documents and queries can be represented as the vector with term weights
- The similarity(query, doc) can be used to retrieve a list of relevant documents

VSM: Vector Similarity Measures

Simple Matching:

$$sim(Q,D) = \sum_{j=1}^{t}(w_{q_j} \cdot w_{d_j})$$

Cosine Coefficient:

$$sim(Q,D) = \frac{\sum_{j=1}^{t}(w_{q_j} \cdot w_{d_j})}{\sqrt{\sum_{j=1}^{t}(w_{q_j})^2 \cdot \sum_{j=1}^{t}(w_{d_j})^2}}$$

Dice's Coefficient:

$$sim(Q,D) = \frac{2 \cdot \sum_{j=1}^{t}(w_{q_j} \cdot w_{d_j})}{\sum_{j=1}^{t}(w_{q_j})^2 + \sum_{j=1}^{t}(w_{d_j})^2}$$

Jaccard's Coefficient:

$$sim(Q,D) = \frac{\sum_{j=1}^{t}(w_{q_j} \cdot w_{d_j})}{\sum_{j=1}^{t}(w_{q_j})^2 + \sum_{j=1}^{t}(w_{d_j})^2 - \sum_{j=1}^{t}(w_{q_j} \cdot w_{d_j})}$$

VSM: Vector Similarity Measures

- Also, we can use different distance measures
- Similarity = 1 – normalized distance

  Step1: we use a distance measure (Euclidean or Manhattan distance) to calculate the distance between query and all document candidates;

  Step2: we normalize the distance results and convert the scale to [0,1]

  Step3: similarity = 1 - normalized distance

IR Evaluations

- Relevance metrics: precision, recall
- Ranking metrics: MRR, NDCG, MAP (Optional)

IR Evaluations: Precision and Recall

- In top-N information retrieval

	Relevant	Not relevant
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

Precision@N = tp / (tp + fp)

Recall@N = tp / (tp + fn)

Summary

- IR: Intro
- Vector Space Model
  - Stop word removal
  - Word Stemming
  - Term weighting by TF.IDF*
  - Vector Similarity Measures*
- IR Evaluations: Precision and Recall at N

Schedule

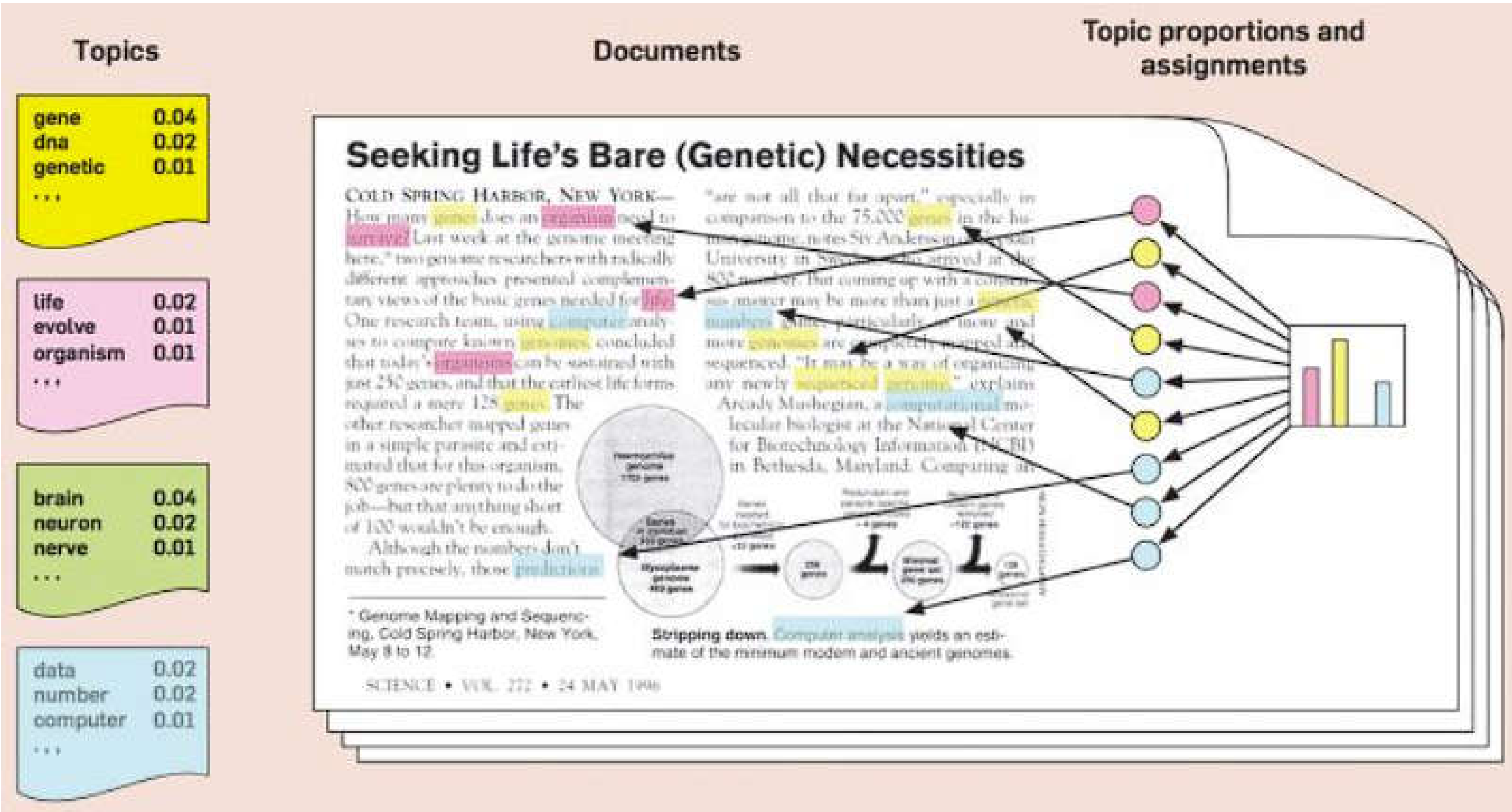
- Text Mining
- Information Retrieval: Vector Space Model
- LDA
- Other Methods

LDA

- Topic modeling is used to discover the topics that occur in a document's body or a text corpus. .
- Latent dirichlet allocation (LDA) is an approach used in topic modeling based on probabilistic vectors of words, which indicate their relevance to the text corpus.

LDA

- Basic Ideas in Topic Modeling

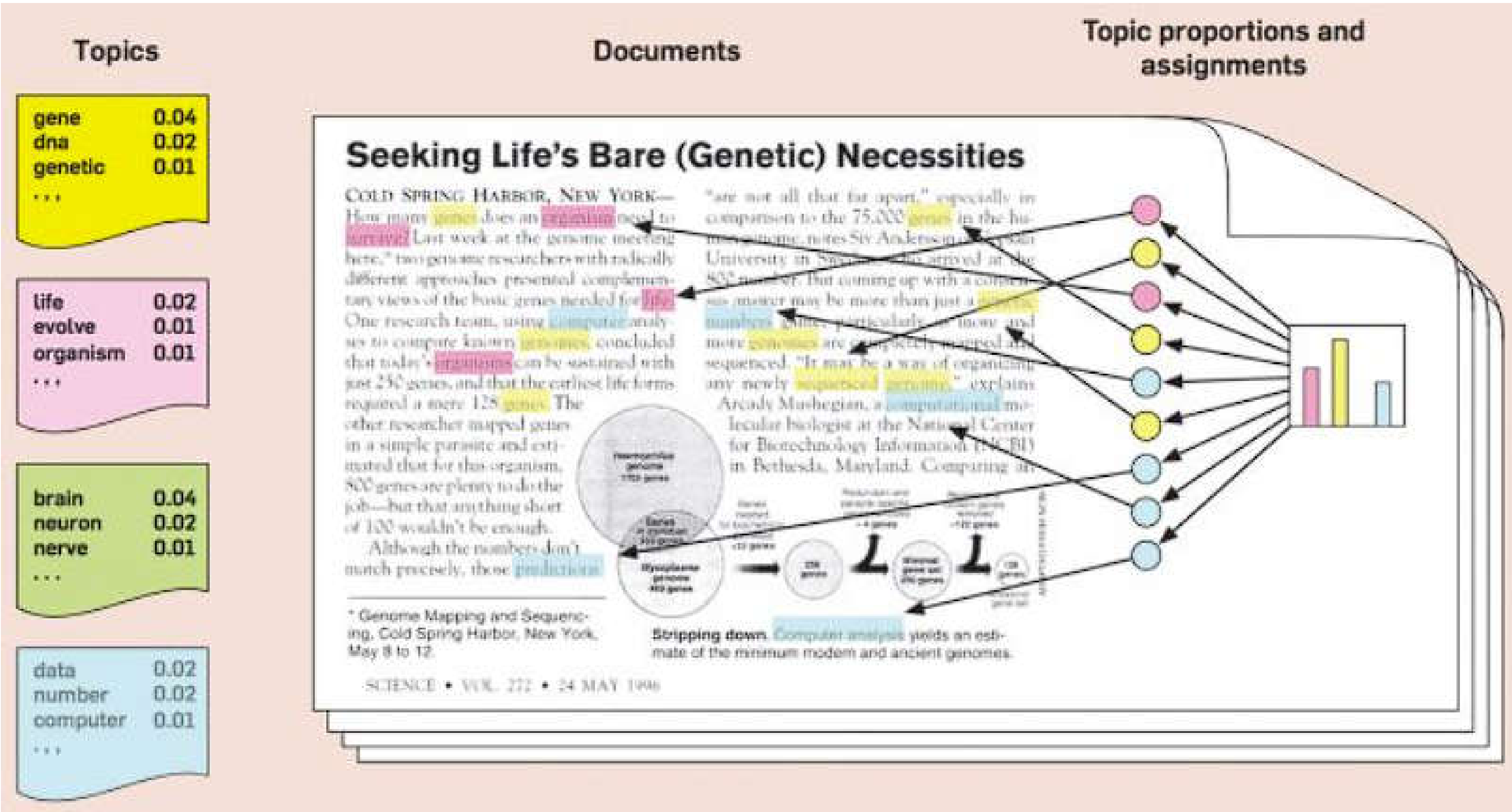


LDA

- Basic Ideas in Topic Modeling
  - A document is composed of a set of words
  - A word is associated with several topics
    For example, “Turkey” => [tourism, politics, econ, …]
  - As a result, a document can be considered as a topic distribution
    - A document is a mixture of topics
    - Each word is selected from a topic over a distribution

LDA

- Basic Ideas in Topic Modeling

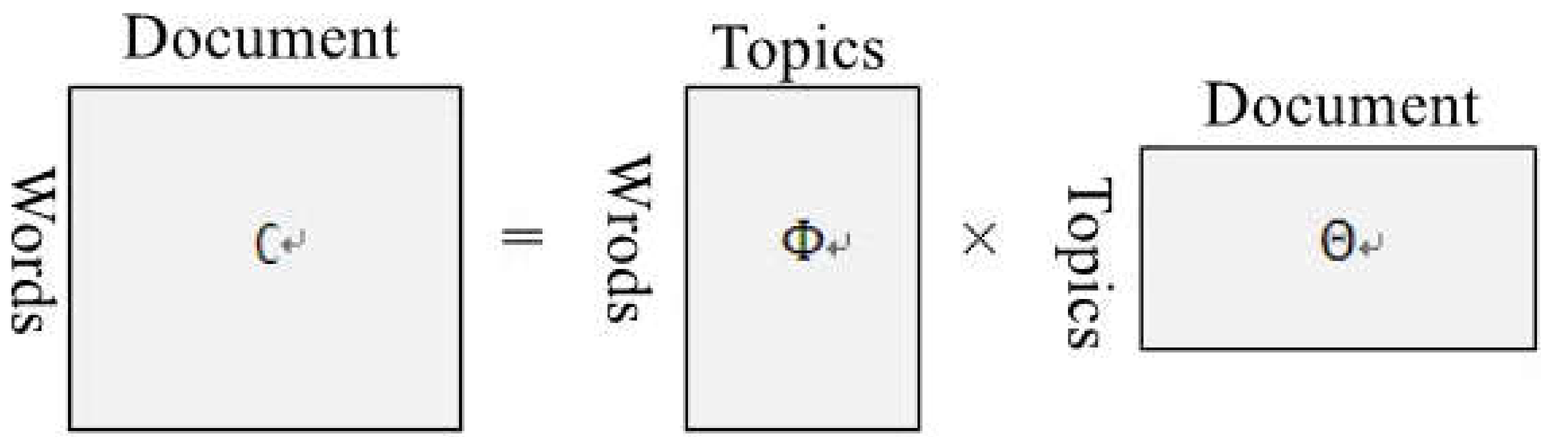


LDA

- Challenges
  - Assume we already know a list of topics
    - A document is a distribution over the topics
      Doc = <0.1, 0.2, 0.25, 0.3, ….>
    - Each word may have correlations to each topic
    - For each topic, we have the top/frequent word/terms
      Example: topic “politic” => USA, Trump, Biden, Russia, …
  - However, we never have the explicit info about topics
  - Therefore, LDA is a process of unsupervised learning

LDA: Unsupervised Learning

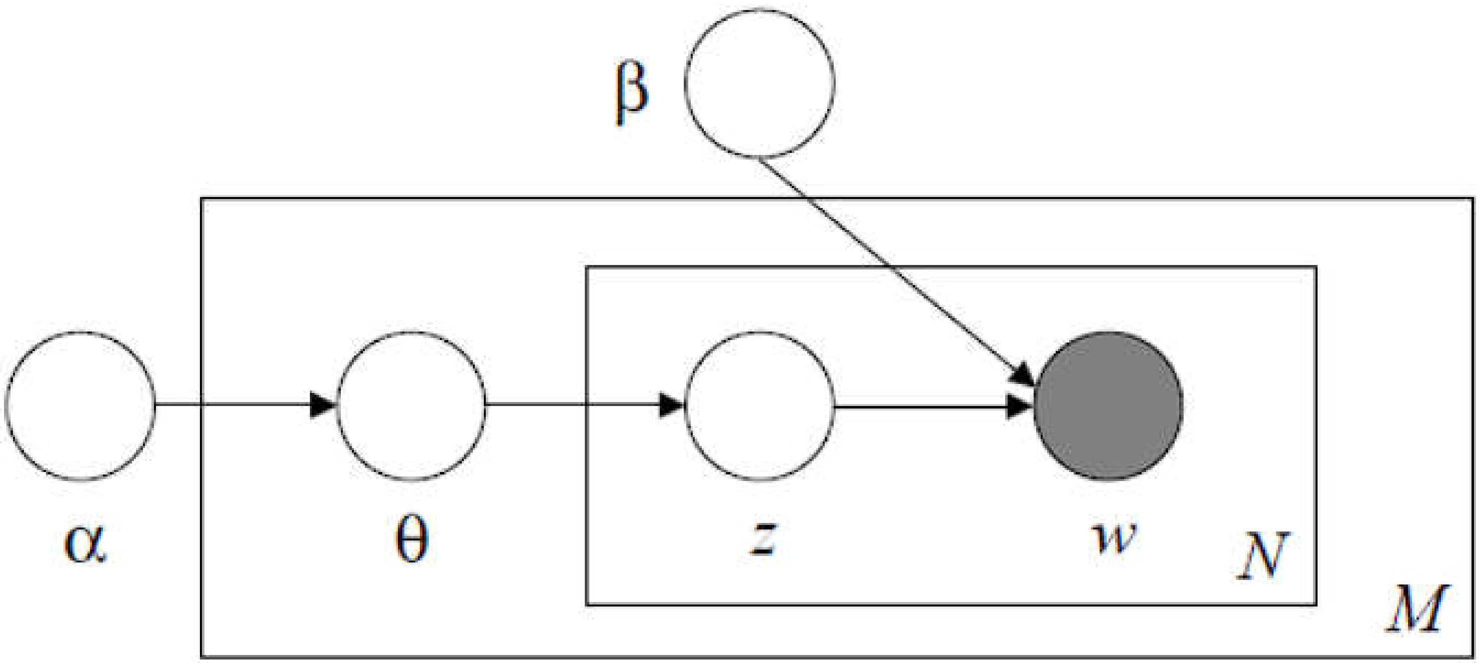
- The Model can be interpreted as a matrix factorization



- However, we do not have the list of topics at hand. We need to learn these “latent” topics

LDA: Unsupervised Learning

- The actual learning is a probabilistic process



$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta)$$

Schedule

- Text Mining
- Information Retrieval: Vector Space Model
- LDA
- Other Methods

Other Methods

- NLP has been advanced by the technique of neural networks and deep learning. There are more efficient and effective ways to help build better models
  - Word2vec
  - Sentence2vec
  - Doc2vec
  - Bert
- You can learn more NLP technique in ITMD 524

Schedule

- Next Class
  - Explanation about exams
  - Python for Text Mining/Similarity
    - VSM
    - LDA