



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Text Mining

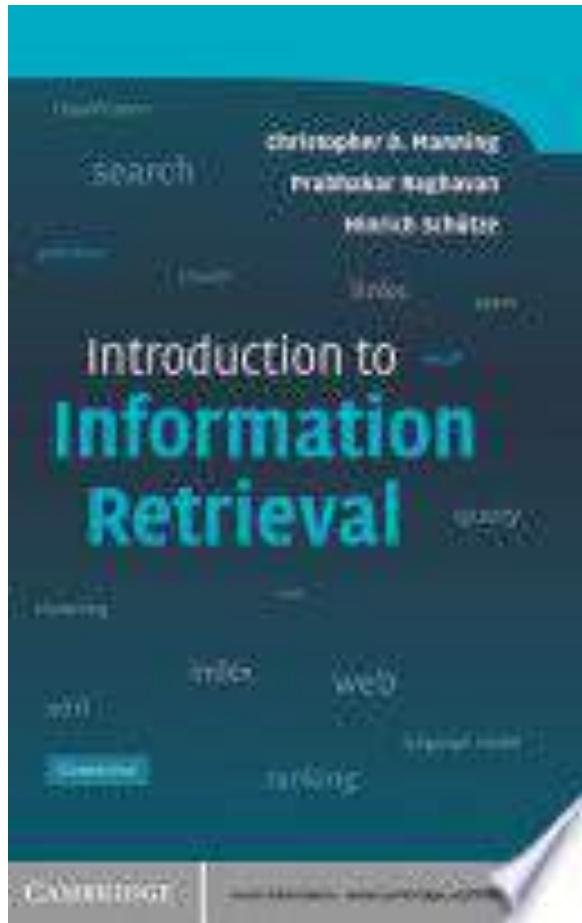
Prof. Aruna Malapati

Overview of the Feature Engineering Course

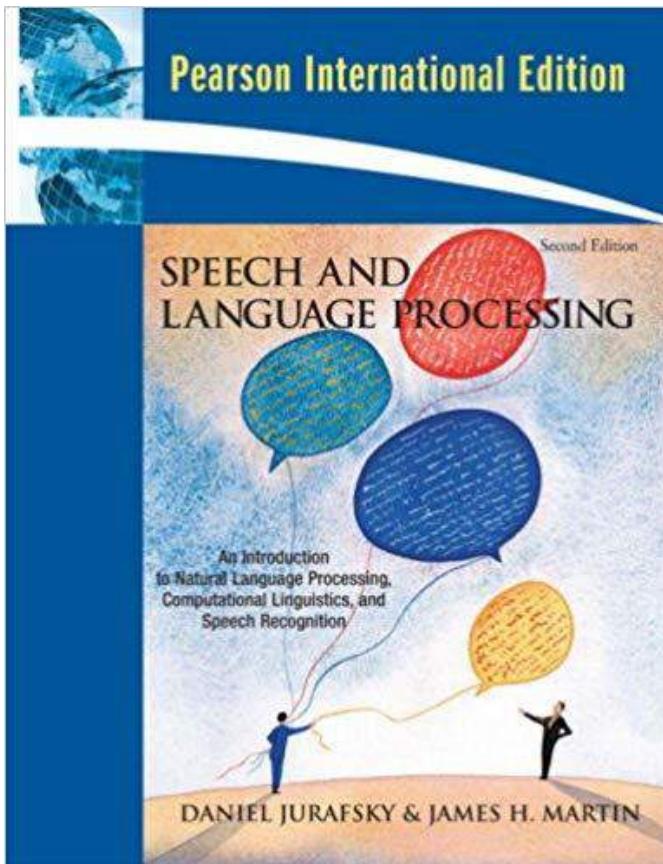
- Introduction to Text Mining
- Parts of speech Tagging
- Topic modelling using Latent Dirichlet Allocation
- Sentiment Analysis
- Recommender systems



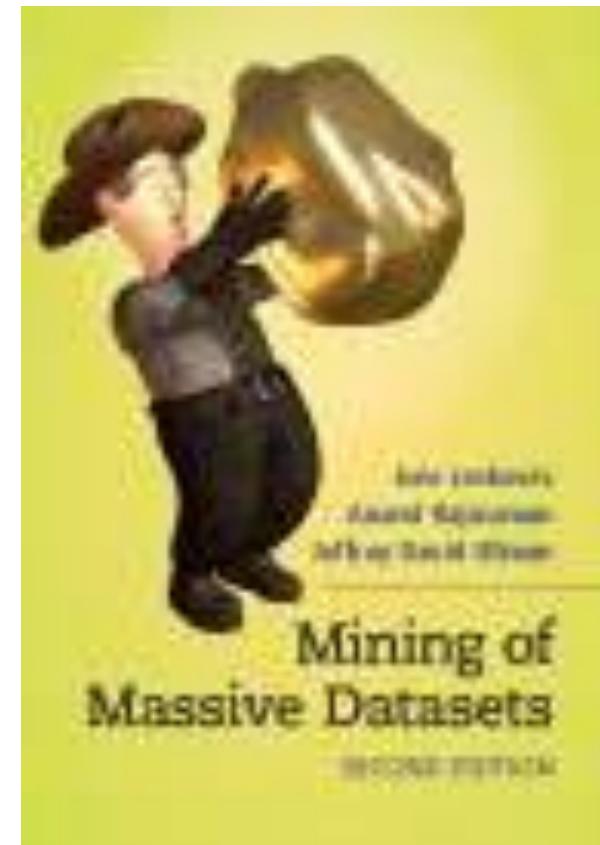
Books



Module-1



Module-2



Module-5

Evaluation

Evaluation Component	Marks	Type
Comprehensive Examination	40%	Closed
Quizzes (2)	24%	Open
2 Minor Projects (Evaluated twice)	24%	Open
Assignments/Exercises (2)	12%	Open



Thank You!

In our next session: Boolean Retrieval model



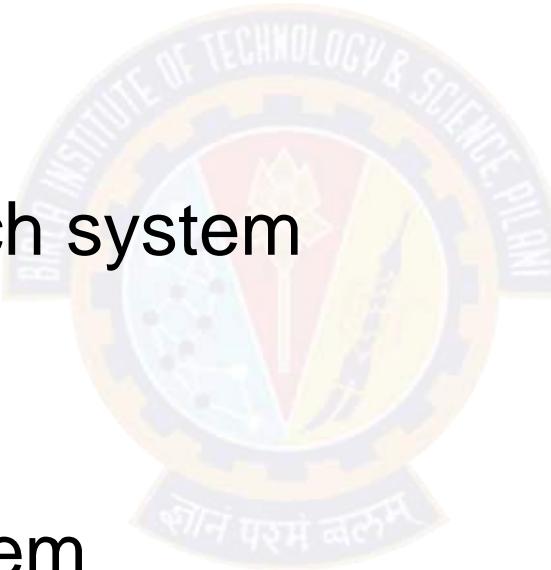
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Boolean Retrieval Model

Prof. Aruna Malapati

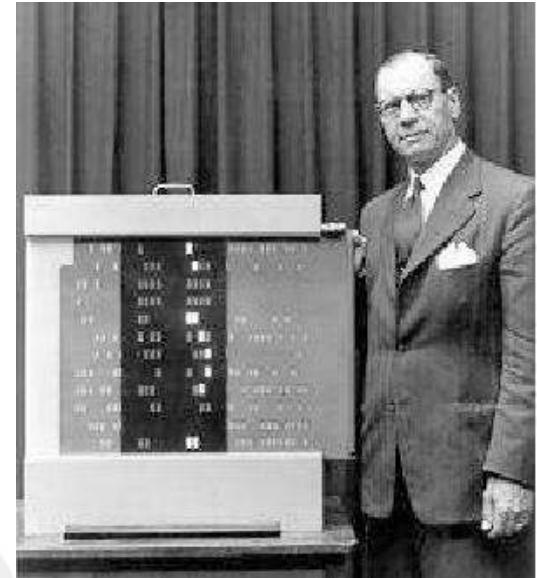
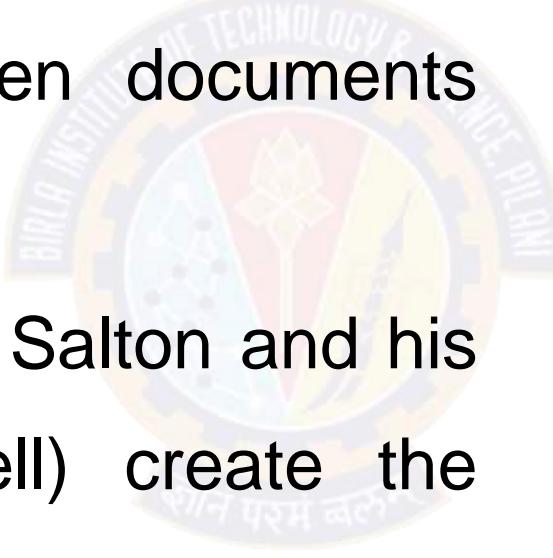
Learning objectives

- History of web search
- Jargons
- Architecture of a search system
- Bag of words model
- Boolean retrieval System



A Brief History of Web Search

- 1957: Hans-Peter Luhn (IBM) uses words as indexing units for documents – Measure similarity between documents by word overlap
- 1960s and 1970s: Gerard Salton and his students (Harvard, Cornell) create the SMART system – Vector space model – Relevance feedback



A Brief History of Web Search

- 1991: Tim Berners-Lee “invents” the World Wide Web
- First Web search engines:
 - Archie: Query file names by regular expressions
 - Architext/Excite: Full text search, simple ranking (1993)
- Until 1998, web search meant information retrieval
- 1998: Google was founded – Exploits link structure using the PageRank algorithm



Jargons

➤ Corpus



Examples

- ✓ Medline / Pubmed document collection
- ✓ Tweets
- ✓ Face books posts
- ✓ Customer Reviews about a product



➤ Information Need



Examples

- ✓ What is the capital of India?
- ✓ Will the Finance Ministry reduce personal taxes?
- ✓ What is the currency in India?

Jargons (Contd..)

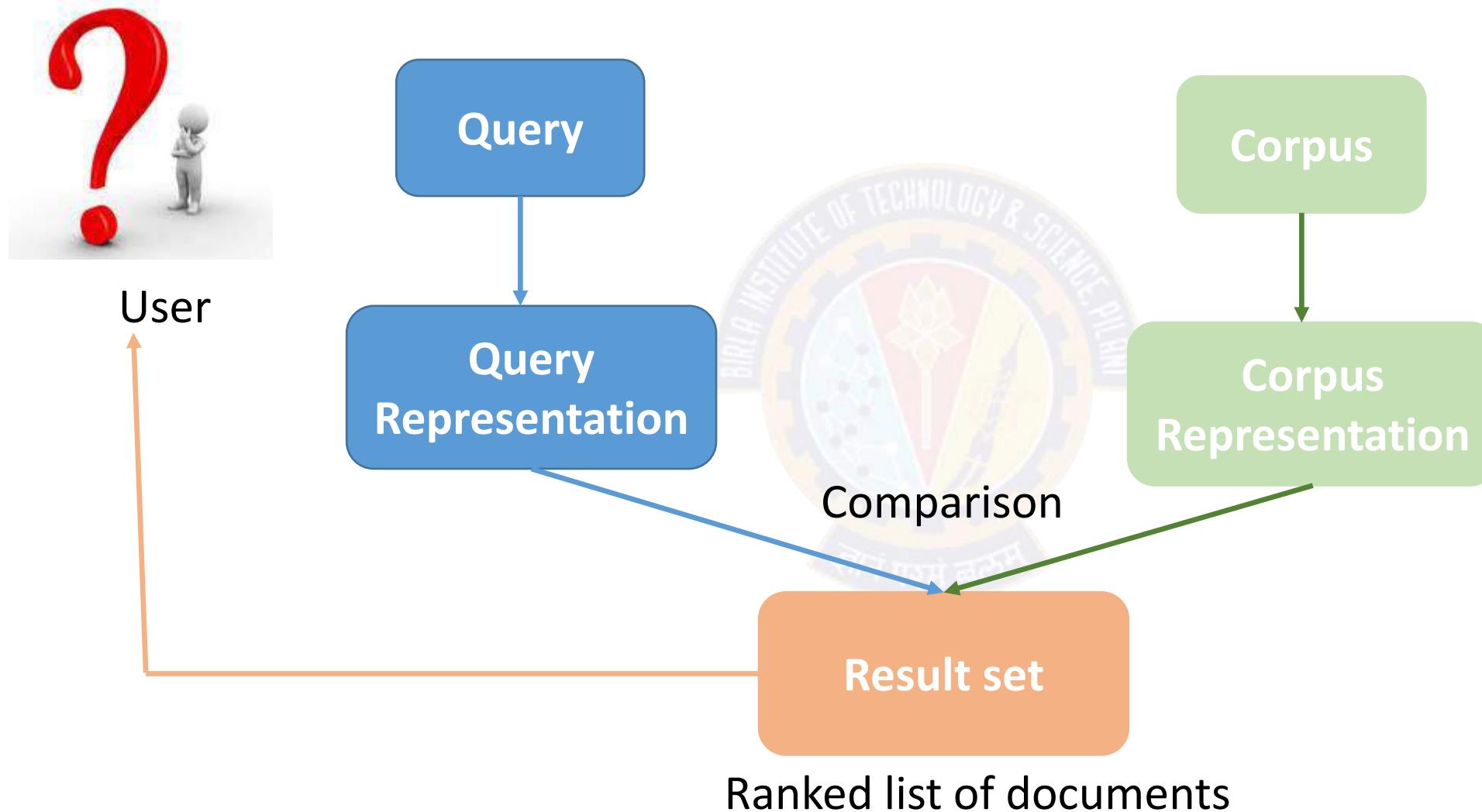
➤Query



Examples

- ✓ Medline / Pubmed document collection

Information Retrieval Systems (IRS)



Bag of Words representation

- A very popular and basic representation of documents is the bag of words model.
 - Each document is represented by a **bag (= multiset) of terms** from a predefined vocabulary.

He was as cunning as
a Jackal

These Grapes are too
sweet but the poor
Jackal could not have it.



Term Incidence Matrix

Two Forms of Term incidence Matrix

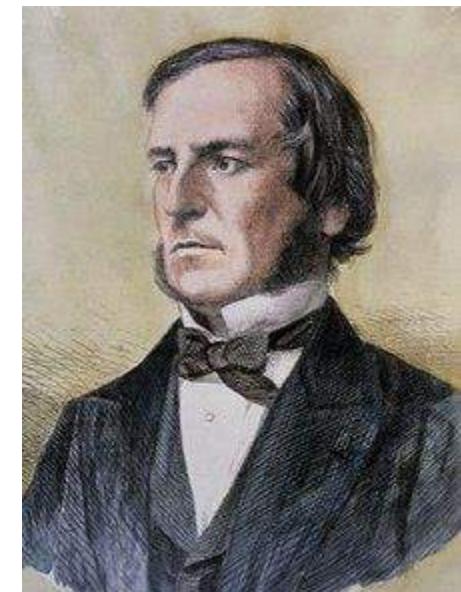
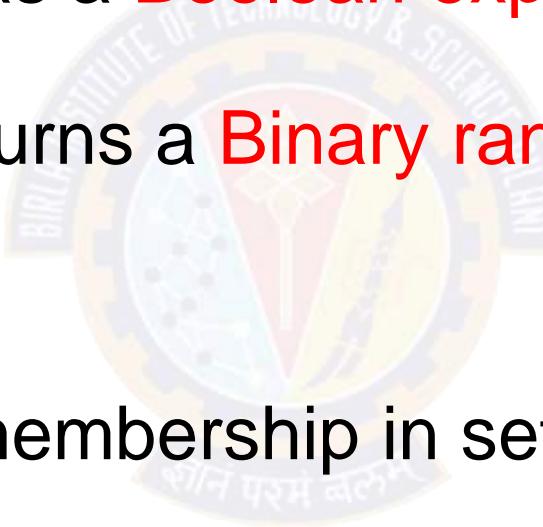
	T1	T2	T3	T4	T5	T6
D1	1	0	0	1	1	0
D2	0	1	0	1	1	0
D3	1	0	1	0	1	1
D4	1	0	1	0	1	1



	T1	T2	T3	T4	T5	T6
D1	6	0	0	2	1	0
D2	0	8	0	5	3	0
D3	2	0	6	0	5	2
D4	5	0	2	0	6	7

Boolean Retrieval Model

- Documents are represented as a **vector of the indexed terms**.
- Query is represented as a **Boolean expressions over index terms**.
- The search system returns a **Binary ranking function**, i.e. 0/1-valued.
- Retrieval is based on membership in sets



Boolean Operators

- The following are the three operators used in Boolean Retrieval model.

- AND (Conjunction or \wedge)
- OR (Disjunction or \vee)
- NOT (Negation or \neg)



\wedge	0	1
0	0	0
1	0	1

\vee	0	1
0	0	1
1	1	1

\neg	
0	1
1	0

Example

<i>doc</i>	<i>t₁</i>	<i>t₂</i>	<i>t₃</i>	<i>t₄</i>	<i>t₅</i>	<i>t₆</i>	<i>t₇</i>	<i>t₈</i>	<i>t₉</i>	<i>t₁₀</i>	<i>t₁₁</i>
<i>D₁</i>	0	0	1	0	1	1	0	0	0	1	0
<i>D₂</i>	1	1	0	1	0	0	0	0	1	0	1
<i>D₃</i>	1	1	0	0	0	1	0	0	0	1	1
<i>D₄</i>	0	0	0	0	0	1	0	0	1	0	1

Query: t1 AND t2 AND NOT t4

Pros and Con's of Boolean Retrieval Model

- + Simple query paradigm, easy to understand
- A binary ranking function returns a set of results, i.e. it is unordered
- Doc-term matrix is **too sparse**
- Controlling the result size is difficult
- Similarity queries are not supported

Westlaw - Online legal research service for US law

- Includes more than 40,000 databases of case law, state and federal statutes, administrative codes, law journals, newspapers ...
 - Offers search by:
 - “Terms and Connectors” – Boolean Search
 - “Natural Language” – Free text querying (added in 1992)
 - Boolean search includes the Boolean operators plus some proximity operators
 - space = OR • /s, /p, /k = matches in the same sentence, paragraph or within k-words respectively
 - & = AND • ! = a trailing wildcard query
 - Example: “trade secret” /s disclos! /s prevent /s employe!
disab! /p access! /s (work-site work-place) (employment /3 place)



Thank You!

In our next session: Information Retrieval Pipeline



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Information Retrieval Pipeline

Prof. Aruna Malapati

Learning objectives

- Information Retrieval pipeline
- Inverted index construction



Information Retrieval Pipeline

Documents collected from various sources



Ram and Shyam are childhood friends.

⋮

Tokenizer

Token stream

Ram

Shyam

childhood

friends

Linguistic modules

DE pluralization

Case folding

Modified tokens/
Stream of normalized
tokens

ram

shyam

childhood

friend

Indexer

childhood

2

4

friend

1

2

ram

13

16

shyam

1

4

Inverted index

Steps during indexing

Doc 1

The Jackal was eyeing
at the grapes

Doc 2

He was as cunning as
a Jackal

Doc 3

These Grapes are too
sweet but the poor
Jackal could not have it.

Sequence
Of tokens,
Document ID
pairs

the	1
Jackal	1
Was	1
eyeing	1
at	1
the	1
grapes	1
he	2
was	2
as	2
cunnin	2
g	2
as	2
a	2
jackal	2
these	3
grapes	3
are	3
too	3
sweet	3
.	
.	
.	

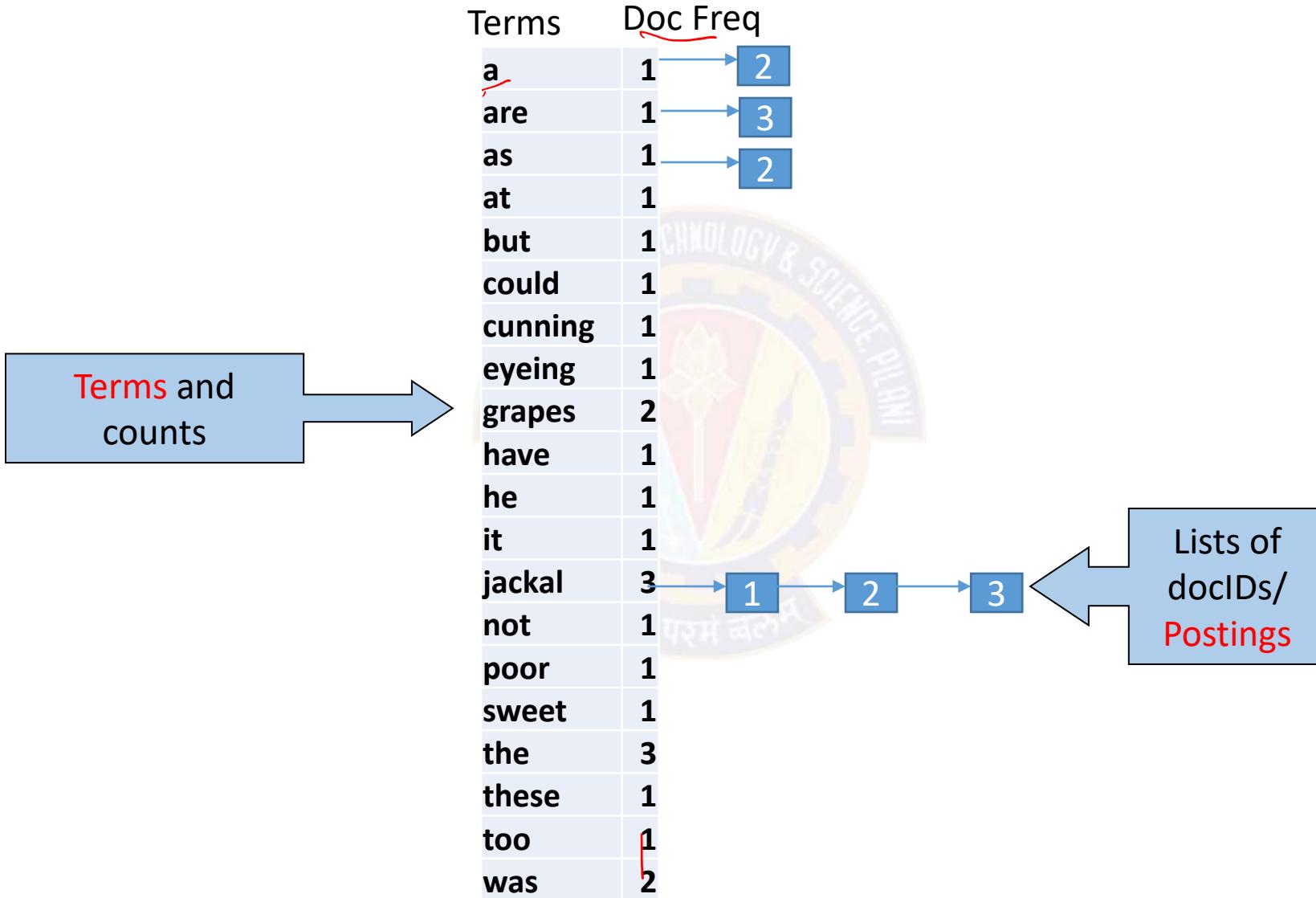
Sort
by
terms

at	1
eyeing	1
grapes	1
jackal	1
the	1
the	1
was	1
a	2
as	2
as	2
cunning	2
he	2
jackal	2
was	2
are	3
but	3
Could	3
.	
.	
.	

Sort
by
Doc ids

a	2
are	3
as	2
as	2
at	1
but	3
could	3
cunning	2
eyeing	1
grapes	1
grapes	3
have	3
he	2
it	3
jackal	1
jackal	2
jackal	3
not	3
poor	3
sweet	3
the	1
the	3
the	1

Inverted Index



Posting list implementations

- Arrays vs Linked list
- Factors that influence the decision
 - Is the corpus fixed?
 - Can we fit the entire posting list in main memory?





Thank You!

In our next session: Merge Algorithm and Query Optimization



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Merge Algorithm

Prof. Aruna Malapati

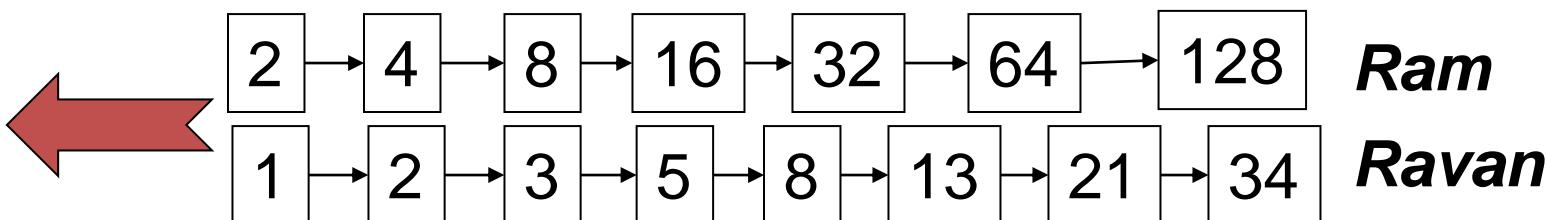
Learning objectives

- Answering Queries using merge algorithm



Query processing: AND

- Consider processing the query:
 - ***Ram AND Ravan***
 - Locate ***Ram*** in the Dictionary;
 - Retrieve its postings.
 - Locate ***Ravan*** in the Dictionary;
 - Retrieve its postings.
 - “Merge” the two postings:



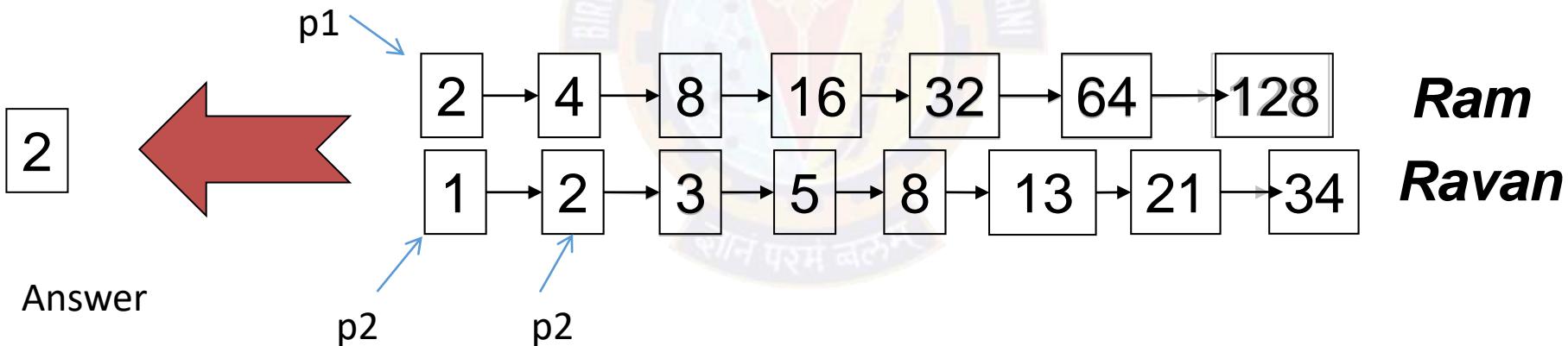
Intersecting two postings lists (a “merge” algorithm)

INTERSECT(p_1, p_2)

```
1  answer  $\leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then ADD(answer,  $\text{docID}(p_1)$ )
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8          then  $p_1 \leftarrow \text{next}(p_1)$ 
9          else  $p_2 \leftarrow \text{next}(p_2)$ 
10 return answer
```

Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1
P2: pointer to current location in list2



```
INTERSECT( $p_1, p_2$ )
1   answer ← ⟨ ⟩
2   while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3     do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4       then ADD(answer, docID( $p_1$ ))
5          $p_1 \leftarrow \text{next}(p_1)$ 
6          $p_2 \leftarrow \text{next}(p_2)$ 
7     else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8       then  $p_1 \leftarrow \text{next}(p_1)$ 
9     else  $p_2 \leftarrow \text{next}(p_2)$ 
10  return answer
```

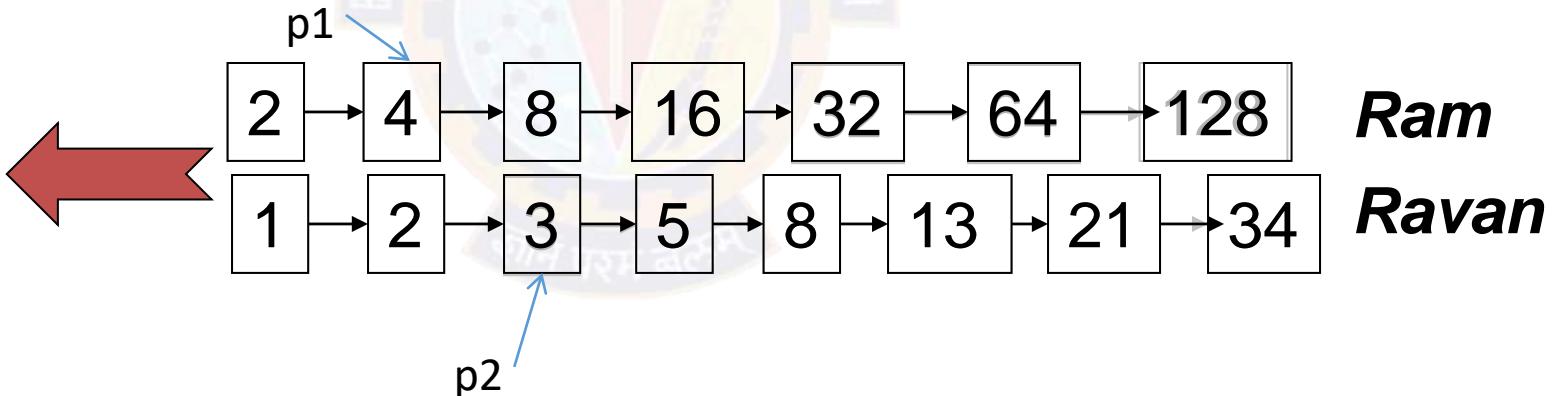
Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1

P2: pointer to current location in list2

2

Answer



INTERSECT(p_1, p_2)

```
1  answer ← ⟨ ⟩  
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$   
3    do if  $\text{docID}(p_1) = \text{docID}(p_2)$   
4      then ADD(answer,  $\text{docID}(p_1)$ )  
5       $p_1 \leftarrow \text{next}(p_1)$   
6       $p_2 \leftarrow \text{next}(p_2)$   
7    else if  $\text{docID}(p_1) < \text{docID}(p_2)$   
8      then  $p_1 \leftarrow \text{next}(p_1)$   
9    else  $p_2 \leftarrow \text{next}(p_2)$   
10   return answer
```

Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1

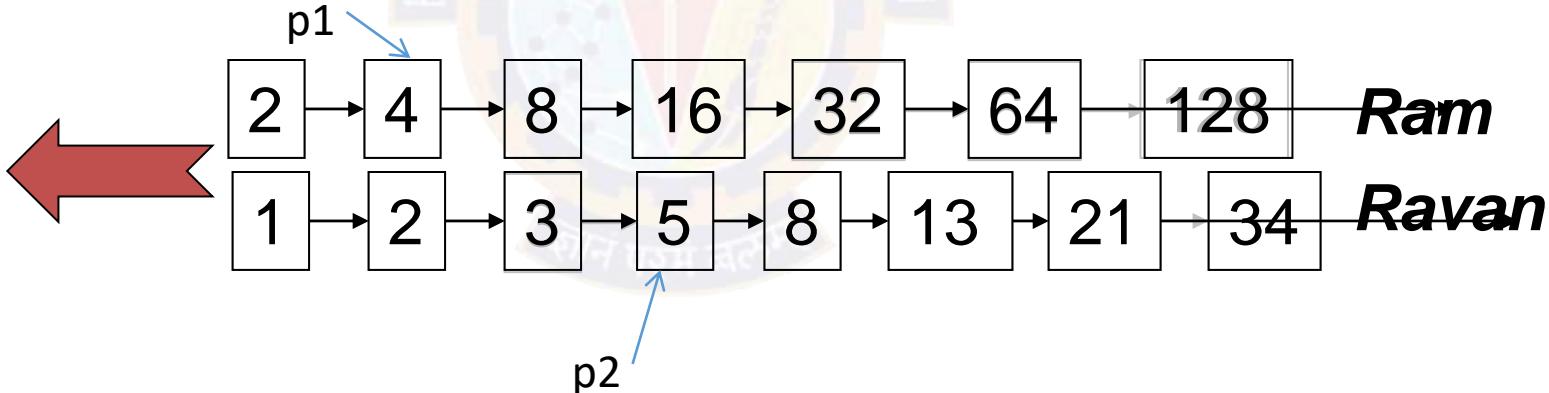
P2: pointer to current location in list2

INTERSECT(p_1, p_2)

```
1  answer ← ⟨ ⟩  
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$   
3    do if  $\text{docID}(p_1) = \text{docID}(p_2)$   
4      then ADD(answer,  $\text{docID}(p_1)$ )  
5       $p_1 \leftarrow \text{next}(p_1)$   
6       $p_2 \leftarrow \text{next}(p_2)$   
7    else if  $\text{docID}(p_1) < \text{docID}(p_2)$   
8      then  $p_1 \leftarrow \text{next}(p_1)$   
9    else  $p_2 \leftarrow \text{next}(p_2)$   
10   return answer
```

2

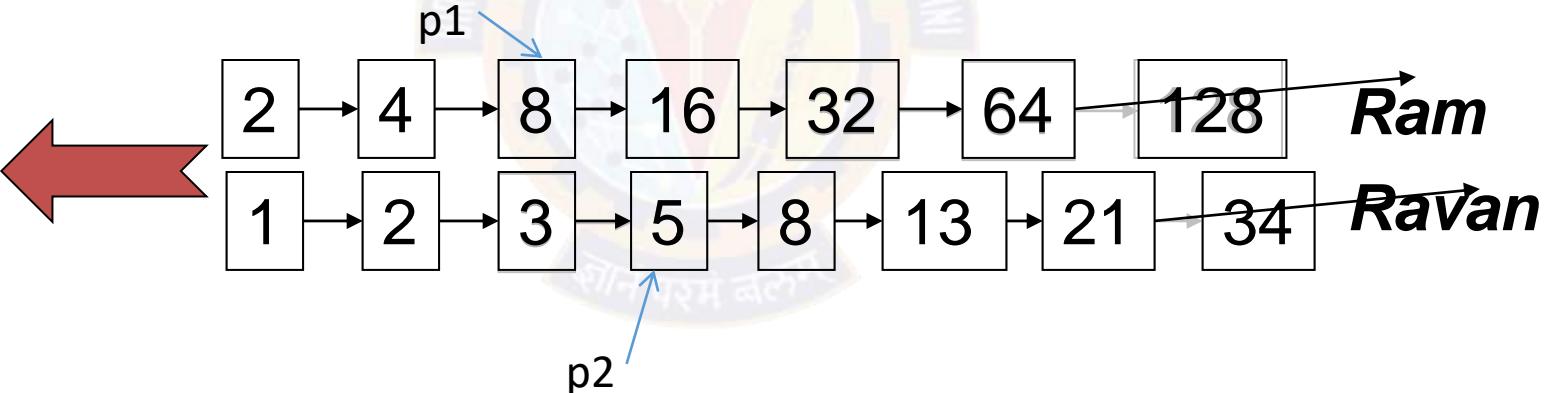
Answer



P1: pointer to current location in list1
P2: pointer to current location in list2

2

Answer

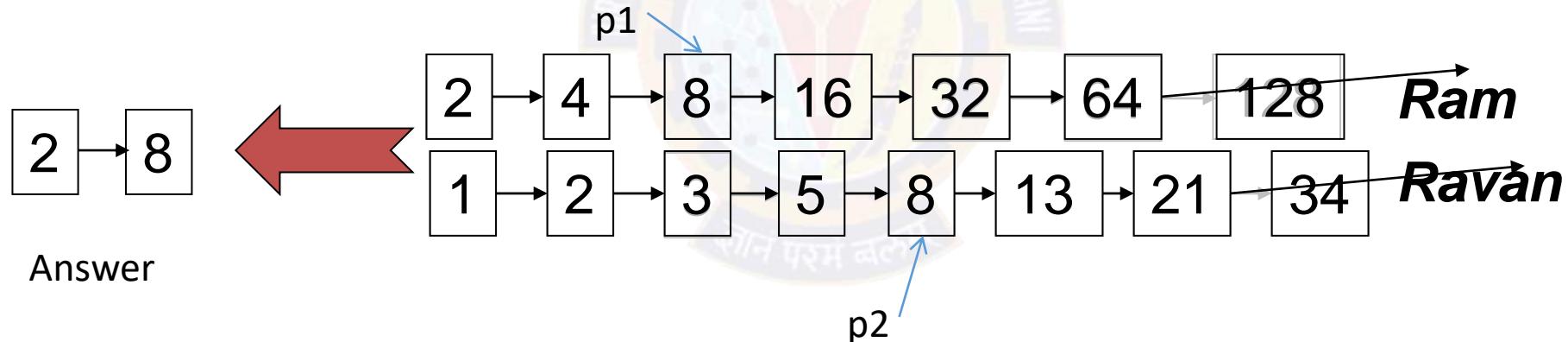


```
INTERSECT(p1, p2)
1  answer ← ⟨ ⟩
2  while p1 ≠ NIL and p2 ≠ NIL
3  do if docID(p1) = docID(p2)
4      then ADD(answer, docID(p1))
5      p1 ← next(p1)
6      p2 ← next(p2)
7  else if docID(p1) < docID(p2)
8      then p1 ← next(p1)
9  else p2 ← next(p2)
10 return answer
```

Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1
P2: pointer to current location in list2

```
INTERSECT( $p_1, p_2$ )
1  $answer \leftarrow \langle \rangle$ 
2 while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3 do if  $docID(p_1) = docID(p_2)$ 
4     then ADD( $answer, docID(p_1)$ )
5      $p_1 \leftarrow next(p_1)$ 
6      $p_2 \leftarrow next(p_2)$ 
7 else if  $docID(p_1) < docID(p_2)$ 
8     then  $p_1 \leftarrow next(p_1)$ 
9     else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```



Postings sorted by DocIds.

More query processing

- Brutus OR Caesar
- NOT Brutus
- Brutus AND NOT Caesar
- Brutus OR NOT Caesar





Thank You!

In our next session: Query Optimization



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Query Optimization

Prof. Aruna Malapati

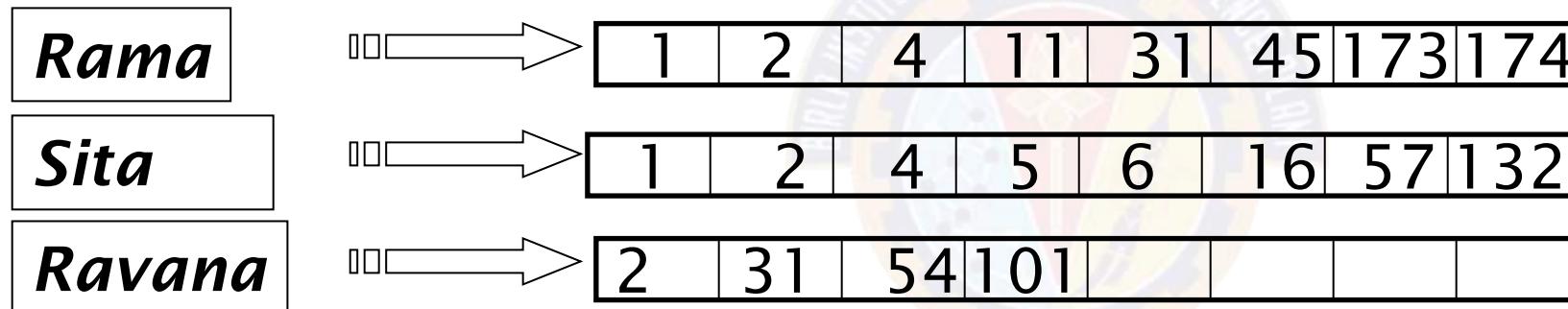
Learning objectives

- Apply query optimization for any type of query



Query Optimization

- Consider a query that is an *and* of t terms.
- For each t terms get the postings list, then AND them together.



QUERY: Rama AND Sita AND Ravana

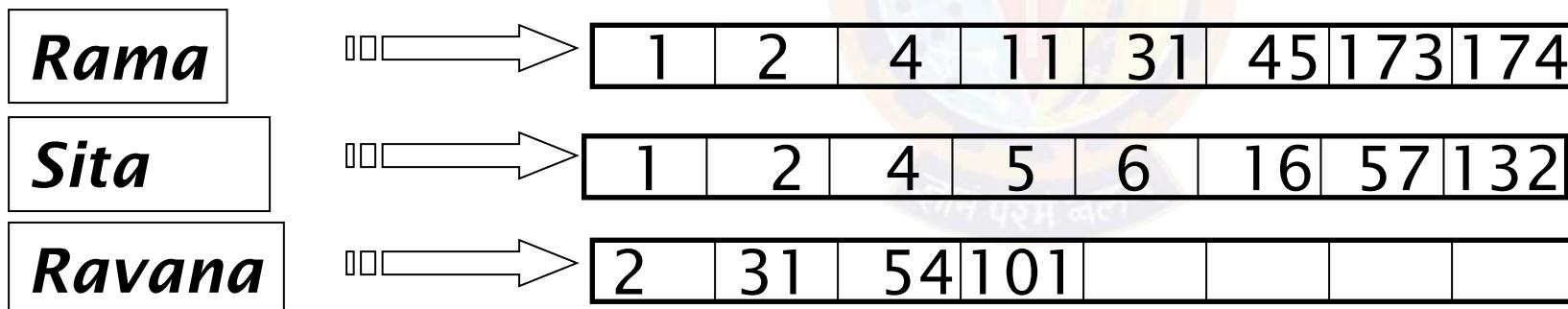
Rama AND (Sita AND Ravana)

(Rama AND Sita) AND Ravana

(Rama AND Ravana) AND Sita

Query Optimization

- Process in the order of increasing document frequency.
- Intersect the two smallest postings list
- All intermediate results will be no bigger than the smallest postings list, so we are likely to minimize the work.



QUERY: Rama AND Sita AND Ravana

Execute the query as (Rama AND Sita) AND Ravana

This is why the doc freq is stored



Thank You!

In our next session: Normalization



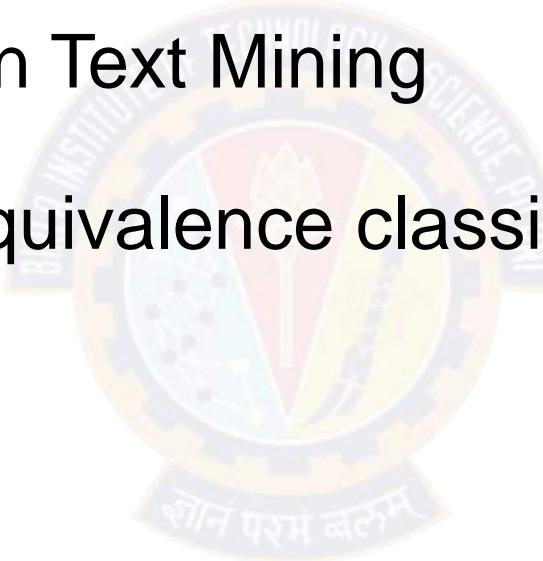
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Tolerant Retrieval using Normalization

Prof. Aruna Malapati

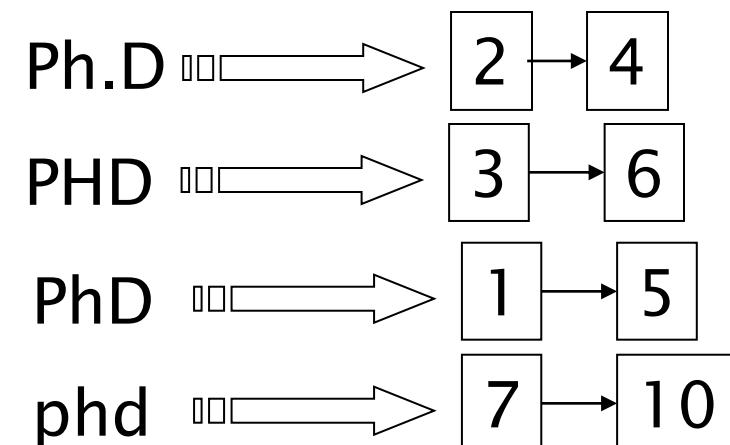
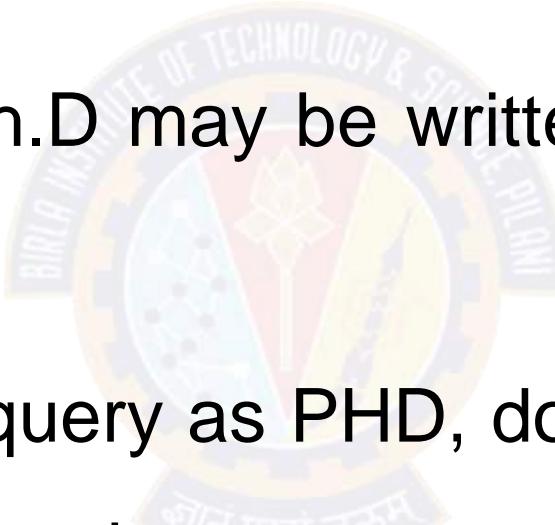
Learning objectives

- Explain Tolerant Retrieval
- Define Normalization in Text Mining
- Normalization using equivalence classing and query expansion



Tolerant Retrieval

- Some words may have different representations in the indexed documents.
- For example the word Ph.D may be written as Ph.D or PHD or PhD or phd
- When the user enters a query as PHD, documents contains all forms of this word must be returned as answer.



Information Retrieval Pipeline

Documents collected from various sources



Ram and Shyam are childhood friends.

⋮

Token stream

Tokenizer

Ram

Shyam

childhood

friends

Linguistic Preprocessing

Case folding

DE pluralization

ram

shyam

childhood

friend

Modified tokens/
Stream of normalized
tokens

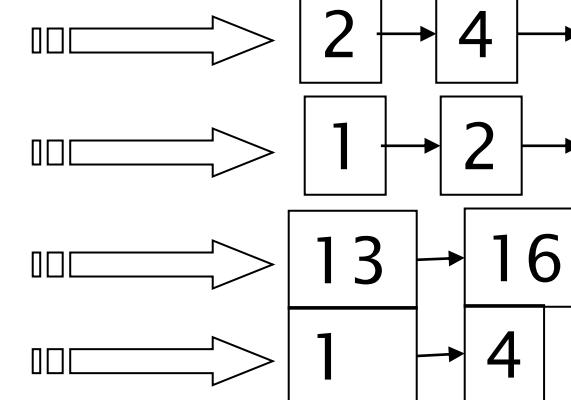
Indexer

childhood

friend

ram

friend



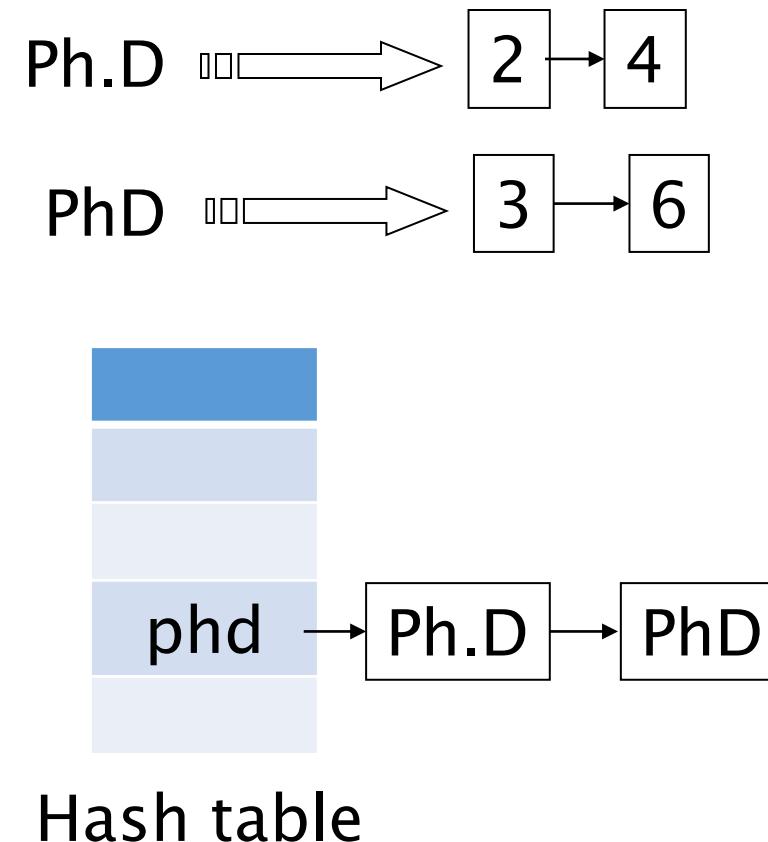
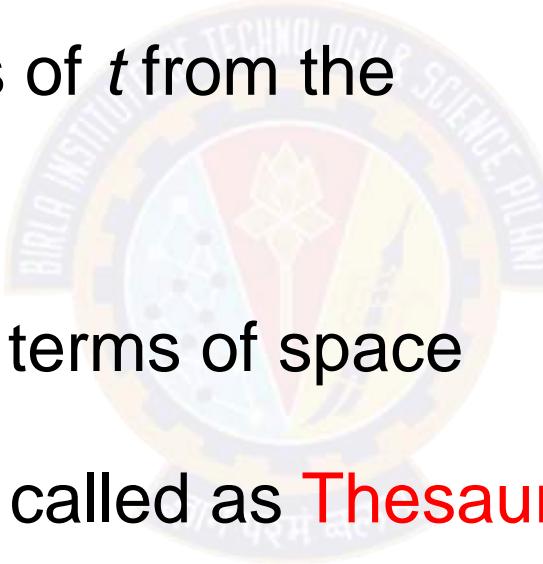
Inverted index

Normalization

- The process of normalization is to **reduce multiple tokens to the same canonical term**, such that matches occur despite superficial differences.
- For example suppose you have variant forms of writing the same word like Ph.D,PhD which gets mapped to a single token as phd.
- **Equivalence classing** is predominantly used technique for normalization.
Deleting the periods, hyphens, Accents etc..

Query expansion / Asymmetric expansion

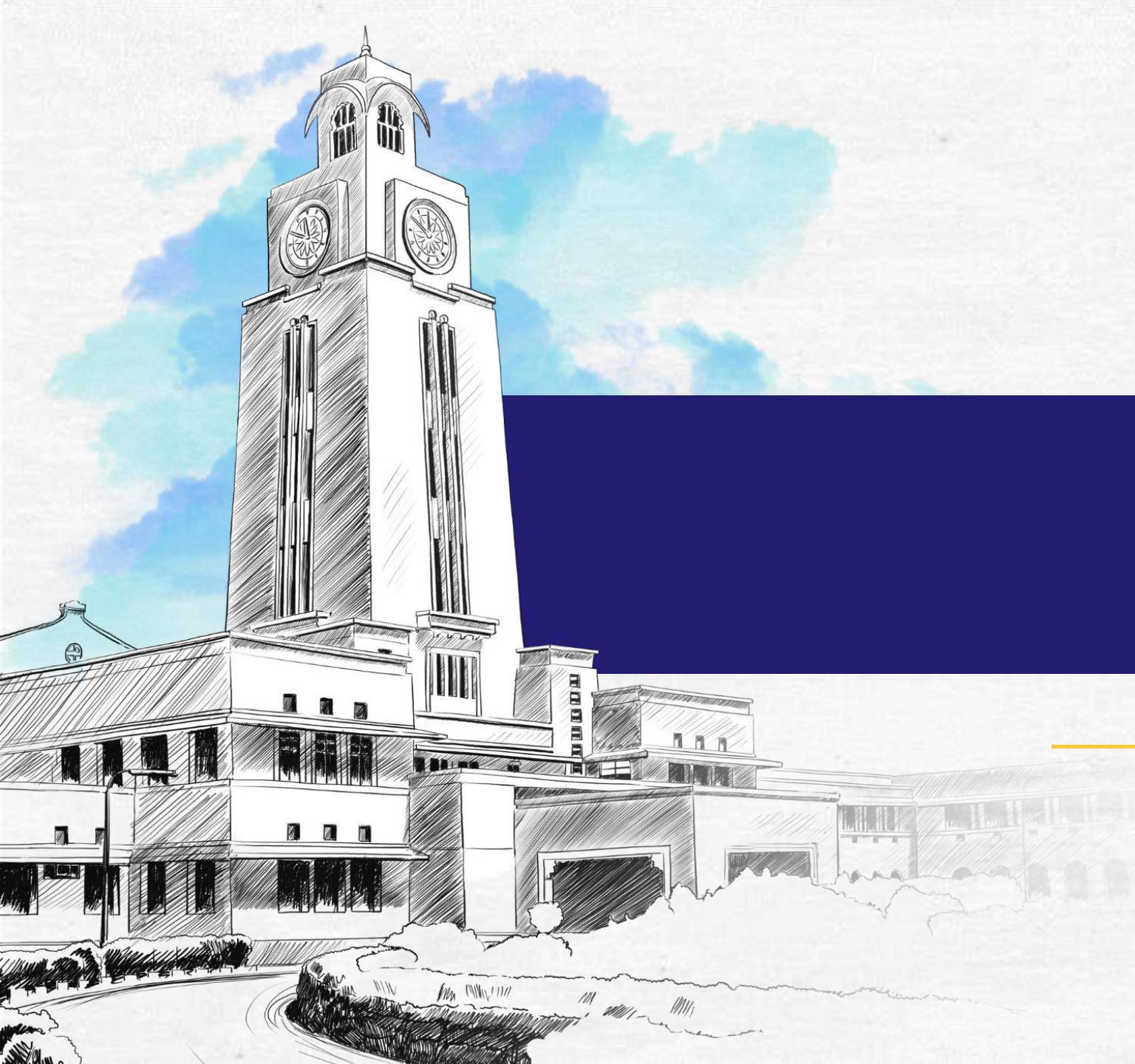
- For each term, t , in a query, expand the query with synonyms and related words of t from the thesaurus.
- Powerful but less efficient in terms of space
- These hand crafted terms is called as **Thesauri**.
- Handle Synonyms and Homonyms





Thank You!

In our next session: Stemming



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stemming

Prof. Aruna Malapati

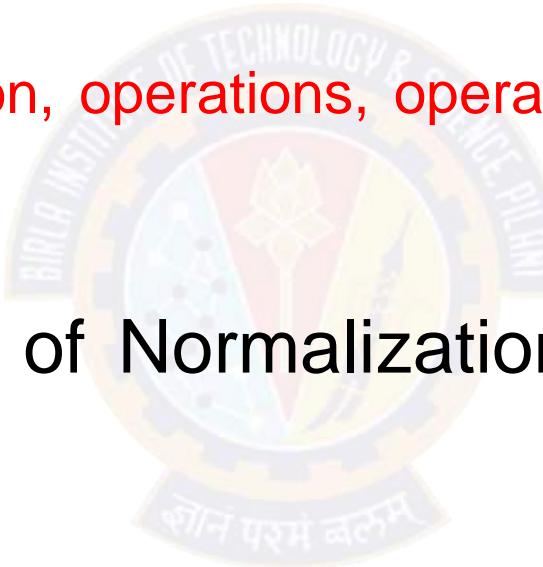
Learning objectives

- Explain stemming
- Apply Porter stemmer
- Analyze the effects of stemming



Stemming

- Stemming is the process of reducing inflectional form of words to their root form.
 - Example words like operation, operations, operational, operating can be reduced to operati (root word)
- Stemming is crude form of Normalization in which the suffixes are removed.
- The advantage of suffix stripping is to reduce the total number of terms in the inverted index resulting in a smaller size and complexity of the data in the system.



Porter Stemmer

- A consonant in a word is a letter other than A, E, I, O or U, and other than Y preceded by a consonant.
- Any letter not a consonant is a Vowel.
- All the words in English are of the form C(VC)^mV where m is measure of any word or word part when represented in this form (VC).

Examples:

m=0 TR, EE, TREE, Y, BY.

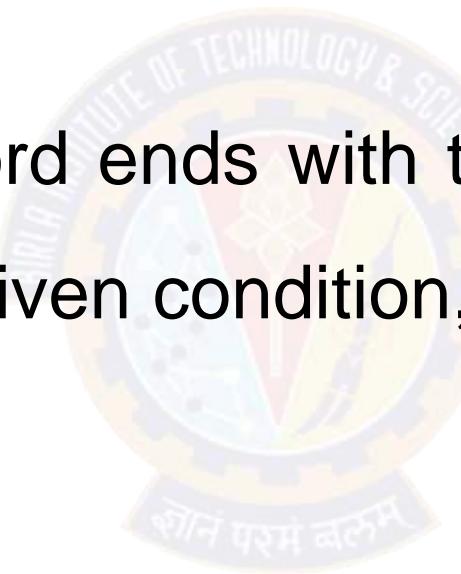
m=1 TROUBLE, OATS, TREES

m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

Porter Stemmer (contd..)

- The rules for removing a suffix will be given in the form
(condition) S1 -> S2
- This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2.

(m > 1) EMENT ->



Porter Stemmer (contd..)

- The 'condition' part may also contain the following:
 - *S - the stem ends with S (and similarly for the other letters).
 - *v* - the stem contains a vowel.
 - *d - the stem ends with a double consonant (e.g. -TT, -SS).
 - *o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

Porter Stemmer (contd..)

Step 1a

SSES -> SS caresses -> caress

IES -> I ponies -> poni
ties -> ti

SS -> SS caress -> caress
S -> cats -> cat

Step 1c

(*v*) Y -> I

happy -> happi
sky -> sky

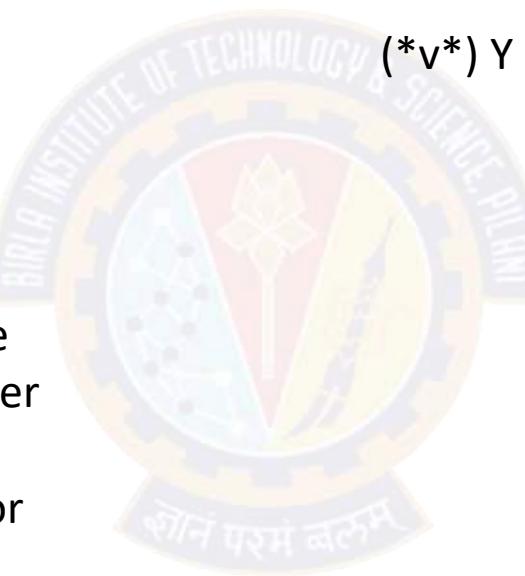
Step 1b

(m>0) EED -> EE feed -> feed

agreed -> agree

(*v*) ED -> plastered -> plaster
bled -> bled

(*v*) ING -> motoring -> motor
sing -> sing



- Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

Effect of stemming

- Suffix stripping of a vocabulary of 10,000 words

Number of words reduced in step 1: 3597

"	2:	766
"	3:	327
"	4:	2424
"	5:	1373

Number of words not reduced: 3650

- The resulting vocabulary of stems contained 6370 distinct entries.
- Thus the suffix stripping process **reduced the size of the vocabulary by about one third.**



Thank You!

In our next session: Lemmatization



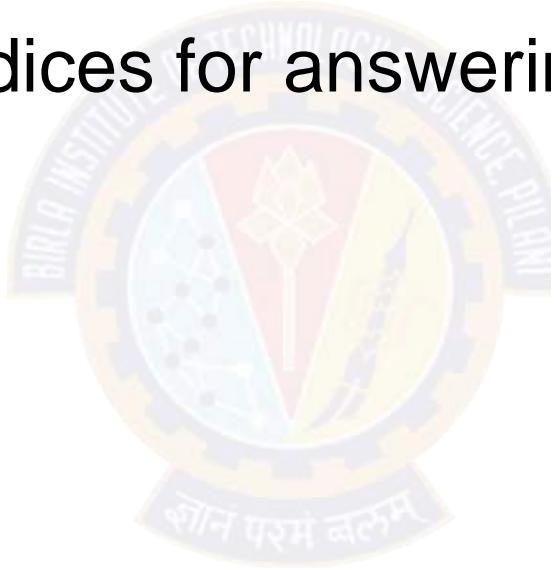
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Types of queries and indexing

Prof. Aruna Malapati

Learning objectives

- List variety of queries that search engine handles
- Identify appropriate indices for answering variant queries



Variety of queries handled by search engine

- Boolean queries — Inverted Index
- Phrase queries — "BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI" "BITS Pilani"
"BITS" | K | "Hyderabad"
BITS | K | Hyderabad
ax
- Positional queries
- Wild card queries

Phrase Queries

- Queries of the form “**BITS HYDERABAD**” where the word order needs to maintained.
- Two approaches
 - Biword Index
 - Positional Index



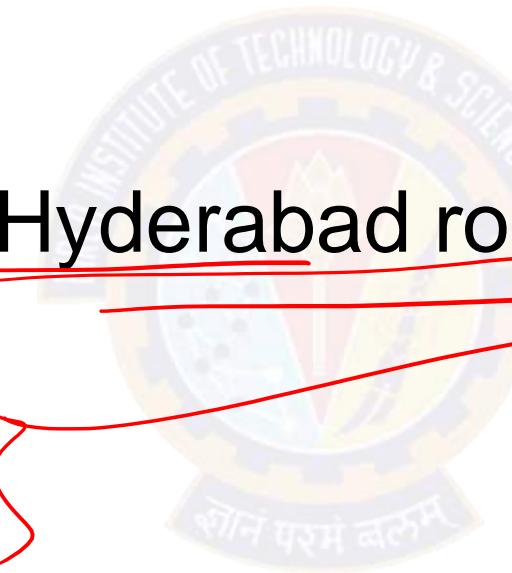
Biword Index

- Index every consecutive pair of terms in the text as a phrase.

- For example “BITS Hyderabad rocks”

➤ BITS Hyderabad

➤ Hyderabad rocks



- Disadvantage: False positives for longer queries

$t_1 t_2 t_3 t_4 t_5 t_6 t_7$

$t_1 t_2$ — (D) t_1
 $t_2 t_3$ — (H) t_2
 $t_3 t_4$ — (D) t_3
 t_4

$t_1 t_2 \times t_2 t_3 \times \dots \times t_3 t_4$

Extended biwords

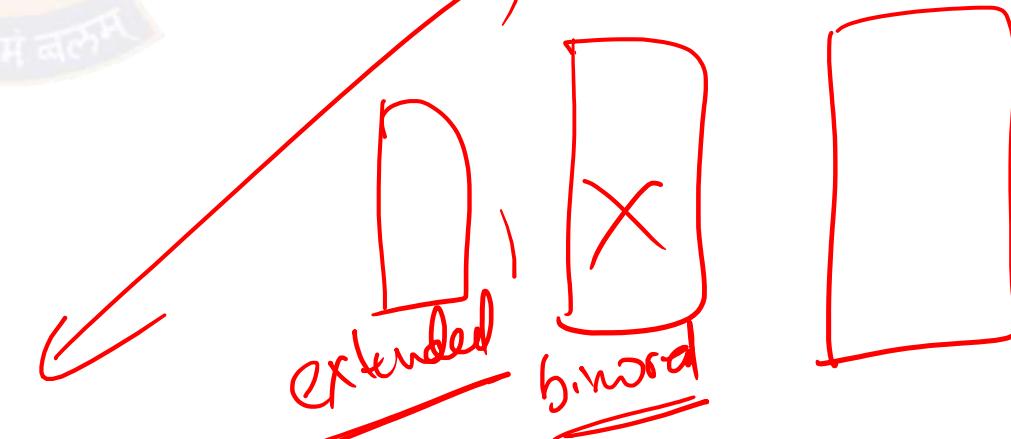
- Parse the indexed text and perform part-of-speech-tagging.
- Bucket the terms into (say) Nouns (N) and articles/prepositions (X).
- Call any string of terms of the form NX*N an extended biword.
 - Each such extended biword is now made a term in the dictionary.

➤ Example: percyJackson and the torch

N X X N

t-percyJackson torch

- Query processing: parse it into N's and X's
 - Segment query into enhanced biwords
 - Look up in index: percyJackson torch





Thank You!

In our next session: Positional Index



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Positional Index

Prof. Aruna Malapati

Positional index

➤ In the postings, store for each **term** the position(s) in which tokens of it appear:

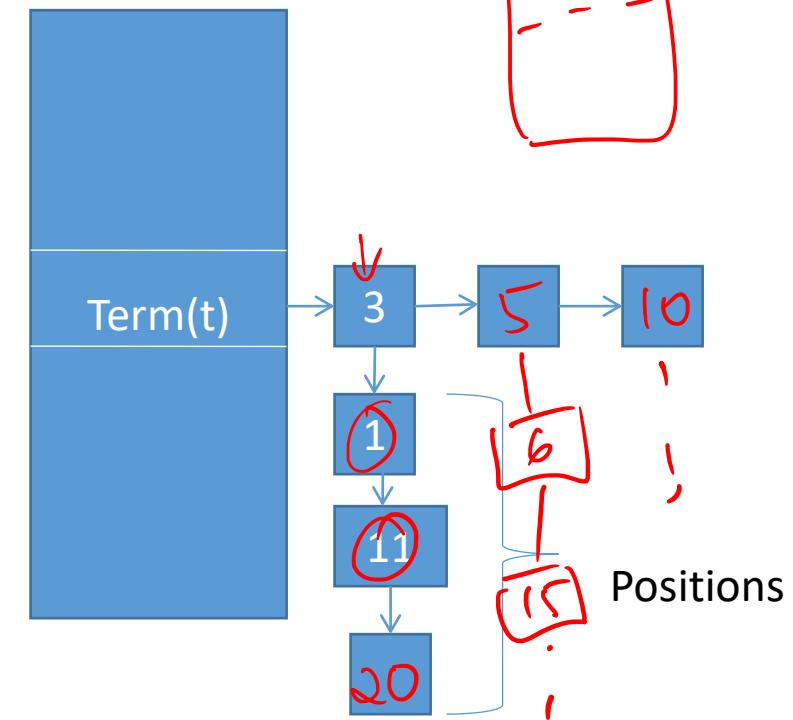
t 100
<term, number of docs containing term;

$doc1$: position₁, position₂ ... ; 20

$doc2$: position₁, position₂ ... ; 15

etc.>

$t_1 \setminus t_2$

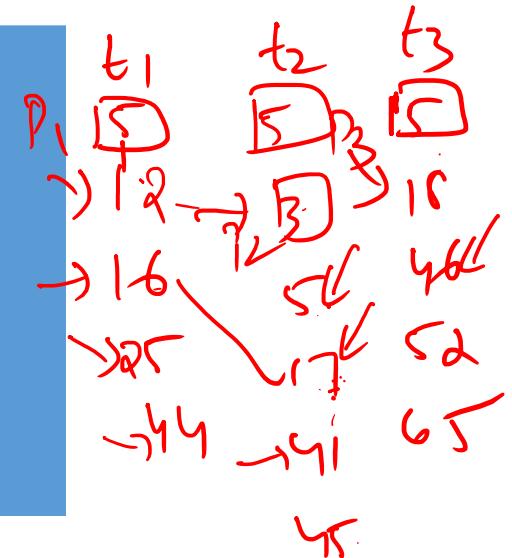


Example

Query = “The undiscovered country”

the:	undiscovered:	country:
3:34,38,55;	3:12,15,19;	3:22,26;
5:12,16,25,44;	5:3,5,17,41,45,96;	5:18,46,52,65;
7:67,87,90,101;	6:21,25,55,62;	7:5,69,91,105;
10:33,39,45,62;	7:4,68,70,85,110;	8:32,42,65,93;
	10:15,34,40,65,81;	10:32,44,75,83;

$t_1 \backslash t_3 t_2$



The occurrence of “The undiscovered country” is found in the following
Doc 5 (16,17,18) and (44,45,46)
Doc 7 (67,68,69)

5 (16, 17, 18)
(44, 45, 46)



Thank You!

In our next session: Wildcard queries



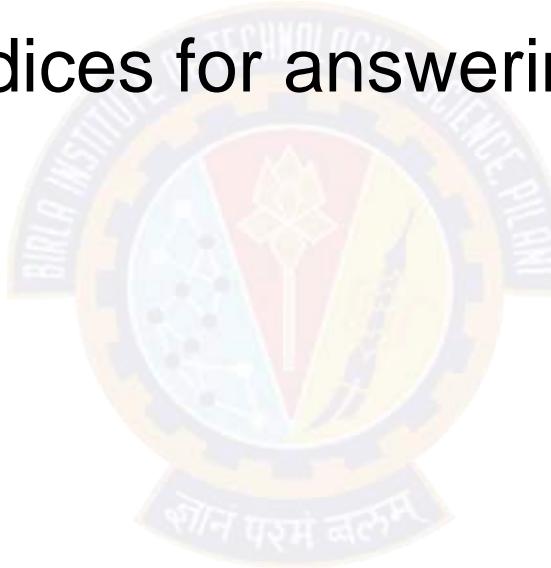
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Wildcard Queries

Prof. Aruna Malapati

Learning objectives

- List and define wildcard queries
- Identify appropriate indices for answering wildcard queries



Wildcard queries

➤ Trailing wildcard query

➤ Ex: a^{*}

➤ Leading wildcard query

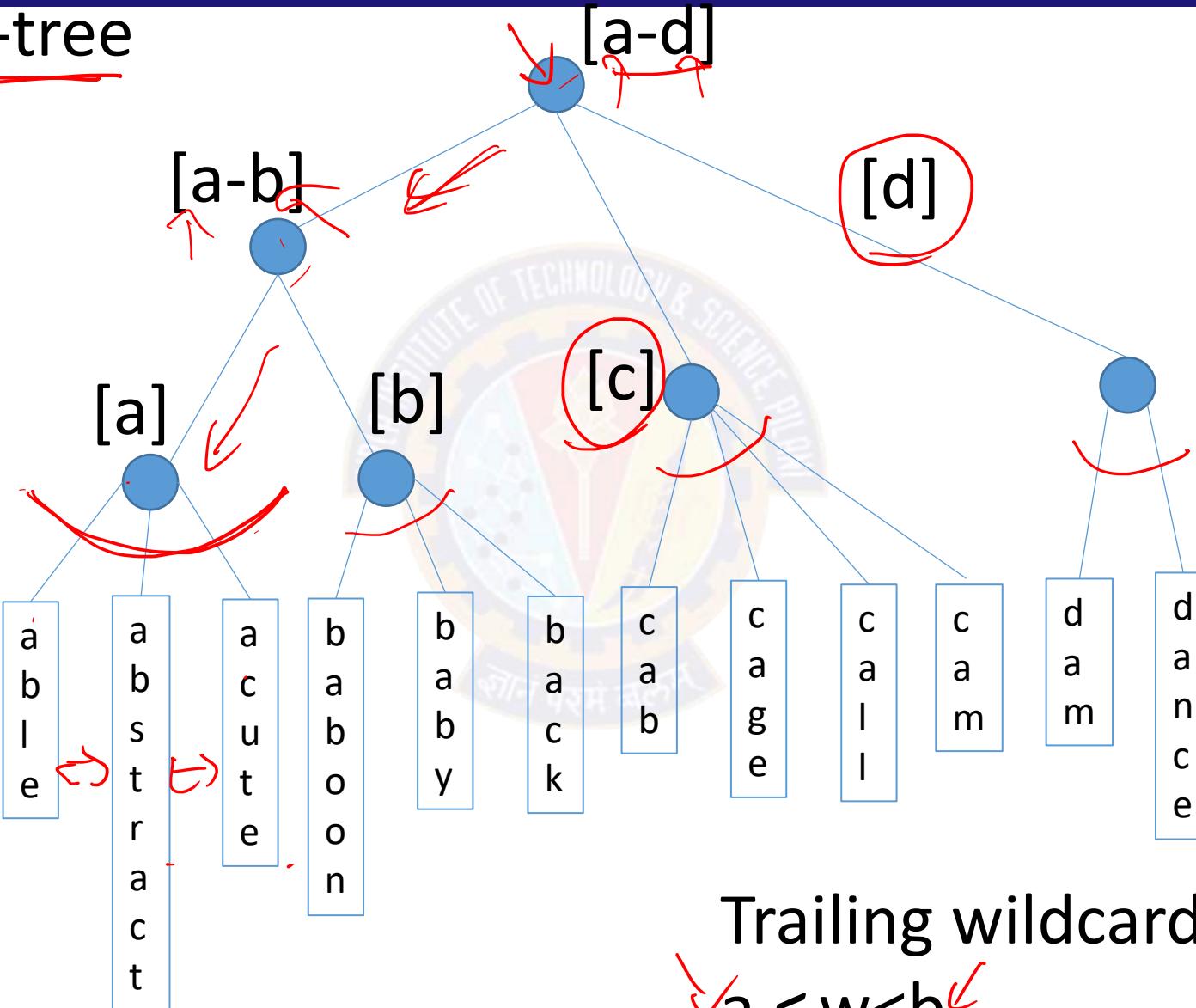
➤ Ex *a



Trailing wildcard query

able
abstract
acute
baboon
baby
back
cab
cage
call
cam
dam
dance

[2-4] B-tree



Inverted index
B-tree

[abc-abl]

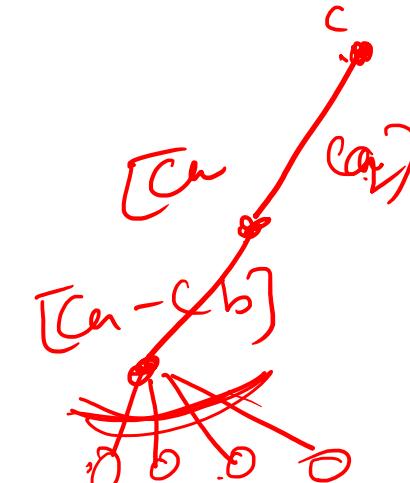
Trailing wildcard query: a^*
 $a \leq w < b$

Wildcard queries

Trailing wild card queries

➤ ca* (e.g cab,cal,cam,can,cap,car,cat)

➤ Walk down the tree following c,a



➤ Retrieve all words w such that: ca ≤ w < cb (i.e. all the words having prefix "ca")

➤ Let set of these terms be W.

➤ Use inverted index to retrieve documents containing terms in W



Thank You!

In our next session: Leading Wildcard queries



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Leading Wildcard Queries

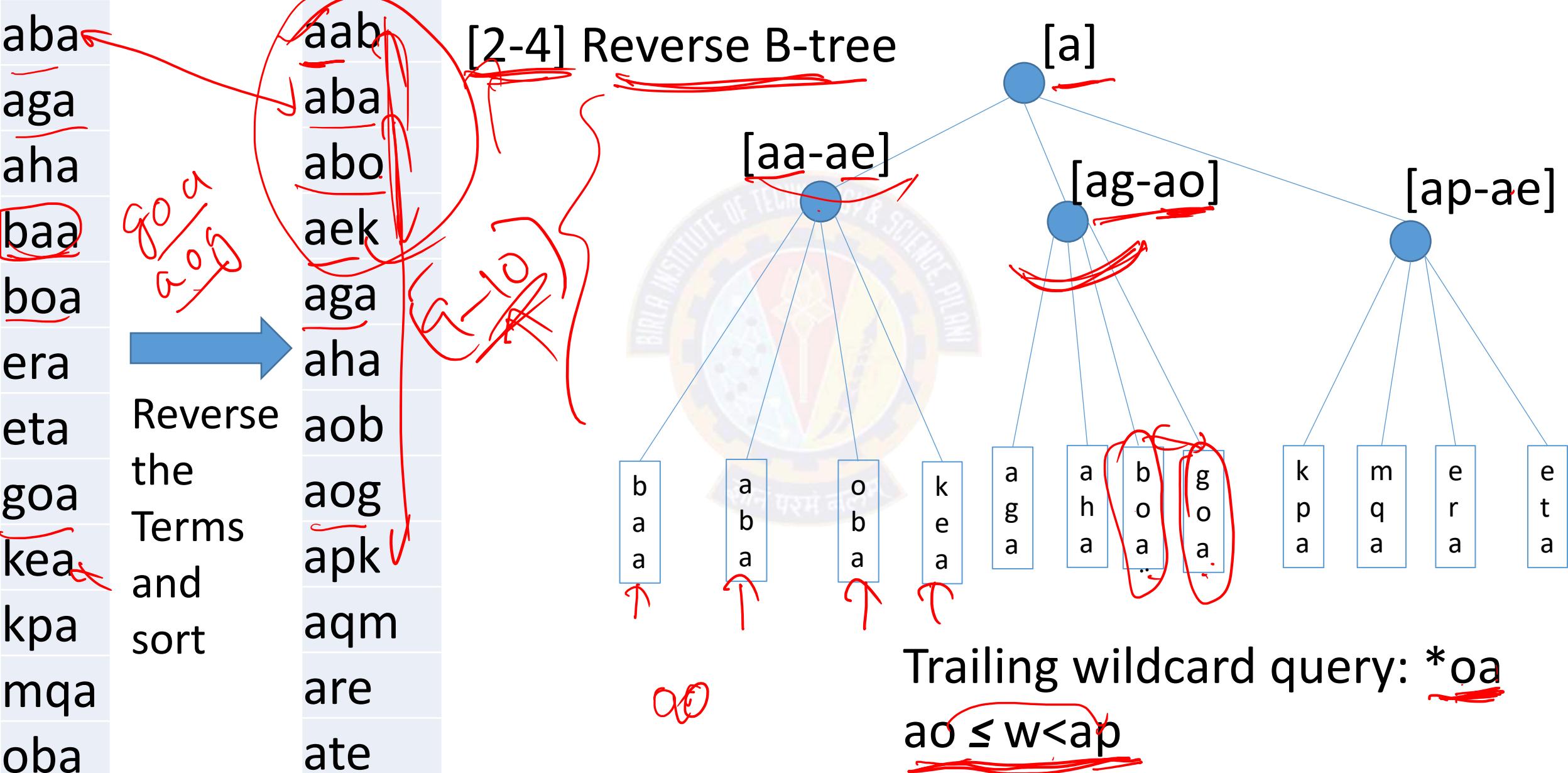
Prof. Aruna Malapati

Learning objectives

- Define a leading wildcard query
- Identify appropriate indices for leading answering wildcard queries



Leading wildcard queries





Thank You!

In our next session: K-Gram Index



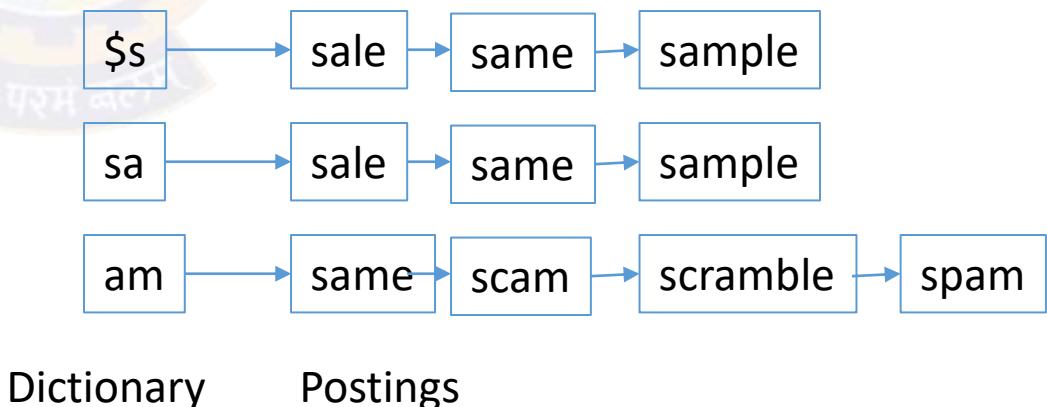
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

K-Gram Index

Prof. Aruna Malapati

K-gram index

- Query: \$sam*
- 2-grams from the query {\$s and sa and am}
- Fetch all the terms found in the posting list of the three K-grams using the merge algorithm



Postprocessing

- Consider using the 3-gram index described for the query
red*





Thank You!

In our next session: Ranked Retrieval using vector space model



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Ranked Retrieval

Prof. Aruna Malapati

Learning objectives

- Define Ranked Retrieval and vector space model
- Relate vector notation to the documents and queries
- Types of vector coefficients
- Compute similarity between query and documents

Ranked Retrieval

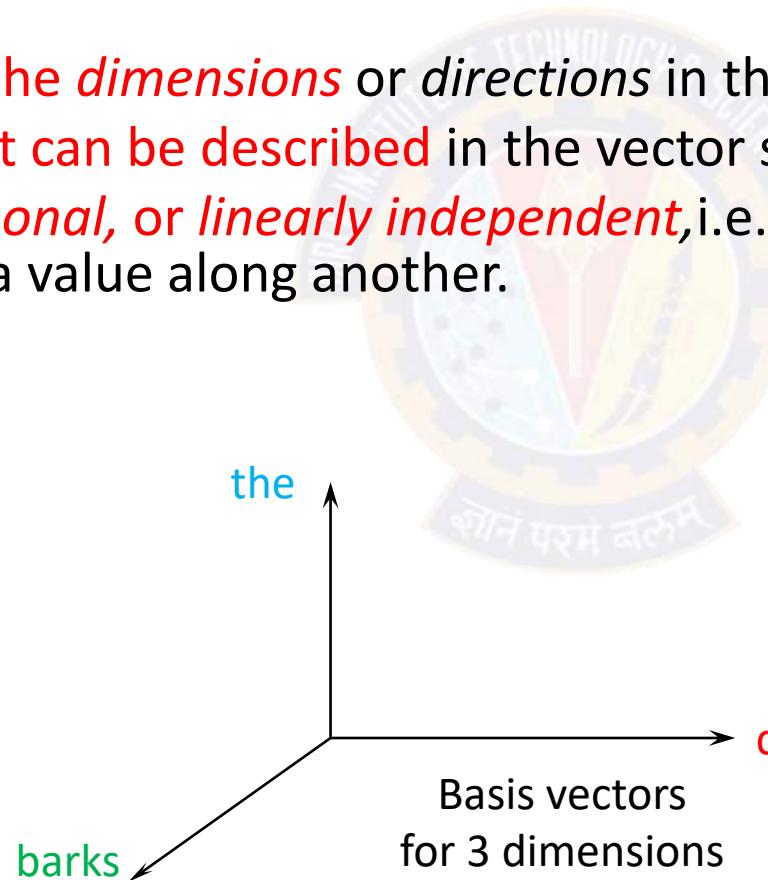
- The user enters a **free text query** and we would like to return the results in a ranked order.
- Hence we need a **scoring scheme** to compute the relevance of the document to the user query.
- A simple scoring scheme in the range [0-1].

Vector Space Model

- Any text object can be represented by a term vector
 - Examples: Documents, queries, sentences,
- Similarity is determined by distance in a vector space
 - Example: The cosine of the angle between the vectors

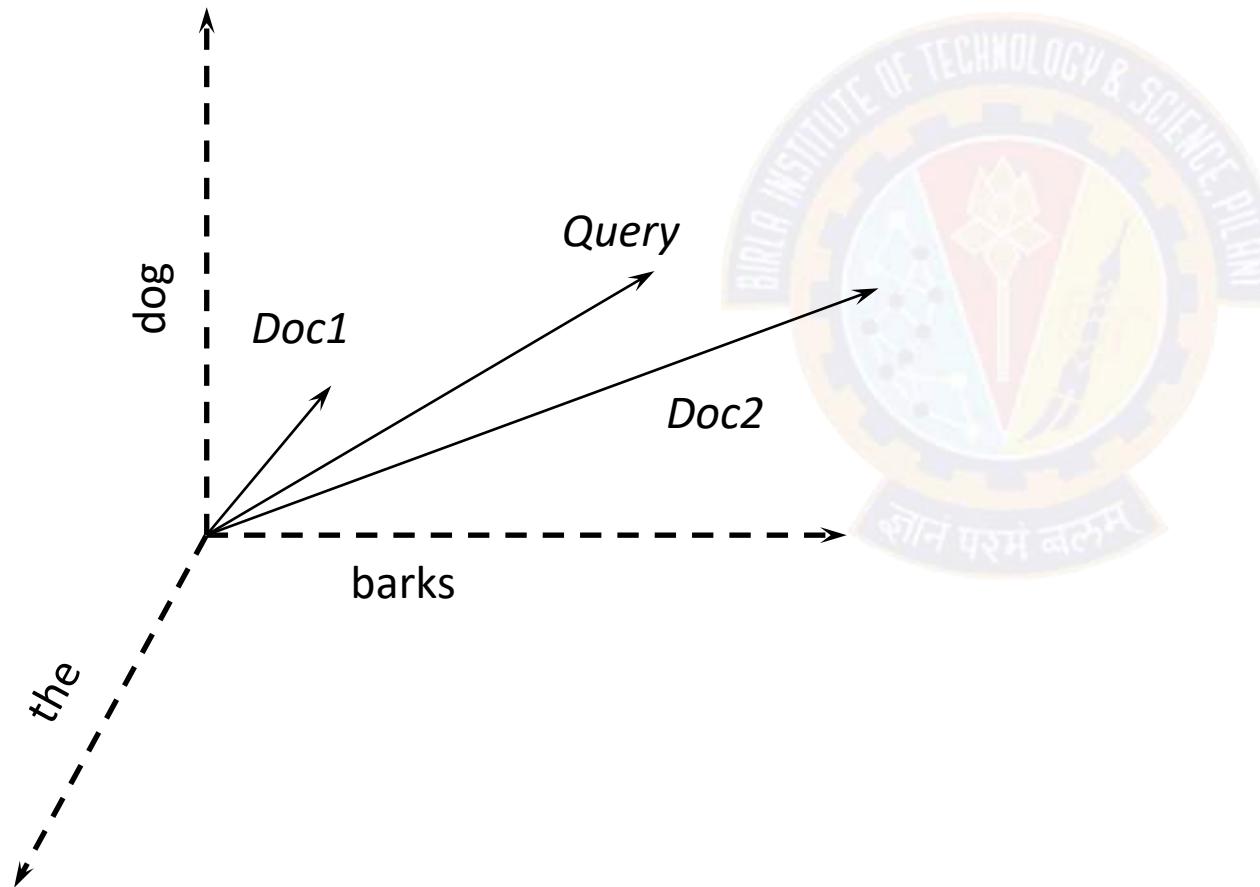
Vector Space Representation from linear algebra perspective

- Formally, a vector space is defined by a set of linearly independent basis vectors.
- Basis vectors:
 - correspond to the *dimensions* or *directions* in the vector space;
 - determine what can be described in the vector space; and
 - must be *orthogonal*, or *linearly independent*, i.e. a value along one dimension implies nothing about a value along another.



Vector Coefficients

- How to represent the documents and queries?



Doc1: the dog barks <1 1 1>

Doc2: the dog dog barks barks barks <1 2 3>

Query: the dog dog barks barks <1 2 2>

Vector Coefficients

- The coefficients (vector elements, term weights) represent term presence, importance, or “representativeness”
- The vector space model does not specify how to set term weights.
- Commonly used coefficients are :
 - Raw term frequency(tf)
 - Term Frequency – Inverse Document Frequency (TF-Idf)

Vector Space Similarity

Sim(X,Y)

Inner product

(# nonzero dimensions)

Dice coefficient

(Length normalized

Inner Product)

Cosine coefficient

(like Dice, but lower
penalty with diff # features)

Jaccard coefficient

(like Dice, but penalizes
low overlap cases)

Binary Term Vectors

$$|X \cap Y|$$

$$\frac{2|X \cap Y|}{|X| + |Y|}$$

$$\frac{|X \cap Y|}{\sqrt{|X|} \sqrt{|Y|}}$$

$$\frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

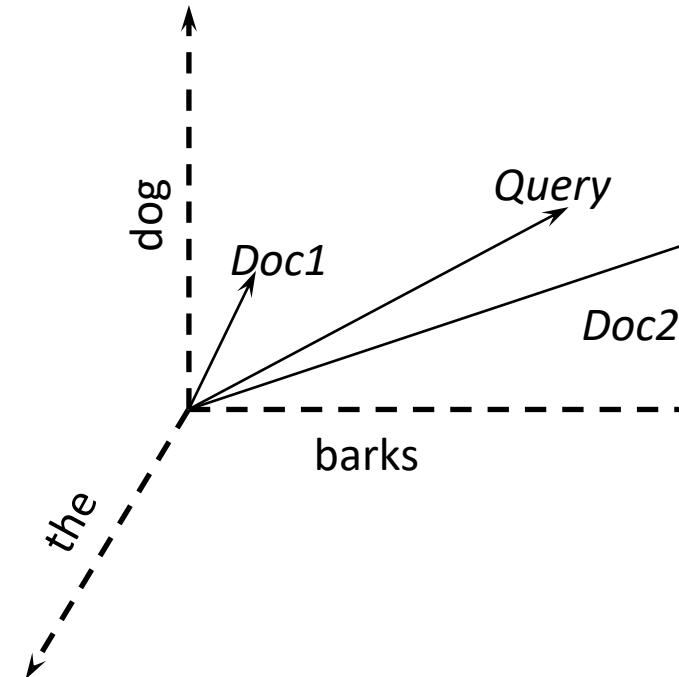
Weighted Term Vectors

$$\sum x_i \cdot y_i$$

$$\frac{2 \sum x_i y_i}{\sum x_i^2 + \sum y_i^2}$$

$$\frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}}$$

$$\frac{\sum x_i \cdot y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i \cdot y_i}$$





Thank You!

In our next session: Vector Space Model



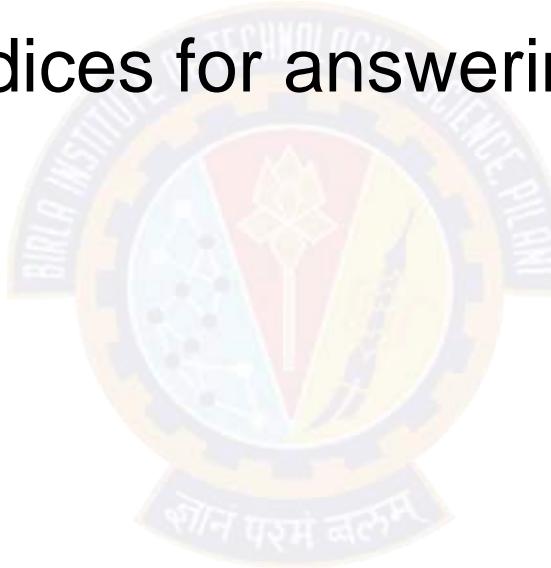
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Wildcard Queries

Prof. Aruna Malapati

Learning objectives

- List and define wildcard queries
- Identify appropriate indices for answering wildcard queries



Wildcard queries

- Trailing wildcard query

- Ex: a*

- Leading wildcard query

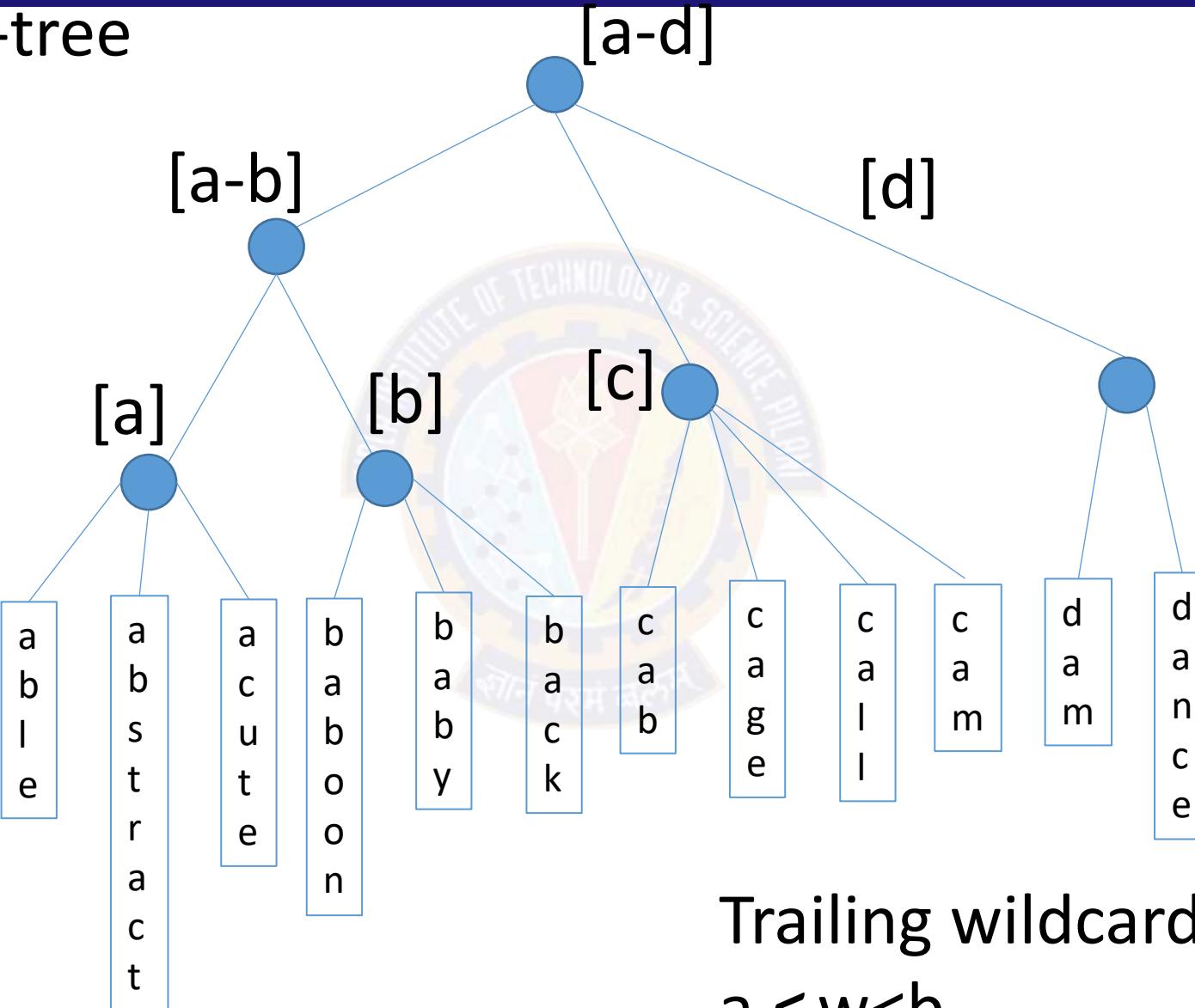
- Ex *a



Trailing wildcard query

able
abstract
acute
baboon
baby
back
cab
cage
call
cam
dam
dance

[2-4] B-tree



Trailing wildcard query: a^*
 $a \leq w < b$

Wildcard queries

Trailing wild card queries

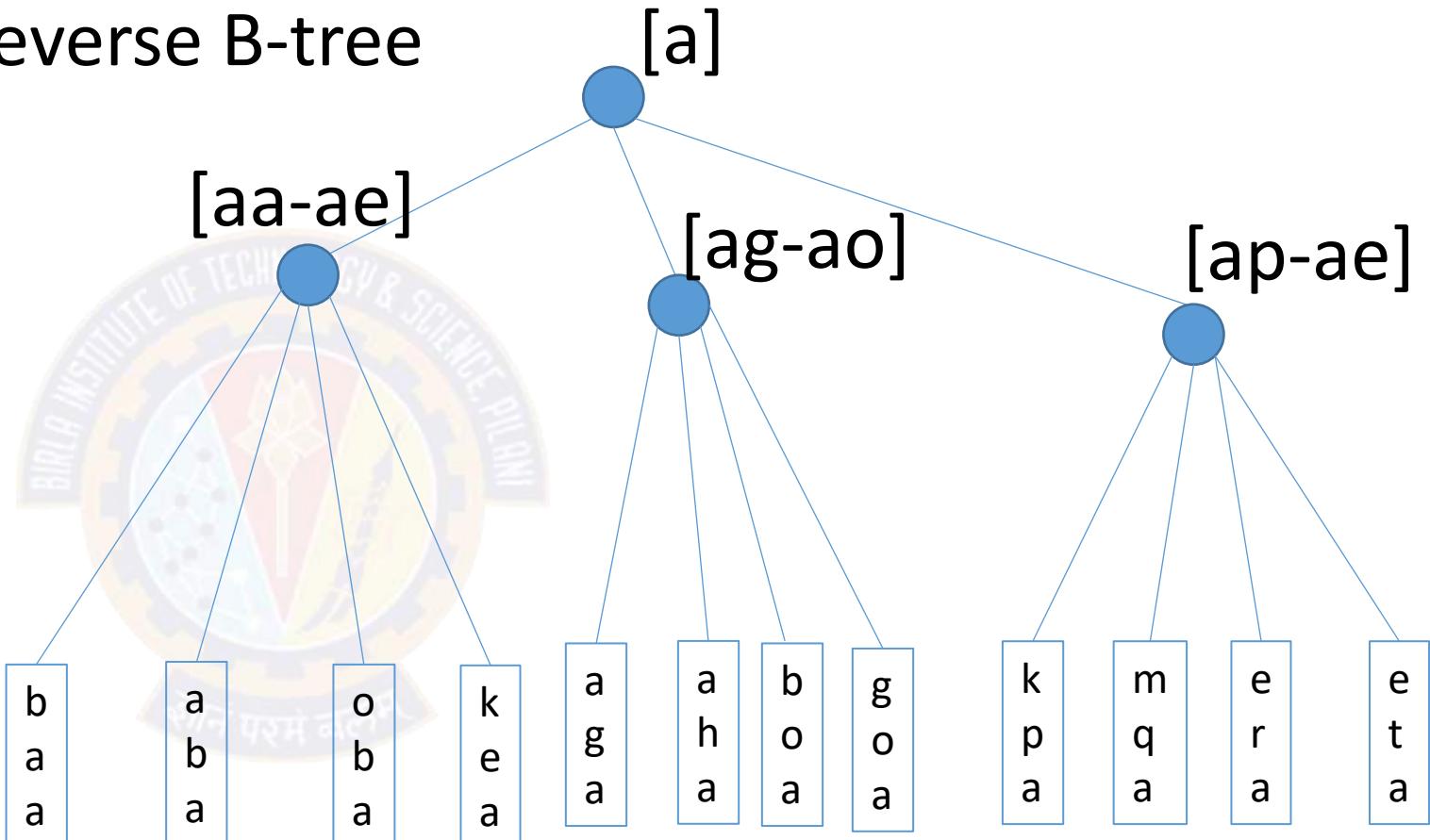
- ca* (e.g cab,cal,cam,can,cap,car,cat)
- Walk down the tree following c,a
- Retrieve all words w such that: $ca \leq w < cb$ (i.e. all the words having prefix “ca”)
- Let set of these terms be W.
- Use inverted index to retrieve documents containing terms in W

Leading wildcard queries

aba	aab
aga	aba
aha	abo
baa	ae
boa	ag
era	ah
eta	ao
goa	aog
kea	apk
kpa	aqm
mqa	are
oba	ate

Reverse
the
Terms
and
sort

[2-4] Reverse B-tree



Trailing wildcard query: *oa
 $ao \leq w < ab$



Thank You!

In our next session: K-Gram Index



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Ranked Retrieval

Prof. Aruna Malapati

Learning objectives

- Define Ranked Retrieval and vector space model
- Relate vector notation to the documents and queries
- Types of vector coefficients
- Compute similarity between query and documents

Ranked Retrieval

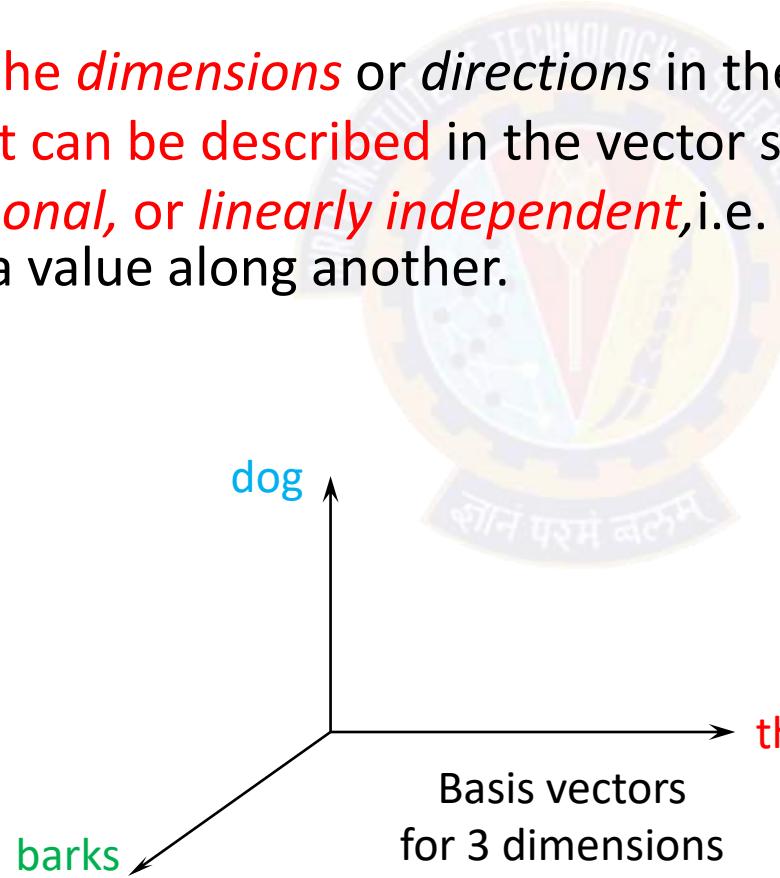
- The user enters a **free text query** and we would like to return the results in a ranked order.
- Hence we need a **scoring scheme** to compute the relevance of the document to the user query.
- A simple scoring scheme in the range [0-1].

Vector Space Model

- Any text object can be represented by a term vector
 - Examples: Documents, queries, sentences,
- Similarity is determined by distance in a vector space
 - Example: The cosine of the angle between the vectors

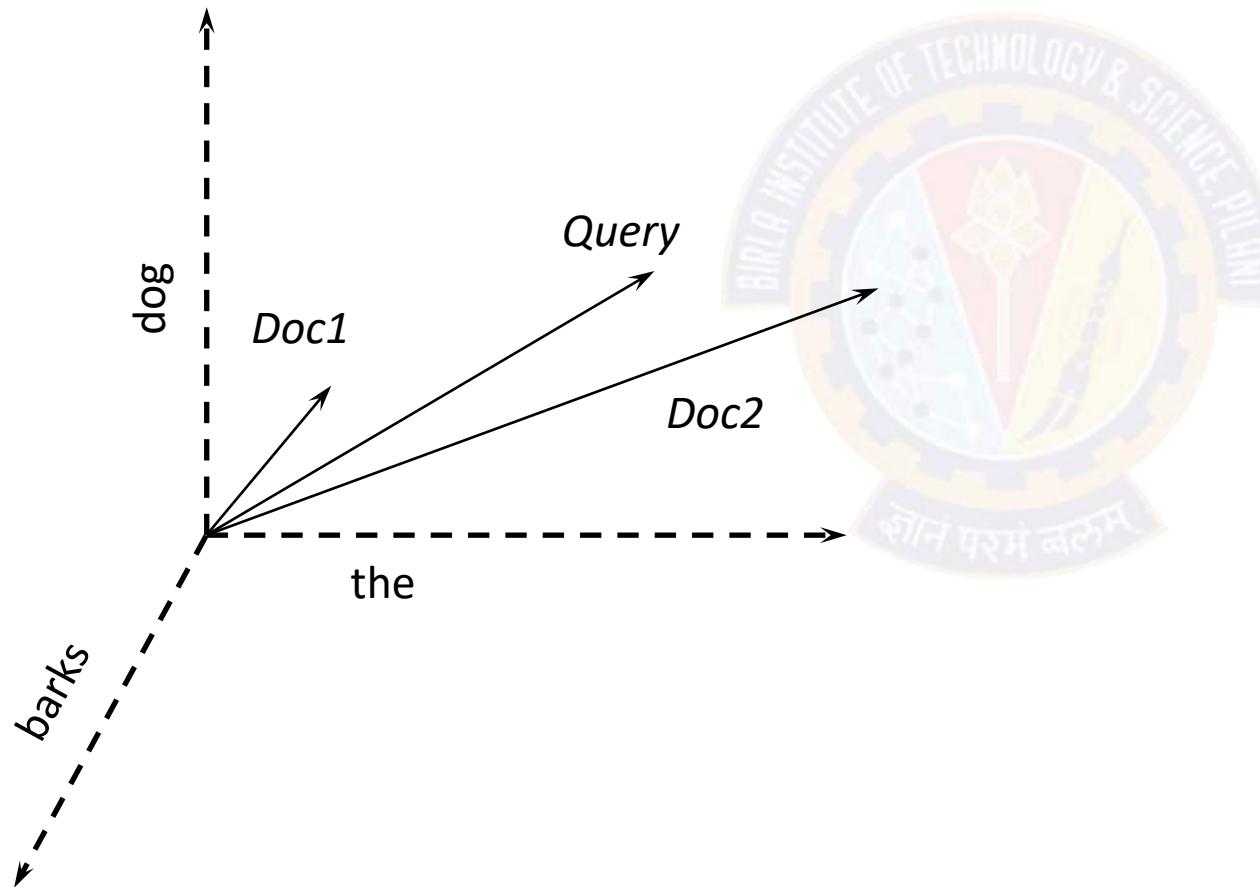
Vector Space Representation from linear algebra perspective

- Formally, a vector space is defined by a set of linearly independent basis vectors.
- Basis vectors:
 - correspond to the *dimensions* or *directions* in the vector space;
 - determine what can be described in the vector space; and
 - must be *orthogonal*, or *linearly independent*, i.e. a value along one dimension implies nothing about a value along another.



Vector Coefficients

- How to represent the documents and queries?



Doc1: the dog barks <1 1 1>

Doc2: the dog dog barks barks barks <1 2 3>

Query: the dog dog barks barks <1 2 2>

Vector Coefficients

- The coefficients (vector elements, term weights) represent term presence, importance, or “representativeness”
- The vector space model does not specify how to set term weights.
- Commonly used coefficients are :
 - Raw term frequency(tf)
 - Term Frequency – Inverse Document Frequency (TF-IDF)
 - Collection Frequency(CF)

Vector Space Similarity

Sim(X,Y)

Inner product

Dice coefficient

Cosine coefficient

Jaccard coefficient

Weighted Term Vectors

$$\sum x_i \cdot y_i$$

$$\frac{2\sum x_i y_i}{\sum {x_i}^2 + \sum {y_i}^2}$$

$$\frac{\sum x_i \cdot y_i}{\sqrt{\sum {x_i}^2} \cdot \sqrt{\sum {y_i}^2}}$$

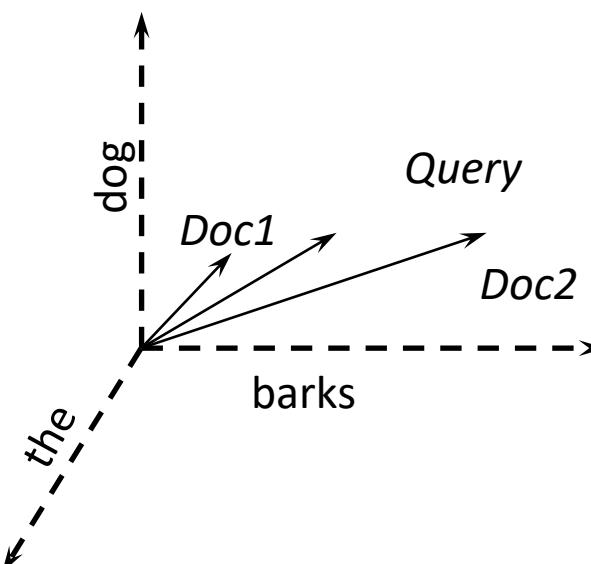
$$\frac{\sum x_i \cdot y_i}{\sum {x_i}^2 + \sum {y_i}^2 - \sum x_i \cdot y_i}$$

Assume that the query vector is denoted by X and Y is the document vector.

Example of similarity with vector coefficient as term frequency

	Term Weights		
Query	1	2	2

	Term Weights		
Doc 1	1	1	1
Doc 2	1	2	3



Inner product $S(Q, D1) = 1*1+2*1+2*1 = 5$

Inner product $S(Q, D2) = 1*1+2*2+2*3 = 11$

Dice coefficient $S(Q, D1) = \frac{2(1*1 + 2*1 + 2*1)}{(1^2 + 2^2 + 2^2) + (1^2 + 1^2 + 1^2)}$

Dice coefficient $S(Q, D2) = \frac{2(1*1 + 2*2 + 2*3)}{(1^2 + 2^2 + 2^2) + (1^2 + 2^2 + 3^2)}$

Cosine coefficient $S(Q, D1) = \frac{1*1 + 2*1 + 2*1}{\sqrt{(1^2 + 2^2 + 2^2)} * \sqrt{(1^2 + 1^2 + 1^2)}}$

Cosine coefficient $S(Q, D2) = \frac{1*1 + 2*2 + 2*3}{\sqrt{(1^2 + 2^2 + 2^2)} * \sqrt{(1^2 + 2^2 + 3^2)}}$

Advantages and Disadvantages

- Simplicity: Easy to implement
- Ability to incorporate any kind of term weights
- Can measure similarities between almost anything:
 - documents and queries, documents and documents, queries and queries, sentences and sentences, etc.
- The vector space model is the most **popular retrieval model (today)**
- Assumes independence relationship among terms.
- The weighting is subjective.



Thank You!

In our next session: Ranked Retrieval using TF-IDF



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Ranked retrieval

Prof. Aruna Malapati

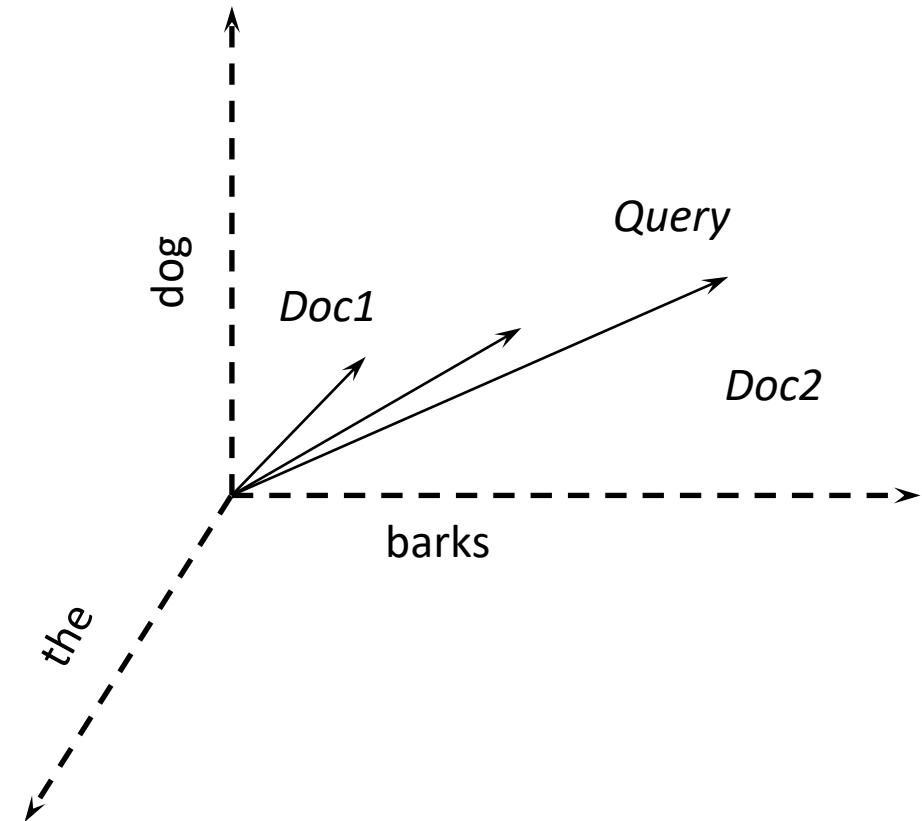
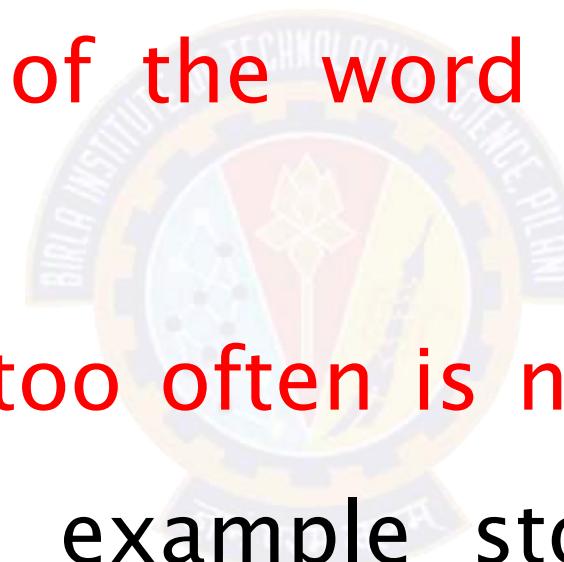
Learning objectives

- List and problems with Term Frequency as the vector co-efficients
- Define and apply TF-IDF vector co-efficients



Limitations of Term frequency and cosine angle

- Term Frequency represents the relative importance of the word in the document.
- A term appearing too often is not very important. For example stop words like a,an,the,etc...



TF Weighting

- The log frequency weight of term t in document d is

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d}, & \text{if } \text{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4$, etc.

- Score for a document-query pair: sum over terms t in both q and d :

$$\text{Score}(q,d) = \sum_{t \in q \cap d} W_{t,d}$$

Inverse Document Frequency (IDF)

- The presence of a rare word is more important than a stop word.
- The rarity of a term is quantified using IDF.
- df_t is the document_frequency of t : the number of documents that contain t
- We define the idf (inverse document frequency) of t by

$$\text{idf}_t = \log_{10} (N/\text{df}_t)$$

tf-idf weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \times \log_{10}(N / \text{df}_t)$$

- Best known weighting scheme in information retrieval
- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

tf-idf weighting for document and queries

Example :D1: The restaurants are in the city.

D2: The resorts are in the outskirts.

V={are, city, in outskirts, resorts, restaurants, the}

Q= {resorts,outskirts}

$$\text{Score}(q, d) = \sum_{t \in q \cap d} \text{tf.idf}_{t,d}$$

	TF-D1	IDF-D1	TF-IDF(D1)	TF-D2	IDF-D2	TF-IDF(D2)	TF-Query	IDF-Query	TF-IDF Query
are	$1+\log 1=1$	$\log 2/2=0$	0	$1+\log 1=1$	$\log 2/2=0$	0	0	$\log 2/2=0$	0
city	$1+\log 1=1$	$\log 2/1=0.3$	0.3	0	$\log 2/1=0.3$	0	0	$\log 2/1=0.3$	0
in	$1+\log 1=1$	$\log 2/2=0$	0	$1+\log 1=1$	$\log 2/2=0$	0	0	$\log 2/2=0$	0
outskirts	0	$\log 2/1=0.3$	0	$1+\log 1=1$	$\log 2/1=0.3$	0.3	$1+\log 1=1$	$\log 2/1=0.3$	0.3
resorts	0	$\log 2/1=0.3$	0	$1+\log 1=1$	$\log 2/1=0.3$	0.3	$1+\log 1=1$	$\log 2/1=0.3$	0.3
restaurants	$1+\log 1=1$	$\log 2/1=0.3$	0.3	0	$\log 2/1=0.3$	0	0	$\log 2/1=0.3$	0
the	$1+\log 2=1.3$	$\log 2/2=0$	0	$1+\log 2=1.3$	$\log 2/2=0$	0	0	$\log 2/2=0$	0

tf-idf weighting has many variants

Term frequency	Document frequency	Normalization
n (natural) $tf_{t,d}$	n (no) 1	n (none) 1
I (logarithm) $1 + \log(tf_{t,d})$	t (idf) $\log \frac{N}{df_t}$	c (cosine) $\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented) $0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf) $\max\{0, \log \frac{N - df_t}{df_t}\}$	u (pivoted unique) $1/u$
b (boolean) $\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$		b (byte size) $1/CharLength^\alpha, \alpha < 1$
L (log ave) $\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$		

- SMART Notation: denotes the combination in use in an engine, with the notation *ddd.ddd*



Thank You!

In our next session: Computing scores



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

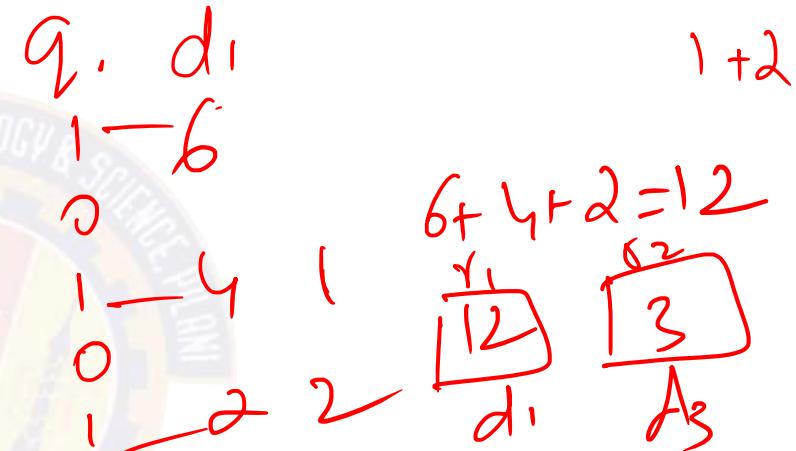
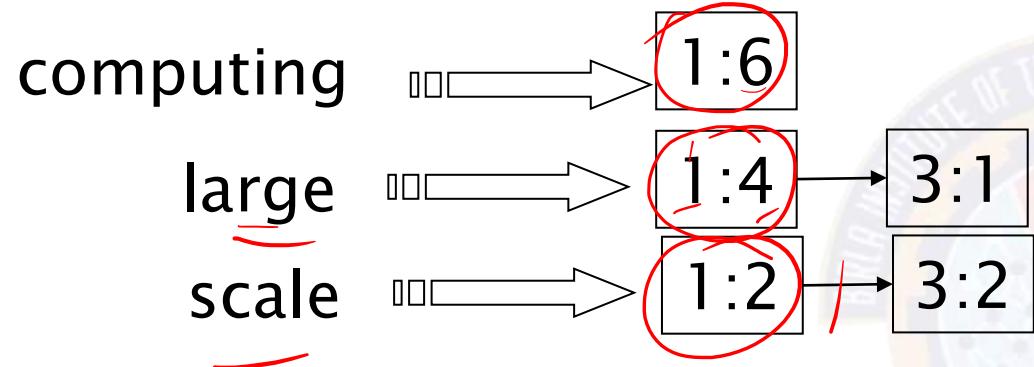
Ranked retrieval

Prof. Aruna Malapati

Document at a Time

Vocabulary {computing, data, large, mining, scale}

Query = <1 0 1 0 1>



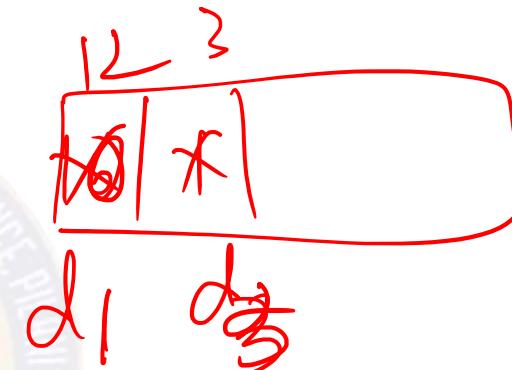
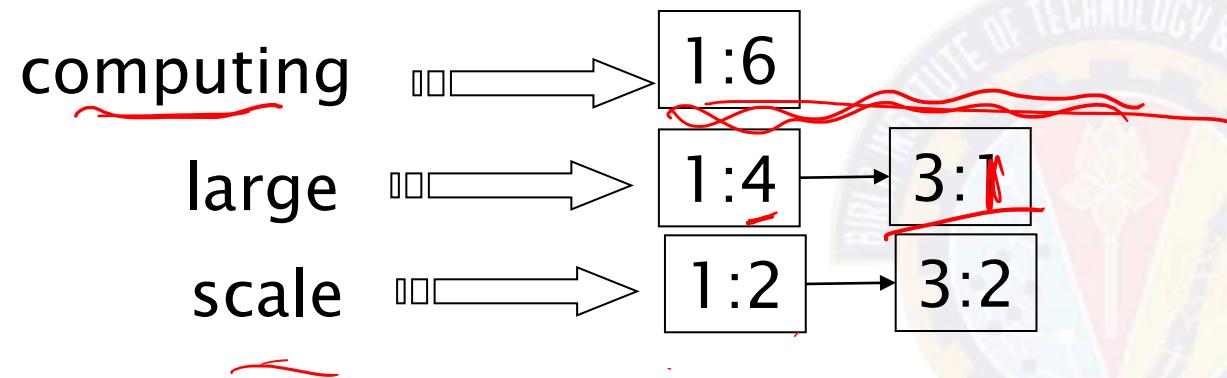
Scanning posting lists of computing, large, scale in parallel

1. Find the smallest docId, say d, among all lists
2. Compute the pointers for all postings whose docid= d
3. Forward the pointers for all postings whose docid = d
4. Repeat steps 1- 3 for next doc

Term at a time

Vocabulary {computing, data, large, mining, scale}

Query = $<1\ 0\ 1\ 0\ 1>$



- Scanning posting lists of computing, large, scale in one by one
- $\text{Sim}(q,d)$ is available after scanning the last term
- Accumulates partial score when each term is scanned



Thank You!

In our next session: Parts of Speech Tagging



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Parts of Speech Tagging

Prof. Aruna Malapati

Learning objectives

- Define the problem of Parts Of Speech tagging (POS)
- Challenges while POS tagging
- Introduce to the benchmarked PennTree dataset
- Applications
- Measuring performance of POS taggers
- Two approaches to POS tagging

Parts Of Speech Tagging (POS)

Aruna saw the saw.

NNP

VB

DT

NN

➤ Annotate each word in a sentence with a part of speech.

Sequence Data

- Till now, we assumed that the data instances are classified independently.
- More precisely, we assumed that the data is iid (identically and independently distributed)
- In many applications, the data arrives sequentially and the classes are correlated – E.g., weather prediction, speech recognition, activity recognition, etc..

What is the challenge in PoS Tagging?

- Tag ambiguous words
 - Solve the lexical ambiguities
 - The/DT wind/NN was/VB too/ADV strong/ADJ to/PRP
wind/VB the/DT sail/NN.
- Tag unknown words
 - The/DT rural/JJ Babbitt/??? who/WP bloviates/???
about/IN progress/NN and/CC growth/NN

Category of classes

- Open: vast number of new members
 - Nouns, Verbs, Adjectives, Adverbs
- Closed: small set of words
 - Determiners: a, an, the
 - Pronouns: she, he, i, you, we
 - Prepositions: on, under, over, near, by,...

Choosing a tagset

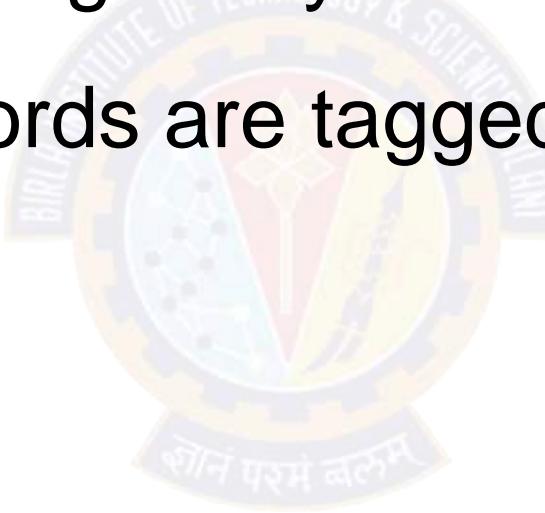
- Need to choose a standard set of tags to do POS tagging.
- Could pick very coarse tagset – N, V, Adj, Adv, Prep.
- More commonly used set is finer-grained.
 - Penn TreeBank II tagset has 36 word tags
 - PRP, PRPS, VBG, VBD, JJR, JJS ...

Penn TreeBank PoS Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	<i>and, but, or</i>	SYM	symbol	<i>+, %, &</i>
CD	cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	determiner	<i>a, the</i>	UH	interjection	<i>ah, oops</i>
EX	existential 'there'	<i>there</i>	VB	verb, base form	<i>eat</i>
FW	foreign word	<i>mea culpa</i>	VBD	verb, past tense	<i>ate</i>
IN	preposition/sub-conj	<i>of, in, by</i>	VBG	verb, gerund	<i>eating</i>
JJ	adjective	<i>yellow</i>	VBN	verb, past participle	<i>eaten</i>
JJR	adj., comparative	<i>bigger</i>	VBP	verb, non-3sg pres	<i>eat</i>
JJS	adj., superlative	<i>wildest</i>	VBZ	verb, 3sg pres	<i>eats</i>
LS	list item marker	<i>1, 2, One</i>	WDT	wh-determiner	<i>which, that</i>
MD	modal	<i>can, should</i>	WP	wh-pronoun	<i>what, who</i>
NN	noun, sing. or mass	<i>llama</i>	WPS	possessive wh-	<i>whose</i>
NNS	noun, plural	<i>llamas</i>	WRB	wh-adverb	<i>how, where</i>
NNP	proper noun, singular	<i>IBM</i>	\$	dollar sign	<i>\$</i>
NNPS	proper noun, plural	<i>Carolinas</i>	#	pound sign	<i>#</i>
PDT	predeterminer	<i>all, both</i>	"	left quote	<i>' or "</i>
POS	possessive ending	<i>'s</i>	"	right quote	<i>' or "</i>
PRP	personal pronoun	<i>I, you, he</i>	(left parenthesis	<i>[, (, {, <</i>
PRP\$	possessive pronoun	<i>your, one's</i>)	right parenthesis	<i>],), }, ></i>
RB	adverb	<i>quickly, never</i>	,	comma	<i>,</i>
RBR	adverb, comparative	<i>faster</i>	.	sentence-final punc	<i>. ! ?</i>
RBS	adverb, superlative	<i>fastest</i>	:	mid-sentence punc	<i>::, ... --</i>
RP	particle	<i>up, off</i>			

POS Tagging performance evaluation

- Percentage of tags predicted correctly.
- Baseline approach: Tag every word with its most common tag and rest of the words are tagged as Noun.



Applications of POS tagging

- Text to speech conversion
- Useful as a preprocessing step of parsing



Two Methods for PoS Tagging

- Rule-based systems
- Statistical sequence models
 - Hidden Markov Models





Thank You!

In our next session: Stochastic Language Models



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stochastic Language Models

Prof. Aruna Malapati

Learning objectives

- Express the need for language models
- Jargons
- Formulate a naïve language model



Language Modelling

$$V = \{A, B, C\}$$

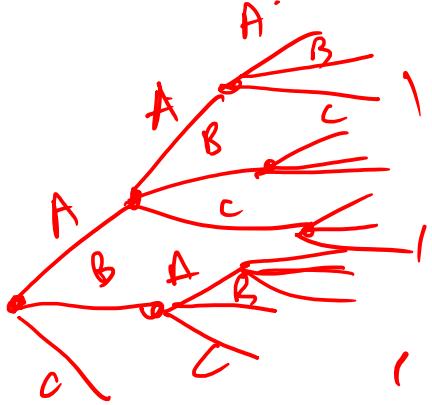
length of start node = 3

A = the

B = dog

C = banks

D = boy

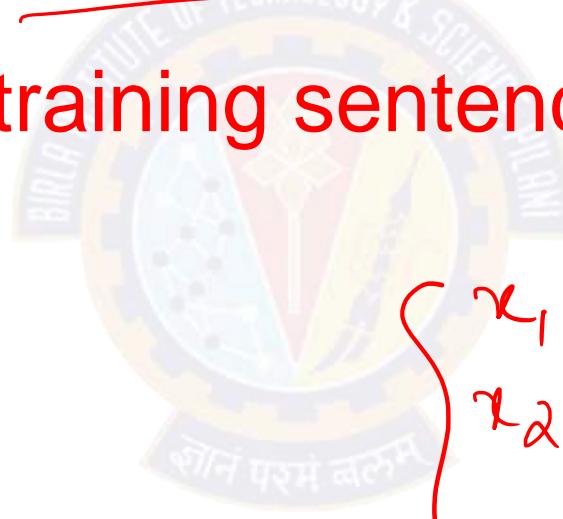


the boy banks

- The task of a language model is to **express the restrictions imposed** on the way in which words can be combined to form sentences.

Stochastic Language Model

- The task of a stochastic language model is to provide estimates of the prior probability of the sentence generation using the training sentences.



$$\left. \begin{array}{l} x_1 = w_1, w_2, \dots, w_n \\ x_2 = w_n, w_{12}, w_6, \dots, w_n \\ \vdots \\ \vdots \end{array} \right\}$$

Jargons

- Corpus: - set of training sentences.
- Vocabulary: finite set of words used in writing sentences.
 - For example $V = \{\text{the}, \text{dog}, \text{barks}, \text{cat}, \text{smiles}, \text{boy}, \text{play}..\}$
- V' = Set of all possible sentences in this language.

the dog barks STOP(well formed sentence)

the cat cat STOP(ill formed sentence)

the the the STOP(ill formed sentence)

the cat smiles STOP(well formed sentence)

STOP

Stochastic Language Model (Contd..)

- Assume that we have a training sentences in English.
- These could be easily collected from online newspapers or webpages.
- Given these training samples the task of Language Model is learn a distribution \underline{P} over sentences in our language.
- P is going to be a function which must satisfy the following two constraints

$$\begin{aligned}1) \text{ For any sentence } s \quad \forall s \in V^+ \quad P(s) > 0 \\2) \quad \sum_{s \in V^+} P(s) = 1\end{aligned}$$

Stochastic Language Model (Contd..)

21
1
Xm

Training sentences



Function P

$P(\text{the dog barks}) = 2 \times 10^{-6}$
 $\underline{P(\text{the boy smiles}) = 3 \times 10^{-2}}$
 $\underline{P(\text{the saw saw}) = 10^{-20}}$
...

- The task of function P is to **assign probability to every sentence in the language.**
- We would prefer a good **language model** to assign high probability to a sentence which occurs in English.

A naïve language model

- Given N training sentences, the task is to learn a distribution P over sentences in the language.
- For any sentence $\underline{w_1, w_2, \dots, w_n}$ let $c(w_1, w_2, \dots, w_n)$ denote the number of times this sentence is seen in the corpus.

$$P(S) = \frac{c(w_1, \dots, w_n)}{N}$$



Thank You!

In our next session: Types of Language Models



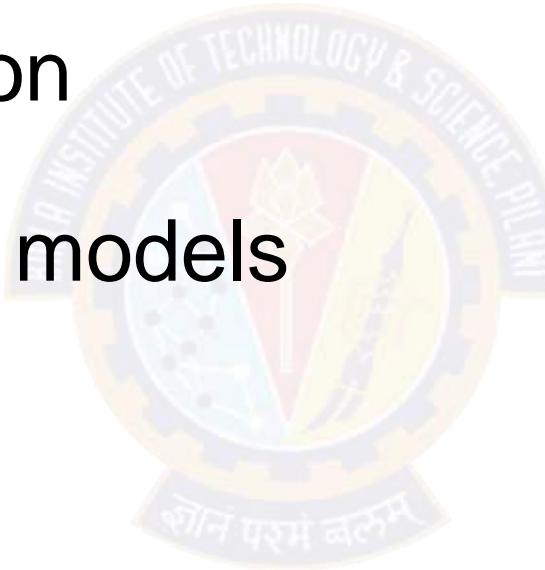
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stochastic Language Models

Prof. Aruna Malapati

Learning objectives

- Express the Language model using the joint probability distribution
- Types of Language models



Markov Process



First Order Markov Process



Second Order Markov Process



How to model variable length sequences?





Thank You!

In our next session: Hidden Markov Model



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Generative model for POS tagging

Prof. Aruna Malapati

Discriminative classification models

- Given a set of training examples $\langle x_i, y_i \rangle$ for $i=1..m$ where each x_i is the input vector and y_i is the class label.
- The modeling task as to **learn a function f mapping the input x to labels $f(x)$.**
- The classification models you have learnt till now are discriminative models where each sample input was considered independent of each other.

Generative model for sequence labelling

- Given a set of training examples $\langle x_i, y_i \rangle$ for $i=1..m$ where each x_i is the input vector and y_i is the class label.
- In the POS tagging problem we have
 - $x_1 = \text{the dog barks}, y_1 = \text{DT NN VB}$
 - $x_2 = \text{the boy smiles}, y_2 = \text{DT NN VB}$
 -
- The task is to learn a function f that maps input x to its corresponding labels $f(x)$.

Generative model for sequence labelling

Using Bayes rule

$$P(t_1, \dots, t_n | w_1, \dots, w_n) = \frac{P(t_1, \dots, t_n)P(w_1, \dots, w_n | t_1, \dots, t_n)}{P(w_1, \dots, w_n)}$$



Submodels:

1. Prior: $P(t_1, \dots, t_n)$
2. Likelihood: $P(w_1, \dots, w_n | t_1, \dots, t_n)$
3. Marginal: $P(w_1, \dots, w_n)$ – can be ignored in argmax search

Markov Assumption

➤ Context model (prior)

$$P(t_1, \dots, t_n) = \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1})$$

➤ Lexical model (likelihood)

$$P(w_1, \dots, w_n | t_1, \dots, t_n) = \prod_{i=1}^n P(w_i | t_i)$$

Model Parameters

- Contextual probabilities : $P(t_i|t_{i-k}, \dots, t_{i-1})$
- Lexical probabilities : $P(w_i|t_i)$
- We can estimate these probabilities from a tagged corpus:

$$\hat{P}_{\text{MLE}}(w_i|t_i) = \frac{c(w_i, t_i)}{c(t_i)} \quad \hat{P}_{\text{MLE}}(t_i|t_{i-k}, \dots, t_{i-1}) = \frac{c(t_{i-k}, \dots, t_{i-1}, t_i)}{c(t_{i-k}, \dots, t_{i-1})}$$

Computing Probabilities

- The probability of a tagging:

$$P(t_1, \dots, t_n, w_1, \dots, w_n) = \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1}) P(w_i | t_i)$$



- Finding the most probable tagging:

$$\operatorname{argmax}_{t_1, \dots, t_n} \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1}) P(w_i | t_i)$$

Two fundamental problems in HMM

- Decoding:
 - How do we compute the best tag sequence given parameters?
- Learning:
 - How do we estimate the parameters?



Example

- Given a sentence of length 3, * * the dog barks STOP and the tag sequence * * DT NN VB * then
- $P(w_1 w_2 w_3, y_1, y_2, y_3) = T(DT|*, *) \times T(NN|*, DT) \times T(VB|DT NN) \times T(STOP|NN VB) \times E(*|*) \times E(*|*) E(\text{the}|DT) \times E(\text{dog}|NN) \times E(\text{barks}|VB) \times E(STOP|*)$
- We can also define $y_{-1} = ^*$ and $y_0 = ^*$ as special symbols.



Thank You!

In our next session: Hidden Markov Model for POS Tagging



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Parts of Speech Tagging using HMM

Prof. Aruna Malapati

Learning objectives

- Define Markov chains
- Define Hidden Markov Model



Markov Chains

- A Markov chain is a model that tells us something about the **probabilities of sequences of random variables**, states, each of which can take on values from some set.
- A Markov Model is a finite state machine with probabilistic state transitions.
- Markov assumption that next state only depends on the current state and independent of previous history.

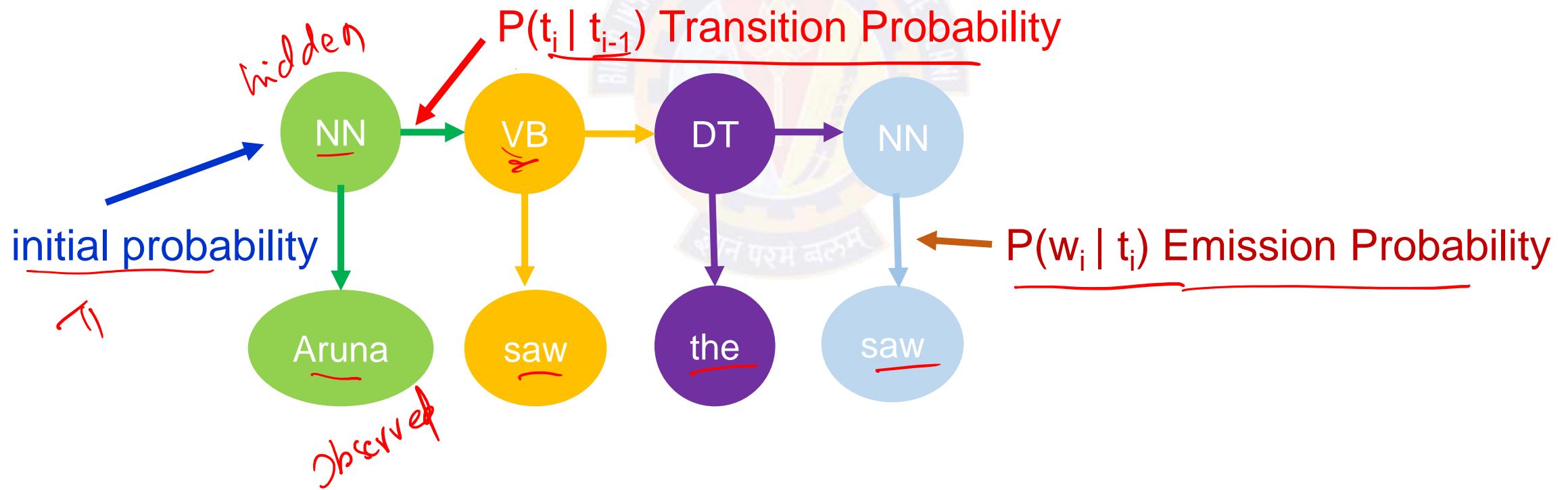
Markov Chain

- Formally, a Markov chain is specified by the following components:

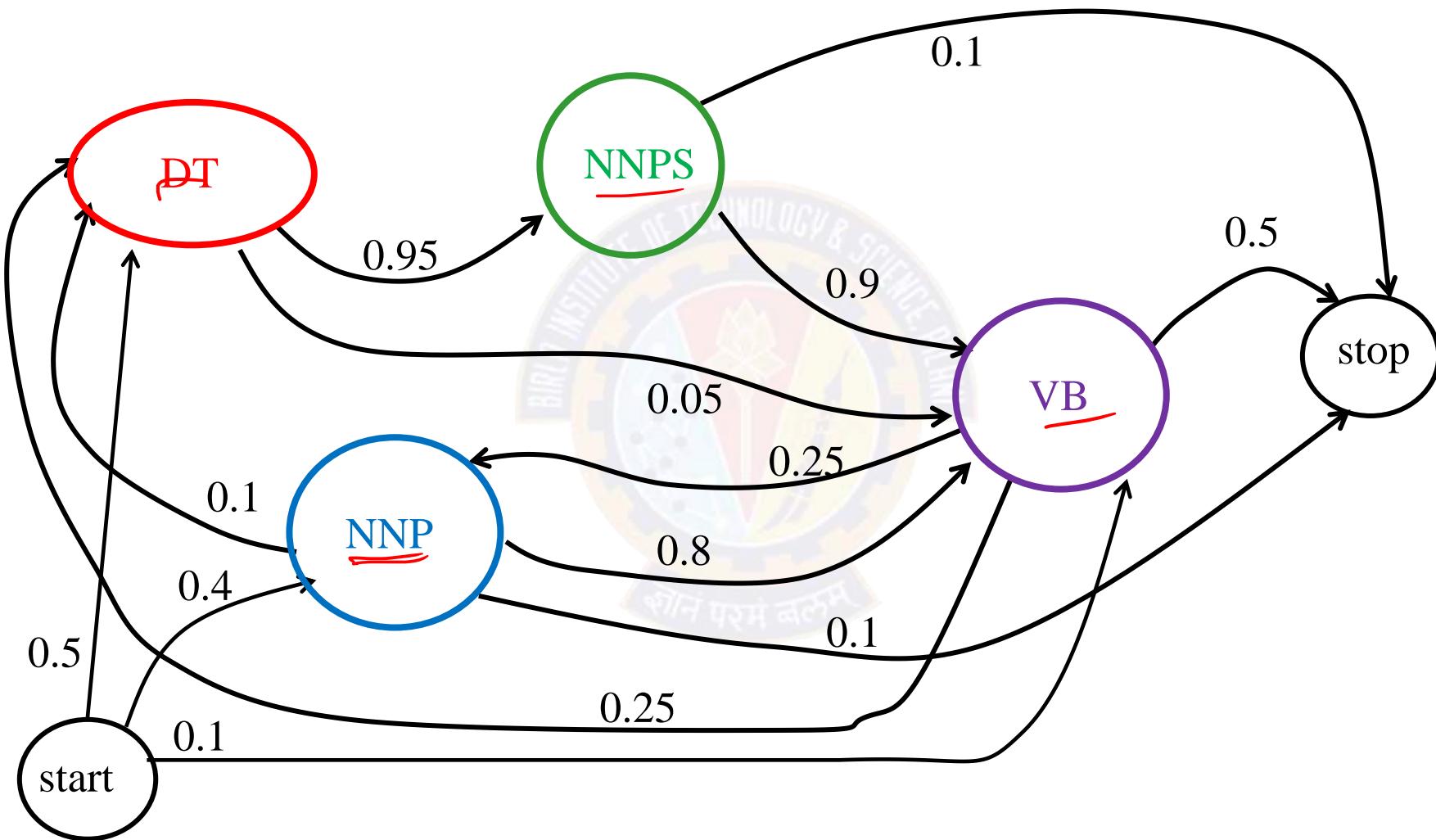
$\underline{Q} = q_1 q_2 \dots q_N$ a set of N states	A set of N states
$\underline{A} = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$	A transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, $\sum_{j=1}^n a_{ij} = 1 \forall i$
$\underline{\pi} = \pi_1, \pi_2, \pi_3, \dots, \pi_n$	An initial probability distribution over states. π_i is the probability that the Markov chain will start in state i . Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. $\sum_{i=1}^n \pi_i = 1$

The Hidden Markov Model

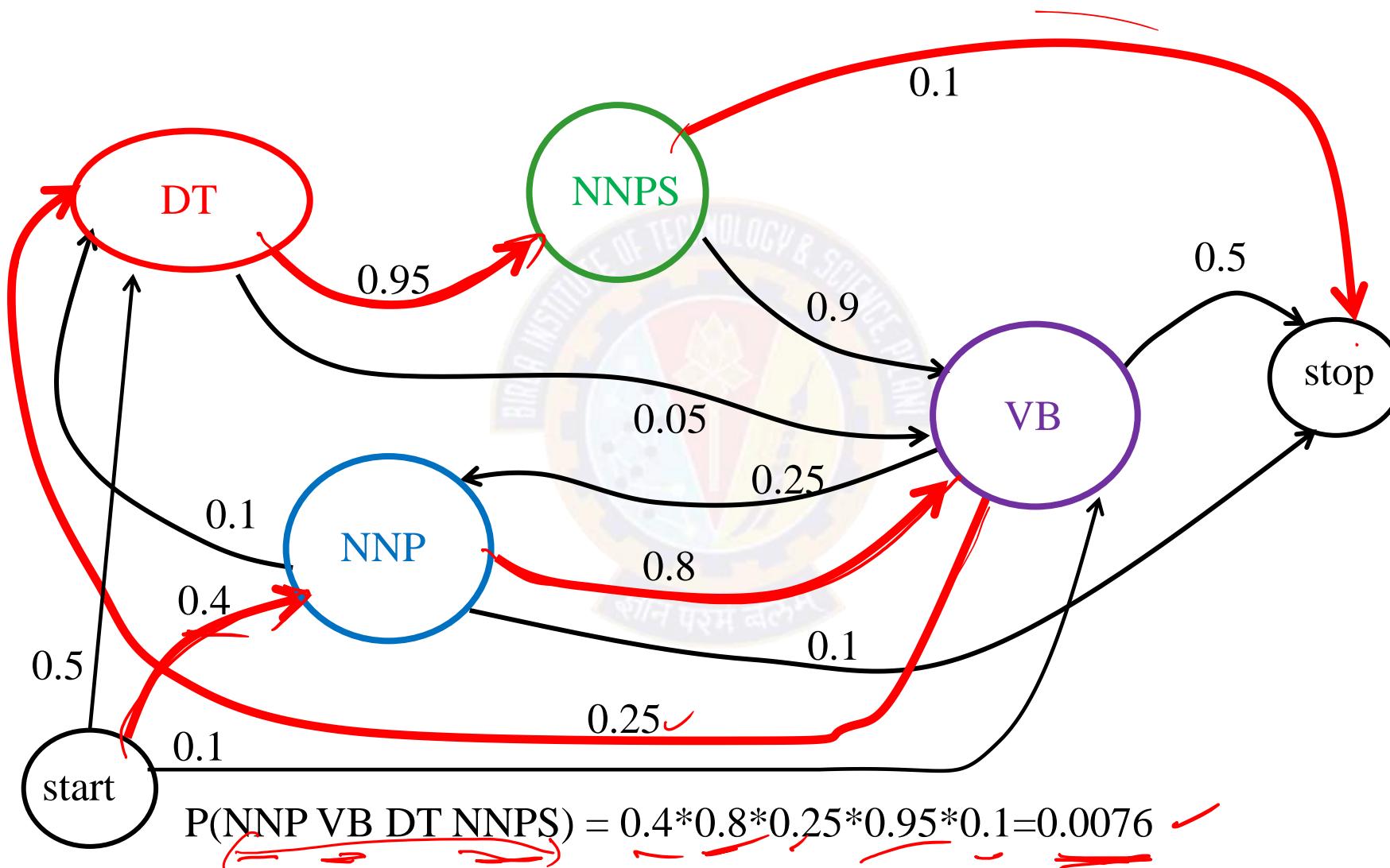
- In many cases the events we are interested in are hidden: we don't observe them directly.



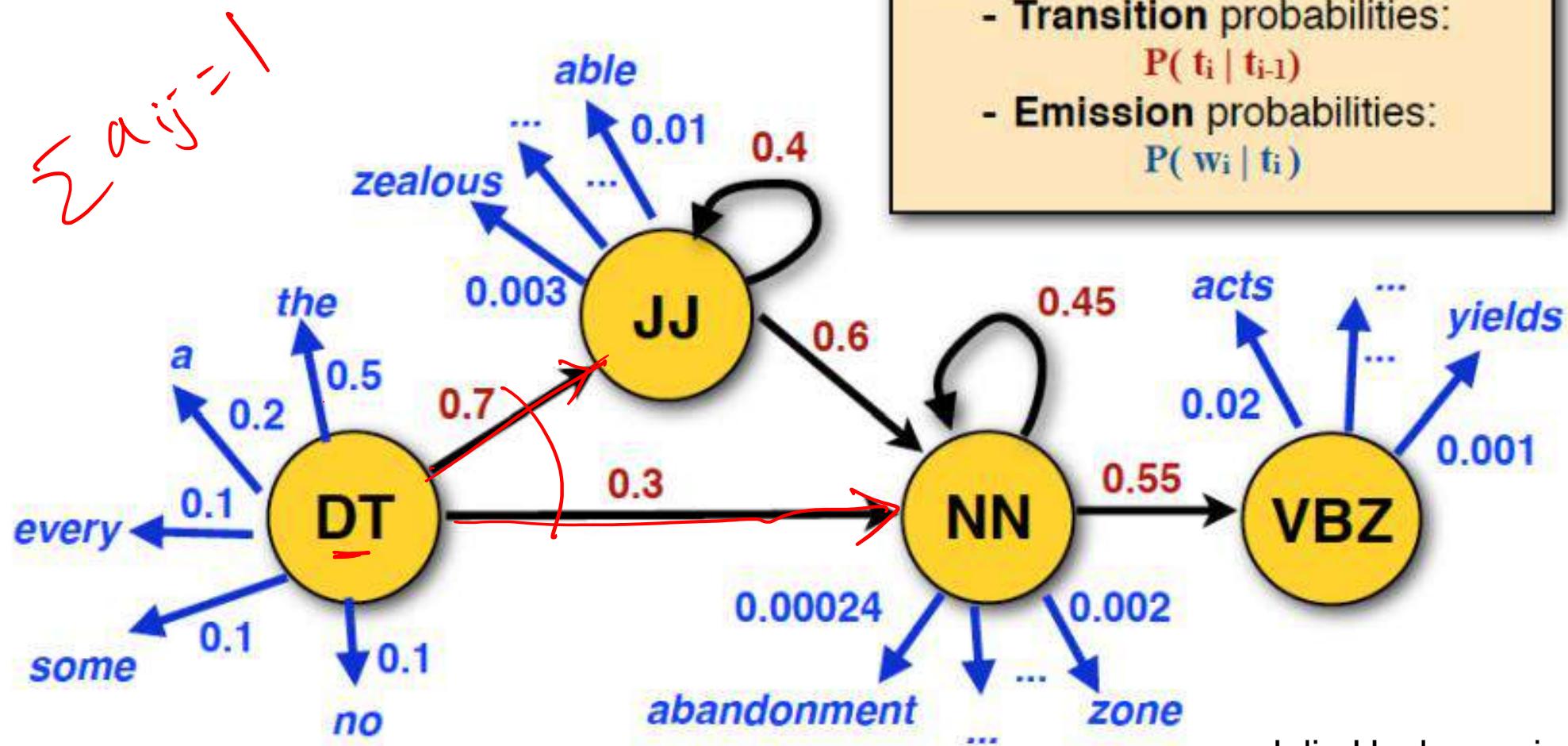
HMMs as probabilistic FSA



HMMs as probabilistic FSA



HMMs as probabilistic FSA



An HMM defines

- **Transition** probabilities:

$$P(t_i | t_{i-1})$$

- **Emission** probabilities:

$$P(w_i | t_i)$$

Hidden Markov Models (formal)

- States $T = t_1, t_2 \dots t_N$;
- Observations $W = w_1, w_2 \dots w_N$;
 - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \dots v_V\}$
- Transition probabilities
 - Transition probability matrix $A = \{a_{ij}\}$
$$a_{ij} = P(t_i = j | t_{i-1} = i) \quad 1 \leq i, j \leq N$$
- Observation likelihoods
 - Output probability matrix $B = \{b_i(k)\}$
$$b_i(k) = P(w_i = v_k | t_i = i) \quad ?(v_k | i)$$
- Special initial probability vector π $\pi_i = P(t_1 = i) \quad 1 \leq i \leq N$

HMM tagging as decoding

➤ HMM model contains hidden variables, the task of determining the hidden variables sequence corresponding to the sequence of observations decoding is called decoding.

➤ Find our best estimate of the sequence that maximizes

$$\underbrace{P(t_1 \dots t_n | w_1 \dots w_n)}$$

$$\hat{t}_1^n = \underbrace{\operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n)}$$

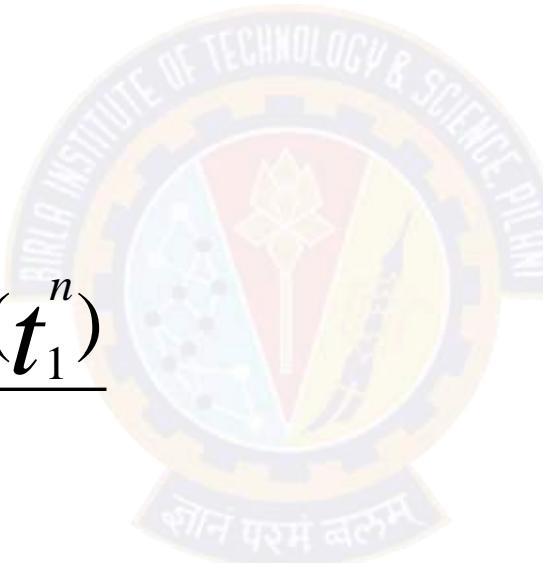
HMM tagging as decoding (Contd...)

Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$



Drop denominator since is the same for all tags we consider

HMM tagging as decoding (Contd...)

➤ A1: P(w) depends only on its own POS

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^n P(w_i | t_i)$$

➤ A2: P(t) depends only on P(t-1)

$$P(t_1^n) \approx \prod_{i=1}^n P(t_i | t_{i-1})$$

likelihood prior

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \overbrace{P(w_1^n | t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

HMM tagging as decoding (Contd...)

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n) \approx \boxed{\operatorname{argmax}_{t_1^n}} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$

Now we have two probabilities to calculate:

- Probability of a word occurring given its tag
- Probability of a tag occurring given a previous tag
- We can calculate each of these from a POS-tagged corpus



Thank You!

In our next session: HMM Example



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

HMM Example

Prof. Aruna Malapati

Learning Objectives

- Construction of Transaction and Emission Matrices



Example

Emission Matrix

	*	DT	NNS	VB	NN	IN	STOP
*	1						
the			3/4				
employees				3/4			
pass					2/4		
an			1/4				
exam						1	
wait				1/4			
for						1	
employers			1/4				
fire				1/4			
.					1		

Tag Translation Matrix

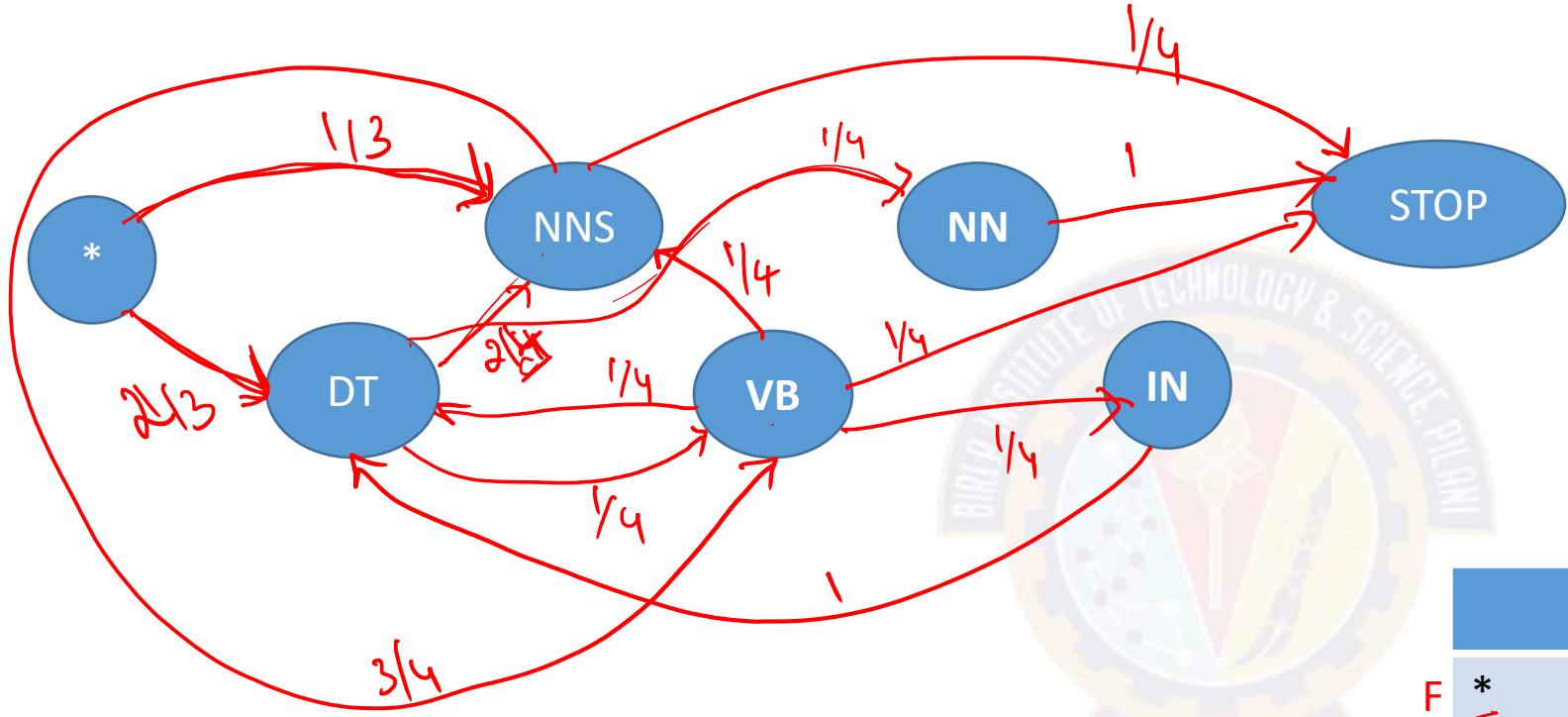
	*	<u>DT</u>	<u>NNS</u>	<u>VB</u>	<u>NN</u>	<u>IN</u>	<u>STOP</u>
F	*	<u>2/3</u>	<u>1/3</u>				
R	<u>DT</u>		<u>2/4</u>	<u>1/4</u>	<u>1/4</u>		
I					<u>3/4</u>		<u>1/4</u>
S	<u>NNS</u>						
T	<u>VB</u>	<u>1/4</u>	<u>1/4</u>			<u>1/4</u>	<u>1/4</u>
A	<u>NN</u>						1
A	<u>IN</u>	1					
G	<u>STOP</u>						

S1: the Employees pass an exam .
T1: DT NNS VB DT NN STOP

S2: the employees wait for the pass .
T2: DT NNS VB IN DT VB STOP

S3: employers fire employees .
T3: NNS VB NNS STOP

Transition diagram



$$\frac{P(DT | \cancel{*} \cancel{*})}{P(NNS | \cancel{DT} \cancel{DT})}$$

Tag Translation Matrix

	*	DT	NNS	VB	NN	IN	STOP
F	*	<u>$2/3$</u>	<u>$1/3$</u>				
R	DT		<u>$2/4$</u>	<u>$1/4$</u>	<u>$1/4$</u>		
I	NNS				$3/4$		$1/4$
S	VB		$1/4$	$1/4$		$1/4$	$1/4$
T	NN						1
A	IN		1				
G	STOP						



Thank You!

In our next session: Viterbi Algorithm



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Parts of Speech Tagging using HMM

Prof. Aruna Malapati

Learning objectives

- Define Markov chains
- Define Hidden Markov Model



Markov Chains

- A Markov chain is a model that tells us something about the **probabilities of sequences of random variables**, states, each of which can take on values from some set.
- A Markov Model is a finite state machine with probabilistic state transitions.
- Markov assumption that next state only depends on the current state and independent of previous history.

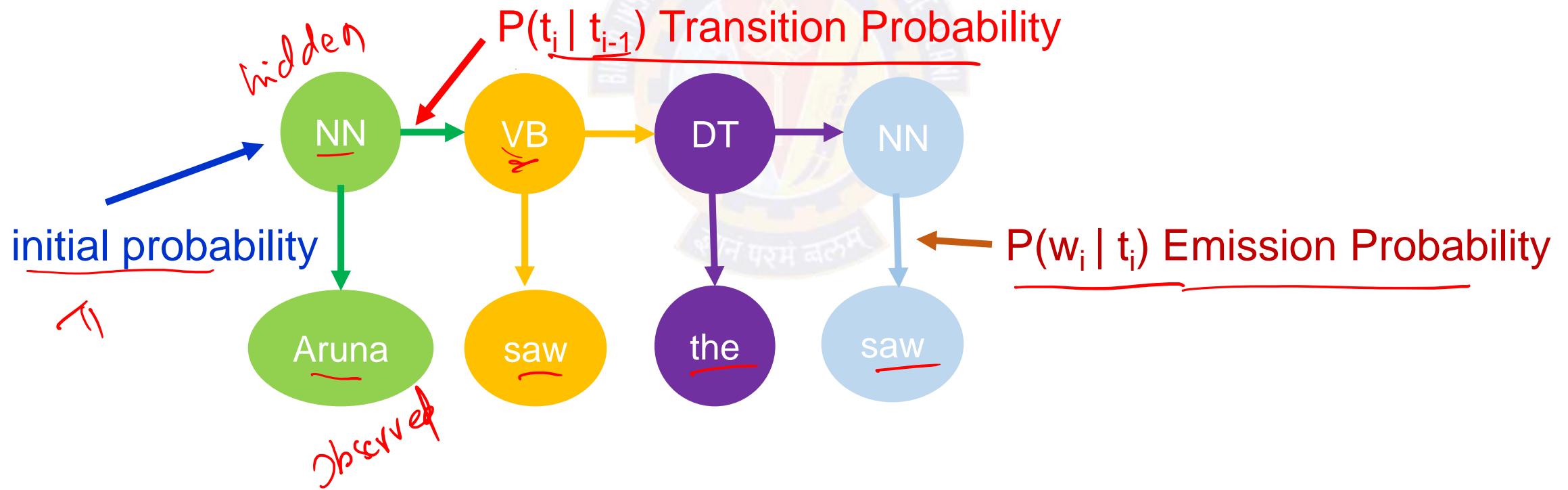
Markov Chain

- Formally, a Markov chain is specified by the following components:

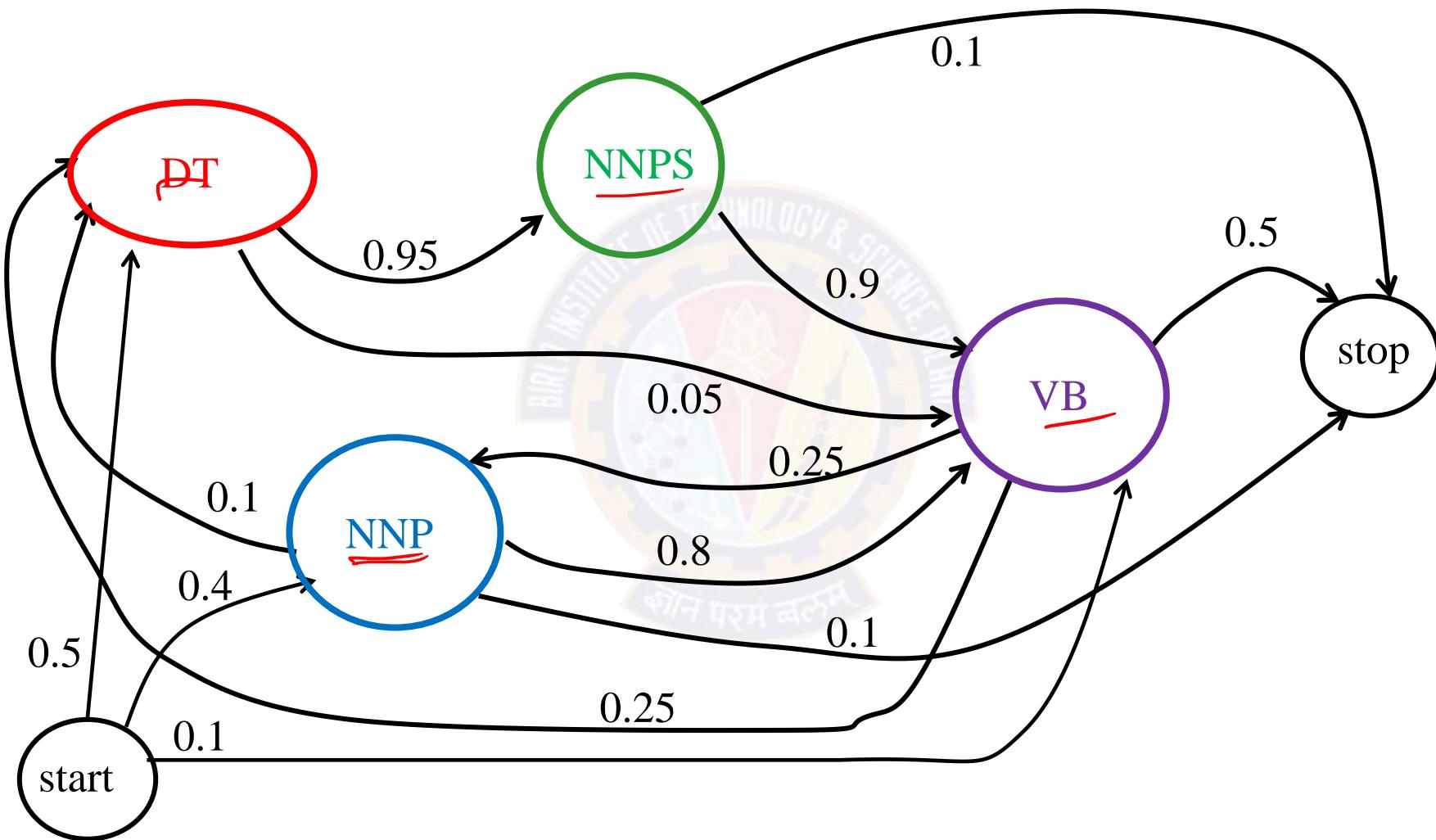
$\underline{Q} = q_1 q_2 \dots q_N$ a set of N states	A set of N states
$\underline{A} = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$	A transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, $\sum_{j=1}^n a_{ij} = 1 \forall i$
$\underline{\pi} = \pi_1, \pi_2, \pi_3, \dots, \pi_n$	An initial probability distribution over states. π_i is the probability that the Markov chain will start in state i . Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. $\sum_{i=1}^n \pi_i = 1$

The Hidden Markov Model

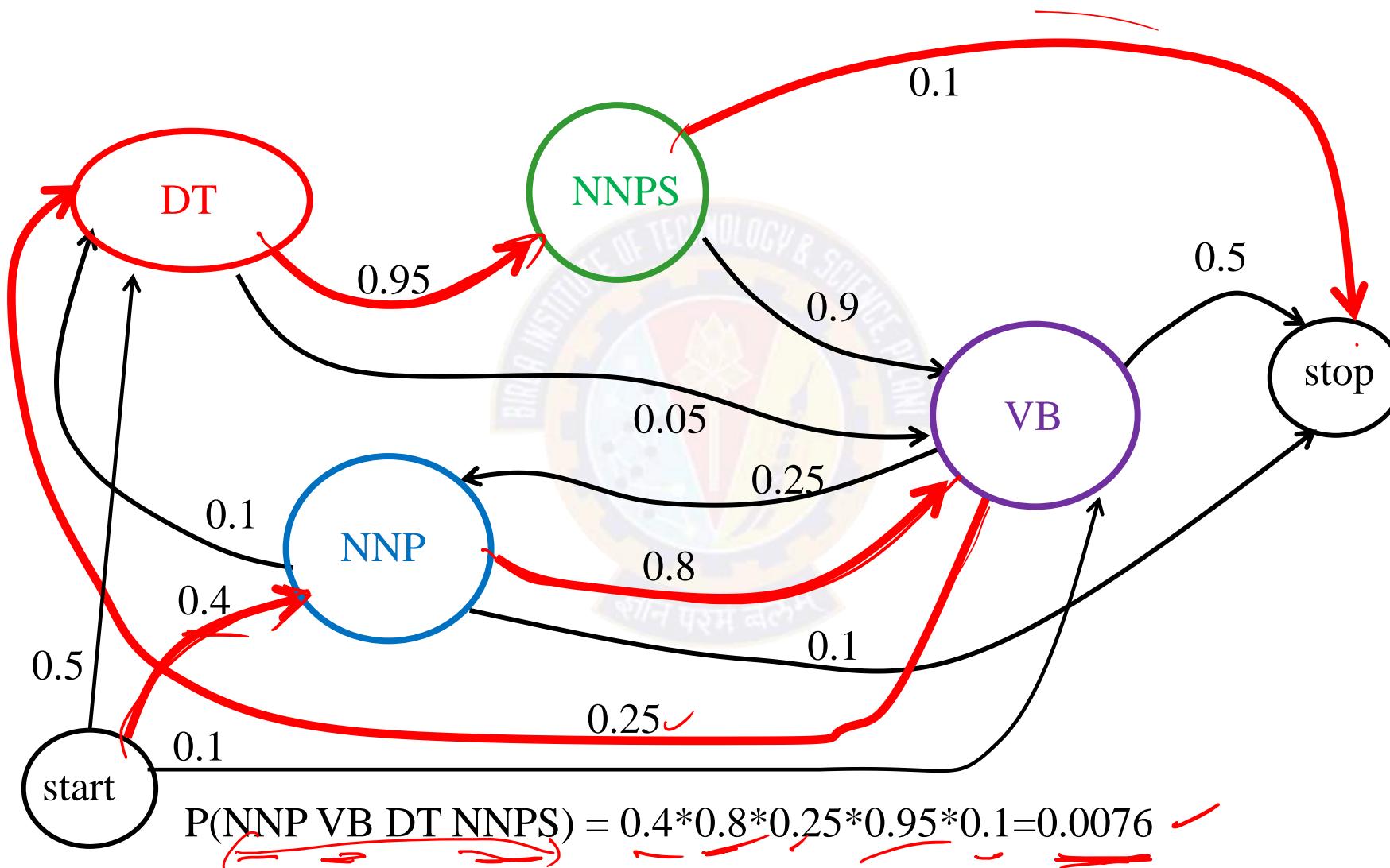
- In many cases the events we are interested in are hidden: we don't observe them directly.



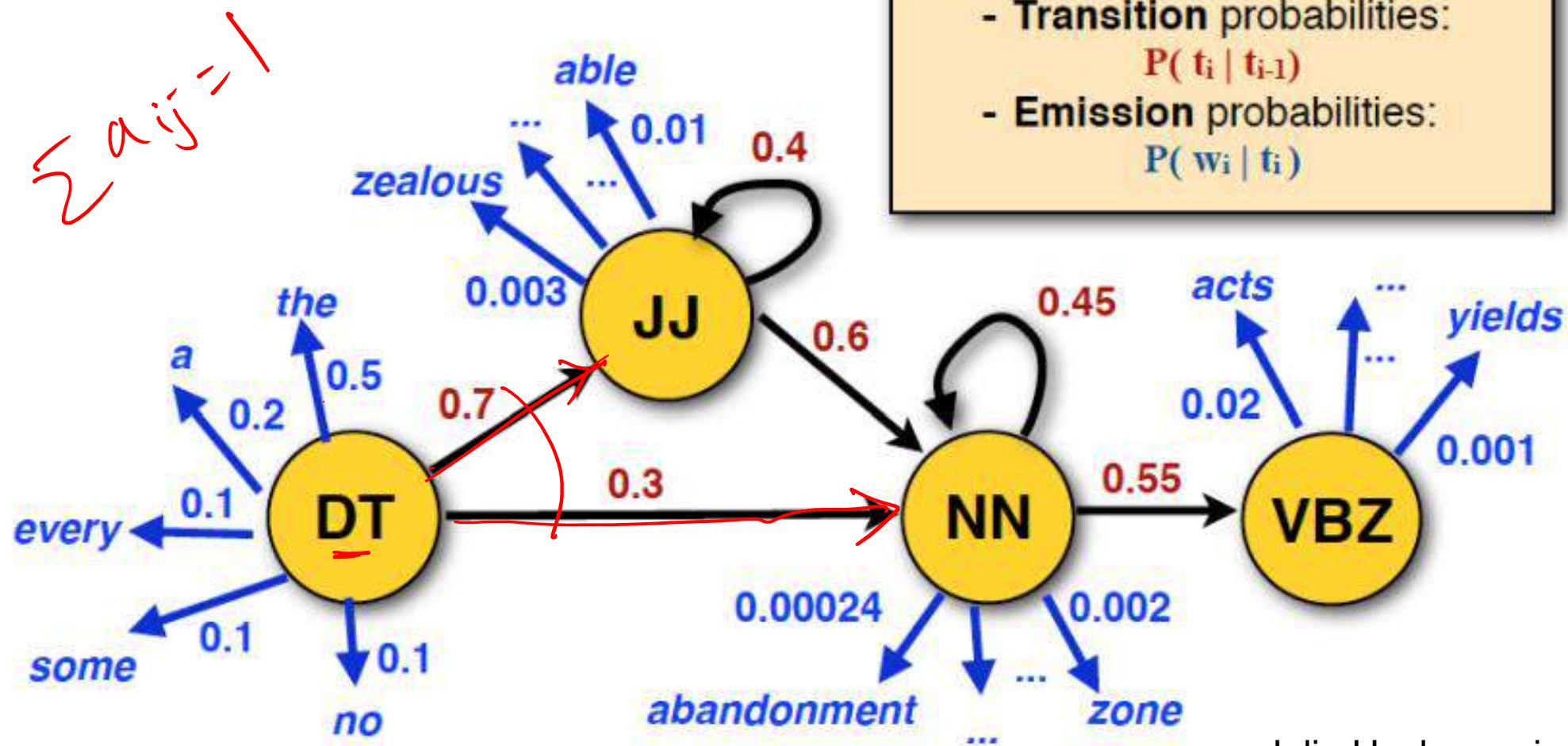
HMMs as probabilistic FSA



HMMs as probabilistic FSA



HMMs as probabilistic FSA



An HMM defines

- **Transition** probabilities:

$$P(t_i | t_{i-1})$$

- **Emission** probabilities:

$$P(w_i | t_i)$$

Hidden Markov Models (formal)

- States $T = t_1, t_2 \dots t_N$;
- Observations $W = w_1, w_2 \dots w_N$;
 - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \dots v_V\}$
- Transition probabilities
 - Transition probability matrix $A = \{a_{ij}\}$
$$a_{ij} = P(t_i = j | t_{i-1} = i) \quad 1 \leq i, j \leq N$$
- Observation likelihoods
 - Output probability matrix $B = \{b_i(k)\}$
$$b_i(k) = P(w_i = v_k | t_i = i) \quad ?(v_k | i)$$
- Special initial probability vector π $\pi_i = P(t_1 = i) \quad 1 \leq i \leq N$

HMM tagging as decoding

➤ HMM model contains hidden variables, the task of determining the hidden variables sequence corresponding to the sequence of observations decoding is called decoding.

➤ Find our best estimate of the sequence that maximizes

$$\underbrace{P(t_1 \dots t_n | w_1 \dots w_n)}$$

$$\hat{t}_1^n = \underbrace{\operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n)}$$

HMM tagging as decoding (Contd...)

Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$



Drop denominator since is the same for all tags we consider

HMM tagging as decoding (Contd...)

➤ A1: P(w) depends only on its own POS

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^n P(w_i | t_i)$$

➤ A2: P(t) depends only on P(t-1)

$$P(t_1^n) \approx \prod_{i=1}^n P(t_i | t_{i-1})$$

likelihood prior

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \overbrace{P(w_1^n | t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

HMM tagging as decoding (Contd...)

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n) \approx \boxed{\operatorname{argmax}_{t_1^n}} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$

Now we have two probabilities to calculate:

- Probability of a word occurring given its tag
- Probability of a tag occurring given a previous tag
- We can calculate each of these from a POS-tagged corpus



Thank You!

In our next session: HMM Example



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

HMM Example

Prof. Aruna Malapati

Learning Objectives

- Construction of Transaction and Emission Matrices



Example

Emission Matrix

	*	DT	NNS	VB	NN	IN	STOP
*	1						
the			3/4				
employees				3/4			
pass					2/4		
an			1/4				
exam						1	
wait				1/4			
for						1	
employers			1/4				
fire				1/4			
.					1		

Tag Translation Matrix

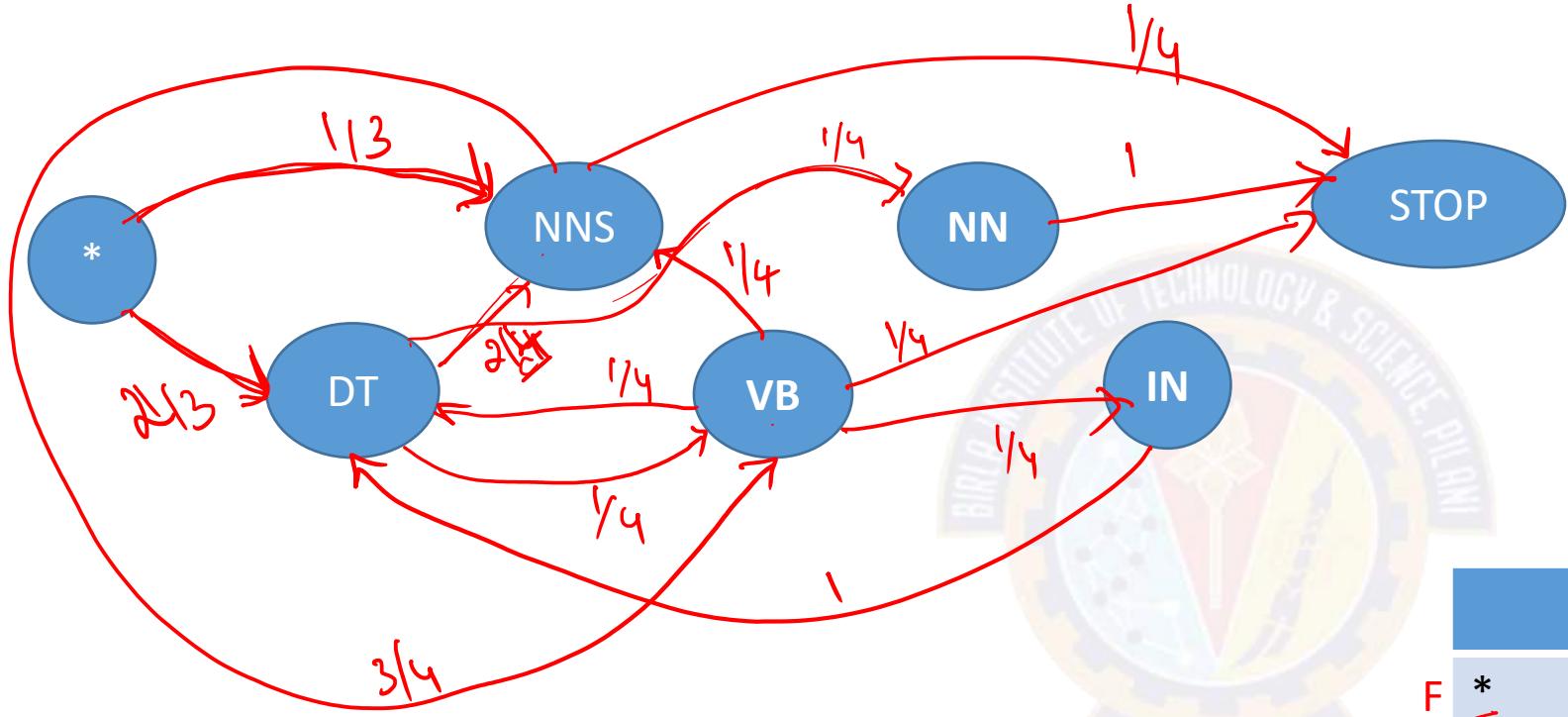
	*	<u>DT</u>	<u>NNS</u>	<u>VB</u>	<u>NN</u>	<u>IN</u>	<u>STOP</u>
F	*	<u>2/3</u>	<u>1/3</u>				
R	<u>DT</u>		<u>2/4</u>	<u>1/4</u>	<u>1/4</u>		
I					<u>3/4</u>		<u>1/4</u>
S	<u>NNS</u>						
T	<u>VB</u>	<u>1/4</u>	<u>1/4</u>			<u>1/4</u>	<u>1/4</u>
A	<u>NN</u>						1
A	<u>IN</u>	1					
G	<u>STOP</u>						

S1: the Employees pass an exam .
T1: DT NNS VB DT NN STOP

S2: the employees wait for the pass .
T2: DT NNS VB IN DT VB STOP

S3: employers fire employees .
T3: NNS VB NNS STOP

Transition diagram



$$\frac{P(DT | \cancel{*} \cancel{*})}{P(NNS | \cancel{DT} \cancel{DT})}$$

Tag Translation Matrix

	*	DT	NNS	VB	NN	IN	STOP
F	*	<u>$2/3$</u>	<u>$1/3$</u>				
R	DT		<u>$2/4$</u>	<u>$1/4$</u>	<u>$1/4$</u>		
I	NNS				<u>$3/4$</u>		<u>$1/4$</u>
S	VB		<u>$1/4$</u>	<u>$1/4$</u>		<u>$1/4$</u>	<u>$1/4$</u>
T	NN						<u>1</u>
A	IN		<u>1</u>				
G	STOP						



Thank You!

In our next session: Viterbi Algorithm



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Viterbi Algorithm

Prof. Aruna Malapati

Learning Objectives

- Motivation for Viterbi
- HMM decoding using Viterbi algorithm
- Example



The intuition behind Viterbi

- Finding the most probable tagging sequence for a test sentence $\langle w_1 \ w_2 \ w_3 \dots, w_n \rangle$

$$\underset{t_1, \dots, t_n}{\operatorname{argmax}} \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1}) P(w_i | t_i)$$

- The argmax is taken over all sequences y_1, y_2, \dots, y_n such that $y_i \in S$ for $i = 1, 2, \dots, n$ and $y_{n+1} = \text{STOP}$
- Lets assume that our tagging model is a bigram model

Brute force search over tag sequences

- Input sentence <the employees wait for an exam>
- S = { DT NNS VB IN NN }
- All possible tag sequences

DT DT DT DT DT DT STOP $\rightarrow 0.002$ \uparrow^6
DT DT DT DT DT NNS STOP $\rightarrow 0.003 \times 10^{-12}$ $|S|^n$
DT DT DT DT DT VB STOP $\rightarrow 0.0006$
....

Viterbi algorithm

- Create a table V with $N+2$ rows and T columns:
 - N – the number of states/tags
 - T – the length of the sequence/sentence
- Initialise the first column
- For each tag t in the tagset compute:

$$\underline{V[t, 1]} = \underline{P(t|start)} \underline{P(w_1|t)}$$

- For each column $j = 2$ to T in the table V :
 - For each tag t in the tagset compute:

$$V[t, j] = \max_{t'} V[t', j-1] \underline{P(t|t')} \underline{P(w_j|t)}$$

Example

Transition matrix: $P(t_i|t_{i-1})$

	NOUN	Verb	Det	Prep	ADV	STOP
<S>	.3	.1	.3	.2	.1	0
Noun	.2	.4	.01	.3	.04	.05
Verb	.3	.05	.3	.2	.1	.05
Det	.9	.01	.01	.01	.07	0
Prep	.4	.05	.4	.1	.05	0
Adv	.1	.5	.1	.1	.1	.1

Emission matrix: $P(w_i|t_i)$

	a	cat	doctor	in	is	the	very
Noun	0	.5	.4	0	0.1	0	0
Verb	0	0	.1	0	.9	0	0
Det	.3	0	0	0	0	.7	0
Prep	0	0	0	1.0	0	0	0
Adv	0	0	0	.1	0	0	.9

(S)
(I)
(C)
+, (S), start

$$V[t, 1] = P(t|start)P(w_1|t)$$

	w1=the	w2=doctor	w3=is	w4=in	STOP
Noun	0				
Verb	0				
Det	.21				
Prep	0				
Adv	0				

$$\begin{aligned}
 V(\text{Noun}, \text{the}) &= P(\text{Noun}|<\text{S}>)P(\text{the}|\text{Noun}) = .3 \times 0 = 0 \\
 V(\text{Verb}, \text{the}) &= P(\text{Verb}|<\text{S}>)P(\text{the}|\text{Verb}) = .1 \times 0 = 0 \\
 V(\text{Det}, \text{the}) &= P(\text{Det}|<\text{S}>)P(\text{the}|\text{Det}) = .3 \times .7 = .21 \\
 V(\text{Prep}, \text{the}) &= P(\text{Prep}|<\text{S}>)P(\text{the}|\text{Prep}) = .2 \times 0 = 0 \\
 V(\text{Adv}, \text{the}) &= P(\text{Adv}|<\text{S}>)P(\text{the}|\text{Adv}) = .2 \times 0 = 0
 \end{aligned}$$

Example (Contd..)

$$V(\text{Noun}, \text{doctor}) = \max_{t'} V(t', \text{the}) \overbrace{XP(\text{Noun}|t')}^{\text{underlined}} X \overbrace{P(\text{doctor}|\text{Noun})}^{\text{underlined}}$$
$$= \max \{ \underline{0}, \underline{0}, \underline{.21} \underline{(.3 \times .4)}, 0, 0 \} = \underline{.0756}$$

$$V(\text{Verb}, \text{doctor}) = \max_{t'} V(t', \text{the}) \overbrace{XP(\text{Verb}|t')}^{\text{underlined}} X \overbrace{P(\text{doctor}|\text{Verb})}^{\text{underlined}}$$
$$= \max \{ \underline{0}, \underline{0}, \underline{.21} \underline{(.01 \times .1)}, 0, 0 \} = \underline{.00021}$$



	w1=-the	w2=doctor	w3=is	w4=in	STOP
Noun	0	.0756			
Verb	0	.00021			
Det	.21	0			
Prep	0	0			
Adv	0	0			

Completed Viterbi matrix

	w1-=the	w2=doctor	w3=is	w4=in	STOP
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	.0000272
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	

Backtracking the Viterbi Matrix

	w1= <u>the</u>	w2= <u>doctor</u>	w3= <u>is</u>	w4= <u>in</u>	STOP
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	.0000272
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	

Det Noun Verb Prep STOP



Thank You!

In our next session: Python Implementation of POS Tagging

Transition matrix: $P(t_i | t_{i-1})$:

	Noun	Verb	Det	Prep	Adv	$</s>$
$< s >$.3	.1	.3	.2	.1	0
Noun	.2	.4	.01	.3	.04	.05
Verb	.3	.05	.3	.2	.1	.05
Det	.9	.01	.01	.01	.07	0
Prep	.4	.05	.4	.1	.05	0
Adv	.1	.5	.1	.1	.1	.1

Emission matrix: $P(w_i | t_i)$:

	a	cat	doctor	in	is	the	very
Noun	0	.5	.4	0	0.1	0	0
Verb	0	0	.1	0	.9	0	0
Det	.3	0	0	0	0	.7	0
Prep	0	0	0	1.0	0	0	0
Adv	0	0	0	.1	0	0	.9

Example

Suppose $W=\text{the doctor is in}$. Our initially empty table:

v	$w_1=\text{the}$	$w_2=\text{doctor}$	$w_3=\text{is}$	$w_4=\text{in}$	$</s>$
Noun					
Verb					
Det					
Prep					
Adv					

Filling in the first column

Suppose $W=\text{the doctor is in}$. Our initially empty table:

v	$w_1=\text{the}$	$w_2=\text{doctor}$	$w_3=\text{is}$	$w_4=\text{in}$	$</s>$
Noun	0				
Verb	0				
Det	.21				
Prep	0				
Adv	0				

$$v(\text{Noun, the}) = P(\text{Noun}|<s>)P(\text{the}|\text{Noun})=.3(0)$$

$$v(\text{Det, the}) = P(\text{Det}|<\!\!s\!\!>)P(\text{the}|\text{Det})=.3(.7)$$

The second column

$v(\text{Noun}, \text{doctor})$

$$= \max_{t'} v(t', \text{the}) \cdot P(\text{Noun}|t') \cdot P(\text{doctor}|\text{Noun})$$

v	$w_1 = \text{the}$	$w_2 = \text{doctor}$	$w_3 = \text{is}$	$w_4 = \text{in}$	$</s>$
Noun	0	?			
Verb	0				
Det	.21				
Prep	0				
Adv	0				

$$P(\text{Noun}|\text{Det}) P(\text{doctor}|\text{Noun}) = .3(.4)$$

The second column

$v(\text{Noun}, \text{doctor})$

$$= \max_{t'} v(t', \text{the}) \cdot P(\text{Noun}|t') \cdot P(\text{doctor}|\text{Noun})$$
$$= \max \{ 0, 0, .21(.36), 0, 0 \} = .0756$$

v	w ₁ =the	w ₂ =doctor	w ₃ =is	w ₄ =in	</s>
Noun	0	.0756			
Verb	0				
Det	.21				
Prep	0				
Adv	0				

$$P(\text{Noun}|\text{Det}) P(\text{doctor}|\text{Noun}) = .9(.4)$$

The second column

$v(\text{Verb}, \text{doctor})$

$$= \max_{t'} v(t', \text{the}) \cdot P(\text{Verb}|t') \cdot P(\text{doctor}|\text{Verb})$$

$$= \max \{ 0, 0, .21(.001), 0, 0 \} = .00021$$

v	$w_1 = \text{the}$	$w_2 = \text{doctor}$	$w_3 = \text{is}$	$w_4 = \text{in}$	$</s>$
Noun	0	.0756			
Verb	0	.00021			
Det	.21				
Prep	0				
Adv	0				

$$P(\text{Verb}|\text{Det}) P(\text{doctor}|\text{Verb}) = .01(.1)$$

The second column

$v(\text{Verb}, \text{doctor})$

$$= \max_{t'} v(t', \text{the}) \cdot P(\text{Verb}|t') \cdot P(\text{doctor}|\text{Verb})$$

$$= \max \{ 0, 0, .21(.001), 0, 0 \} = .00021$$

v	$w_1 = \text{the}$	$w_2 = \text{doctor}$	$w_3 = \text{is}$	$w_4 = \text{in}$	$</s>$
Noun	0	.0756			
Verb	0	.00021			
Det	.21	0			
Prep	0	0			
Adv	0	0			

$$P(\text{Verb}|\text{Det}) P(\text{doctor}|\text{Verb}) = .01(.1)$$

The third column

$v(\text{Noun}, \text{is})$

$$= \max_{t'} v(t', \text{doctor}) \cdot P(\text{Noun}|t') \cdot P(\text{is}|\text{Noun})$$

$$= \max \{ .0756(.02), .00021(.03), 0, 0, 0 \} = .001512$$

v	$w_1 = \text{the}$	$w_2 = \text{doctor}$	$w_3 = \text{is}$	$w_4 = \text{in}$	$</s>$
Noun	0	.0756	.001512		
Verb	0	.00021			
Det	.21	0			
Prep	0	0			
Adv	0	0			

$$P(\text{Noun}|\text{Noun}) P(\text{is}|\text{Noun}) = .2(.1) = .02$$

$$P(\text{Noun}|\text{Verb}) P(\text{is}|\text{Noun}) = .3(.1) = .03$$

The third column

$v(\text{Verb}, \text{is})$

$$= \max_{t'} v(t', \text{doctor}) \cdot P(\text{Verb}|t') \cdot P(\text{is}|\text{Verb})$$

$$= \max \{ .0756(.36), .00021(.045), 0, 0, 0 \} = .027216$$

v	$w_1 = \text{the}$	$w_2 = \text{doctor}$	$w_3 = \text{is}$	$w_4 = \text{in}$	$</s>$
Noun	0	.0756	.001512		
Verb	0	.00021	.027216		
Det	.21	0	0		
Prep	0	0	0		
Adv	0	0	0		

$$P(\text{Verb}|\text{Noun}) P(\text{is}|\text{Verb}) = .4(.9) = .36$$

$$P(\text{Verb}|\text{Verb}) P(\text{is}|\text{Verb}) = .05(.9) = .045$$

The fourth column

$v(\text{Prep, in})$

$$= \max_{t'} v(t', \text{is}) \cdot P(\text{Prep}|t') \cdot P(\text{in}|\text{Prep})$$

$$= \max \{ .001512(.3), .027216(.2), 0, 0, 0 \} = .005443$$

v	$w_1 = \text{the}$	$w_2 = \text{doctor}$	$w_3 = \text{is}$	$w_4 = \text{in}$	$</s>$
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	
Prep	0	0	0	.005443	
Adv	0	0	0		

$$P(\text{Prep}|\text{Noun}) P(\text{in}|\text{Prep}) = .3(1.0)$$

$$P(\text{Prep}|\text{Verb}) P(\text{in}|\text{Prep}) = .2(1.0)$$

The fourth column

$v(\text{Prep}, \text{in})$

$$= \max_{t'} v(t', \text{is}) \cdot P(\text{Prep}|t') \cdot P(\text{in}|\text{Prep})$$

$$= \max \{ .000504(.004), .027216(.01), 0, 0, 0 \} = .000272$$

v	$w_1=\text{the}$	$w_2=\text{doctor}$	$w_3=\text{is}$	$w_4=\text{in}$	$</s>$
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	

$$P(\text{Adv}|\text{Noun}) P(\text{in}|\text{Adv}) = .04(.1)$$

$$P(\text{Adv}|\text{Verb}) P(\text{in}|\text{Adv}) = .1(.1)$$

End of sentence

$v(</s>)$

$$= \max_{t'} v(t', \text{in}) \cdot P(</s>|t')$$

$$= \max \{ 0, 0, 0, .005443(0), .000272(.1) \} = .0000272$$

v	$w_1=\text{the}$	$w_2=\text{doctor}$	$w_3=\text{is}$	$w_4=\text{in}$	$</s>$
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	.000027 2
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	

$$P(</s>|\text{Prep})=0$$

$$P(</s>|\text{Adv})=.1$$

Completed Viterbi Chart

v	$w_1=\text{the}$	$w_2=\text{doctor}$	$w_3=\text{is}$	$w_4=\text{in}$	$</s>$
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	.000027 2
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

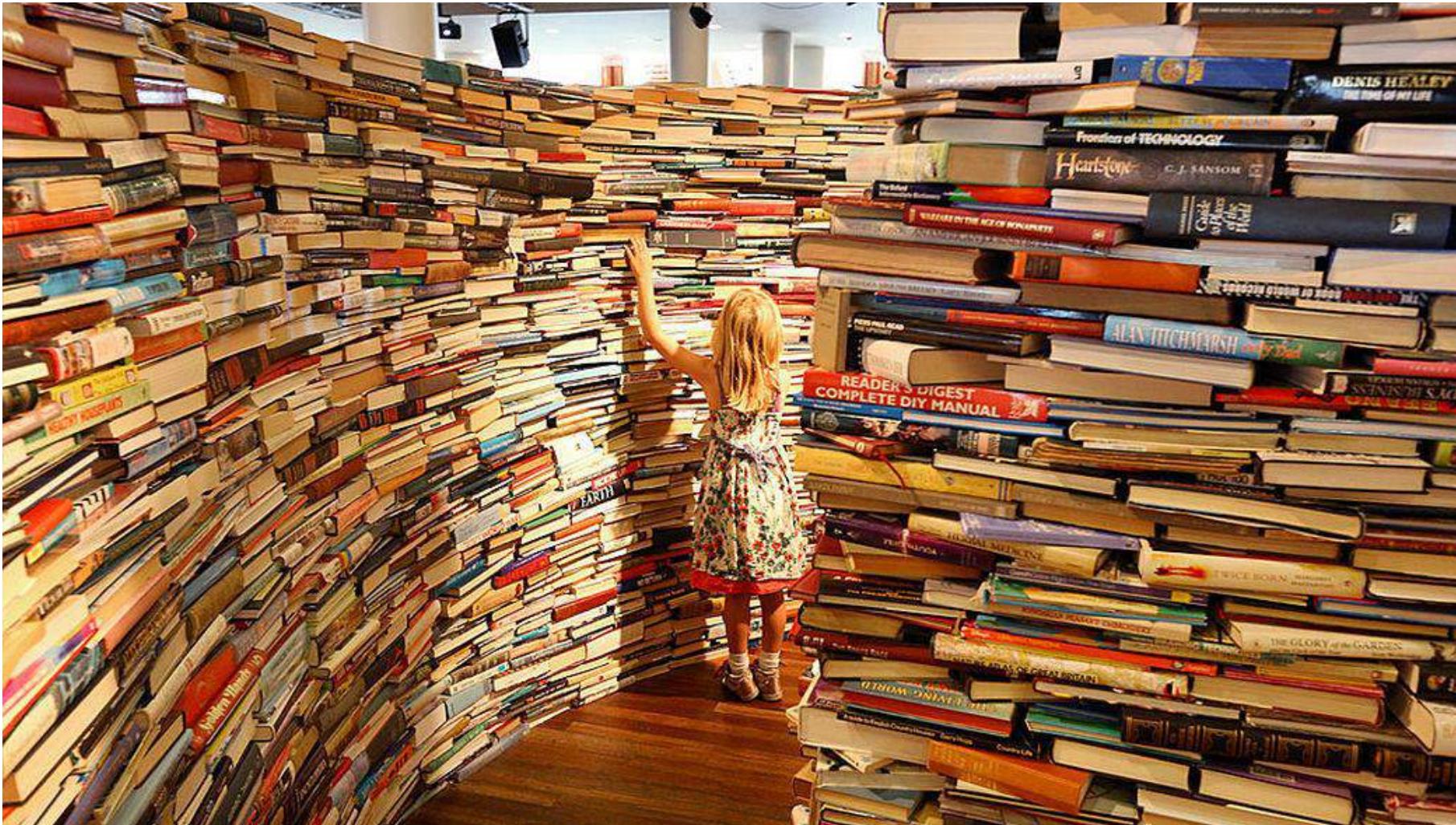
Introduction to Topic Modelling

Prof. Aruna Malapati

Learning Objectives

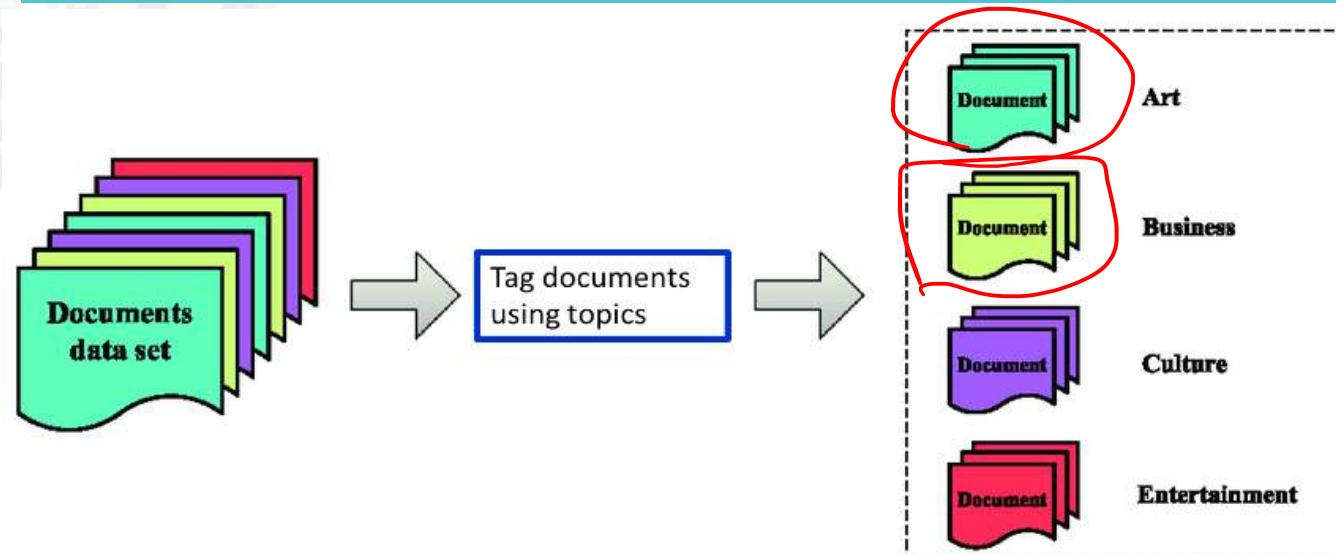
- Motivation for Topic Modelling
- Objectives of Topic Modelling
- Generative model for Topic Model
- The posterior distribution

Motivation for Topic Modelling



Objectives of Topic Modelling

- Use these annotations to organize, summarize and search the documents.



Sample output from the LDA

- Four topics learned from the S&P 500 stock market data
- Goal is to find groups of stocks that tend to move together.

Topic 1	Topic 2	Topic 3	Topic 4
Southwestern Energy	Penneys	Capital One	Simon Property
Range Resources	Macys	BNY Mellon	Kimco Realty
Cabot Oil & Gas	Kohls	Discover	Equity Residential
EOG Resources	Nordstrom	Northern Trust	AvalonBay Communities
Chesapeake Energy	Target	Janus	Apartment Investment
Pioneer Resources	Limited	JPMorgan Chase	Vornado Realty Trust
Devon Energy	Lowes	State Street	Boston Properties
Peabody Energy	Home Depot	Wells Fargo	Public Storage
Anadarko Petroleum	American Express	PPL	Host Hotels
Massey Energy	Abercrombie	T. Rowe Price	HCP Inc.

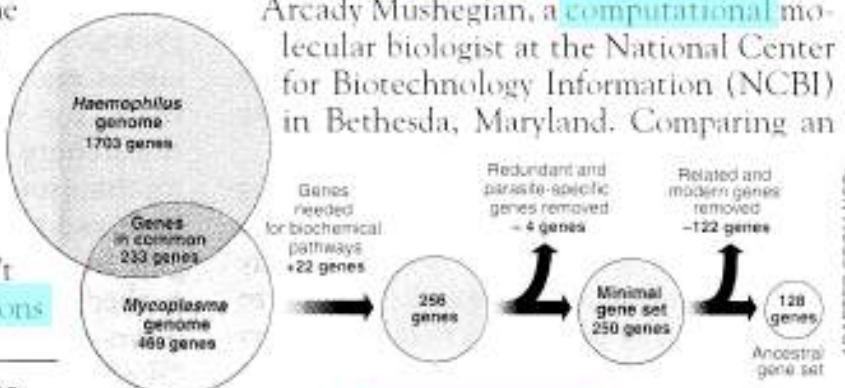
- The topic model does not provide any label to these group of words.

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

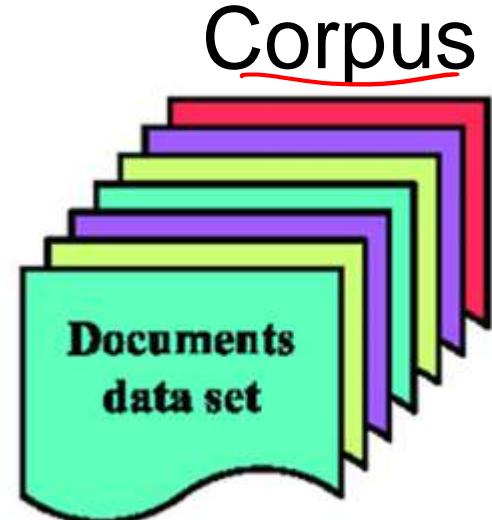
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Genetics

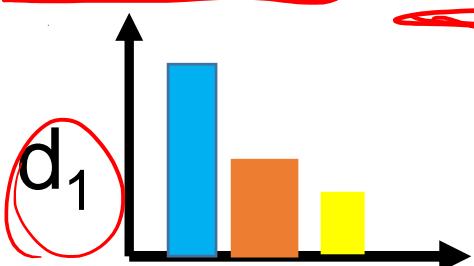
Evolutionary
biology

Data Analysis

Overall schematic

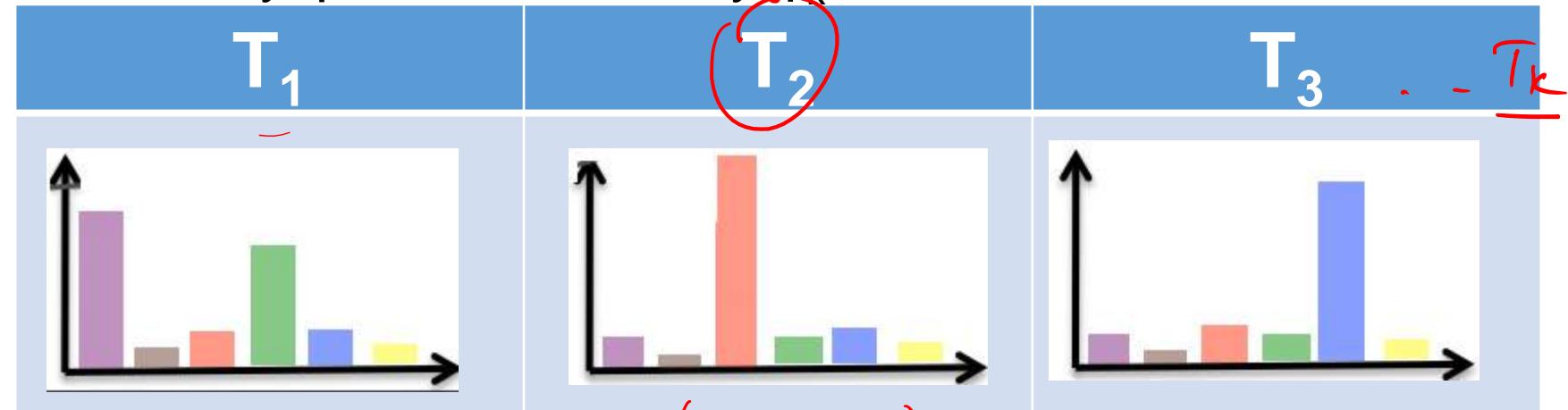


Documents(d₁...d_n)

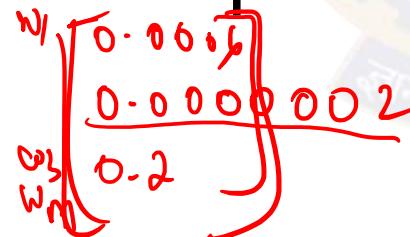


Each document has a distribution over K topics

Each topic is defined as a Multinomial distribution over the vocabulary, parameterized by ϕ_k



K-Topics (Hyper Parameter)

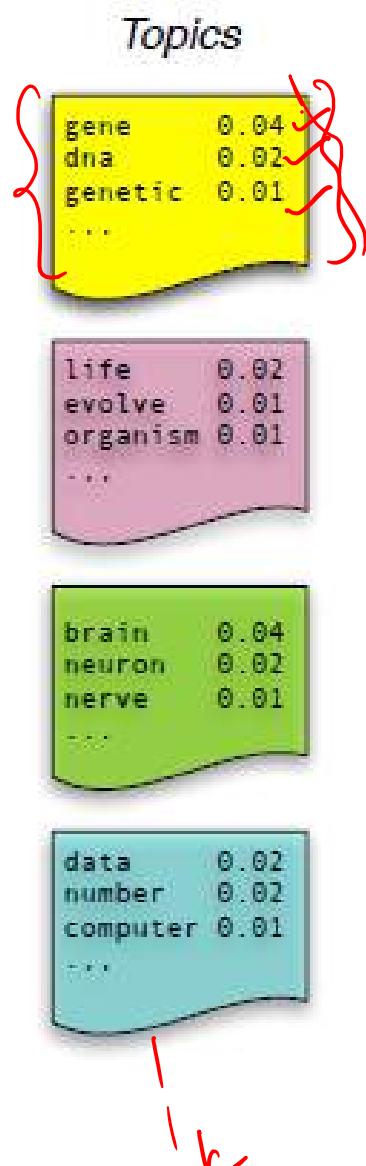


Parameter



Vocabulary($W_1 \dots W_n$)

Generative model for Topic Model



Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

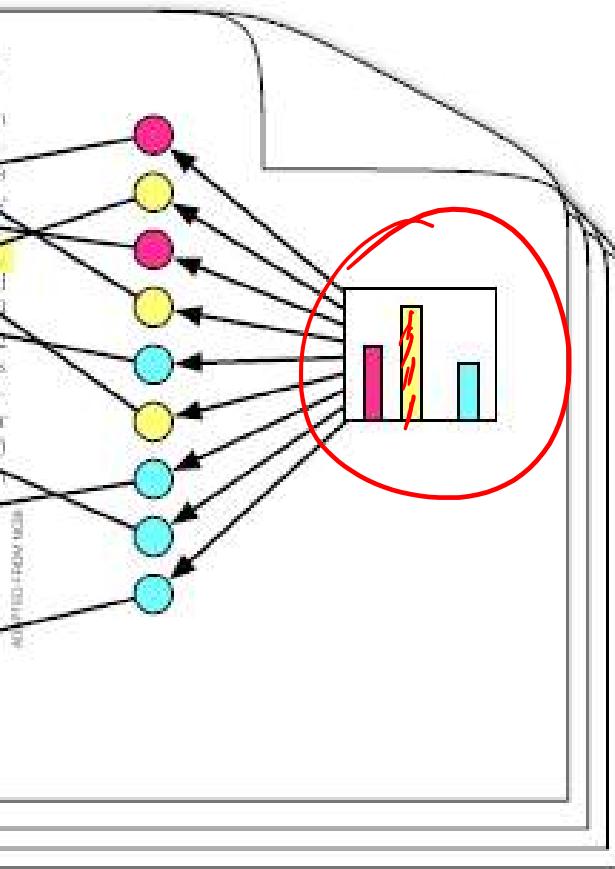
Although the numbers don't match precisely, these predictions

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson, a geneticist at the University in Stockholm, Sweden, who arrived at the Sanger Center. But coming up with a consensus answer may be more than just a numbers game. Surprisingly, more and more genomes are completely sequenced. "It may be a way of organizing my newly sequenced genome," explains Arshad Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all

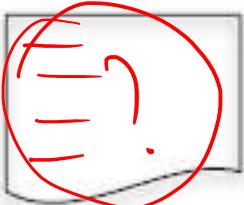


Topic proportions and assignments



The posterior distribution

Topics



Documents

Seeking Life's Bare (Genetic) Necessities

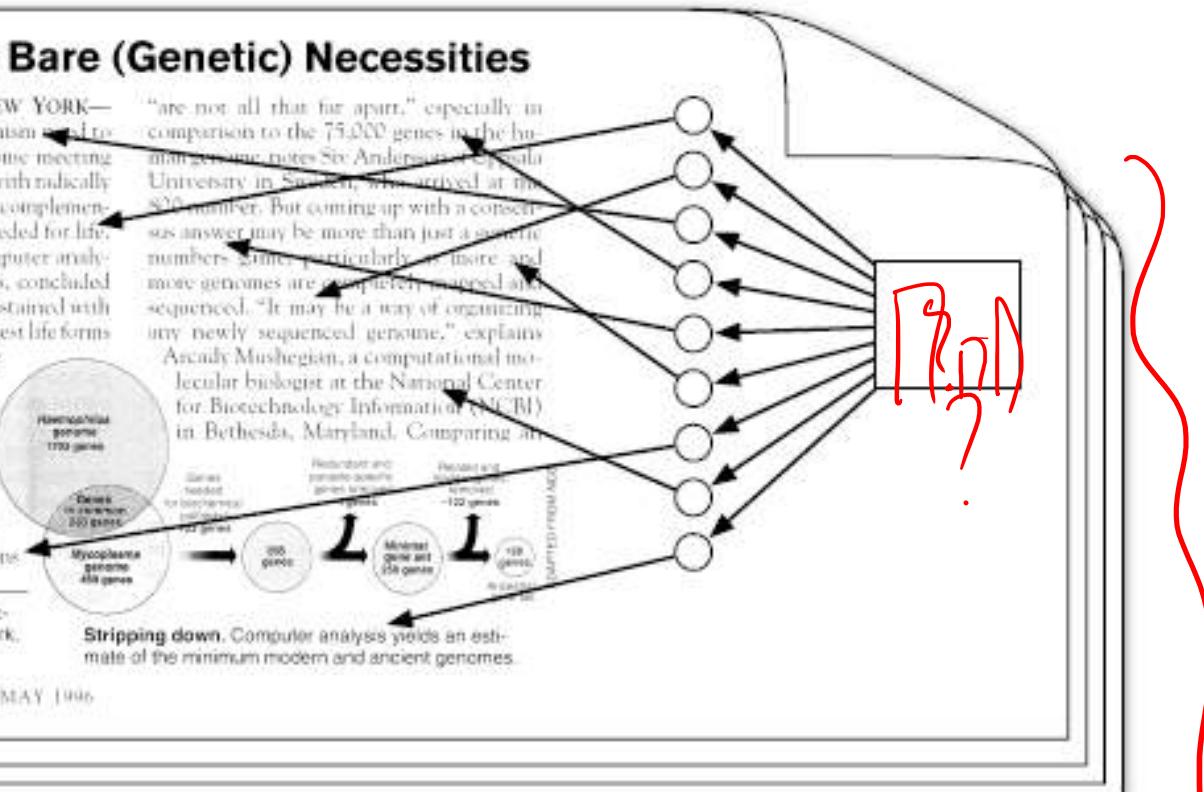
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two-genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Svante Andersson, a geneticist at Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a generic numbers game; particularly as more and more genomes are completely sequenced and sequenced. "It may be a way of organizing many newly sequenced genomes," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Topic proportions and assignments





Thank You!

In our next session: Mathematical basis for LDA



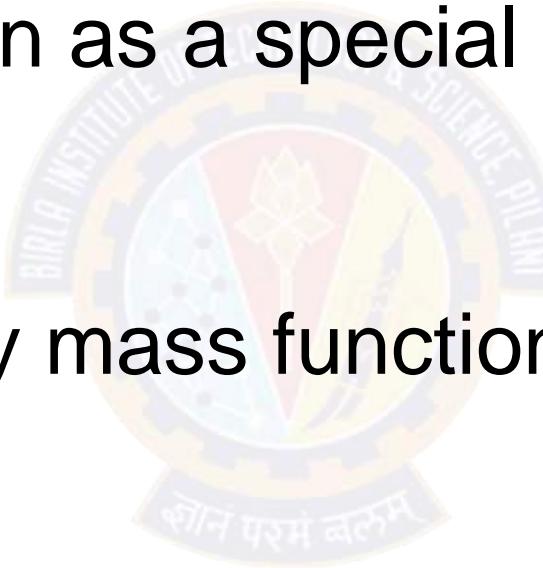
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Probability Density Functions

Prof. Aruna Malapati

Learning Objectives

- Bernoulli trial
- Bernoulli distribution as a special case of Binomial Distribution
- Bernoulli probability mass function
- Beta Distribution



Bernoulli Trial

- Any single trial with two possible outcomes can be modeled as a Bernoulli trial: team wins/loses, pitch is a strike/ball, coin comes up heads or tails, etc.
- A Bernoulli trial uses Bernoulli distribution to calculate the probability of either outcome.

Bernoulli trial



$$P(X=1) = \theta \quad P(X=0) = (1 - \theta)$$

Bernoulli: A Special Case of the Binomial Distribution

Binomial Trail: Chance of getting
n heads in a row(n=3)



Bernoulli Trail: Chance of getting
a heads on a single flip

Bernoulli - Distribution Notation

- The probability mass function of the Bernoulli distribution is

$$P(X=1) = \theta \quad P(X=0) = (1-\theta)$$

$$f(x) = P(X=k) = \theta^k (1-\theta)^{1-k}, \quad k=\{0,1\}$$

- The only parameter of the bernoulli distribution is θ , which defines the probability of success during a bernoulli trial.

Binomial distribution

$$k=0 \\ P(x=0) = \theta^0 (1-\theta)^{1-0} = 1-\theta$$

$$k=1 \\ P(x=1) = \underline{\theta^1} (1-\theta)^{1-1} = \underline{\underline{\theta}}$$

$$P(x_1=1, \underset{P}{x_2=1}, \underset{P}{x_3=0}) = \theta \times \theta (1-\theta) = \theta^2 (1-\theta)$$

$\frac{n=5}{\{ \text{D} \text{ D} \text{ D} \text{ D} \text{ D} \}}$

$$P\left(\sum_{i=1}^{n=3} x_i = 2\right) = P(1, 1, 0) + P(1, 0, 1) + P(0, 1, 1) \\ = \theta^2 (1-\theta) + \theta^2 (1-\theta) + \theta^2 (1-\theta) \\ = 3\theta^2 (1-\theta)$$

$$P\left(\sum_{i=1}^n x_i = k\right) = \binom{n}{k} \theta^k (1-\theta)^{n-k} \quad \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Beta Distribution

- The probability distribution function for the beta distribution

$$f(\theta; \alpha, \beta) = \frac{\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}}{B(\alpha, \beta)} \propto \theta^{(\alpha-1)}(1-\theta)^{\beta-1}$$

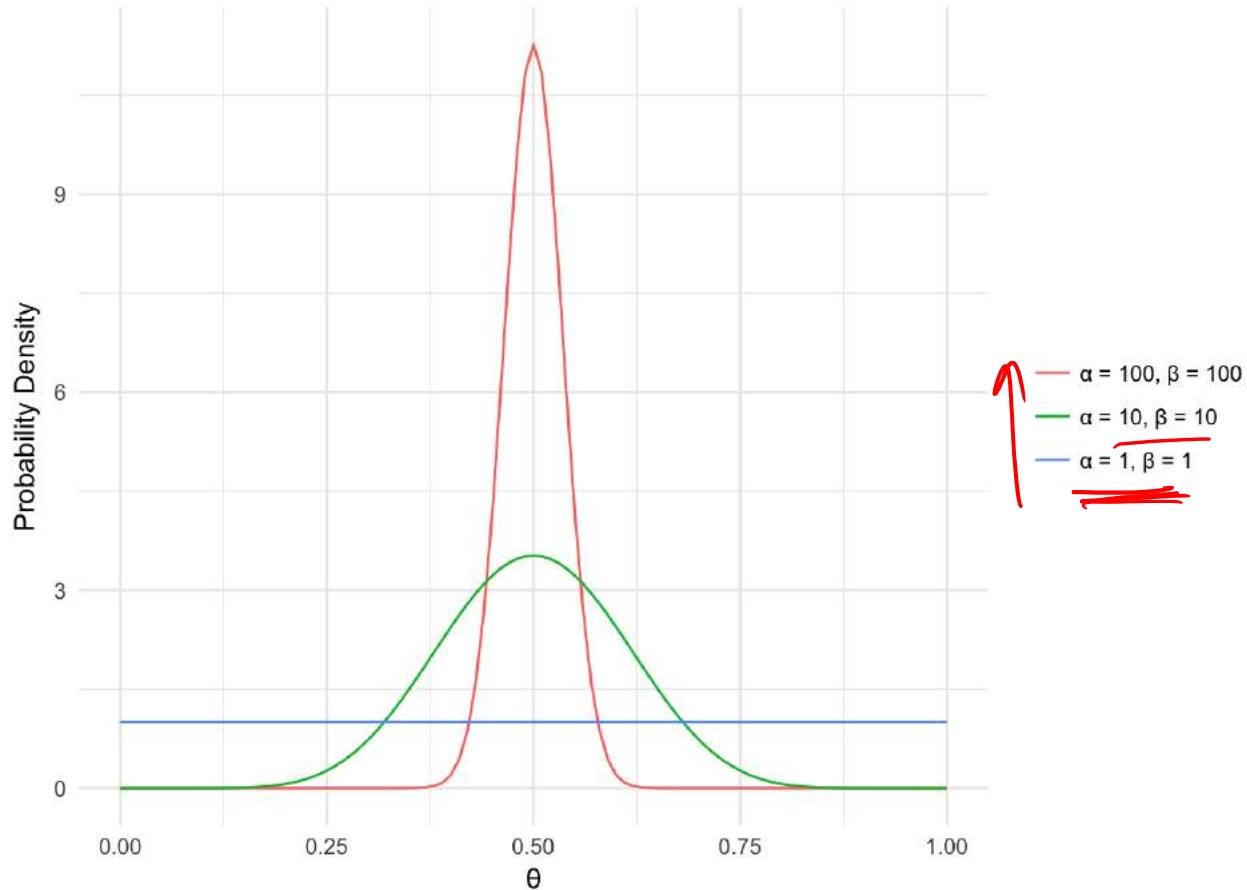
$\theta \in [0, 1]$

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

$$\Gamma(a) = (a-1)!$$

Beta Distribution

- The beta distribution can be thought of as a **probability distribution of probabilities.**



Beta function as a function of Gamma





Thank You!

In our next session: Multinomial Distribution



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Multinomial Distribution

Prof. Aruna Malapati

Learning Objectives

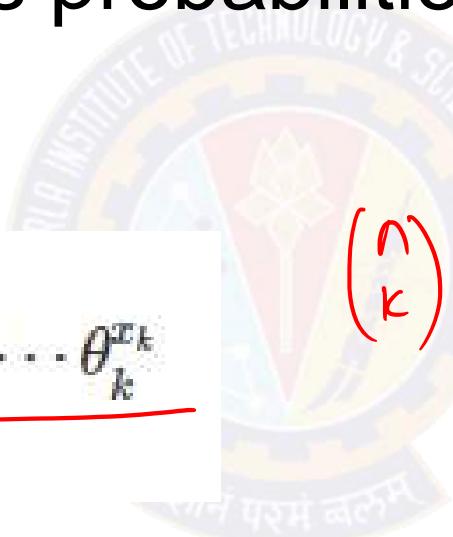
- Multinomial distribution
- Parameter Estimation



From Dice to words

- Suppose we roll our die of words having k sides(vocabulary) where each side takes probabilities $\Theta_1, \dots, \Theta_k$ respectively.

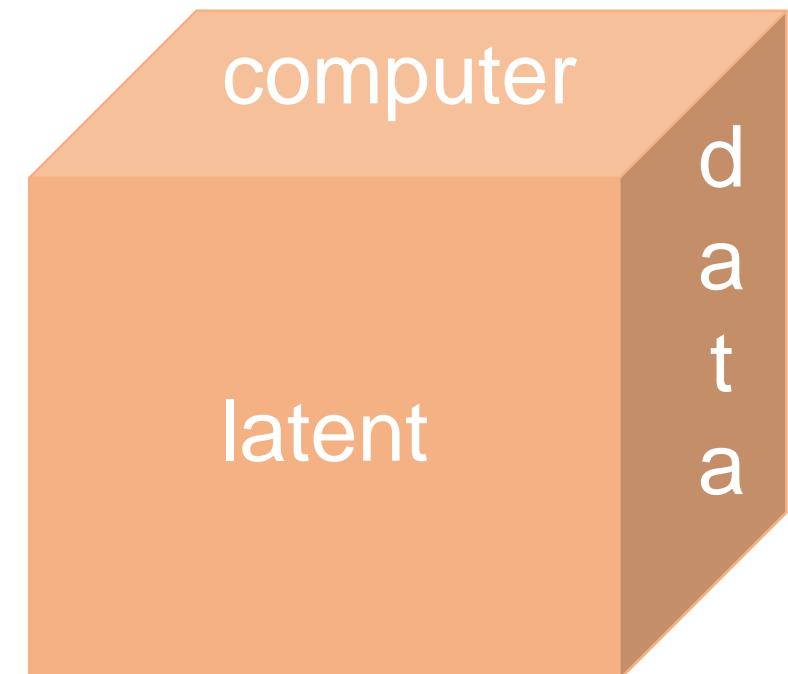
$$f(x) = \frac{n!}{x_1!x_2!\dots x_k!} \theta_1^{x_1} \theta_2^{x_2} \dots \theta_k^{x_k}$$



$$\binom{n}{k} \theta^k (1-\theta)^{n-k}$$

k - number of sides on the die

n - number of times the die will be rolled

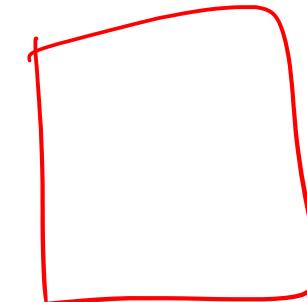


The Building Blocks of inferring the parameters

➤ Parameter estimation

$$p(\theta|D) = \frac{\underbrace{p(D|\theta)}_{posterior} \underbrace{p(\theta)}_{likelihood\ prior}}{\underbrace{p(D)}_{evidence}}$$

$w_1 = \{ \}$
 $w_2 = \{ \}$
 $w_3 = \{ \}$



- Maximum Likelihood
- Maximum a Posterior (MAP)
- Bayesian Inference ✓



Thank You!

In our next session: Conjugate Prior



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Conjugate Prior

Prof. Aruna Malapati

Conjugate Prior

$$\underbrace{p(\theta|D)}_{\text{posterior}} = \frac{\underbrace{p(D|\theta) p(\theta)}_{\text{likelihood prior}}}{\underbrace{p(D)}_{\text{evidence}}}$$

$p(D|\theta) \sim \text{Normal}$ and $p(\theta) \sim \text{Normal}$

$\Rightarrow p(\theta|D) \sim \text{Normal}$

$p(D|\theta) \sim \text{Normal}$ and $p(\theta) \sim \text{Gamma}$

$\Rightarrow p(\theta|D) \propto \text{Normal} / \text{Gamma}$

$p(D|\theta) \sim \text{Bernoulli}$ $\theta^k (1-\theta)^{N-k}$

$p(\theta) \sim \theta^{a-1} (1-\theta)^{b-1}$

Posterior $\sim \text{Beta}$



Thank You!

In our next session: Dirichelet distribution



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Dirichlet distributions

Prof. Aruna Malapati

Dirichlet distributions

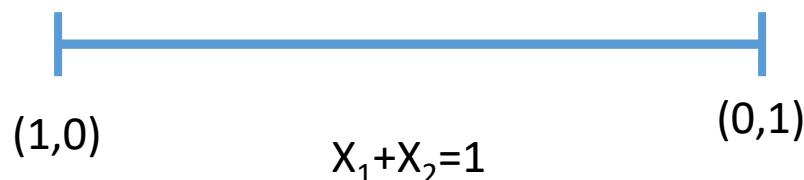
- Dirichlet distributions are probability distributions over multinomial parameter vectors
- They are called Beta distributions when k = 2
- The Dirichlet probability density function is defined as

$$Dir(\vec{\theta} | \vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K \theta_i^{\alpha_i - 1}$$

$$\therefore Dir(\vec{\theta} | \vec{\alpha}) = \frac{1}{B(\alpha)} \prod_{i=1}^K \theta_i^{\alpha_i - 1} \quad \text{where} \quad \frac{1}{B(\alpha)} = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)}$$

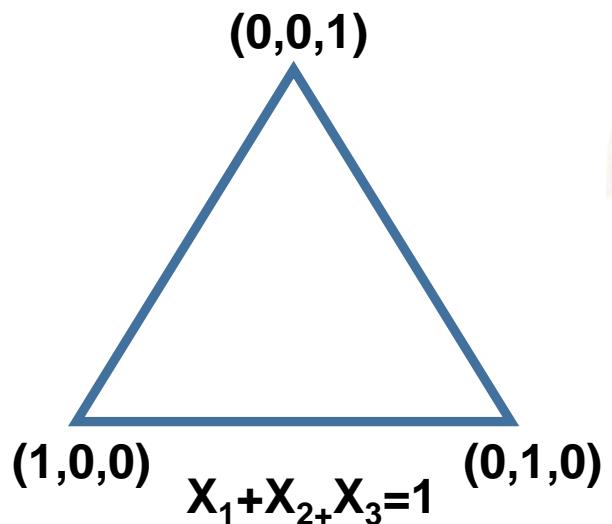
Visualization of the simplex

- This is often referred as **simplex** and a most convenient way to visualize this is using a certain shapes depending upon the number of topics.
- Suppose K=2 topics which can be modeled as 1-simplex and can be visualized using a line.



Visualization of the simplex(Contd...)

- Suppose K=3 topics which can be modeled as 2-simplex and can be visualized using a triangle.



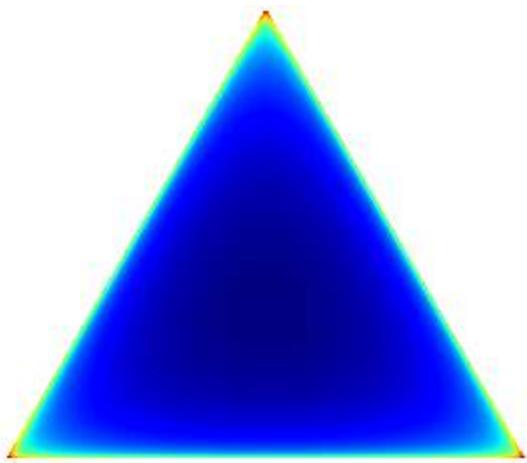
- If we have **K** topics this can be generated using **K-1** simplex.

Dirichlet distribution is parametrized by α

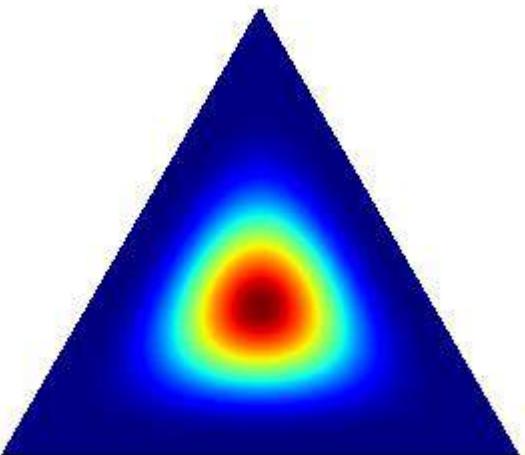


Shape of the Dirichlet distribution

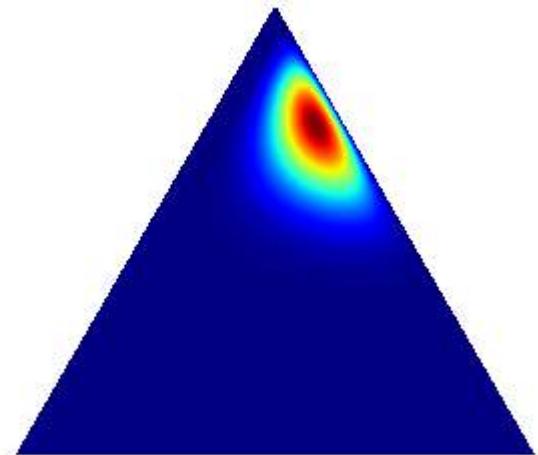
Dirichlet(0.999, 0.999, 0.999)



Dirichlet(5, 5, 5)



Dirichlet(2, 5, 15)





Thank You!

In our next session: Mathematical modelling of LDA



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mathematical modelling of LDA

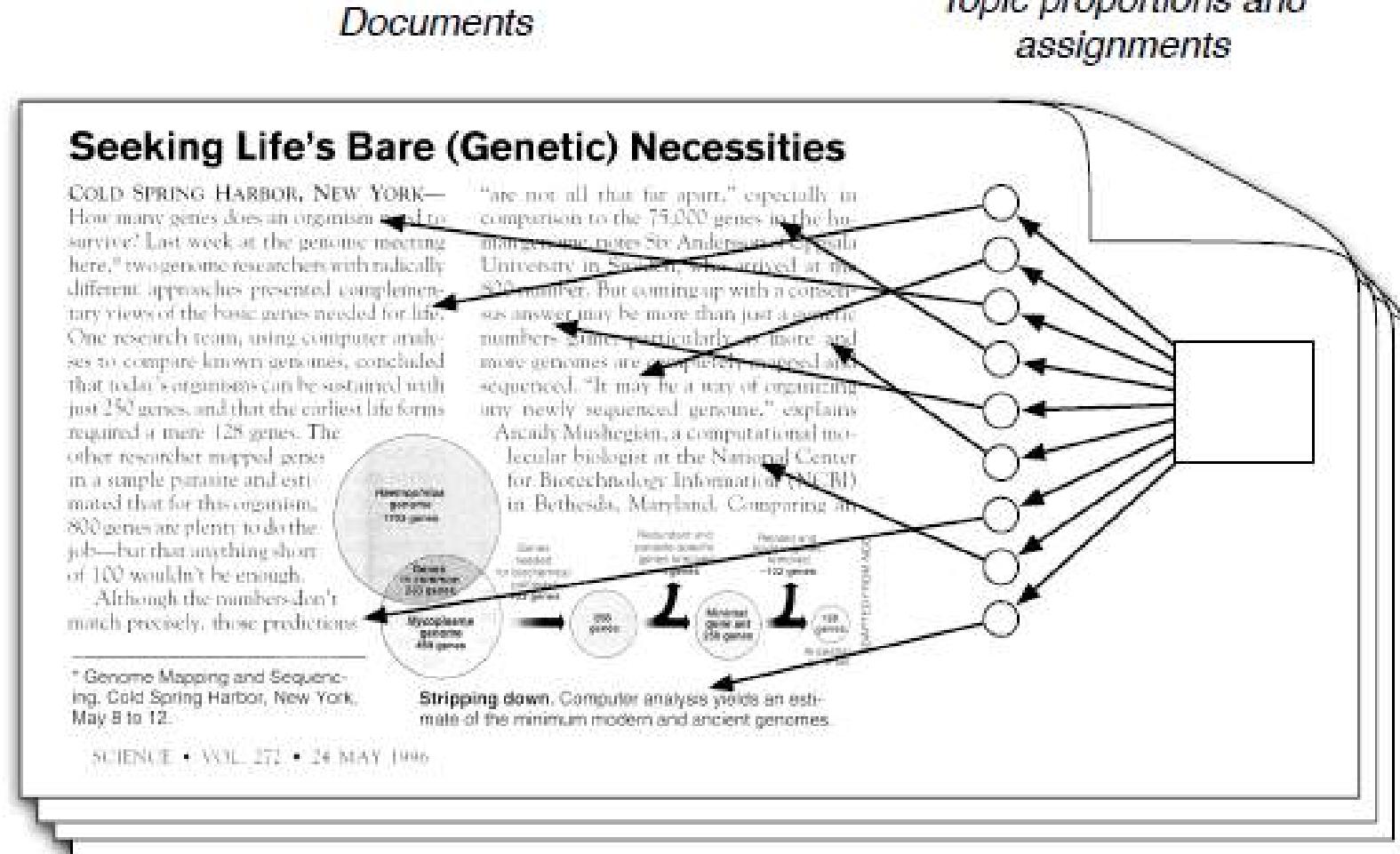
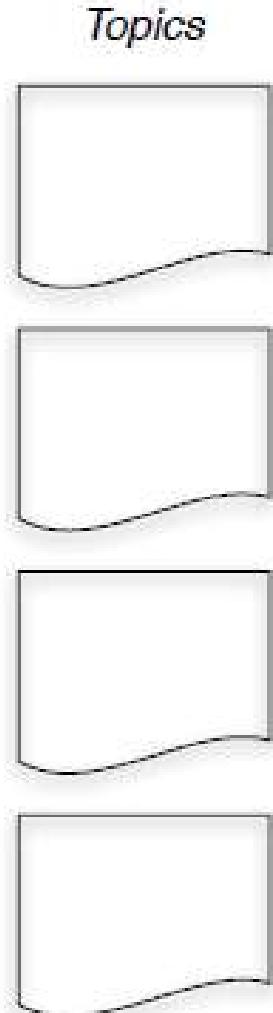
Prof. Aruna Malapati

Learning Objectives

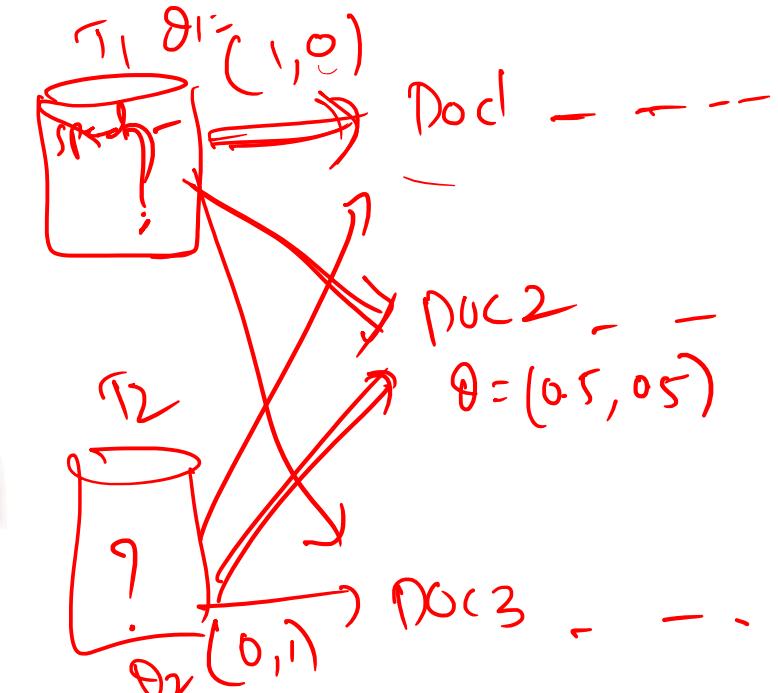
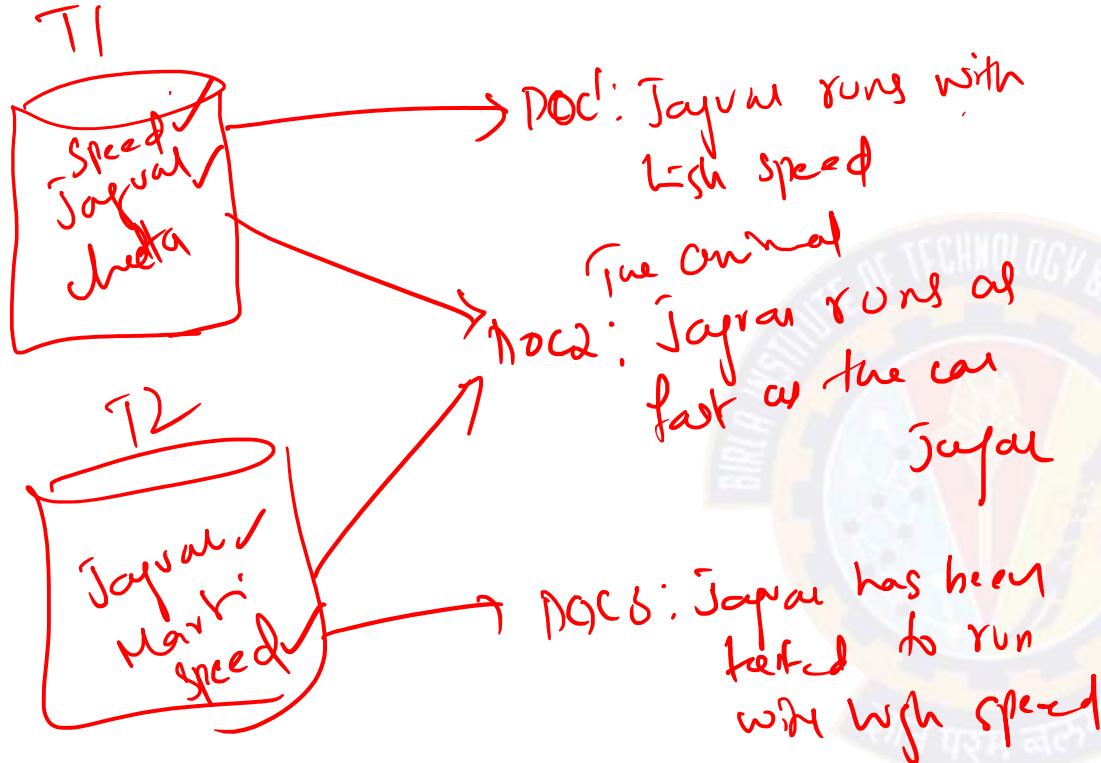
- Generative process of modelling LDA



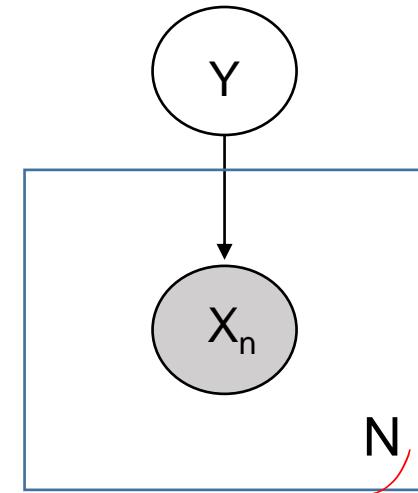
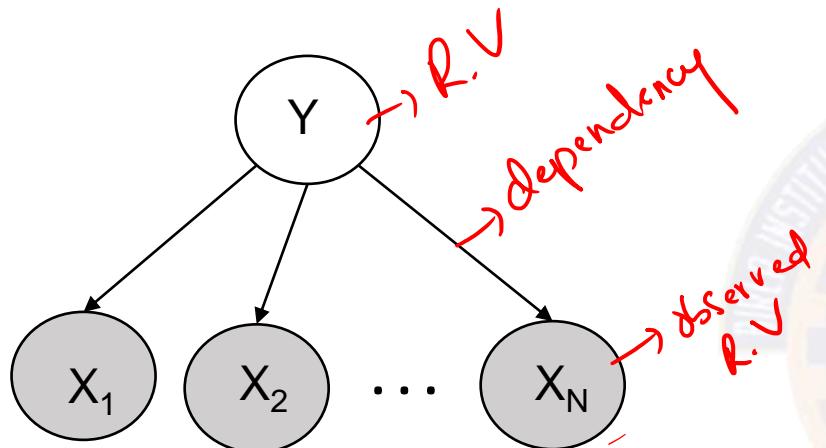
The posterior distribution



Statistical Inference



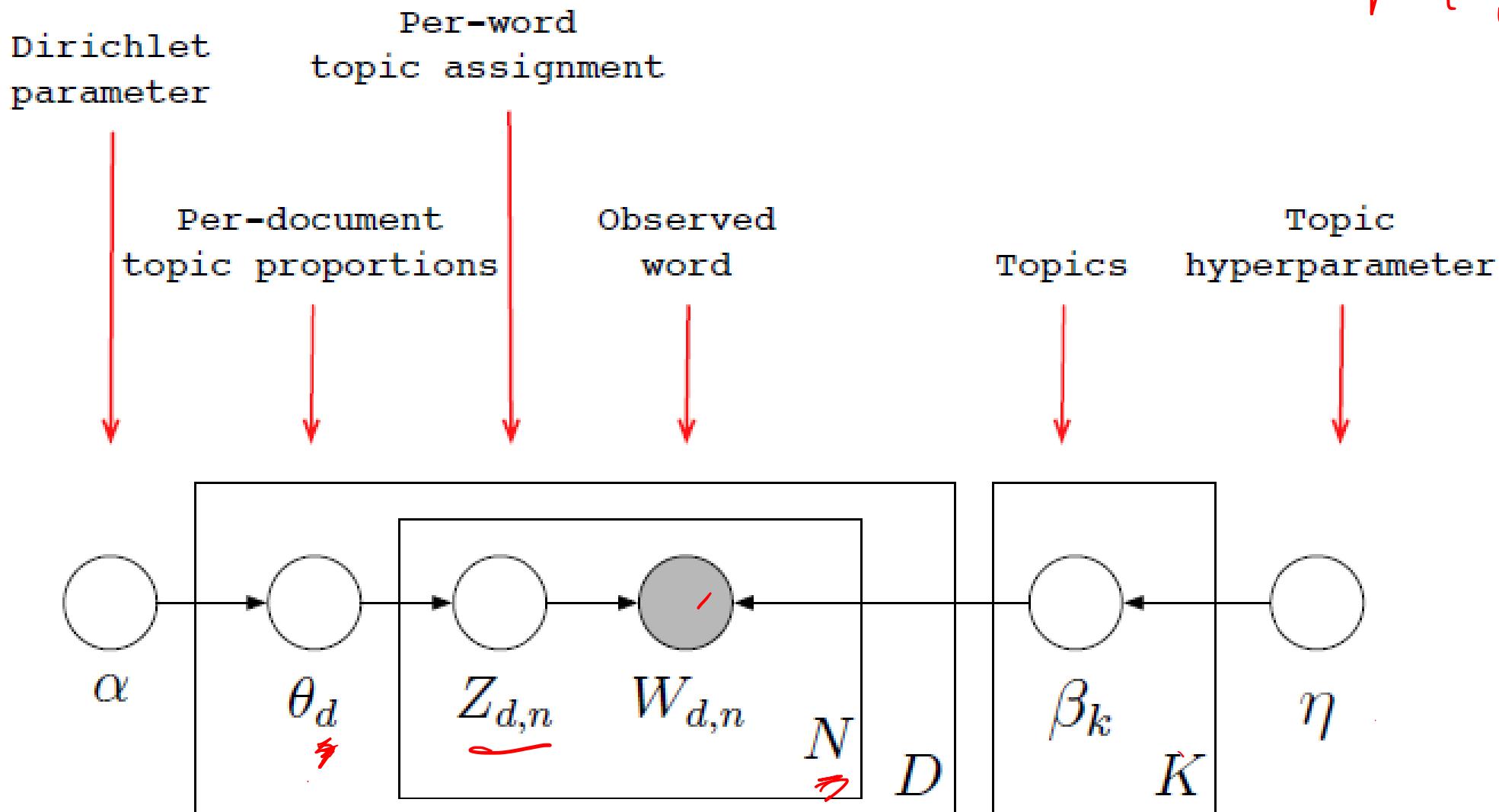
Directed graphical model



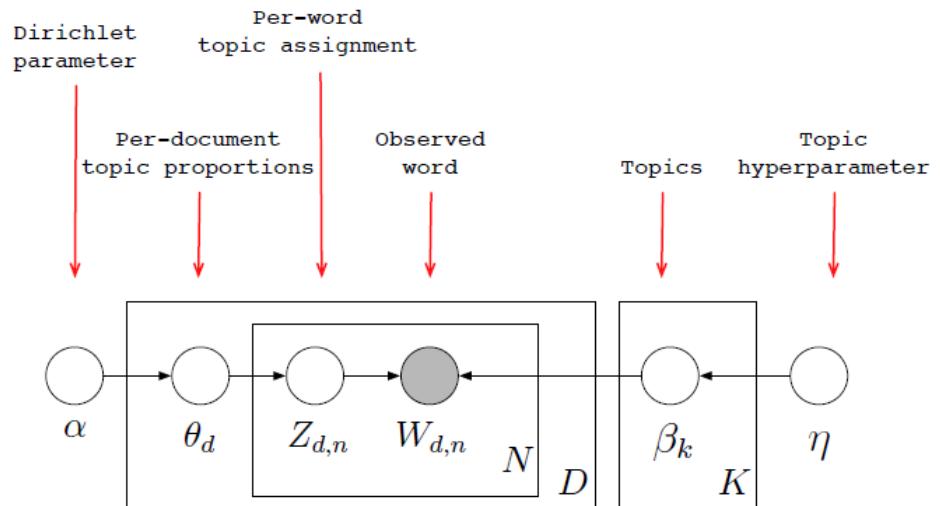
$$P(Y, x_1, x_2, \dots, x_N) = P(Y) \prod_{n=1}^N P(x_n|y)$$

Directed graphical model of LDA

$$\beta_r = \{ \beta_{r,u} \}_{u=1}^U$$

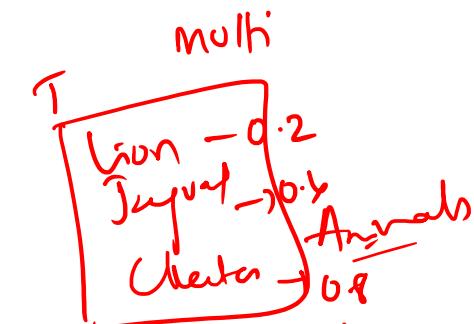


Directed graphical model of LDA (Contd..)



$$\prod_{k=1}^K P(\beta_k | n) \left(\prod_{d=1}^D P(\theta_d | \zeta) \right) \left(\prod_{n=1}^N P(z_{d,n} | \theta_d) \right) P(w_{d,n} | z_{d,n}, \beta_1, \dots, \beta_K)$$

$\sim \text{Dir}^K$ $\sim \text{Dir}^D$



- Draw each topic $\beta_i \sim \text{Dir}(n)$, for $i \in \{1, 2, \dots, K\}$

➤ For each document

➤ Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$

➤ Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$

- Draw $W_{d,n} \sim \text{Multinomial}(\theta_{z_{d,n}})$

$$\left\{ \begin{array}{l} t_1 = \beta_1 (-, -, -) \sim \text{Dir}(1) \\ t_2 = \beta_2 (-, -, -) \\ t_3 \\ \vdots \\ t_k = \beta_k (-, -, -) \\ Z_{d,n} \sim \text{mult}(t_i(\theta_d)) \\ \theta_d = 2 \\ d_1 = \theta_1 (-, -, -) \\ d_2 = \theta_2 \\ \vdots \\ d_k = \theta_k \end{array} \right.$$

Directed graphical model of LDA (Contd..)

- Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$
- For each document
 - Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$
 - Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$
 - Draw $W_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$



Thank You!

In our next session: Gibbs sampling



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mathematical modelling of LDA

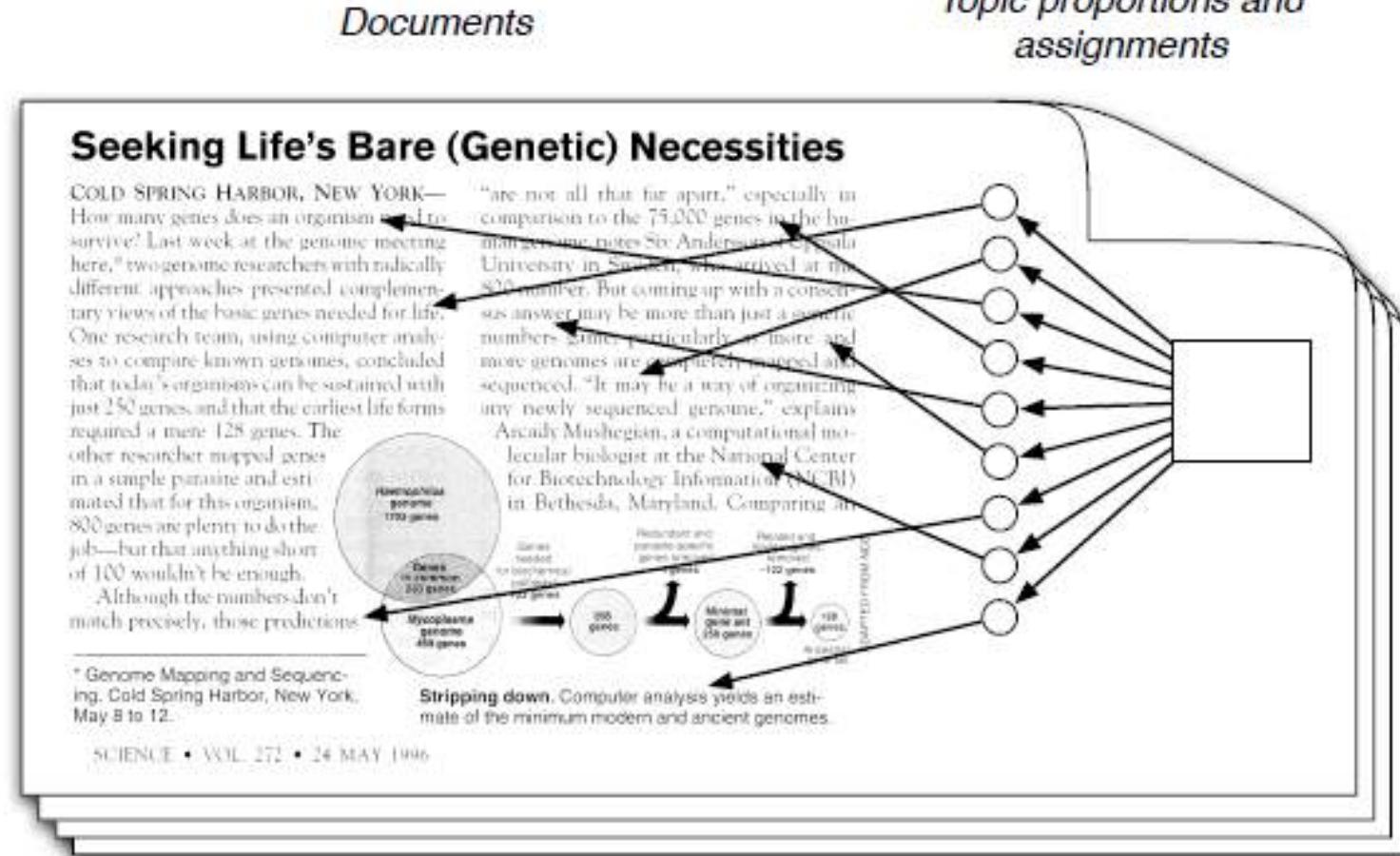
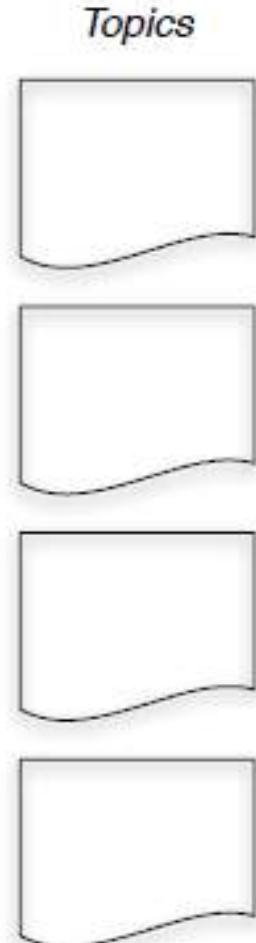
Prof. Aruna Malapati

Learning Objectives

- Generative process of modelling LDA

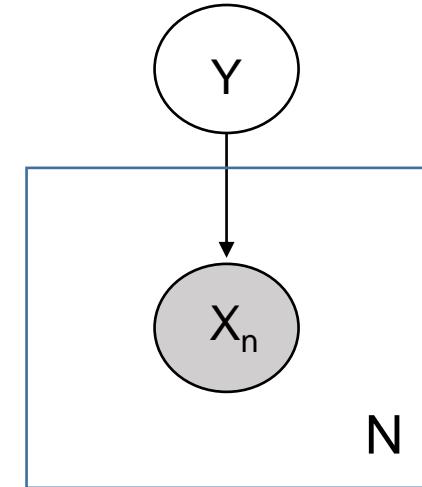
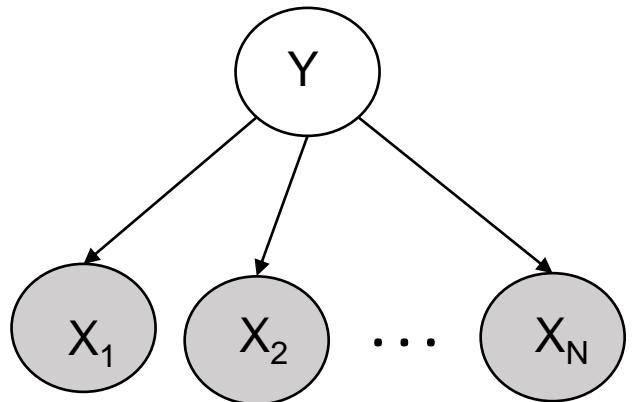


The posterior distribution

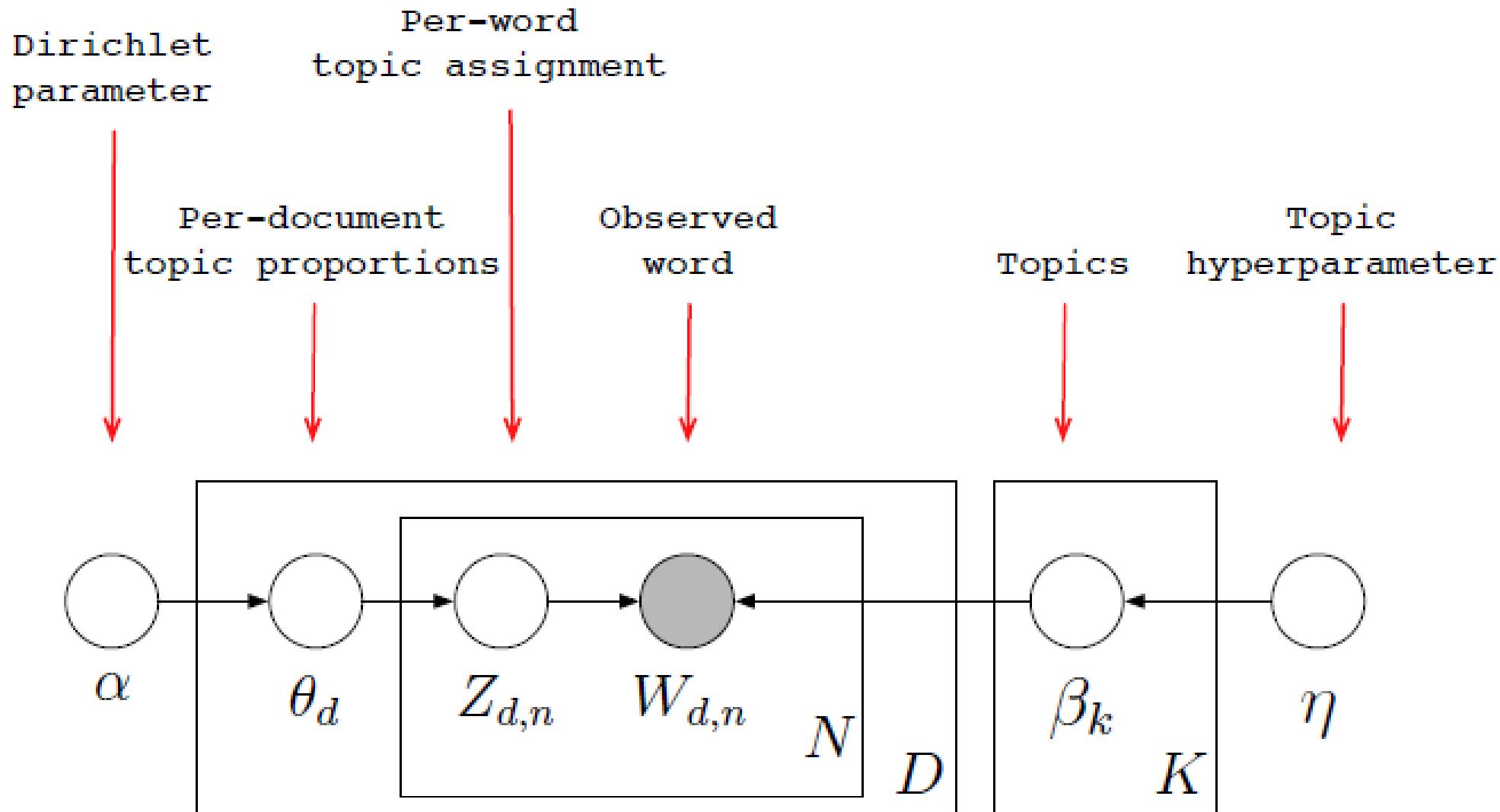




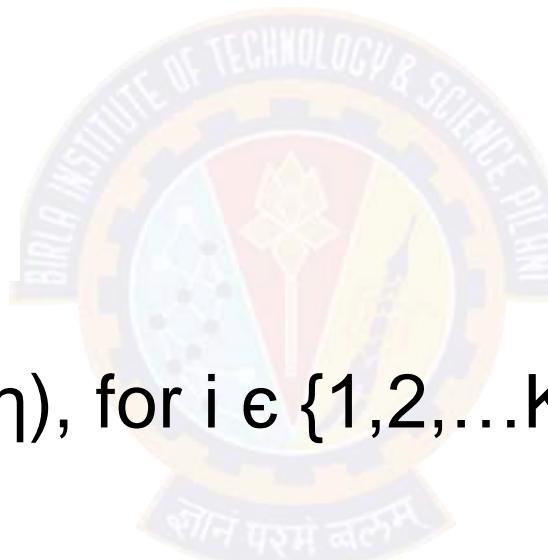
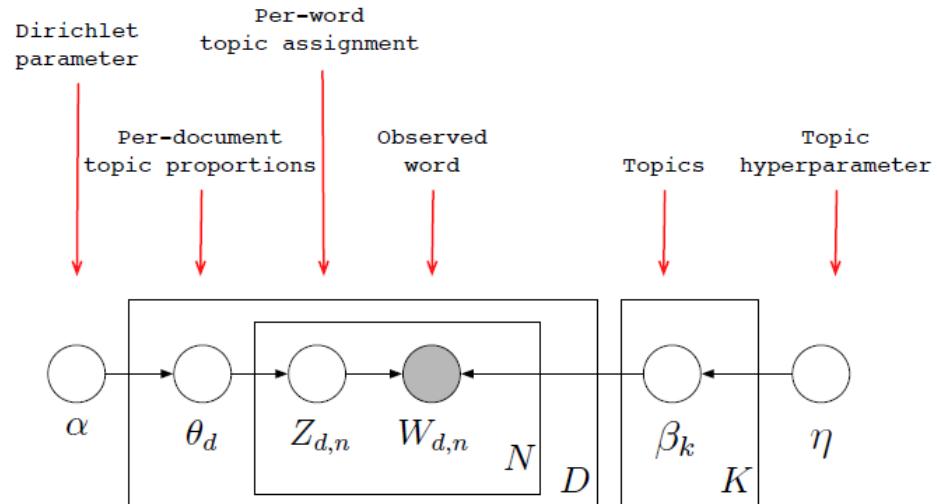
Directed graphical model



Directed graphical model of LDA



Directed graphical model of LDA (Contd..)



- Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$
- For each document
 - Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$
 - Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$
 - Draw $W_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$

- Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$
- For each document
 - Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$
 - Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$
 - Draw $W_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$



Thank You!

In our next session: Gibbs sampling



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Gibbs Sampling for Parameter Estimation

Prof. Aruna Malapati







Algorithm

- Step1: Assign a random topic [1...T] for each word
- Step2: For each word token, a new topic is sampled as per $P(z_i=j|z_{-i}, w_i, d_i)$ and the matrices C_{wt} and C_{dt} are updated.
- One iteration over all word token in the document is a Gibbs Sample
- Each iteration may have correlation with the next hence these samples are saved at spaced intervals.

Estimate for θ and β





Thank You!

In our next session: Sentiment analysis



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Sentiment Analysis

Prof.Aruna Malapati

Learning Objectives

- Motivation for Sentiment Analysis
- Facts Vs Opinions
- Sentiment Analysis definition
- Applications of Sentiment Analysis



Motivation for Sentiment Analysis

➤ What others think has always been an important piece of information.

“Which mobile should I buy?”

“Which colleges should I apply to?”

“Which employer is best to work?”

“Whom should I vote for?”



Whom should I ask?

- Pre Web
 - Friends and relatives
 - Acquaintances
 - Consumer Reports
- Post Web
 - "...I don't know who..but apparently it's a good phone. It has good battery life and..."
 - Blogs (google blogs, livejournal)
 - E-commerce sites (amazon, ebay)
 - Review sites (CNET, PC Magazine)
 - Discussion forums (forums.craigslist.org, forums.macrumors.com)
 - Friends and Relatives (occasionally)



Too many opinions?????



Different terms for the sentiment analysis

Review
Mining

Subjectivity Analysis

Sentiment Analysis

Appraisal
Extraction

Opinion Mining

Facts Vs Opinions



Text Data

Facts

Opinions

Sentiment Analysis

- Computational study of opinions, sentiments, evaluations, attitudes, appraisal, affects, views, emotions, subjectivity, etc., expressed in text.



Lots of applications

- Businesses and organizations
- Individuals
- Ads placements
- Election campaigns
- Policy Acceptance





Thank You!

In our next session: Modelling Sentiment Analysis



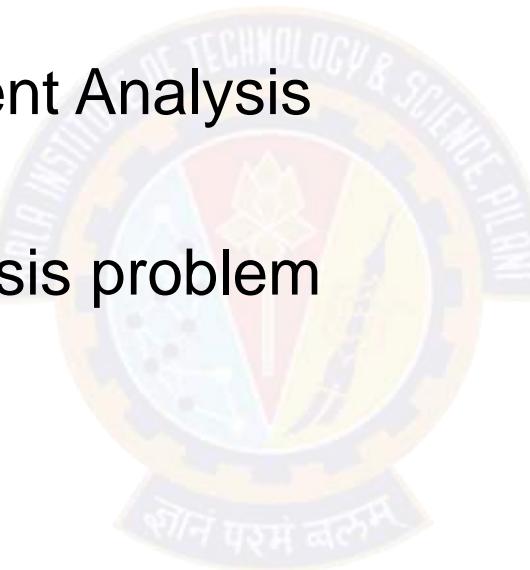
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Modelling Sentiment Analysis

Prof. Aruna Malapati

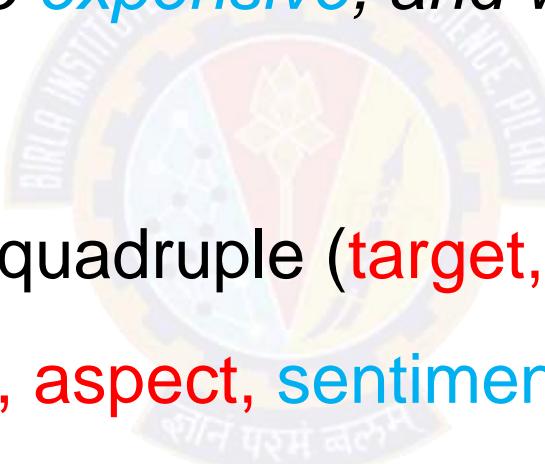
Learning Objectives

- Sentiment Analysis Problem definition
- Different Levels of Sentiment Analysis
- Modelling Sentiment Analysis problem



The Problem of Sentiment Analysis

“(1) I bought an *iPhone* a few days ago. (2) It was such a *nice phone*. (3) The *touch screen* was really *cool*. (4) The *voice quality* was *clear* too. (5) Although the *battery life* was *not long*, that is ok for me. (6) However, *my mother* was mad with me as I did not tell her before I bought it. (7) She also thought the *phone* was too *expensive*, and wanted me to return it to the shop.”



Definition: An opinion is a quadruple (**target**, **sentiment**, **holder**, **time**)

Practical Definition: (**entity**, **aspect**, **sentiment**, **holder**, **time**)

e.g, (iPhone,touch_screen,+,John,29-01-2020)

The goal of sentiment analysis is to **mine all quintuples** from the opinion documents.

Different Levels of Sentiment Analysis

- Three levels of granularity
 - Document level
 - Sentence level
 - Entity and Feature/Aspect level



Modelling Sentiment Analysis problem

- The solution to the Sentiment Analysis problem depends on the granularity of the sentiment
- Positive / Negative or 1 to 5 stars : (Binary Classification / Multiclass classification)
 - Naïve Bayes, and support vector machines (SVM), Logistic regression and Maximum Entropy etc..
- Regression if the Sentiment is a continuous value between 1 to 5



Thank You!

In our next session: Sentiment Resources: Lexicons and Datasets



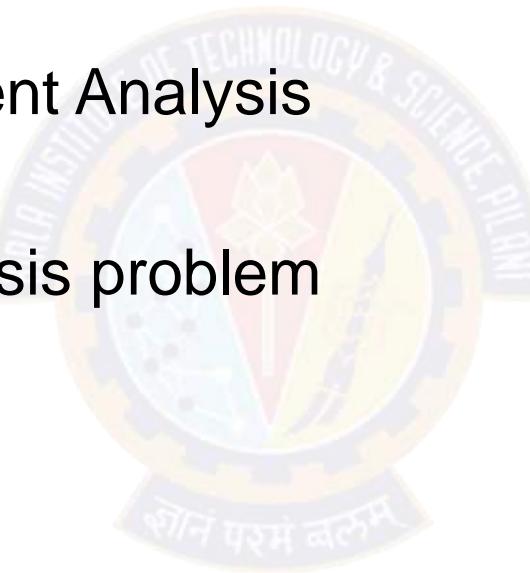
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Modelling Sentiment Analysis

Prof. Aruna Malapati

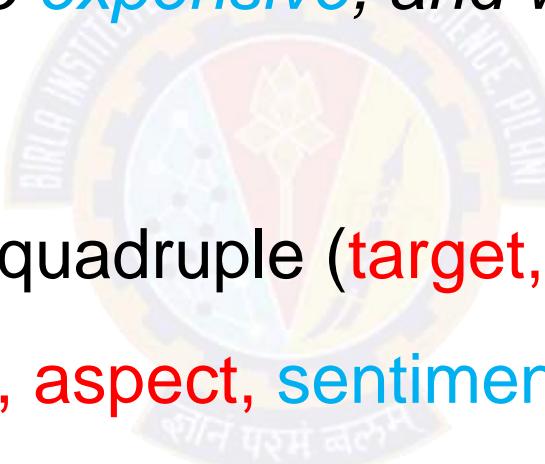
Learning Objectives

- Sentiment Analysis Problem definition
- Different Levels of Sentiment Analysis
- Modelling Sentiment Analysis problem



The Problem of Sentiment Analysis

“(1) I bought an *iPhone* a few days ago. (2) It was such a *nice phone*. (3) The *touch screen* was really *cool*. (4) The *voice quality* was *clear* too. (5) Although the *battery life* was *not long*, that is ok for me. (6) However, my mother was mad with me as I did not tell her before I bought it. (7) She also thought the *phone* was too *expensive*, and wanted me to return it to the shop.”



Definition: An opinion is a quadruple (**target**, **sentiment**, **holder**, **time**)

Practical Definition: (**entity**, **aspect**, **sentiment**, **holder**, **time**)

e.g, (iPhone,touch_screen,+,John,29-01-2020)

The goal of sentiment analysis is to **mine all quintuples** from the opinion documents.

Different Levels of Sentiment Analysis

- Three levels of granularity
 - Document level
 - Sentence level
 - Entity and Feature/Aspect level



Modelling Sentiment Analysis problem

- The solution to the Sentiment Analysis problem depends on the granularity of the sentiment
- Positive / Negative or 1 to 5 stars : (Binary Classification / Multiclass classification)
 - Naïve Bayes, and support vector machines (SVM), Logistic regression and Maximum Entropy etc..
- Regression if the Sentiment is a continuous value between 1 to 5

The Multinomial Naive Bayes' Classifier

Given a document $d = \{w_1, \dots, w_n\}$

Class $C \in \{0, 1\}$

$$P(\text{Sentiment} | w_1, \dots, w_n) = \frac{P(\text{sentiment}) \prod_{i=1}^n P(w_i | \text{sentiment})}{P(w_1, \dots, w_n)}$$

$$P(\text{sentiment} | w_1, \dots, w_n) \propto P(\text{sentiment}) \prod_{i=1}^n P(w_i | \text{sentiment})$$

$$\log P(\text{Sentiment} | w_1, \dots, w_n) \propto \log P(\text{sentiment}) + \log \prod_{i=1}^n P(w_i | \text{sentiment})$$

$$\stackrel{i \leftarrow 0}{\stackrel{-1 \text{ if}}{\stackrel{0 \text{ if}}{}}} P(\text{Sentiment}) \propto P(w_i | \text{sentiment})$$

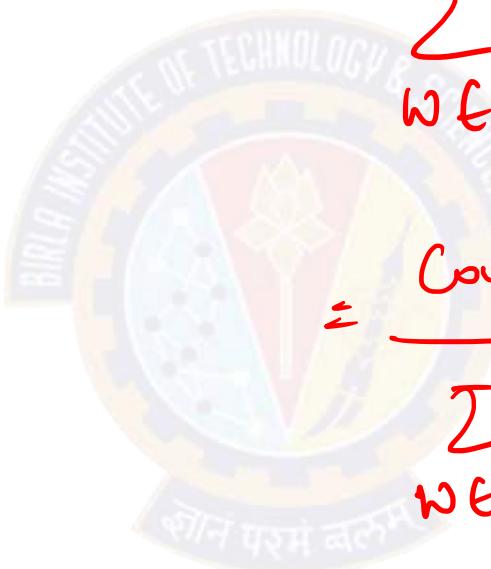
$$(d_1, C_1)(d_2, C_2) \dots (d_n, C_n)$$

$$P(C_i) = \frac{\text{no of document of class } c}{\text{total no of documents in the training dataset}}$$

$$= \frac{N_c}{N_{\text{doc}}}$$

The Multinomial Naive Bayes' Classifier

$$P(w_i|c) = \frac{\text{Count}(w_i, c)}{\sum_{w \in V} \text{Count}(w, c)}$$
$$= \frac{\text{Count}(w_i, c) + 1}{\sum_{w \in V} \text{Count}(w, c) + |V|}$$
$$= \frac{\text{Count}(w_i, c) + 1}{\sum_{w \in V} \text{Count}(w, c) + |V|}$$





Thank You!

In our next session: Sentiment Resources: Lexicons and Datasets



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Sentiment Lexicon Resources

Prof. Aruna Malapati

Learning Objectives

- Lexicons and their use
- Resources for sentiment lexicons

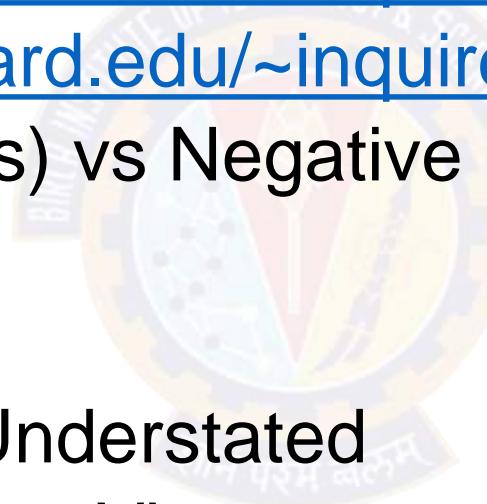


Lexicon

- Many sentiment applications rely on lexicons to supply features to a model.
- A lexicon is a **resource with information about words**.
- A sentiment lexicon has information such as list of words which are positive and negative.

General Inquirer (GI)

- Harvard General Inquirer Database (Stone, 1966)
 - Total of 11,788 terms
 - http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm
 - <http://www.wjh.harvard.edu/~inquirer/homecat.htm>
 - Positive (1915 words) vs Negative (2291 words)
 - Strong vs Weak ✓
 - Active vs Passive
 - Overstated versus Understated
 - Pleasure, Pain, Virtue, Vice
 - Motivation, Cognitive Orientation, etc



1-5
12345

A red bracket is drawn from the top of the brace down to the number 5, with the numbers 1, 2, 3, 4, and 5 written in red next to it.

MPQA Subjectivity Cues Lexicon

➤ Home page:

http://www.cs.pitt.edu/mpqa/subj_lexicon.html

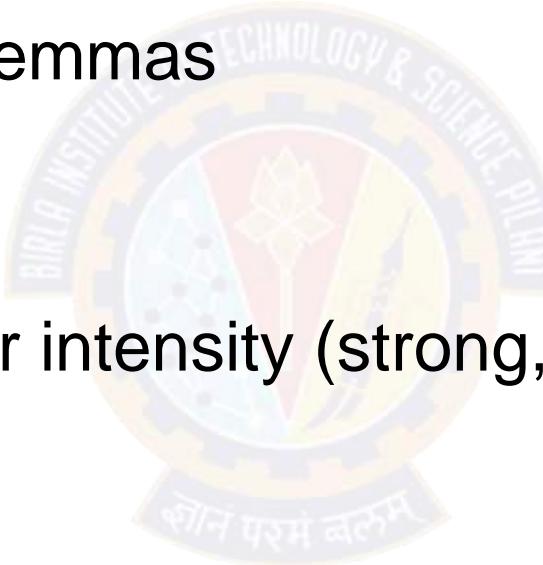
➤ ~~6885 words from 8221 lemmas~~

➤ 2718 positive

➤ 4912 negative

➤ Each word annotated for intensity (strong, weak)

➤ GNU GPL



Theresa Wilson, Janyce Wiebe, and Paul Hoffmann (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proc. of HLT-EMNLP-2005.

Riloff and Wiebe (2003). Learning extraction patterns for subjective expressions. EMNLP-2003.

LIWC Linguistic Inquiry & Word Count

- Home Page: <http://www.liwc.net/>
- 2300 word > 70 classes
- Affective Processes
- Negative emotion (bad, weird, hate, problem,tough) ✓
- Positive emotion (love,nice,sweet) ✓
- Cognitive Processes

Bing Liu Opinion Lexicon

- [Bing Liu's Page on Opinion Mining](#)
- <http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar>
- 6786 words
 - 2006 positive
 - 4783 negative



Disagreements between polarity lexicons

	Opinion Lexicon ✓	General Inquirer	SentiWordNet
MPQA ✓	33/5402 (<u>0.6%</u>)	49/2867 (<u>2%</u>)	1127/4214 (<u>27%</u>)
Opinion Lexicon ✓		32/2411 (<u>1%</u>)	1004/3994 (<u>25%</u>)
General Inquirer ✓			520/2306 (<u>23%</u>)
SentiWordNet ✓			

Christopher Potts, [Sentiment Tutorial](#), 2011



Thank You!

In our next session: Automatically generate lexicons



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Generating custom based sentiment lexicons

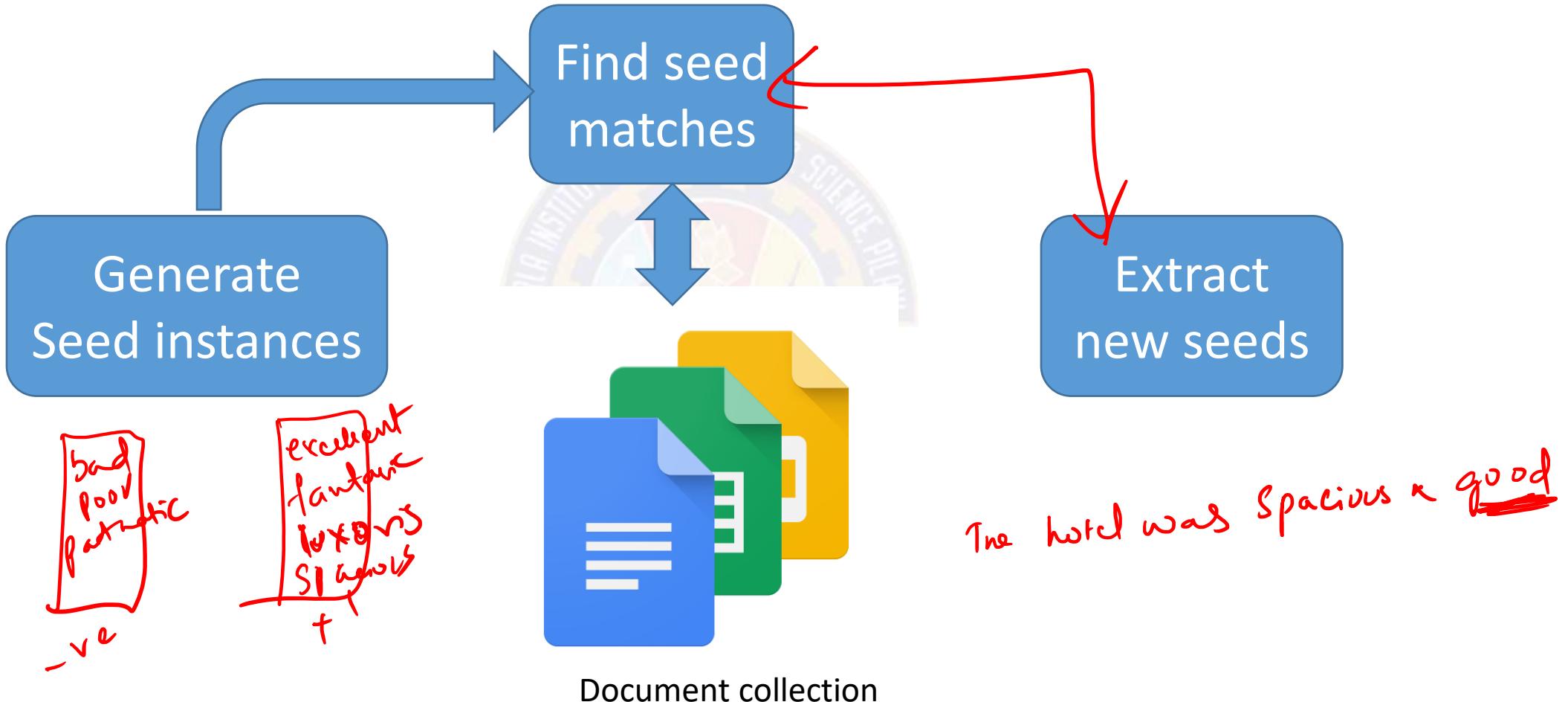
Prof. Aruna Malapati

Learning Objectives

- Bootstrapping
- Corpus based lexicon generation



Bootstrapping Architecture



Corpus-based lexicon generation

- A more sophisticated technique is a corpus-based approach which relies on syntactic or co-occurrence patterns together with a seed list of opinion words.
- The technique starts with a list of seed opinion adjective words, and uses them and a set of linguistic constraints or conventions on connectives to identify additional adjective opinion words and their orientations.
- For example “This house is beautiful and spacious”

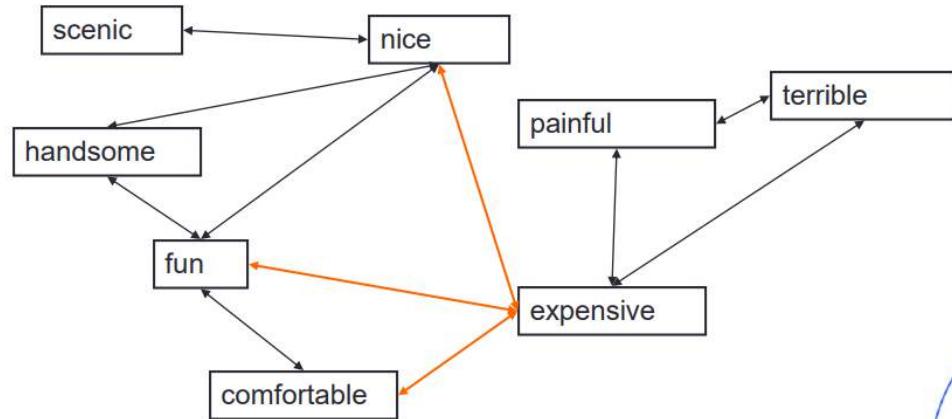
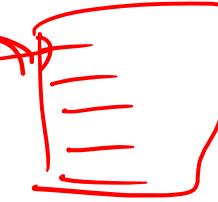
~~spacious and lux
good but bad pair Adjective And~~



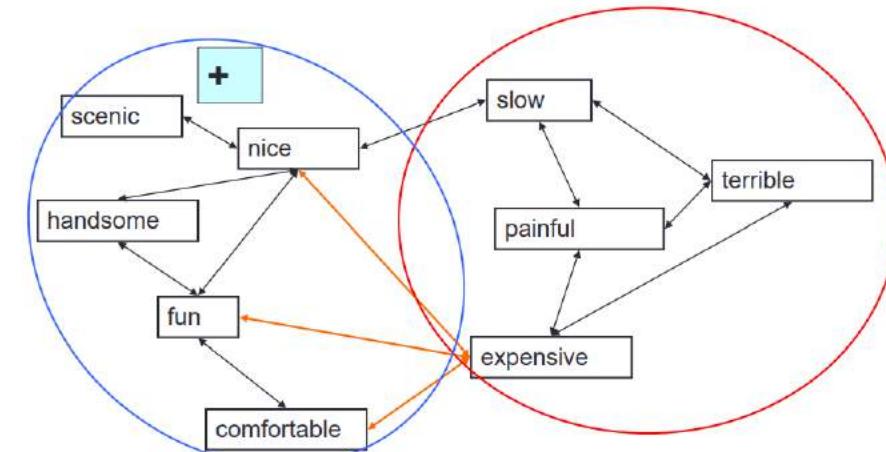
Algorithm

- Generate a Labeled seed set of adjectives
- Expand seed set to conjoined adjectives by looking up in a corpus/web search
- A supervised learning algorithm builds a graph of adjectives linked by the same or different semantic orientation

Spacious and
but



1977



- A clustering algorithm partitions the adjectives into two subsets

Turney Algorithm

- Extract a phrasal lexicon from reviews
- Learn polarity of each phrase
- Rate a review by the average polarity of its phrases

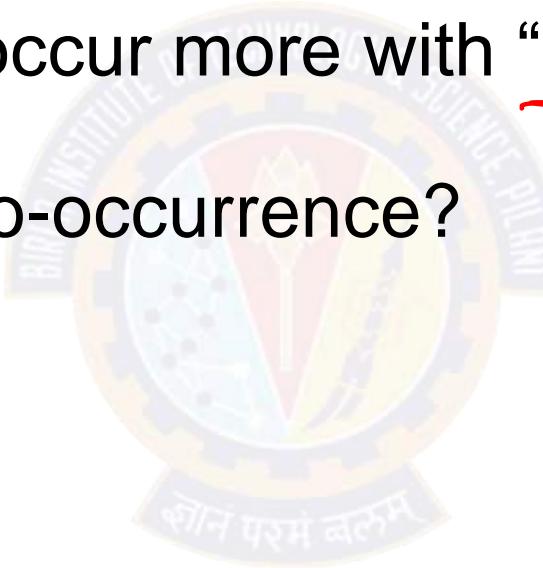


First Word	Second Word	Third Word (not extracted)
JJ <i>Adj</i> <i>Adjective</i>	NN or NNS	<u>anything</u>
RB, RBR, RBS <i>Adverb</i>	JJ <i>Adj</i>	Not NN nor NNS <u> </u>
JJ <i>Adj</i>	JJ <i>Adj</i>	<u>Not NN or NNS</u>
NN or NNS	JJ	Not NN nor NNS
RB, RBR, or RBS	VB, VBD, VBN, VBG <i>Verb</i>	anything

Two-word phrases with adjectives

How to measure polarity of a phrase?

- Positive phrases co-occur more with “excellent”
- Negative phrases co-occur more with “poor”
- But how to measure co-occurrence?



Pointwise Mutual Information

- Pointwise mutual information: How much more do events x and y co-occur than if they were independent?

$$\text{PMI}(\underline{\text{word}}_1, \underline{\text{word}}_2) = \log_2 \frac{P(\text{word}_1, \text{word}_2)}{P(\text{word}_1)P(\text{word}_2)}$$

- If two words are statistically independent, $\text{PMI}=0$ $\log_2 \frac{P(w_1)P(w_2)}{P(w_1)P(w_2)} = \log_2 1 = 0$
- If two words tend to not at all co-occur , PMI is negative $\log_2 \frac{P(\tilde{w}_1, \tilde{w}_2)}{P(w_1)P(w_2)} = -\infty$
- If two words tend to co-occur , PMI is positive

Does phrase appear more with “poor” or “excellent”?

➤ Polarity(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor")

SO(phrase) =

$$\log_2 \left[\frac{\text{hits}(\text{phrase NEAR "excellent"}) \text{ hits}(\text{"poor"})}{\text{hits}(\text{phrase NEAR "poor"}) \text{ hits}(\text{"excellent"})} \right]$$



Spacious rooms are excellent

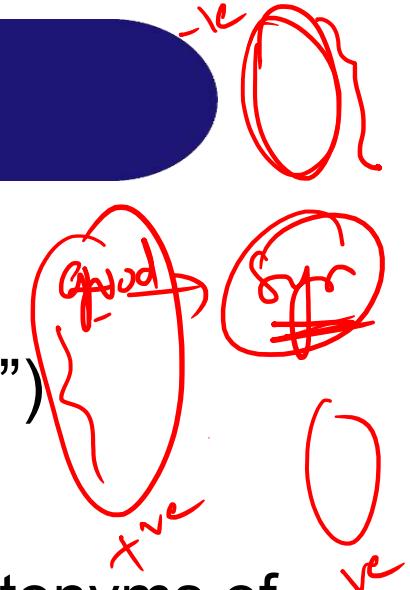
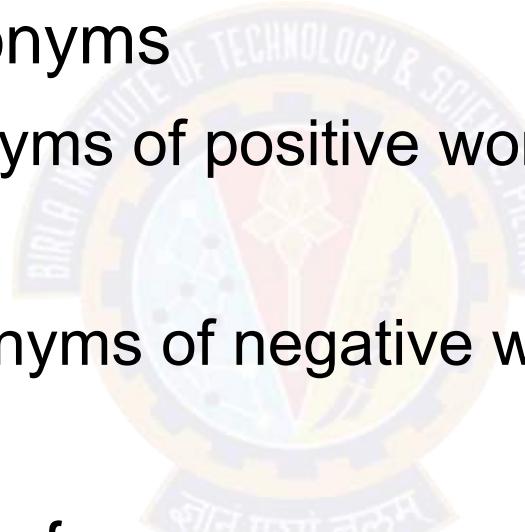
Two reviews for Positive and Negative phrases

Phrase	POS tags	Polarity
online service	JJ NN	2.8
online experience	JJ NN	2.3
direct deposit	JJ NN	1.3
local branch	JJ NN	0.42
...		
low fees	JJ NNS	0.33
true service	JJ NN	-0.73
other bank	JJ NN	-0.85
inconveniently located	JJ NN	-1.5
Average		0.32

Phrase	POS tags	Polarity
direct deposits	JJ NNS	5.8
online web	JJ NN	1.9
very handy	RB JJ	1.4
...		
virtual monopoly	JJ NN	-2.0
lesser evil	RBR JJ	-2.3
other problems	JJ NNS	-2.8
low funds	JJ NNS	-6.8
unethical practices	JJ NNS	-8.5
Average		-1.2

Wordnet based polarity estimation

- WordNet: online thesaurus indexing words by synonyms
- Create positive (“good”) and negative seed-words (“terrible”)
- Find Synonyms and Antonyms
 - Positive Set: Add synonyms of positive words (“well”) and antonyms of negative words
 - Negative Set: Add synonyms of negative words (“awful”) and antonyms of positive words (“evil”)
- Repeat, following chains of synonyms
- Filter





Thank You!

In our next session: Aspect based Sentiment Analysis



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Aspect Based Sentiment Analysis

Prof. Aruna Malapati

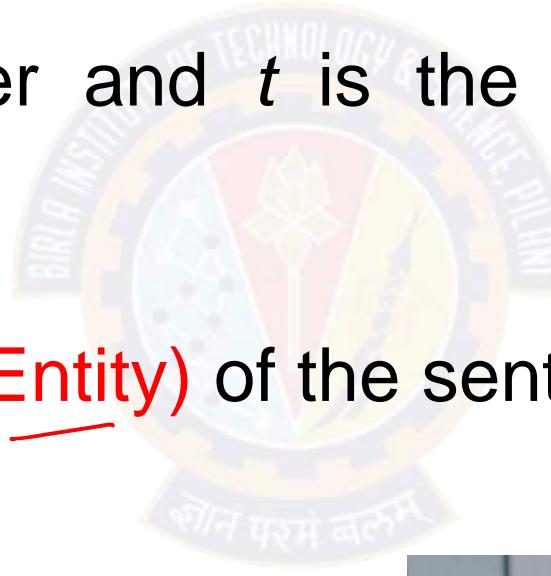
Learning Objectives

- Aspect Based Sentiment Analysis (ABSA)
- Frequency based Aspect Extraction



Aspect Based Sentiment Analysis (ABSA)

- Each opinion is defined as quintuple (e, a, s, h, t) , where e is an entity and a is one of its aspects, s is the sentiment on the aspect a , h is the opinion holder and t is the time when the opinion is expressed.



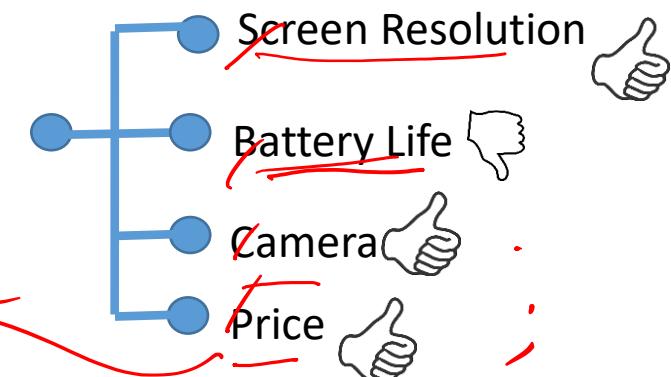
I bought an iPhone and the voice quality was extremely good.

- Find the target(Aspect/Entity) of the sentiment.

- Two approaches

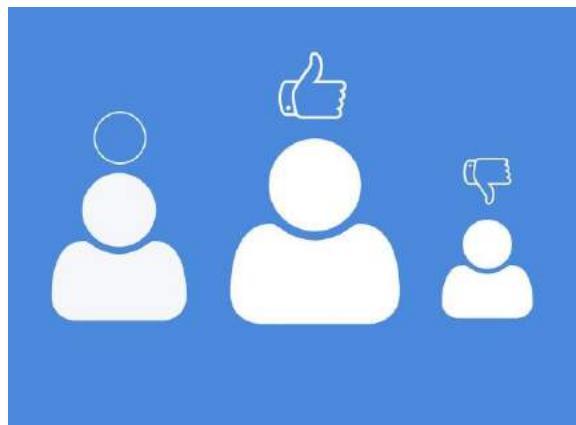
- Find most common noun phrases

- Build a classifier



Frequency-Based Aspect Extraction

- A key characteristic is that an **opinion always has a target**.
- Exploit **syntactic structures** to depict opinion and target relationships



Review corpus



Association Rule Mining

$t_1 \leftarrow$
.....
.....
.....

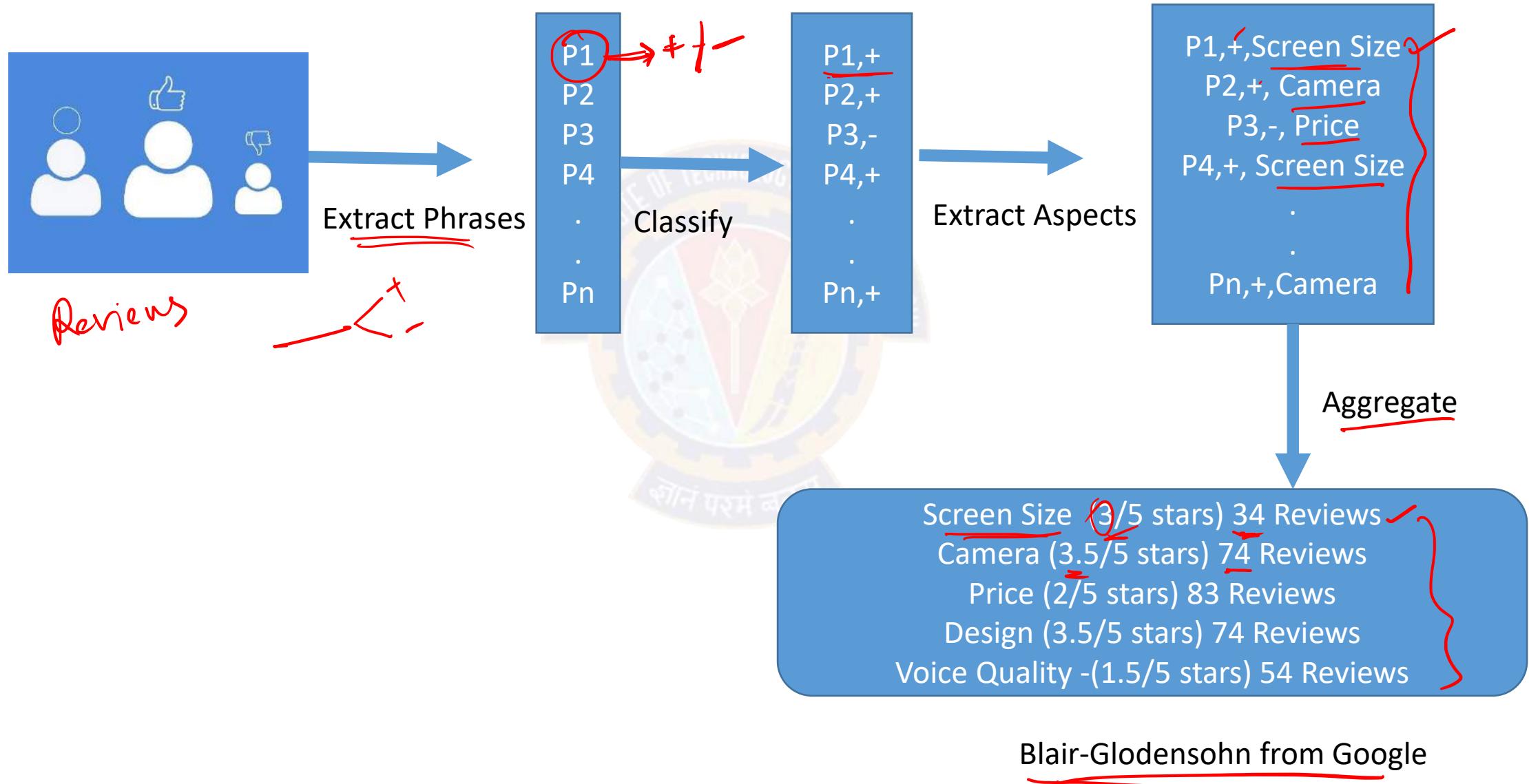
Screen Size – 100/500 ✓
Camera Resolution – 300/500
Battery Life – 350/500
Price - 450/500
Voice clarity – 325/500

Examples of aspects extracted

Entity	Aspects extracted
Casino	Casino, <u>buffet</u> , <u>pool</u> , <u>resort</u> , <u>beds</u>
Department store	Selection, department, sales, shop, clothing
Greek Restaurant	Food, Wine, Service, Appetizer, lamb

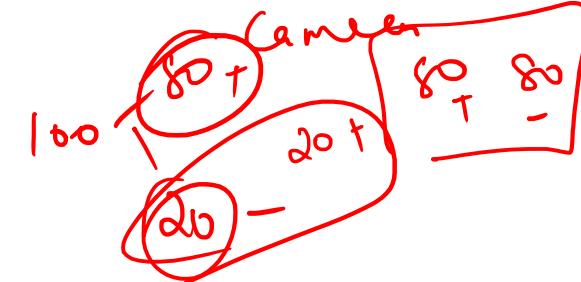
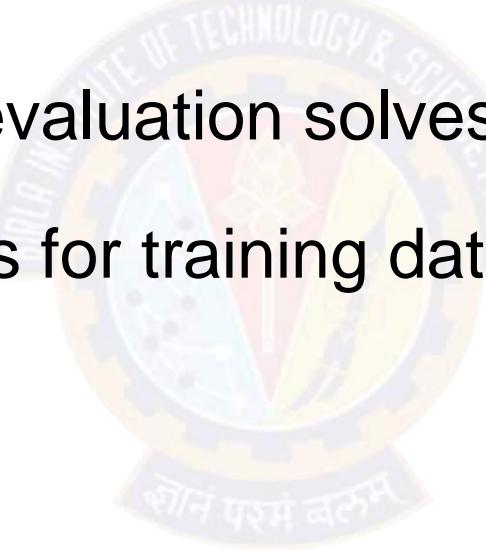
- Those **candidate aspects with the highest frequency counts** are almost always the most **important aspects of the product**.
- Assumption: Corpus has **reasonable number of reviews** and belong to same product.

Architecture for Aspect Based Sentiment Analysis



Assumptions

- The baseline algorithm assumes that the number of reviews for Positive and Negative sentiment are of equal frequencies.
- Usage of F-Score for evaluation solves this problem.
- Use Sampling methods for training data



How to deal with star ratings?

8

- Binarization of the star ratings. $\begin{cases} 0 & \text{if } r < 2.5 \\ 1 & \text{if } r \geq 2.5 \end{cases}$
- Use regression instead of a binary classifier.

~~5~~
~~+~~
~~-~~
10





Thank You!

In our next session: Opinion Spamming



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Opinion Spamming

Prof. Aruna Malapati

Opinion Spamming

- Types of Spam
 - Type 1 (fake reviews)
 - Type 2 (reviews about brands only)
 - Type 3 (non-reviews)



Types of Data, Features and Detection

- Three main types of data have been used for review spam detection:
 - Review content
 - Meta-data about the review
 - Product information



Supervised Spam Detection

- Opinion spam detection can be formulated as a classification problem with two classes, fake and non-fake.
- Due to the fact that there is no labeled training data for learning, Jindal and Liu (2008) exploited duplicate reviews.
- In their study of 5.8 million reviews and 2.14 million reviewers from amazon.com, a large number of duplicate and near-duplicate reviews were found.

Four categories to handle duplicates and near duplicates

- Duplicates from the same user-id on the same product
- Duplicates from different user-ids on the same product
- Duplicates from the same user-id on different products
- Duplicates from different user-ids on different products

Feature engineering for fake reviews

- Review centric features
- Reviewer centric features
- Product centric features



Some interesting observations from the study

- Only reviews of some products are likely to be fake.
- Top-ranked reviewers are more likely to be fake reviewers.
- Products of lower sales ranks are more likely to be spammed.



Thank You!

In our next session: Introduction to Recommender Systems



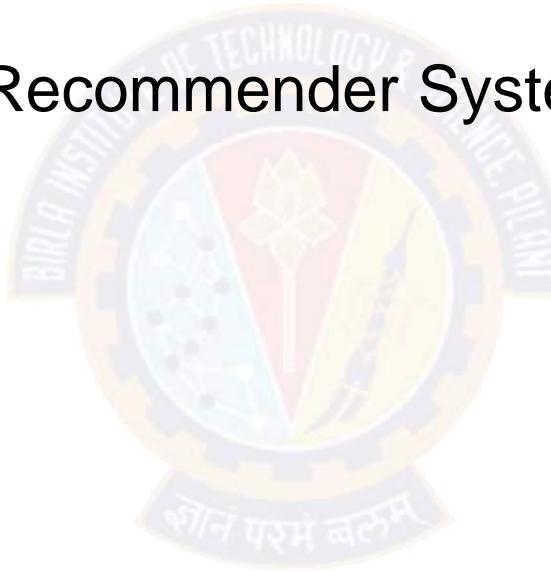
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Recommender Systems

Prof. Aruna Malapati

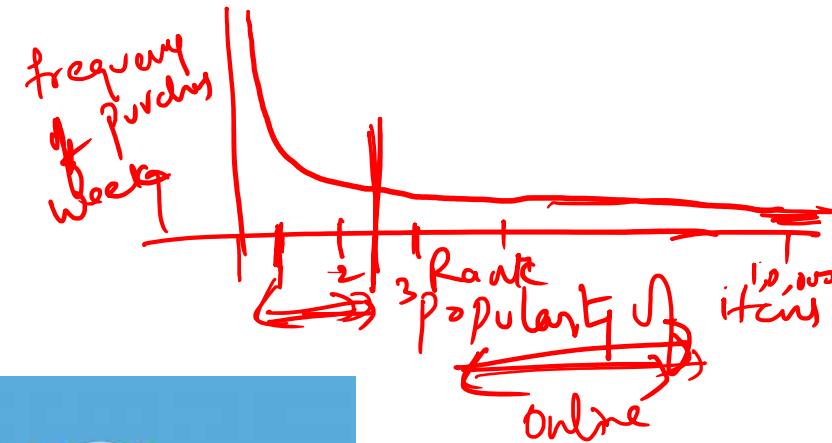
Learning Objectives

- Motivation for Recommender Systems
- Modelling the problem of Recommender Systems



➤ Motivation for Recommender Systems

Search Vs Recommender systems



Commercial Interests for Recommender systems

- Netflix: 2/3 of the movies watched
- Amazon: 35% sales
- Google news: recommendations ⇒ 38% more click through
- Choicestream: 28% of people would buy more music if they found they like the recommendation.

Modelling Recommender Systems

➤ $U = \{\underline{\text{USERS}}\}$

U I

➤ $I = \{\underline{\text{ITEMS}}\}$

➤ F is a utility function, measures the usefulness of items I to user U.

$F: \underline{U} \times \underline{I} \rightarrow R$ where R is the rating
0,1
0 - 5

➤ Characteristics of a good Utility function

➤ Personalized

➤ Diverse

➤ Serendipity

Netflix Utility Matrix



Movie/User	Movie-1	Movie-2	Movie-3	Movie-4
User-1	4	3.5	5	5
User-2	4	3	4.5	?
User-3	4.5	4	2	3

2006-2009

Improvement in RMSE by 10%

Prize: \$1 Million

Netflix dataset: over 17K movies and 500K+ Users!

100
100

1-5

Problems faced while building recommender systems

- Gathering the rating from users.
 - ~~Implicit~~ ^{Explicit (1-5)}
- Developing right models to learn function from known ratings
- Evaluating the models for unknown rating



Thank You!

In our next session: Collaborative Filtering



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Collaborative Filtering

Prof. Aruna Malapati

Learning Objectives

- Baseline method for predicting the ratings
- Define collaborative filtering problem
- User/Item based collaborative filtering



Baseline approach for rating prediction

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?

bew Avg - μ

$$\text{Predicted Rating}(\underline{E}, \underline{\text{Equilibrium}}) = \underline{\mu} + \underline{b_E} + \underline{b_{\text{Equilibrium}}}$$

Where μ is the global average

b_E is the deviation of the user E

$b_{\text{Equilibrium}}$ is the deviation of the Movie Equilibrium

$$\text{Predicted Rating} = \underline{3.78} + \underline{0.22} + \underline{0.22} = \underline{4.22}$$

Collaborative Filtering

- The task of **predicting** user preferences on new items is by **collecting taste of similar users.**

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?

- Each user has expresses an **opinion** for some items
 - Explicit opinion: Rating score ✓
 - Implicit: Purchase records or listen to the tracks ✗

User Based Collaborative Filtering

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?

- Identify the set of items rated by the target user.
- Identify which other users have rated the same items as target user.
- Compute similarity of each user to the target user.
- Select top K similar users

Finding similar users

➤ Let r_x and r_y be the vector of users x and y 's ratings

➤ Jaccard similarity measure between x and y

➤ Problem: Ignores the value of the rating

➤ Cosine similarity measure

$$\text{sim}(x, y) = \cos(r_x, r_y) = \frac{r_x \cdot r_y}{\|r_x\| \cdot \|r_y\|}$$

➤ Problem: Treats missing ratings as “negative”

➤ Pearson correlation coefficient

➤ S_{xy} = items rated by both users x and y

$$Sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

\bar{r}_x, \bar{r}_y ... avg.
rating of x, y

any
 $x \cup y$

$k=2$
 $C \rightarrow A, D$

Estimating the Predicted Rating

- Let N be the set of k users most similar to x who have rated item i
 - Prediction for item s of user x :

$$r_{x_i} = \frac{1}{k} \sum_{y \in N} r_{yi} \rightarrow$$

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$$

$$C=2$$

Item Based Collaborative Filtering

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?



User Based Vs Item Based

- User based Similarity is more **dynamic** and can be used if
 - Item base is smaller than user base
 - Item base rapidly changes rapidly
- Item based Similarity is **static** and recommends new items that were also liked by the same users
 - Good if the use base is small

||| |||
==

Issues in implementing Collaborative Filtering

- Many Items to choose from
- Few data per user
- No data for new users
- Very large datasets





Thank You!

In our next session: Recommender Systems Prerequisite for Matrix factorization



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Recommender Systems Prerequisite for Matrix factorization-1

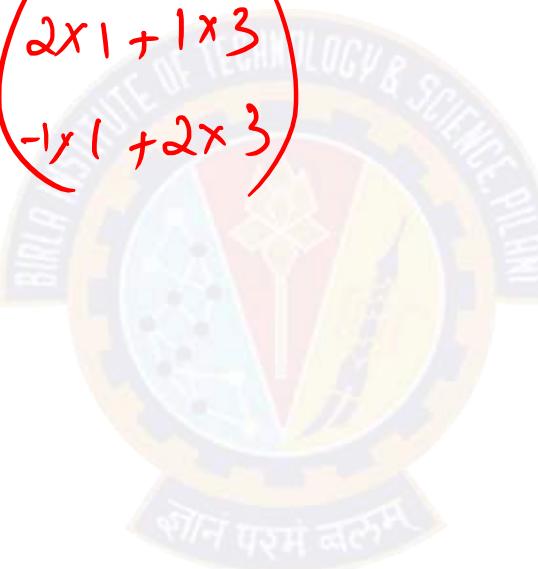
Prof. Aruna Malapati

Matrix vector multiplication

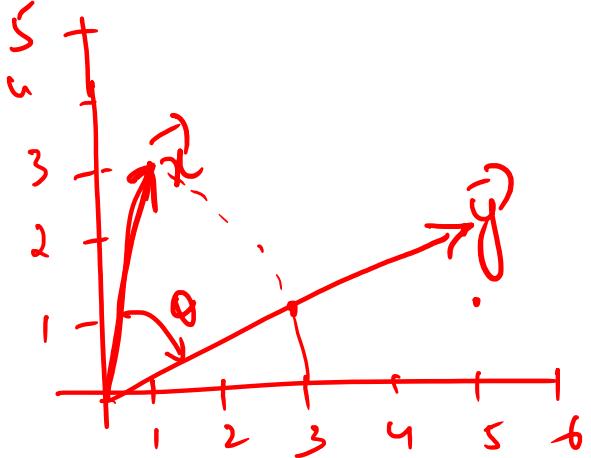
$$\vec{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad M = \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}$$

$$\vec{y} = M\vec{x} = \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \times 1 + 1 \times 3 \\ -1 \times 1 + 2 \times 3 \end{pmatrix}$$

$$\vec{y} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$



Geometric intuition



$$M = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

$$M = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix} \quad \alpha \rightarrow \text{stretching}$$

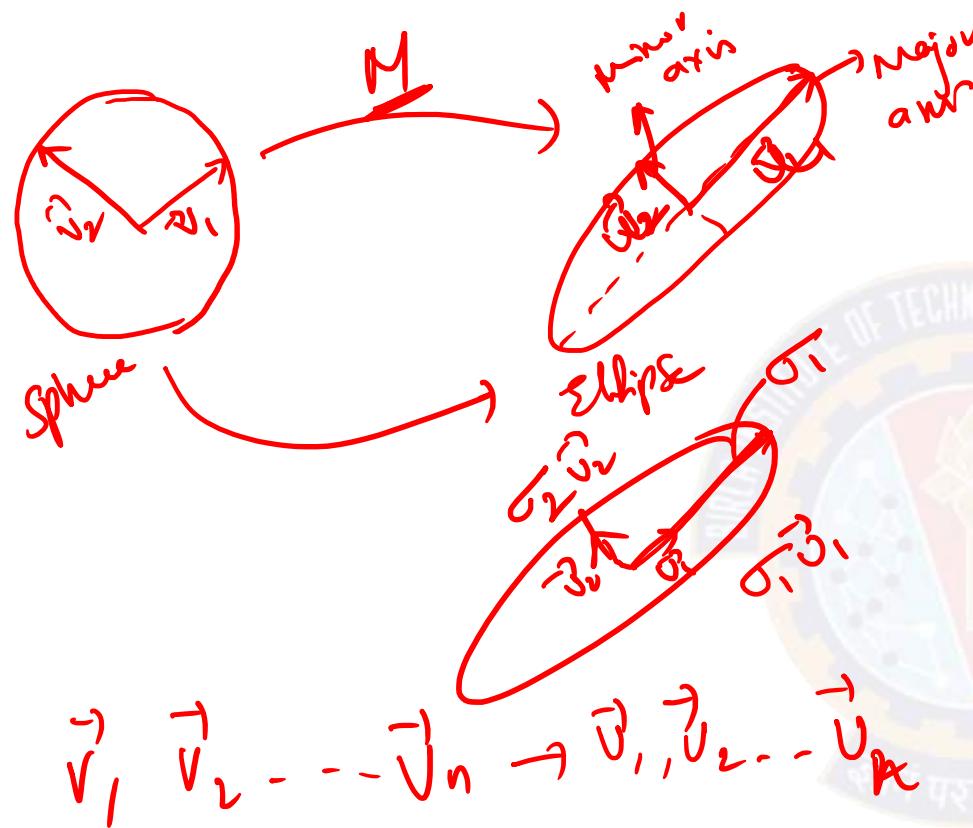
$\alpha > 1$
 $\alpha < 1$

$$\vec{y} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \times 1 + 0 \times 3 \\ 0 \times 1 + 2 \times 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

$$\alpha = 0$$



Matrix vector multiplication in higher dimensions



new coordinate space
 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n \xrightarrow{\text{Matrix multr}} \vec{u}_1, \vec{u}_2, \dots, \vec{u}_n$ principal axis
 $\sigma_1, \sigma_2, \dots, \sigma_n$ stretch factor

$$M \vec{v}_i = \sigma_i \vec{u}_i$$
$$A \vec{x} = \lambda \vec{x}$$
$$M \vec{v}_j = \sigma_j \vec{u}_j \quad j=1 \dots n$$

Matrix vector multiplication in higher dimensions(contd..)

$$\begin{bmatrix} & \vdots \\ \vdots & \end{bmatrix} \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}_{n \times n} = \begin{bmatrix} \vec{u}_1 \ \vec{u}_2 \ \dots \ \vec{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

$$AV = \hat{u} \sum \hat{v}$$

↑ rotation
↑ stretch

$$A_{m \times n} = V \sum_{m \times n} U^T$$

Unitary matrix

$$AA^T = I$$

!!

$$AA^T = \bar{U}$$

$$V^{-1} = V^*$$
$$U^{-1} = V^*$$
$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$$

$$A = \hat{u} \sum \hat{v}^*$$
$$= \hat{u} \sum \hat{v}^T$$



Thank You!

In our next session: Recommender Systems using SVD



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Recommender Systems Using SVD

Prof. Aruna Malapati

SVD Theorem

$$\underline{\mathbf{A}}_{[m \times n]} = \mathbf{U}_{[m \times \cancel{r}]} \Sigma_{[\cancel{r} \times \cancel{r}]} (\mathbf{V}_{[n \times \cancel{r}]})^T$$

A: **Input data matrix**

- $m \times n$ matrix (e.g., m users, n movies)

U: **Left singular vectors**

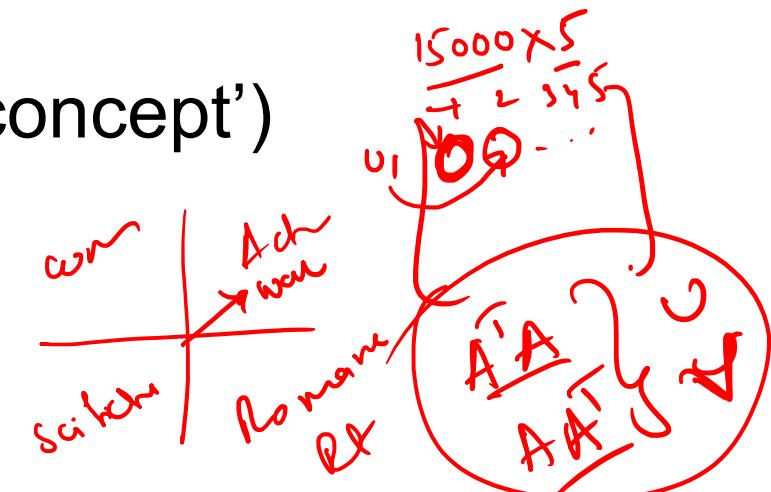
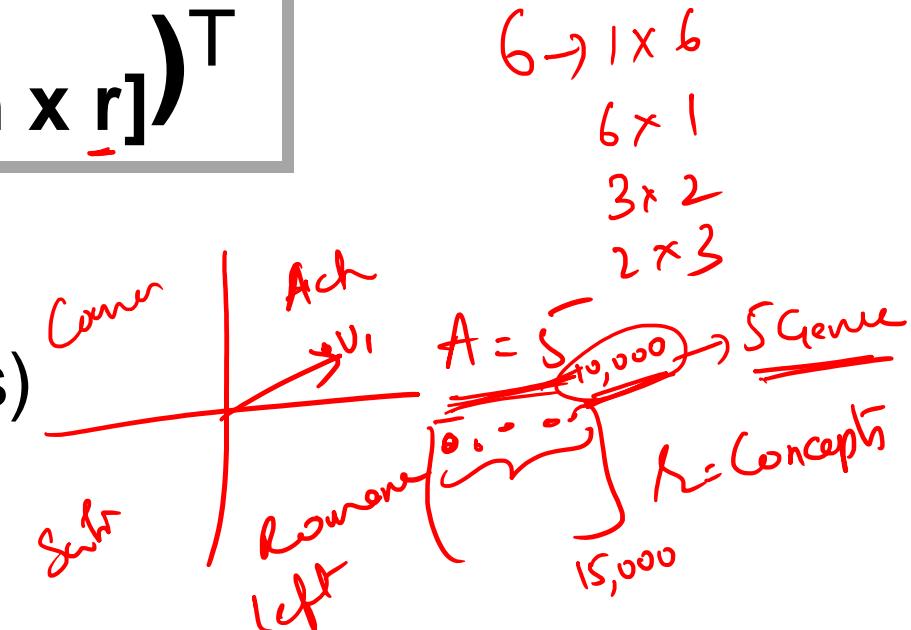
- $m \times r$ matrix (m users, r concepts)

Σ : **Singular values**

- $r \times r$ diagonal matrix (strength of each 'concept')
(r : rank of the matrix \mathbf{A})

V: **Right singular vectors**

- $n \times r$ matrix (n movies, r concepts)



Singular Value Decomposition

➤ The key issue in an SVD decomposition is to find a lower dimensional feature space where the new features represent “concepts” and the strength of each concept in the context of the collection is computable.

- The core of the SVD algorithm lies in the following theorem
- It is always possible to decompose a given matrix A into $\underline{A = U \Sigma V^T}$

Example

User to Movie Utility Matrix 3×4

$$\begin{bmatrix} 4 & 1 & 1 & 4 \\ 1 & 4 & 2 & 0 \\ 2 & 1 & 4 & 5 \end{bmatrix} = \begin{bmatrix} u_1 & c_1 & c_2 & c_3 \\ u_2 & -0.61 & 0.28 & -0.74 \\ u_3 & -0.29 & -0.95 & -0.12 \\ & -0.74 & 0.14 & 0.66 \end{bmatrix} \times \begin{bmatrix} 8.87 & 0.00 & 0.00 & 0.00 \\ 0.00 & 4.01 & 0.00 & 0.00 \\ 0.00 & 0.00 & 2.51 & 0.00 \end{bmatrix} \times \begin{bmatrix} -0.47 & -0.28 & -0.47 & -0.69 \\ 0.11 & -0.85 & -0.27 & 0.45 \\ -0.71 & -0.23 & 0.66 & 0.13 \\ -0.52 & 0.39 & -0.53 & 0.55 \end{bmatrix}$$

User to Concept 3×4

Strength of each concept 4×1

Movie to Concept 4×4

$$\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2} = \sqrt{(-0.61)^2 + (0.28)^2 + (-0.74)^2 + (0.14)^2 + (8.87)^2 + (4.01)^2 + (2.51)^2 + (-0.47)^2 + (0.11)^2 + (-0.71)^2 + (-0.52)^2 + (-0.28)^2 + (-0.27)^2 + (0.66)^2 + (0.39)^2 + (-0.53)^2 + (0.45)^2 + (0.13)^2 + (0.55)^2}$$

B

$$\begin{bmatrix} 2.69 & 0.57 & 2.22 & 4.25 \\ 0.78 & 3.93 & 2.21 & 0.04 \\ 3.17 & 1.38 & 2.92 & 4.78 \end{bmatrix} = \begin{bmatrix} -0.61 & 0.28 \\ -0.29 & -0.95 \\ -0.74 & 0.14 \end{bmatrix} \times \begin{bmatrix} 8.87 & 0.00 \\ 0.00 & 4.01 \end{bmatrix} \times \begin{bmatrix} -0.47 & -0.28 & -0.47 & -0.69 \\ 0.11 & -0.85 & -0.27 & 0.45 \end{bmatrix}$$

Reconstructed Matrix

User to Concept

Strength of each concept

Movie to Concept

$$\rightarrow \begin{bmatrix} -0.23 & -0.81 \end{bmatrix} \quad 3 \times 2$$

$\|B\|_F \approx \sqrt{\sum_{ij} B_{ij}^2}$

$$\|A-B\|_F = \sqrt{\sum_{ij} (A_{ij}-B_{ij})^2}$$

Recommendations for new User

➤ How to use SVD for recommendations?

$$\mathbf{u}_{new} = \mathbf{u} \times \mathbf{V}_{m \times r} \times \mathbf{S}^{-1}_{r \times r}$$

$$\begin{bmatrix} -0.23 & -0.89 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} -0.47 & 0.11 \\ -0.28 & -0.85 \\ -0.47 & -0.27 \\ -0.69 & 0.45 \end{bmatrix} \times \begin{bmatrix} 0.11 & 0.00 \\ 0.00 & 0.25 \end{bmatrix}$$

new
user
feature
vector

$\mathbf{U}_{4 \times 4}$ $\mathbf{V}_{4 \times 2}$ \mathbf{S}^{-1}

$$\begin{bmatrix} 8.87 & 0.00 \\ 0.00 & 4.01 \\ \cancel{0.00} & \cancel{0.00} \end{bmatrix} \Sigma$$

Strength of each concept

$$\begin{bmatrix} \mathbf{v}_1 & -0.47 & -0.28 & -0.47 & -0.69 \\ \mathbf{v}_2 & 0.11 & -0.85 & -0.27 & 0.45 \end{bmatrix}$$

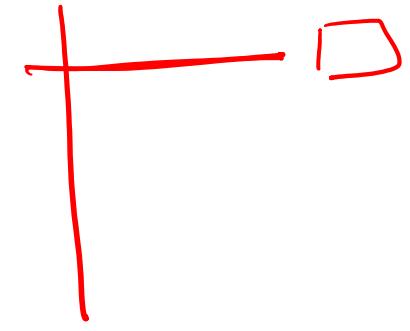
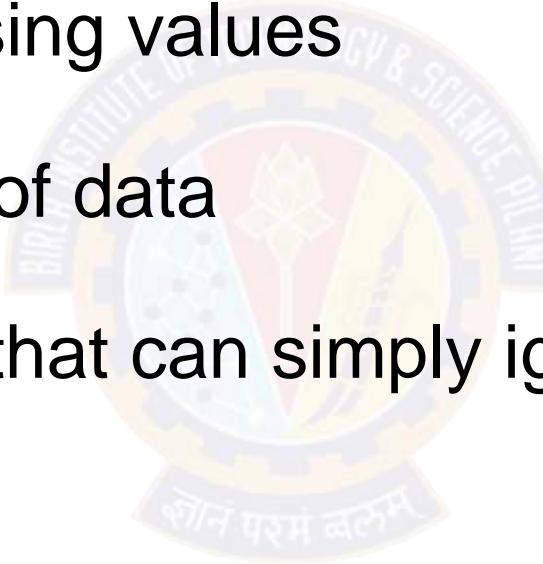
Movie to Concept

$$\sqrt{\Sigma}$$

$$\begin{aligned} & (1x -0.47 + 4x -0.28 + 1x -0.47 + 0x -0.69) \begin{pmatrix} 1 \\ 8.87 \\ 0 \\ 0.45 \end{pmatrix} \\ & \begin{bmatrix} 2.06 & -3.54 \end{bmatrix} \mathbf{S}^{-1} \\ & 1 \times 2 = \begin{bmatrix} -0.23 & -0.89 \end{bmatrix} \end{aligned}$$

Drawbacks of SVD

- Conventional SVD is undefined for incomplete matrices!
- Imputation to fill in missing values
- Increases the amount of data
- We need an approach that can simply ignore missing ratings





Thank You!

In our next session: Latent Factor Model



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Recommender Systems Using Latent Factor Models

Prof. Aruna Malapati

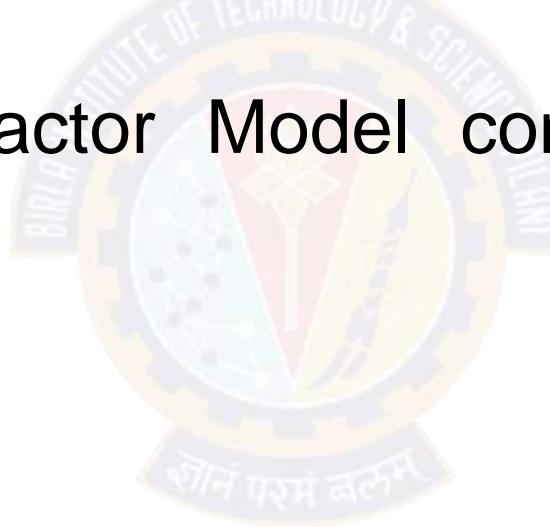
Learning Objectives

- Motivation for Latent Factor Models
- Extracting the Latent Factors for the utility matrix
- Computing the User and Movie Latent factors using Stochastic Gradient Descent



Drawback of SVD

- Though SVD gives us the best rank approximation, the major problem is that it is **not defined for missing entries.**
- Hence the Latent Factor Model comes handy which draws inspiration from SVD.



User – Movie interactions in real world

$U_2 = U_3 + U_4$

dependent

User/ Movie	M1	M2	M3	M4	M5
User1	1	3	2	5	4
User2	2	1	1	1	5
User3	3	2	3	1	5
User4	2	4	1	5	2



User/ Movie	M1	M2	M3	M4	M5
User1	4	4	4	4	4
User2	4	4	4	4	4
User3	4	4	4	4	4
User4	4	4	4	4	4

User/ Movie	M1	M2	M3	M4	M5
User1	1	3	2	5	4
User2	2	1	1	1	5
User3	3	2	3	1	5
User4	2	4	1	5	2

User/ Movie	M1	M2	M3	M4	M5
User1	3	1	1	3	1
User2	1	2	4	1	3
User3	3	1	1	3	1
User4	4	3	5	4	4

$U_1 = U_2 + U_3 + U_4$ $M_1 = M_2 \dots M_5$

independant

$$U_1 = U_3 \\ M_1 = M_4$$

$$\frac{U_2 + U_3}{M_5} = U_4 \\ M_5 = \text{Avg}(M_2, M_3)$$

Movie / User Features

➤ Movie Features



Comedy



Action



Horror



Drama

➤ User Preferences

➤ Likeliness of genre of movies (0/1)

Movie / User Features

Movie/User	Comedy	Action
M1	3	1
M2	1	2
M3	1	4
M4	3	1
M5	1	3

f_1 f_2

User/Movie	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

f_1 f_2

$$1 \times 1 + 1 \times 3 + 1 + 3 = 4$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

movie 1 3 1
8 1 3

$$v_{11} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$3 \times 1 + 1 \times 0 = 3$$

$$v_{31} = \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad 3 \times 1 + 1 \times 0 = 3$$

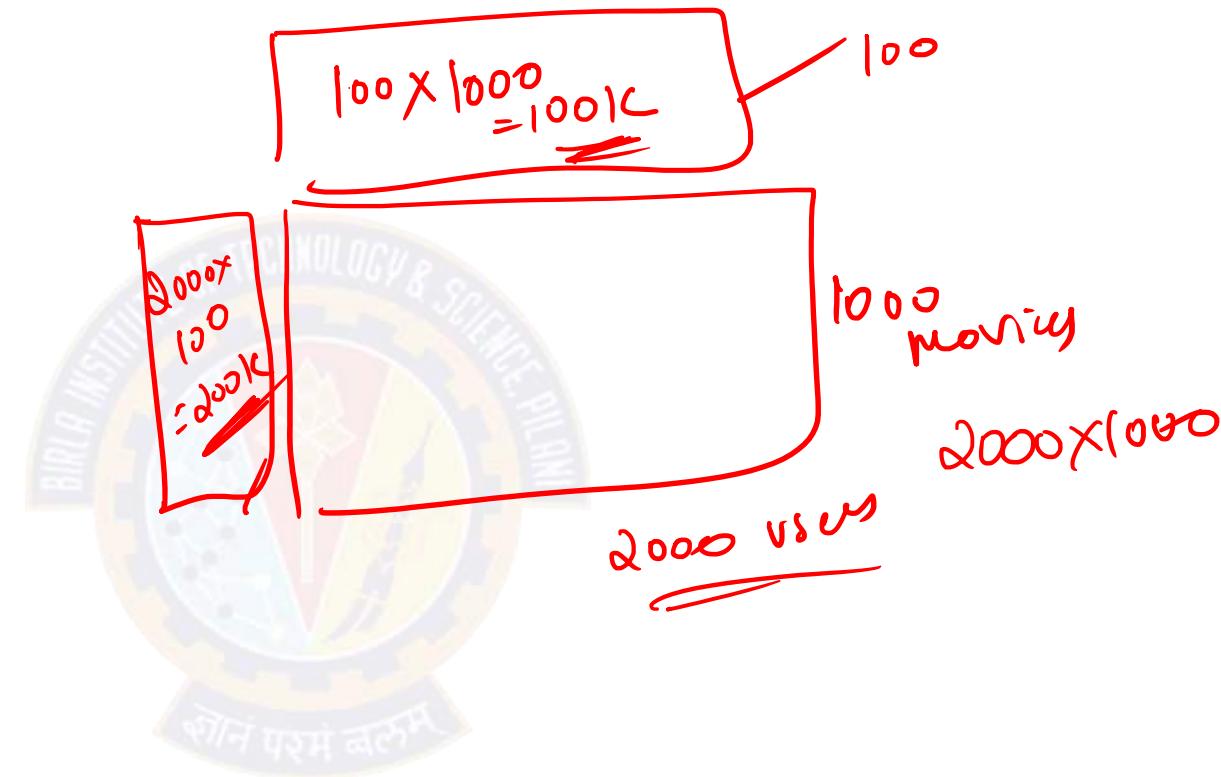
Latent Factors using Matrix Factorization

Movie/ Features	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

User/Features	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

User/ Movie	M1	M2	M3	M4	M5
User1	3	1	1	3	1
User2	1	2	4	1	3
User3	3	1	1	3	1
User4	4	3	5	4	4

Benefit of Matrix Factorization-Storage



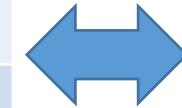
Stochastic Gradient Descent (SGD)

User/Features	F1	F2
User1	0.2	0.5
User2	0.3	0.4
User3	0.7	0.8
User4	0.4	0.5

Movie/Features	M1	M2	M3	M4	M5
Comedy	1.2	3.1	0.3	2.5	0.2
Action	2.4	1.5	4.4	0.4	1.1

complaint
visibility matrix

User/Movie	M1	M2	M3	M4	M5
User1	1.44	1.37	2.26	0.7	0.59
User2	1.32	1.53	1.85	0.91	0.5
User3	2.76	3.37	3.73	2.07	1.02
User4	1.68	1.99	2.32	1.2	0.63



User/Movie	M1	M2	M3	M4	M5
User1	3	1	1	3	1
User2	1	2	4	1	3
User3	3	1	1	3	1
User4	4	3	5	4	4

Err_{uv} = $(3 - 1.44)^2 + (1 - 1.37)^2 + \dots$

derivative

Rating predictions

User/Features	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

m x k

Movie/ Features	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

k x n

User/ Movie	M1	M2	M3	M4	M5
User1	3		1	3	1
User2	1		4	1	
User3	3	1		3	1
User4		3		4	4

m x n

$$\begin{aligned}
 & U_i M_j \\
 & (1 \ 0) \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 3 \times 1 + 1 \times 0 = 3
 \end{aligned}$$



Thank You!

In our next session: Mathematical formulation of Latent Factor Model



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mathematical formulation of Latent Factor Model

Prof. Aruna Malapati

Learning Objectives

- Formulate the objective function for Latent Factor Model
- Apply Stochastic Gradient Descent to solve the objective function



Objective function

$$R \approx P_u Q_i^T = \hat{R}$$

P

Movie/ Features	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

$$\hat{r}_{ui} = P_u Q_i^T = \sum_{k=1}^K P_{uk} Q_{ki}^T$$

User/Features	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

User/ Movie	M1	M2	M3	M4	M5
User1	3		1		1
User2	1		4	1	
User3	3	1		3	1
User4		3		4	4

Our goal is to find two matrices P and Q such that the following objective function is minimized on **test data**:

$$e_{ui}^2 = (r_{ui} - \hat{r}_{ui})^2$$

$$= \left(r_{ui} - \sum_{k=1}^K p_{uk} q_{ki}^T \right)^2$$

Stochastic Gradient Descent

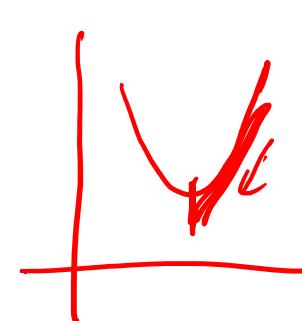
$$\frac{\partial e_{ij}^2}{\partial p_{uk}} = -2(\underline{r_{ui}} - \hat{r}_{ui})(q_{ki})$$
$$= -2e_{ui}q_{ki}$$

$$\frac{\partial e_{ij}^2}{\partial q_{ki}} = -2(\underline{r_{ui}} - \hat{r}_{ui})(\underline{p_{uk}})$$
$$= -2e_{ui}\underline{p_{uk}}$$

$$p^{\text{new}} = p^{\text{old}} - \eta \nabla p$$

$$q^{\text{new}} = q^{\text{old}} - \eta \nabla q$$

$$e_{ui}^2 = (\underline{r_{ui}} - \hat{r}_{ui})^2$$
$$p^{\text{new}} = p^{\text{old}} - \eta (-2e_{ui}q_{ki})$$
$$= p_{uk} + 2\eta e_{ui}q_{ki}$$



$$q^{\text{new}} = q^{\text{old}} - \eta (-2e_{ui}p_{uk})$$
$$= q_{ki} + 2\eta e_{ui}p_{uk}$$

Stochastic Gradient Descent (Contd..)

6 Parikh

$$\underline{e_{ui}} = \left(r_{ui} - \sum_{k=1}^K p_{uk} q_{ki} \right)^2 + \lambda \left(\| \underline{p_u} \|^2 + \| \underline{q_i} \|^2 \right)$$

$$p_u = \begin{bmatrix} p_{u1} \\ p_{u2} \\ p_{u3} \end{bmatrix} \quad q_i = \begin{bmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \end{bmatrix}$$

$$\begin{aligned} p_{uk}^{new} &= p_{uk}^{old} + \eta \frac{\partial e_{ui}}{\partial p_{uk}} = p_{uk} + \eta (2e_{ui} q_{ki} - \lambda p_{uk}) \\ &= q_{ki} + \eta (2e_{ui} p_{ui} - \lambda q_{ki}) \end{aligned}$$

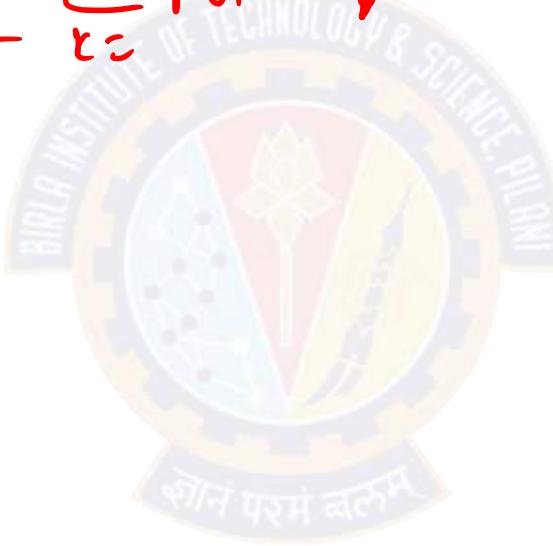
2006-2007

$$\hat{r}_{ui} = \mu + b_u + b_i + \sum_{k=1}^K p_{uk} q_{ki}$$

Other Parameters

$$\hat{q}_{ij} = \mu + b_v(t) + b_i(t) + \sum_{k=1}^K p_{vk} q_{kj}$$

Q_v
~~520~~





Thank You!

In our next session: Metrics used for evaluating Recommender Systems



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Metrics used for evaluating Recommender Systems

Prof. Aruna Malapati

Accuracy of estimated rating / Error Based

- Mean Absolute Error (MAE)
- Mean square error (MSE)
- Root Mean Squared Error(RMSE)

User/Movie	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6
User1✓	2			4	4✓	
User2	5		4			1
User3			5		2	
User4		1		5		4
User5			4			2
User6	4	5		1		

Training Set Test Set

Accuracy of estimated rating / Error Based (Contd..)

test set

User Id	Movie Id	Actual	Predicted	MAE	MSE	RMSE ✓
User1 ✓	4 ✓	4 ✓	2	2	4	4
User1	5	4	1	3	9	9
User2	6	1	1.5	0.5	0.25	0.25
User3	5	2	2	0	0	0
User4	4	5	4.5	0.5	0.25	0.25
User4	6	4	3	1	1	1
User5	6	2	2	0	0	0
User6	4	1	3	2	4	4
				9/8	18.5/8	Sqrt(18.5/8)

Σ

$$MAE = \frac{\sum |P - R|}{\# \text{ ratings}}$$

$$MSE = \frac{\sum (P - R)^2}{\# \text{ ratings}}$$

$$RMSE = \sqrt{\frac{\sum (P - R)^2}{\# \text{ ratings}}}$$

Precision at rank K

Movie	User 1	Actual	Predicted
1	1	4	2.3
2	1	2	3.6
3	1	3	3.4
4	1	?	4.4
5	1	5	4.5
6	1	?	2.3
7	1	2	4.9
8	1	?	4.3
9	1	?	3.3
10	1	4	4.3

Ignore the values whose actual is not known and sort the items in Descending order of the predicted rating.

≥ 3.5

~~Note~~
 Item7 2/4.9
 item5 5/4.5
 item10 4/4.3
 item2 2/3.6
 item3 3/3.4
 item1 4/2.3



Let's assume that all rating ≥ 3.5 are relevant then the recommender system will suggest the following

Item7 2/4.9
 item5 5/4.5
 item10 4/4.3

compute the precision-at-Rank 3

Recall 7/3 5/3 10/3
 Precision 1/1 1/2 2/3



Thank You!

In our next session: Deep Learning