IQRA NATIONAL UNIVERSITY PESHAWAR

NAME: RAIMA ZULFIQAR AHMED
ID: 14321
SUBJECT: Natural Language Processing
Department : Computer Science (BS-SE)
SEMESTER: 5TH
SUBMITTED TO : MAAM ASMAA

Question No. 1:

Explain objectives of NLP? Write the name of 2 Applications of NLP with example? Write the name of 2 Challenges of NLP with example?

ANSWER:

Natural Language Processing, usually shortened as **NLP**, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate **objective of NLP** is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.

APPLICATIONS:

Spell and Grammar Checking

- Checking spelling and grammar
- Suggesting alternatives for the errors

8

Information Extraction

- Includes named-entity recognition

Helicopters will patrol the temporary no-fly zone around New Jersey's MetLife Stadium Sunday, with F-16s based in Atlantic City **ready** to be scrambled if an unauthorized aircraft does enter the restricted airspace.

Down below, **bomb-sniffing** dogs will patrol the trains and buses that are expected to take approximately 30,000 of the **80,000-plus** spectators to Sunday's Super Bowl between the Denver Broncos and Seattle Seahawks.

The Transportation Security Administration said it has added about two dozen dogs to monitor passengers coming in and out of the airport around the Super Bowl.

http://cogcomp.cs.illinois.edu/page/demo_view/Wikifier

19

CHALLENGES:

Ambiguity

- One word/sentence can have different meanings
  - Fall
    - The third season of the year
    - Moving down towards the ground or towards a lower position
  - The door is open.
    - Expressing a fact
    - A request to close the door

51

Syntax and ambiguity

- I saw the man with a telescope.
  - Who had the telescope?

(http://www.realtytrac.com/landing/2009-year-end-foreclosure-report.html)

53

Question No. 2:

Specify the text strings using the below regular expressions:

a. /a(bc)

- Given string: ab abc ac acb a0b a2b a42c A87d

ANS:

ab abc ac acb a0b a2b a42c A87d

b. /[abc]

- Given string: ab abc ac acb a0b a2b a42c A87d

ANS:

ab abc ac acb a0b a2b a42c A87d

c. /abc+

- Given string: ab abc abcc babc

ANS:

ab abc abcc babc

d. /abc*

- Given string: ab abc abcc babc

ANS:

ab abc abcc babc

e. /[^a-z A-Z 0-9]

- Given string: a89 opx cfff $1!

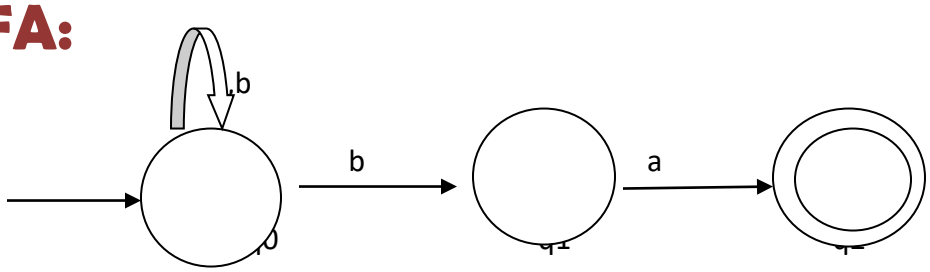
ANS:

a89 opx cfff $1!

Question No. 3: (10)

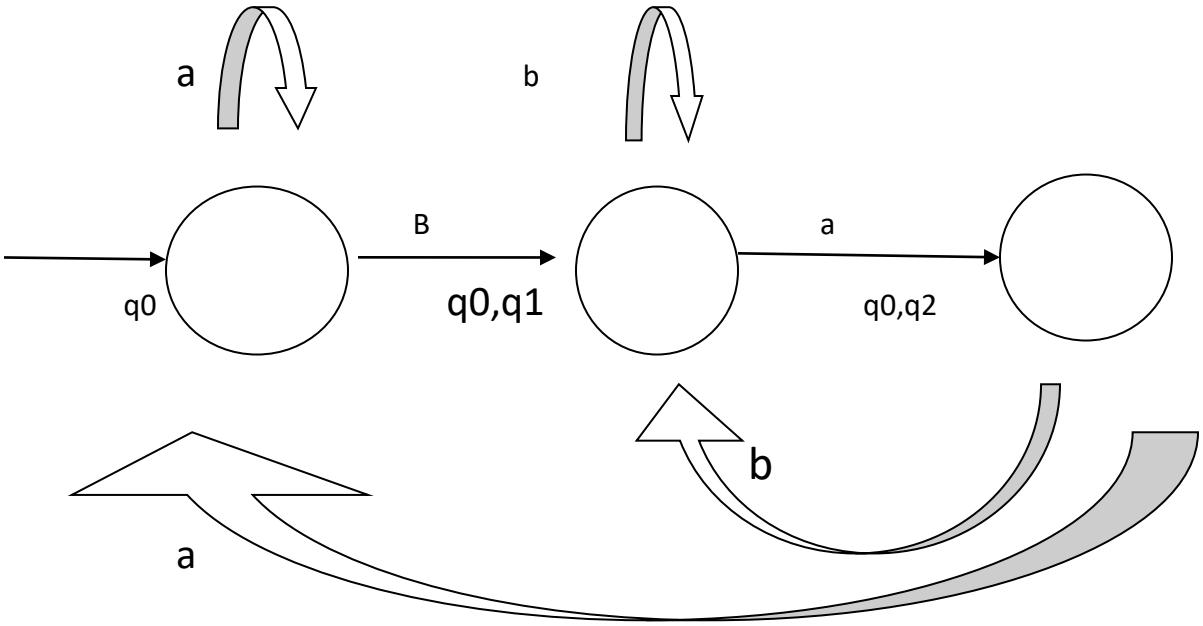
a) Design an NFA over an alphabet ∑= {a, b} such that every string accepted must end with a string --ba. Identify its tuples and also convert it into DFA.

NFA:



E/Q	a	b
Q0	Q0	Q0,Q 1
Q1	Q2	null
Q2	null	null

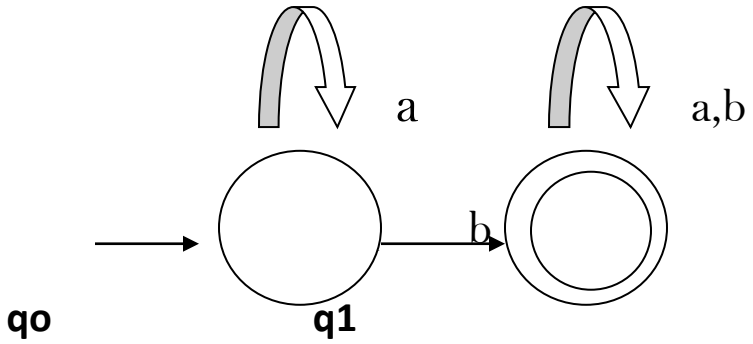
DFA:



E/Q	a	b
q0	q0	{ q0,q1}
{ q0,q1}	{ q0,q2}	{ q0,q1}
{ q0,q2 }	q0	{ q0,q1}

b) Design an NFA for the regular expression : $a^* b(a+b)^*$

ANSWER:



QUESTION NO 4:

a) Explain Part of Speech Tagging (POS) and explain POS tag ambiguity with two examples.

ANSWER:

- **POS Tagging:**

  It is a process of converting a sentence to forms – list of words, list of tuples (where each tuple is having a form *(word, tag)*). The tag in case of is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

  Default tagging: is a basic step for the part-of-speech tagging. It is performed using the

  Default Tagger class. The Default Tagger class takes 'tag' as a single argument. **NN** is the tag for a singular noun. Default Tagger is most useful when it gets to work with most common part-of-speech tag. That's why a noun tag is recommended.

- **POS tag ambiguity:**

  Common parts of speech in English are noun, verb, adjective, adverb, etc. The POS tagging problem **is to determine the POS tag for a particular instance of a word**. The main problem with POS tagging is **ambiguity**.

- **For Example**

POS Tag Ambiguity

In English : I $bank_1$ on the $bank_2$ on the river $bank_3$ for my transactions.

$Bank_1$ is verb, the other two banks are noun

b) State difference between open vs. closed classes in POS tagging.

ANSWER:

- **Open class (CONTENT/LEXICAL)**
  - Lexical words deal with content and vocabulary.
  - They have concrete meaning that goes beyond their function in a sentence.
- **Closed class (Grammatical/Function)**
  - Grammatical words deal with the formation of sentences.
  - They have ambiguous meaning and serve to express grammatical relationships with other words within a sentence.

c) Apply Viterbi Algorithm on the below given bigram and lexical probabilities;

Initial Probabilities	
Noun	1\3
Verb	0
Other	1\3

Bigram Probabilities			
	Noun	Verb	Other
Noun	1\4	1\4	0
Verb	1\4	0	1\4
Other	1\3	0	1\3

Lexical Probabilities					
	O1=time	O2=flies	O3=like	O4=an	O5=arrow
Noun	1\5	1\5	0	0	1\5
Verb	1\5	2\5	1\5	0	0
Other	0	0	1\5	2\5	0

ANSWR:

INITIAL PROBIBILITES:

NOUN	**1/3**
VERB	**0**
OTHER	**1/3**

BIGRAM PROBILITIES:

	noun	**verb**	**other**
noun	**1/4**	**1/4**	**0**
verb	**1/4**	**0**	**1/4**
other	**1/3**	**0**	**1/3**

POS Tags:

^N V A N N. ^ N V N A R A.

Hi	P1(h1)	P2(h2)	P3(h3)	P4(h4)	P5(h5)
Noun	1/3x1/5=1/15	1/5x1/4x1/15=1/300	0x1/150=0	0	1/5x1/4x1/16875
Verb	0	2/5x1/4x1/15=1/150	1/5x1/4x1/150=1/3000	0	0
other	0	0	1/5x1/3x1/150=1/2250	1/3x2/5x1/2250=1/16875	0

Question No. 5: (15)

a) Apply Bayesian theorem over the below given string:

^John got many NLP books. ^He found them all very interesting.

Where for lexical probabilities assume John=0.5, got=0.3, many=0.2, NLP=0.1 and books=0.

Ans:

P(T)=PWIT)=tt(ti/ti-1)-P(wi)tiP1(ti/ti-1)=P(Wi/ti)

Le corpus :^ John got many NLP books found all very interesting

POS tagged

^N V N N.^N V N A R A

Recording numbers

	^	N	V	A	R	.
^	0	2	0	0	0	0
N	0	1	2	1	0	1
V	0	1	0	1	0	0
A	0	1	0	0	1	1
R	0	0	0	1	0	0
.	1	0	0	0	0	0

Bigram Probabilties P(N/V)=# (^=N)/#^

	^	N	V	A	R	.
^	0	1	0	0	0	0
N	0	1/5	2/5	1/5	0	1/5
V	0	1/2	0	1/5	0	0
A	0	1/3	0	0	1/3	1/3

R	0	0	0	1	0	0
.	1	0	0	0	0	0

Lexical probability

	John	Got	Many	NLP	books
^	0.5	0.3	0.2	0.1	0
N	0.5	0.3	0.2	-	-
V	0.5	0.3	0.2	-	-
A	0.5	-	-	-	-
R	0.5	-	-	-	-
.					

P(John/^) = P(Wi=John/ti=^)
=# (John,^)/ #^
(m y,^)/#^ = 0.5x2/2 = 0.5

=0.2x2/2 = # (got ,^)/ # ^
=0.2 = 0.3x2/2 =0.3
=(NLP,^)/^

=0.1x2/2 = <John,N)/N

=0.1 =0.5x5/5=0
(book,^)/^
=0x2=0.

b) Find the CFG of the string “abaabaa” using the production rules

S→a, S→aAS, A→bS

ANS:
S⟶ a (Rule:1)
S⟶ aAS (Rule:2)

A → bS (Rule:3)

Compute the string W=”abaabaa”with the left most deivation

S ⇒ aAS (Rule:2)

⇒ abS (Rule:3)

⇒ abaAS (Rule:1)

⇒ abaabSS (Rule:2)

⇒ abaabaS (Rule:3)

⇒ abaabaa (Rule:3)