Ali Haider
14259

Final Term Assignment
Course: Natural Language Processing
Instructor: Mam Aasma Khan
Date: June 22, 2020

Note: Attempt all Questions.

Question No. 1: (5)

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#1:

NLP:

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. Most NLP techniques rely on machine learning to derive meaning from human languages.

Application of NLP:

- Machine translation (MT), process of translating one source language or text into another language, is one of the most important applications of NLP.
- Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.
- Personal assistant applications such as OK Google, Siri, Cortana, and Alexa.
- Email filters are one of the most basic and initial applications of NLP online. It started out with spam filters, uncovering certain words or phrases that signal a spam message. But filtering has upgraded, just like early adaptations of NLP.

Challenges:

- Syntax and Ambiguity

  Example: I saw the man with a telescope

  - Who had a telescope.

- Semantics

  Example: The astronomer loves the star.

  - Star in sky Or celebrity

Question No. 2:

ANSWER#2:

Specify the text strings using the below regular expressions: (5)

Note: *the **Red** word are the answer of the questions.*

a. /a(bc)
   - Given string: ab **abc** ac acb a0b a2b a42c A87d

b. /[abc]
   - Given string: **ab** **abc** **ac** **acb** **a**0**b** **a**2**b** **a**42**c** A87d

c. /abc+
   - Given string: ab **abc** **abcc** b**abc**

d. /abc*
   - Given string: **ab** **abc** **abcc** b**abc**

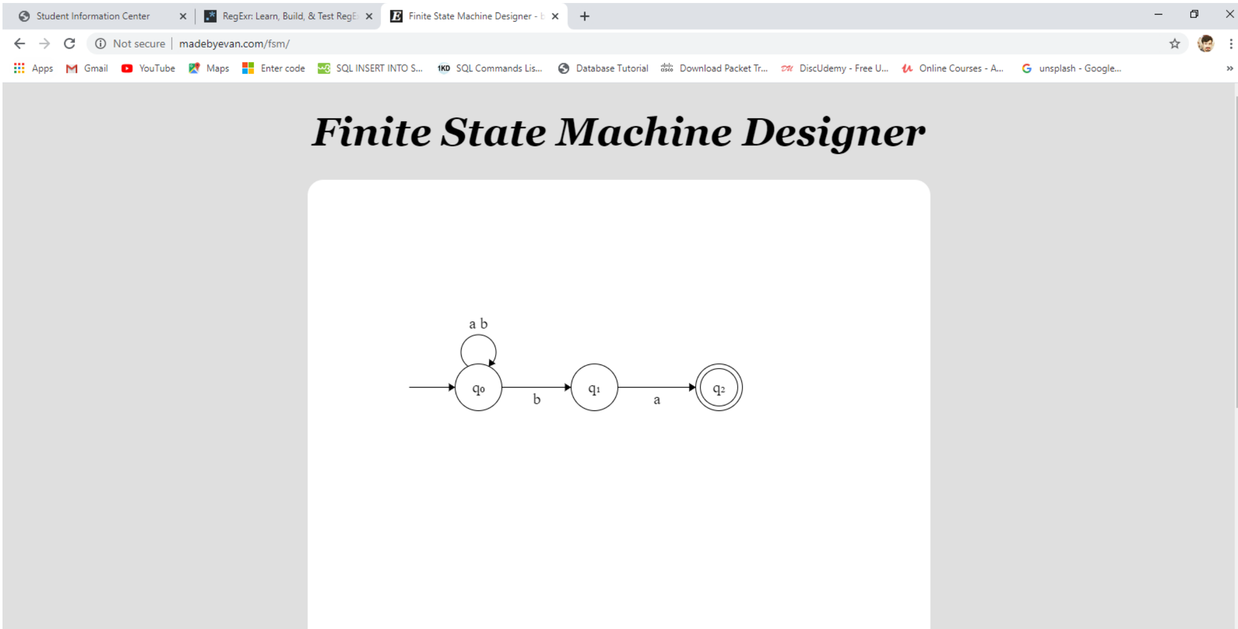
e. /[^a-z A-Z 0-9]
   - Given string: a89 opx cfff **$**1**!**

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. (05)

ANSWER#3a:

NFA:



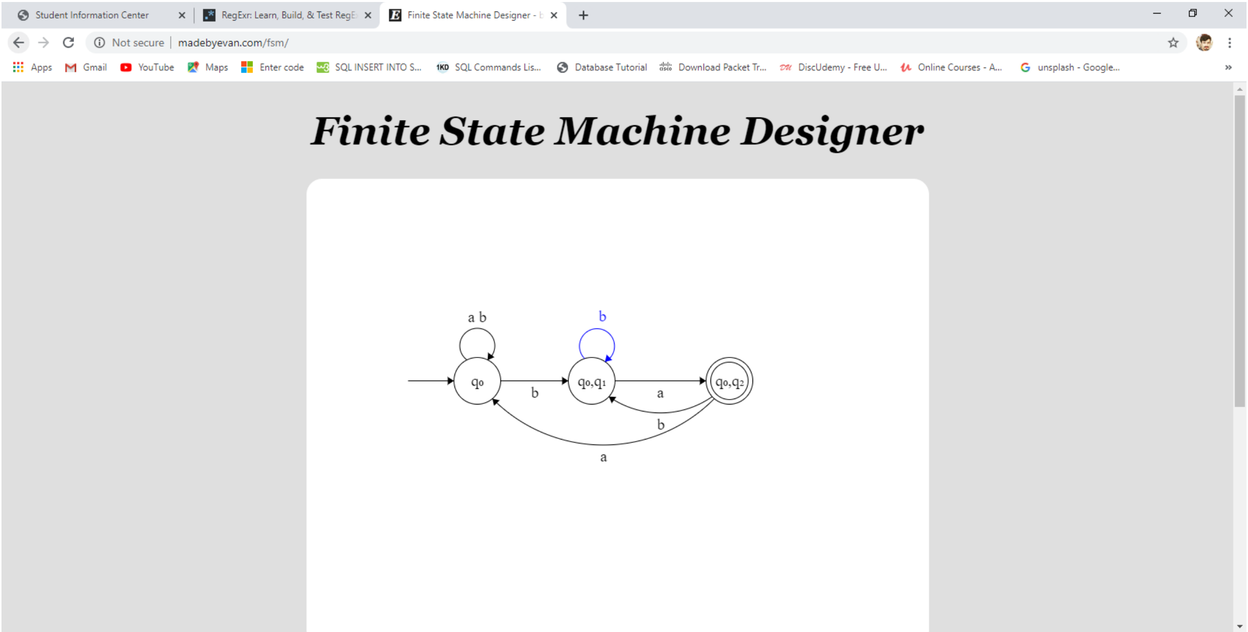
FSA Transition Table:

Q	**a**	**a**
qo	qo	qo,q1
q1	q2	Null
q3	null	Null

DFSA Transition Table:

Q	**a**	**b**
qo	qo	{qo,q1}
{qo,q1}	{qo,q2}	{qo,q1}
{qo,q2}	qo	{qo,q1}

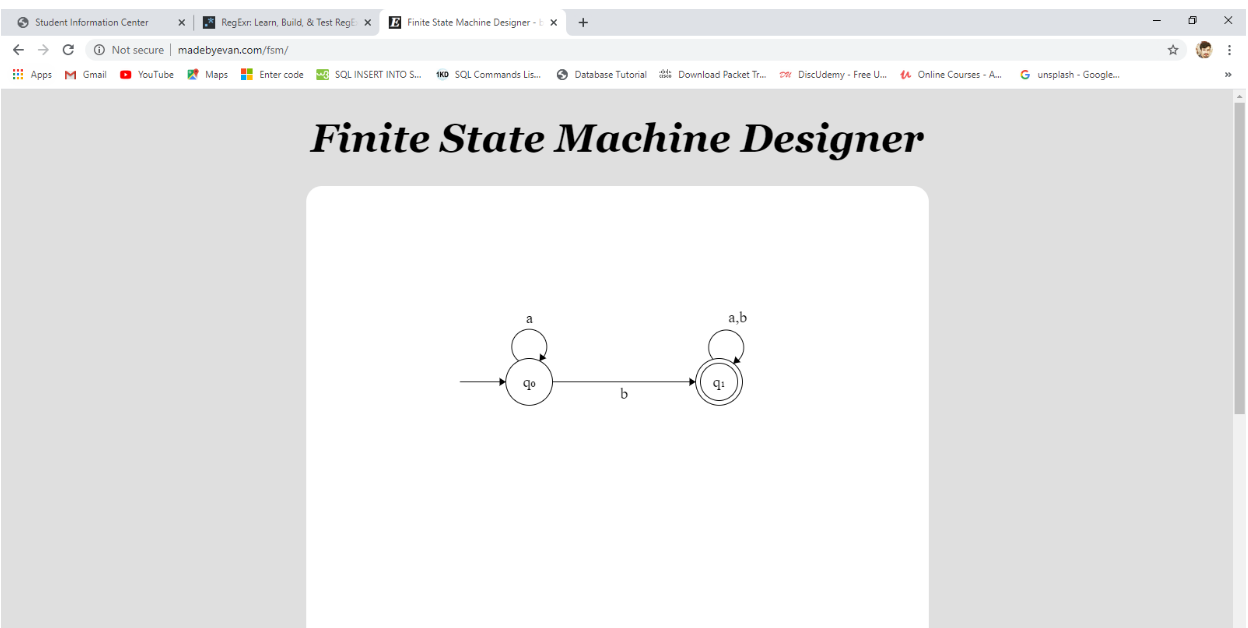
DFA:



b) Design an NFA for the regular expression : a* b(a+b)* (05)

ANSWER#3b:

NFA For The Regular Expression:



Question No. 4: (15)

a) Explain Part of Speech Tagging (POS) and explain POS tag ambiguity with two examples. (03)

ANSWER#4a:

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.

Types of POS Tagger: Most of POS-tagging algorithms fall under

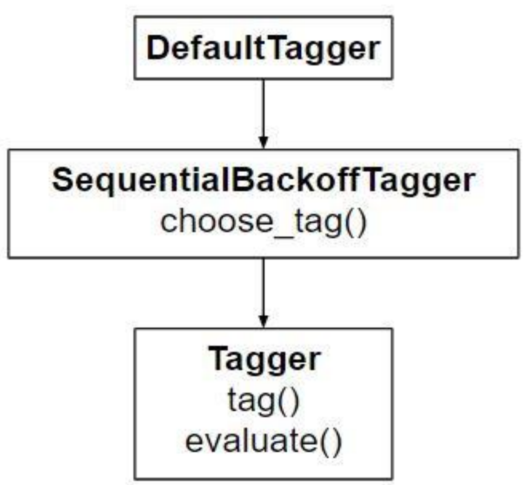
Rule-Based POS Taggers

Stochastic POS Taggers

Transformation based POS Taggers

Hidden Markov Model POS Taggers

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**. In English, many common words have multiple meanings and therefore multiple POS. The job of a POS tagger is to resolve this ambiguity accurately based on the context of use.

Examples:

- 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

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB

b) State difference between open vs. closed classes in POS tagging. (02)

ANSWER#4b:

Open class (CONTENT/LEXICAL)

- Lexical words deal with content and vocabulary.
- They have concrete meaning that goes beyond their function in a sentence.
- These words refer to things, people, actions, descriptions, or other ideas that have more than just a grammatical usage.

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.
- They signal the structural relationships that words have to one another and are the glue that holds sentences together.
- Thus, they serve as important elements to the structure of sentences.

c) Apply Viterbi Algorithm on the below given bigram and lexical probabilities; (10)

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

ANSWER#4C:

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
	time=noun	**flies=verb**	**like=other**	**an=other**	**arrow=noun**

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.

ANSWER#5a:

POS Tags:

^N V A 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 Probability:

Bigram Probability= P(x|y)=P(a.b)/P(a)

	^	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	1/2	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

Where lexical probability is given
i.e. John=0.5, got=0.3, many=0.2, NLP=0.1 and books=0.
Now putting all the values in **Bayes theorem**
i.e. **P(T)=P(W/T)=TTP(Ti-/Ti-1)x P(Wi/Ti)**
so, for **john** 1/5x0.5=0.1 for **got** 0x0.3=0 for **many** 0x0.2=0 for **nlp** 1/5x0.1=0.02 for **book**=0

b) Find the CFG of the string "abaabaa" using the production rules

S→a, S→aAS, A→bS

ANSWER#5b:

Obtain Parse Tree For String:

Test Results for CFG

#	String	Matches	
1	"abaabaa"	Yes	See Derivation

Rule	Application	Result
Start → S	Start	S
S → aAS	S	aAS
A → bS	aAS	abSS
S → a	abSS	abaS
S → aAS	abaS	abaaAS
A → bS	abaaAS	abaabSS
S → a	abaabSS	abaabaS
S → a	abaabaS	abaabaa

Tree For String: