Vivekanand Education Society's Institute of Technology, Chembur, Mumbai,
Department Of Computer Engineering,
Year:2019-20 (Even Sem)
Solution Test No.- 1

Class : D17A/B/C	**Division: A, B and C**
Semester :VIII	**Subject: Natural Language Processing**
Date: 20/02/2020	**Time: 1 hr**

<table>
<tr><td>Q.1</td><td></td><td>(Attempt any five of the following)</td></tr>
<tr><td></td><td>a)</td><td>Explain Lexical and Syntactic ambiguity in NLP with example.
Answer Key :
● Lexical Ambiguity: ambiguity of single word.
Eg: I saw a bat.
saw: past tense 'see'/'saw':instrument(tool)
bat: Cricket bat/ Animal

● Syntactic Ambiguity: The structural ambiguities were syntactic ambiguities.
(old men and women) or ((old men) and women)</td></tr>
<tr><td></td><td>b)</td><td>Explain any two applications of NLP.
1. Language translator
2. Information Retrieval
3. Question Answering
4. Information Extraction
5. Speech Recognition System</td></tr>
<tr><td></td><td>c)</td><td>Differentiate between inflectional and derivational morphemes with an example each
Answer Key :
<table><tr><td>Inflectional Morphology</td><td>Derivational Morphology</td></tr><tr><td>Inflectional morphology is the study of the modification of words to fit into different grammatical contexts</td><td>derivational morphology is the study of the formation of new words that differ either in syntactic category or in meaning from their bases</td></tr><tr><td>Affixes that merely serve as grammatical markers and indicate some grammatical information about a word</td><td>Affixes that are capable of either changing the meaning or the grammatical category of the word</td></tr><tr><td>Create new forms of the same word. Eg : bring, brings, brought, bringing</td><td>Creates new words. Eg. : logic, logical, illogical, logician.</td></tr></table></td></tr>
</table>

	d)	What are the challenges of POS tagging? POS tag the following sentence "The little yellow dog barked at the cat" Answer key: 1. Multiple tags and multiple words 2. Multi-part tags Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N The/[Det] little/[Adj] yellow/[Adj] dog/[N] barked/[V] at/[Prep] the/[Det] cat/[N]
	e)	Define N-gram model and explain with example bigram probabilities **Answer Key:** N-gram model : a contiguous sequence of n tokens from a given piece of text. It assumes each word depends only on the last n-1 words (Markov assumption) $P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$ <s>I am nere </s> <s>who am I </s> <s>I would like to know </s> *Estimating bigrams* P(I|<s>) = 2/3 P(</s>|here) =1 P(would | I) = 1/3 P(here | am) = 1/2 P(know | like) = 0
	f)	What is lemmatization and stemming explain with example **Answer Key :** **Lemmatization:** uses vocabulary and morphological analysis of words, normally aiming to remove inflectional endings.and return base or dictionary form of word, lemma. eg: see or saw, for token saw **Stemming in English** : car, cars, car's, cars' => car
Q.2	a)	Describe various POS tagging Approaches. **Answer Key :** Parts of speech (POS) tagging means assigning grammatical classes i.e. appropriate parts of speech tags to each word in a natural language sentence. Assigning a POS tag to each word of an unannotated text by hand is very time consuming, which results in the existence of various approaches to automate the job. So automated POS tagging is a technique to automate the annotation process of lexical categories. Approaches are : 1. Rule based POS Tagging : a. Start with a dictionary b. Assign all possible tags to words from the dictionary c. Write rules by hand to selectively remove tags d. Leaving the correct tag for each word. **ENGTWOL Rule-Based Tagger**

First Stage: Run words through a morphological analyzer to get all parts of speech.

Example: *Pavlov had shown that salivation ...*

Pavlov	**PAVLOV N NOM SG PROPER**
had	**HAVE V PAST VFIN SVO**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**

Second Stage: Apply constraints.

Constraints used in negative way.

Example: Adverbial "that" rule

Given input: "that"

If

(+1 A/ADV/QUANT)

(+2 SENT-LIM)

(NOT -1 SVOC/A)

Then eliminate non-ADV tags

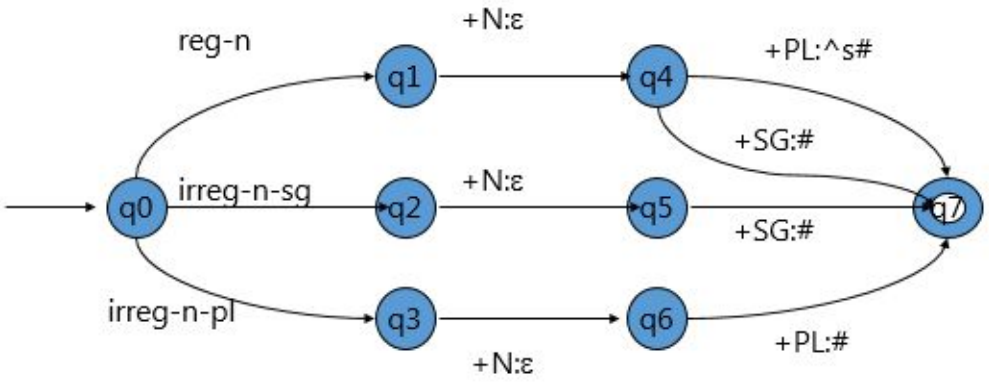
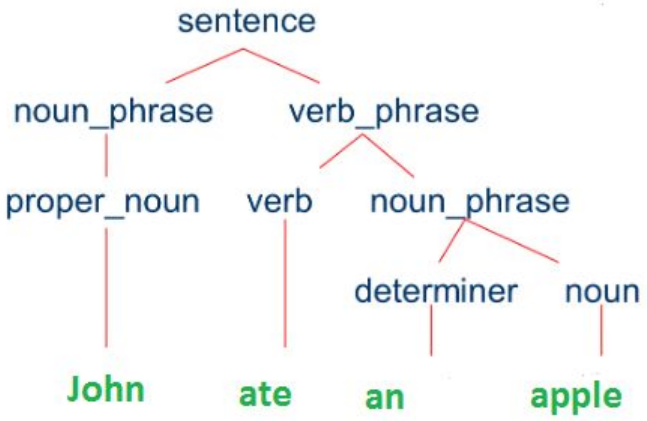
Else eliminate ADV

2. Stochastic POS Tagging
    a. Word Frequency Approach
    b. Tag sequence Probabilities
3. Transformation based tagging.
    a. Combination of **Rule-based and stochastic tagging** .
    b. **Automatic tagging of POS** to the given text
    c. TBL, allows us to have linguistic knowledge in a readable form, **transforms one state to another state by using transformation rules.**
    d. Like **rule-based** : because rules are used to specify tags in a certain environment
    e. Like **stochastic approach** : because machine learning is used—with tagged corpus as input

OR

b) Write short note on Morphological analysis and Role of FST.

Answer Key :

		Morphological Analysis : Individual words are analysed into their component and nonword tokens. Punctuation are separated from word. eg : Carried = carry + ed. **Two Options :** **1. Full-form lexicon** **2. Root-form lexicon and Unknown words.** •FSTs map between one set of symbols and another using an FSA whose alphabet S is composed of pairs of symbols from input and output alphabets • In general, FSTs can be used for : • Translator (Hello : Ciao) • Parser/generator (Hello : How may I help you?) • To map between the lexical and surface levels of Kimmo's 2-level morphology 
Q.3	a)	Consider following grammar 1. S→ NP VP 2. VP→V NP 3.NP→NAME 4. NP→ART N 5. NAME→John 6. V→ate 7. ART → an|the 8. N→ apple Derive the sentence "John ate an apple" using top down **or** bottom up parsing. Compare Top down parsing with Bottom up parsing. **Answer Key :** 

The bottom up Shift reduce Parsing table is as shown below:

Stack	Input remaining	Action
()	John ate an apple	shift
(John)	ate an apple	reduce, Name → John
(Name)	ate an apple	Shift
(Name ate)	an apple	reduce, V->ate
(Name V)	an apple	shift
(Name V an)	apple	reduce, ART → an
(Name V ART)	apple	Shift
(Name V ART apple)		reduce, N →apple
(Name V ART N)		reduce, VP → Verb NP
(Name V NP)		reduce, NP→ Name
(NP VP)		Reduce, S→NP VP
(S)		SUCCESS!

Top-down Parsing

- Only searches for trees that can be answers
- But suggests trees that are not consistent with the words
- Guarantees that tree starts with S as root
- Does not guarantee that tree will match input words

Bottom-up Parsing

- Only forms trees consistent with the words
- Suggest trees that make no sense globally
- Guarantees that tree matches input words
- Does not guarantee that parse tree will lead to S as a root.

OR

b) .Explain CFG with suitable example. Describe following potential problem in CFG such as : 1)Agreement 2)sub Categorization 3)Movement

Answer Key :

1) **Agreement:**verbs agree in number with their subjects:
What flights leave in the morning?
*What flight leave in the morning?

This dog * This dogs
Those dogs *those dog
This dog eats *This dog eat
Those dogs eat *Those dogs eat

2) **Subcategorization:**

More examples:

- *find* is subcategorized for an NP (can take an NP complement)
- *want* is subcategorized for an NP or an infinitival VP
- *bet* is subcategorized for NP NP S

A listing of the possible sequences of complements is called the subcategorization frame for the verb.
As with agreement, the obvious CFG solution yields rule explosion:
$VP \rightarrow V_{intr}$
$VP \rightarrow V_{tr}$ NP
$VP \rightarrow V_{ditr}$ NP NP

Movement :

•Core example

–$[[\text{My travel agent}]_{NP}$ [booked [the flight]$_{NP}]_{VP}]_{S}$

•I.e. "book" is a straightforward transitive verb. It expects a single NP arg within the VP as one of its arguments, and a single NP arg as the subject.

What about?

–Which flight do you want me to have the travel agent book_?

– The direct object argument to "book" isn't appearing in the right place. It is in fact a long way from where its supposed to appear.

And note that its separated from its verb by 2 other verbs.