LP1

Artificial Intelligence & Robotics.

Assignment B 3

Date of Complution:- 16.10.2020

Title:- Syntax Analysis

Problem Statement :- Implement syntax analysis for the assertive English statements.
The stages to be executed are,
Sentence segmentation
Word tokenization
Part of speech/morpho syntactic tagging
Syntactic parsing (use any of the parser like Stanford).

Objectives :- To understand sentence segmentation, tokenization concepts.
To implement syntax analysis on the text.

Outcome :- Students will be able to implement different text analysis and parser techniques.

Theory:-
Sentence segmentation:-
1. Breaking ~~Reference~~ text apart into separate sentences.

Word tokenization

1. Breaking the sentences into ~~work~~ words ie tokens is tokenization.
2. These words are then captured for further analysis.
3. Sentences are separated on space.
4. Punctuation marks are treated as separate tokens since punctuations also have meaning.

Identifying Stop Words.

1. Many words in english like a, the, it are filler word and have no particular particular meanings or add no semantic to sentences.
2. These words are called stop words.
3. Identifying and removing them is important for optimizing speed and quality of model.
4. They are filtered out by comparing with list of words which can vary from application to application.

Text Part of speech tagging.

In corpus linguistics, a part of speech tagging (POS, PoS tagging or POST), also called grammatical tagging or word category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech based on both its definition and its context. ie its relation with adjacent and related words in a phrase, sentence or paragraph.

Syntactic Parsering:-

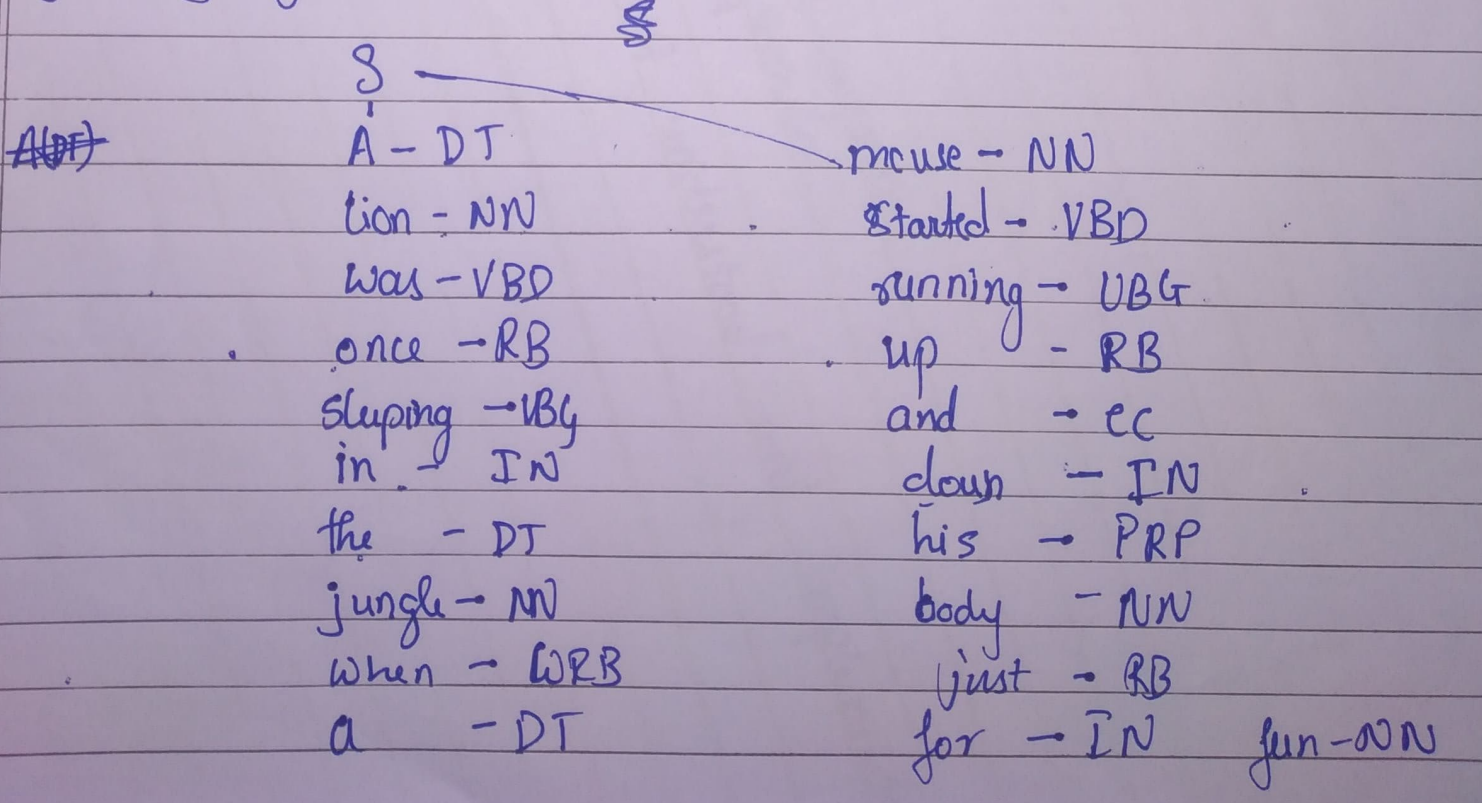
Parsing, syntax analysis is the process of analysis a string of symbols either in natural language, computer language or data structures, conforming rules of a formal grammer.

Standard parser:-
A statistical parser.
A natural language parser is a program the works out the grammatical structure of sentences, for instance which group of words go together (as "phrases") and which words are the subject or object of a verb.
Probabilistic parsers use knowledge of langugs.

~~Algorithm~~

Test Case:-

"A lion was once sleeping in the jungle when a mouse started running up and down his body just for fun".



NN - Noun Singular
DT - determiner
VBD
RB - adverb
VBG -
IN - preposition
WRB -
CC - coordinating conjunction
PRN - Personal pronoun.

Conclusion :- Thus I understood and implemented text analyzer.

TEXT ANALYSIS

CODE:

```
import nltk
nltk.download('state_union')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
from nltk.corpus import state_union
from nltk.tokenize import PunktSentenceTokenizer,sent_tokenize,word_tokenize

train_text = state_union.raw("2005-GWBush.txt")
sample_text = "A lion was once sleeping in the jungle when a mouse started running up and down his body just for fun."

custom_sent_tokenizer = PunktSentenceTokenizer(train_text)

tokenized = custom_sent_tokenizer.tokenize(sample_text)

def process_content():
    try:
        for i in tokenized:
            words = nltk.word_tokenize(i)
            tagged = nltk.pos_tag(words)
            chunkGram = r"Chunk: {<RB.?><VB.?><NNP>+<NN>?}"
            chunkParser = nltk.RegexpParser(chunkGram)
            chunked = chunkParser.parse(tagged)
            namedEnt = nltk.ne_chunk(tagged, binary=True)
            namedEnt.draw()

            for subtree in chunked.subtrees(filter=lambda t: t.label() == 'Chunk'):
                print(subtree)
    except Exception as e:
        print(str(e))
process_content()
```

OUTPUT: