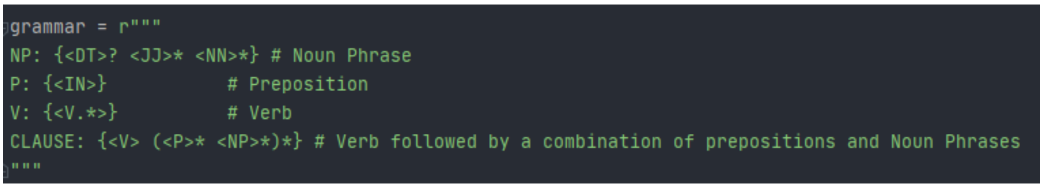
Lab 4 TDT4310

Vebjørn Ohr

March 19 2021

Exercise 1

The grammar used is shown in Figure 1. The RegexpParser of NLTK for used to createh the NP chunker. The Clause of the grammar finds all chunks with a verb followed by a combination of prepositions and nounphrases (NP). The pretagged sents from the Brown corpus was used as input. The resulting tree was iterated through to find all subtrees with the clause. Noun phrases are shortened to the string 'NP'. The resulting chunks can be seen in Figure 2



```
grammar = r"""
NP: {<DT>? <JJ>* <NN>*} # Noun Phrase
P: {<IN>}           # Preposition
V: {<V.*>}          # Verb
CLAUSE: {<V> (<P>* <NP>*)*} # Verb followed by a combination of prepositions and Noun Phrases
"""
```

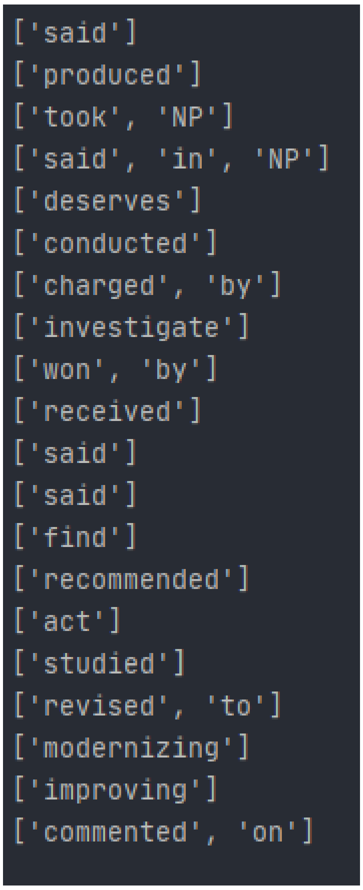
Figure 1: Task 1 grammar

Exercise 2

The "SpaceX.txt" file was loaded and made into a list of sentences, with each sentence being one line in the text file. Escape characters were removed and the sentences tokenized and made lowercase. The NLTK pos_tag function was used to POS tag the sentences. The sentences were then parsed one by one all matching the 'NP' clause from the provided grammar were added to the results. The NP chunk from the first sentence can be seen in Figure 3, showing that one NP chunk was found in the first sentence with the provided grammar ("musk shot").

Exercise 3

The context free grammar (CFG) was made using the grammar shown in in Figure 4. Both of the provided sentences consists of a noun phrase followed by a verb. NP is defined by a determiner (demonstrative) followed by a noun. The nouns and verbs of the sentences are then included in the grammar. The NLTK ChartParser was used to parse the sentences, and the results are shown in 5

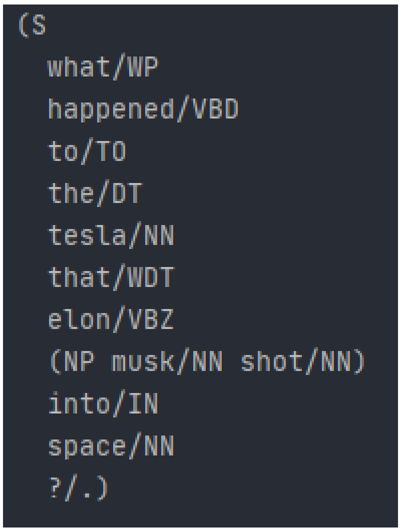


```
['said']
['produced']
['took', 'NP']
['said', 'in', 'NP']
['deserves']
['conducted']
['charged', 'by']
['investigate']
['won', 'by']
['received']
['said']
['said']
['find']
['recommended']
['act']
['studied']
['revised', 'to']
['modernizing']
['improving']
['commented', 'on']
```

Figure 2: Task 1 results

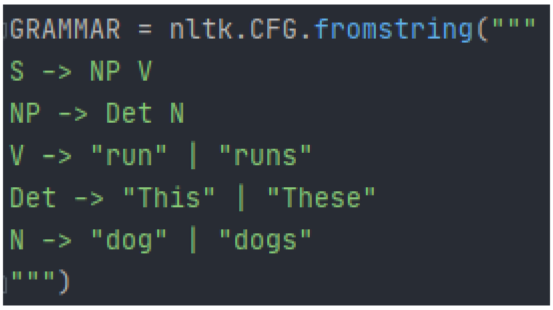
Exercise 4

The tweets were loaded and tokenized as in Lab 3. The NLTK RegexpTokenizer was used to remove punctuation marks, and another regular expression was used to remove simple URLS. This resulted in some empty tweets (links only) which were removed. The NLTK LM module (nltk.lm) was used for preprocessing the tokens and training the language model. The model used was based on maximum likelihood estimation (MLE) with trigrams. The *padded_everygram_pipeline* function was used to pre-process the sentences, again with trigrams, resulting in a training set and a vocabulary. The MLE model was then fitted with these. A simple function was made to take a sentence as input and find the most likely completion of the sentence. This was done by adding one word at a time until the '¡/s¿' tag appeared indicating the end of the sentence. Using the provided example sentence ”make America” resulted in ”make America great again”



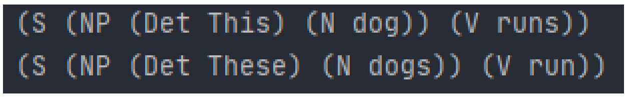
```
(S
  what/WP
  happened/VBD
  to/TO
  the/DT
  tesla/NN
  that/WDT
  elon/VBZ
  (NP musk/NN shot/NN)
  into/IN
  space/NN
  ?/.)
```

Figure 3: Task 2 first sentence results



```
GRAMMAR = nltk.CFG.fromstring("""
S -> NP V
NP -> Det N
V -> "run" | "runs"
Det -> "This" | "These"
N -> "dog" | "dogs"
""")
```

Figure 4: Task 3 Context free grammar



```
(S (NP (Det This) (N dog)) (V runs))
(S (NP (Det These) (N dogs)) (V run))
```

Figure 5: Task 3 results