Section 1: Linguistic Structure, Data and Analysis

Consider the following sentence: "the man with the hat and tie came after the man in the dark suit"

How many types and tokens are there in the sentence?

The number of tokens is the total number of words in the sentence. In this case, the number of tokens is 15.

The number of types is the total number of unique words on the sentence. In this case, the number of types is 12.

- Question 1B: Define a constituency (phrase) grammar and lexicon that analyses the following sentence by using the non-terminal symbols 'S, NP, VP, PP' and the pre-terminal symbols 'Det, Noun, Verb, Prep'

The minister visited the power plant in the south of the country

Draw a constituency (phrase) structure tree for this sentence, using the grammar and lexicon you defined

The lexicon we need to analyse the sentence above is given by the types of the sentence. This is $L = \{$the, minister, visited, power, plant, in, south, of, country$\}$ together with their ~~respective~~ non-terminal symbols:

~~Grammar~~

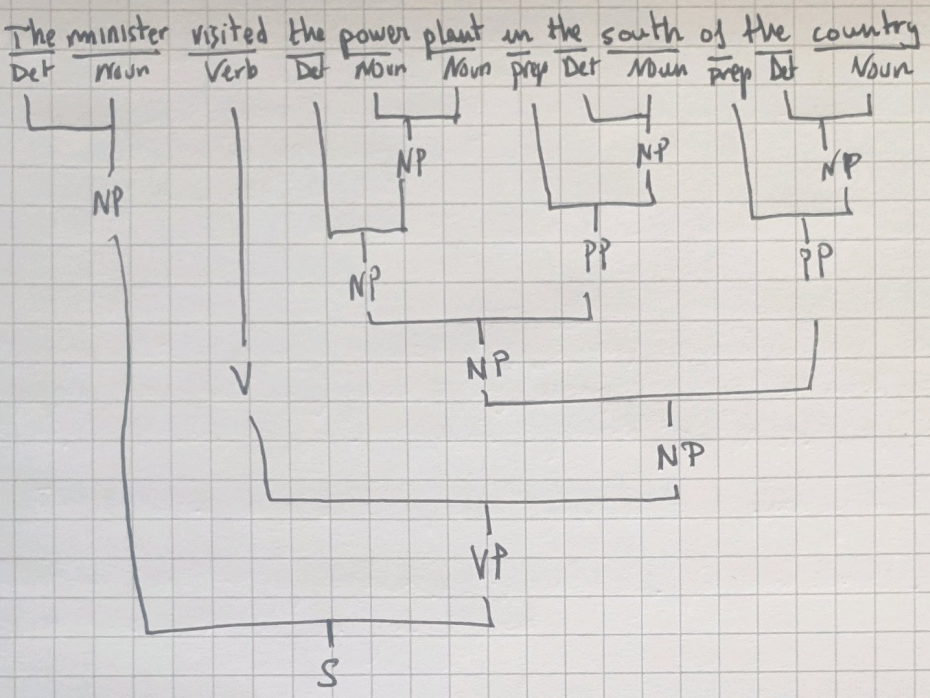
- Det → the
- Noun → minister, power, plant, south, country
- Verb → visited
- Prep → in, of

Now, with this set of terminal, non-terminal symbols and lexicon I define the following constituency grammar

- NP → det Noun | Noun Noun | Det NP | NP PP
- PP → Prep Noun | Prep NP
- VP → Verb | Verb NP | VP PP
- S → NP VP

• Question 1B

With this grammar, the following constituency structure tree can be drawn from the given sentence:



• Question 1C: What is the difference between a parallel corpus and a comparable corpus?

A parallel corpus is a collection of translated documents while a comparable corpus is a collection of documents on the same topic in different language

Section 2: Textual Similarity

s_1 = Tusk swipes at May for better border talk

s_2 = Tusk asks May for better border idea

Question 2A: Calculate the following similarities for s_1 and s_2:

→ Dice similarity using a bag-of-words model

→ Jaccard similarity using a bag-of-words model

→ The length of the longest common subsequence

Jaccard similarity is given by $J_{s_1,s_2} = \frac{|S_1 \cap S_2|}{|S_1 \cap S_2|}$

where $\begin{cases} |S_1 \cap S_2| = |\{Tusk, May, for, better, border\}| = 5 \\ |S_1 \cup S_2| = |\{Tusk, swipes, asks, at, May, for, better, border, talk, idea\}| = 10 \end{cases}$

Therefore $J_{s_1,s_2} = \frac{1}{2} = 0.5$

From here, it is easy to calculate the Dice similarity as $D = \frac{2 \cdot 5}{1+5} = \frac{1}{1.5} = \frac{2}{3} \approx 0.66$

Finally, the length of the longest common subsequence is 4 and is given by the length of "May for better border".

Question 2B: Recall the Damerau-Levenshtein Edit Distance:

$$d(i,j) = \min \begin{cases} d(i-1,j)+1 \\ d(i,j-1)+1 \\ d(i-1,j-1)+1 \\ d(i-1,j-1) \text{ if } s_{1,i} = s_{2,j} \\ d(i-2,j-2)+1 \text{ if } s_{1,i} = s_{2,j-1} \wedge s_{1,i-1} = s_{2,j} \end{cases}$$

What is the Damerau-Levenshtein distance between these sentences? Explain your method

/

Section 3 Language modelling

Consider the following poem by AA Milne as a corpus. Treat each line as a new sentence. Ignoring punctuation, it is 100 words long

The wind on the hill.
No one can tell me nobody knows where the wind comes from where the wind goes
It's flying from somewhere as fast as it can I couldn't keep up with it not if I ran.
But if I stopped holding the string of my kite, it would blow with the wind for a day and a night.
And then when I found it, wherever it blew, I should know that the wind had been going there too
So then I could tell them where the wind goes
But where the wind comes from nobody knows

Question 3A: Calculate the unigram probabilities ignoring case for the words: "been", "had", "the", "wind", "where"

As there is a total of 100 words, given a word w, the unigram probability of w can be calculated by:

$$P(w) = \frac{\#\text{number of occurrences of } w}{100}$$

This is:

$P(\text{"been"}) = 1/100$ $P(\text{"wind"}) = 7/100$

$P(\text{"had"}) = 1/100$ $P(\text{"where"}) = 4/100$

$P(\text{"the"}) = 8/100$

Question 3B: Calculate the bigram probability ignoring case for the combinations that are not provided in the following table

$P(w_2 \mid w_1)$	w_2 = been	w_2 = had	w_2 = the	w_2 = where	w_2 = wind
w_1 = been	0	0	0	0	0
w_1 = had	1	0	0	0	0
w_1 = the	0	0	0	0	7/8
w_1 = where	0	0	1	0	0
w_1 = wind	0	1/7	0	0	0

• Question 3B

The probability of a bigram (w_1, w_2) can be calculated as:

$$P(w_2|w_1) = \frac{count(w_1, w_2)}{count(w_1)}$$

count(had, been) = 1, count(had) = 1 ⇒ P(been | had) = 1
count(the, where) = 0, count(the) = 8 ⇒ P(where | the) = 0
count(the wind) = 7, count(the) = 8 ⇒ P(wind | the) = 7/8
count(where, the) = 4, count(where) = 4 ⇒ P(~~where~~ the | where) = 1
count(wind, had) = 1, count(wind) = 7 ⇒ P(had | wind) = 1/7

• Question 3C:

State the formula for a bigram language model applied to the sentence "the wind had been". Using this bigram language model calculate the probability of the line "the wind had been".

In general, using a bigram language model we can approximate the probability of a sequence of words by:

$$P(w_1 \cdots w_n) \approx p(w_n|w_{n-1}) \cdot p(w_{n-1}|w_{n-2}) \cdot \cdots p(w_2|w_1) p(w_1)$$

In this specific sentence, this is p("the wind had been") = $\underbrace{p(\text{been}|\text{had})}_{1} \underbrace{p(\text{had}|\text{wind})}_{1/7} \underbrace{p(\text{wind}|\text{the})}_{7/8} p(\text{the})$

This is p("the wind had been") = $1/8 \cdot p(\text{"the"})$ where $p(\text{"the"}) = p(\text{"the"}|\text{start}) = 1/7$

Therefore $p(s) = 1/56$

• Question 3D

p("The wind had been there") = 0 given the bigram model. Briefly explain why and suggest a model that produces a non-zero probability for this sentence.

P("The wind had been there") = P("there"|"been") P("been"|"had") P("had"|"wind") P("wind"|"the") P("the"|

As the bigram ("been", "there") does not exist in the corpus this probability will be always zero (P("there"|"been") = 0).

An alternative would be using Add-One Smoothing in which any bigram is counted one more time that the occurrences in the corpus.

In this case

$$P(\text{"there"}|\text{"been"}) = \frac{c(\text{been, there})+1}{c(\text{been}) + \#\text{1-gram}} = \frac{1}{0+99} = \frac{1}{99}$$

- Section 4: Information Extraction

Consider the following sentences:

S_1 = Shares in Smurfit Kappa have risen by over 18pc in early trading on the London ~~Exchan~~ Stock Exchange

S_2 = Adidas shares were up 1.6%, marking the biggest increase among the largest shares in Germany.

S_4 = Shares of Jaypee Infratech climbed over 4 per cent on Tuesday morning.

S_3 = United Technologies shares rise 2% as a "well-known" activist takes a position

S_5 = "Shares of Commonwealth Bank declined 0.92 percent"

- Question 4A: Annotate sentences S_2 and S_4 with Named Entities of type COMPANY, NUMBER, COUNTRY and TIME where appropriate. Use the following annotation format:

[COMPANY Adidas] shares were up [NUMBER 1.6]% ...

[COMPANY Adidas] shares were up [NUMBER 1.6]% making the biggest increase among the largest shares in [COUNTRY Germany]

[Shares of [COMPANY Jaypee Infratech] climbed over [NUMBER 4] per cent on [TIME Tuesday morning].

- Question 4B: Assume we want to extract the following relations from the ~~sent~~ sentences above:

Shares-Up-Percentage(COMPANY, NUMBER)

Shares-Down-Percentage(COMPANY, NUMBER)

For instance: Shares-Up-Percentage(Smurfit Kappa, 18)

Provide patterns that can be used to extract this information for all companies mentioned in the sentences above

Hearst Patterns for Shares-Up-Percentage:

[COM] have risen over [NUMBER], [COM] were up [NUM], [COM] shares rise [NUM], [COM] climbed over [NUM]

for Shares-Down-Percentage: [COM] declined [NUM]

Question 4C What is the core difference between 'open information extraction' (open IE) and 'knowledge base population (KBP)'?

Open IE refers to the extraction of relation tuples, typically binary relations, from plain text (Unsupervised Relation Extraction)

KBP refers to the task of discovering new facts about entities from a large corpus, and augmenting a knowledge base with these facts (Supervised / based on a knowledge graph)