INTRODUCTION TO NATURAL LANGUAGE PROCESSING

SoSe 2025

ASSIGNMENT -6

Submitted by,

Group – 30

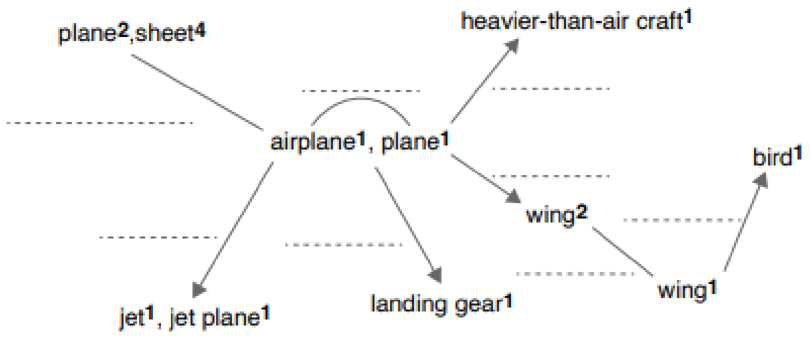
Arundas Mohandas (127115)

Ganga Sunil (127161)

Glen Paul Chirayth Shaju (127125)

Exercise 1 : Lexical Relations & Word Sense Disambiguation (3+2=5 Points)

You are given the following section from the WordNet-graph and four selected glosses.



Synset	Gloss
airplane[1], plane[1]	an aircraft that has a fixed wing and is powered by propellers or jets
	" the flight was delayed due to trouble with the airplane "
plane[2], sheet[4]	(mathematics) an unbounded two-dimensional shape
	" any line joining two points on a plane lies wholly on that plane "
wing[1]	a movable organ for flying (one of a pair)
wing[2]	one of the horizontal airfoils on either side of the fuselage of a plane

(a) Fill in the missing lexical relations (dotted lines) in the WordNet-graph.
Chose the relations from the following:
`polysemy` `synonymy` `antonymy` `homonymy`
`hyponym` `hypernym` `meronym` `holonym`

(b) The lecture introduced LESK – a word sense disambiguation algorithm. For this exercise, we will use NAIVE LESK, a simplified version of the original algorithm. The algorithm computes the Jaccard similarity $J(\{w_1, \ldots, w_n\}, c_j)$ between the context of a word in a sentence $\{w_1, \ldots, w_n\}$ and the gloss c_j without normalizing and preprocessing the text (except for removing punctuation). Consider the following sentence:

`with one wing damaged, the model plane was flying downwards in a spiral`

Disambiguate the words `plane` and `wing` in the given sentence using NAIVE LESK. Which sense is the most likely for each of the two words?

Answer:

1P

(a) The dotted edges in the graph represent missing lexical relations. The correct relations are:

From Synset	To Synset	Lexical Relation
airplane	jet	Hyponym
plane	plane	Homonymy
airplane	wing	Meronym
wing	wing	Homonymy —

the missing

A *hyponym* relation exists from airplane to jet because a jet is a specific kind of airplane. *Homonymy* relation links plane (geometric plane) and plane (airplane) as they are unrelated meanings of the same word. The link between airplane and wing is a *meronym*, since a wing is part of an airplane. Finally, wing and wing are *homonyms*, referring to different meanings (aircraft part vs. biological organ).

6P

(b) Sentence:
"With one wing damaged, the model plane was flying downwards in a spiral."

To disambiguate the words "plane" and "wing", I applied the Naive Lesk algorithm, which selects the most appropriate sense of a word based on the overlap between its dictionary definition (gloss) and the surrounding context.

For ***plane***, there are two possible senses: one refers to an aircraft, and the other to a flat, two-dimensional surface. The context includes words like "model", "flying", "wing", and "spiral", all of which are strongly related to the concept of an aircraft. Therefore, the intended meaning of *plane* in this sentence is the one referring to ***an aircraft.***

For ***wing***, one sense refers to a bird's or animal's wing, and another refers to the structural part of an airplane. Since the sentence describes a "model plane" with a damaged "wing", and also involves flying, it clearly refers to the part of the plane. Thus, the correct sense of *wing* is the one related to ***an aircraft component.***

Jaccard similarity?

Hence,

- *plane* refers to an aircraft.
- *wing* refers to a part of an aircraft.

Exercise 2 : Syntax (1+2+1+1+2+1=8 Points)

Consider the following ambiguous sentence:

`I ate the fish in the freezer`

(a) Provide two unambiguous paraphrases for two alternative meanings for this sentence.

(b) Draw two possible syntax trees for the above sentence, where all the construction rules you use are either *unary* or *binary*. Note which tree corresponds to which of the (disambiguated) meanings you paraphrased in part (a) of this question.

(c) Is the ambiguity in this sentence an example of attachment ambiguity or coordination ambiguity? Briefly explain your answer.

(d) Write a small Context Free Grammar (CFG) that covers the grammar rules you need to build the two syntax trees in part (b) of this exercise.

(e) Now suppose that you turn your CFG into a Probabilistic Context Free Grammar (PCFG): that is, each rule in the grammar is assigned a probability. The probabilities of only a subset of your rules would discriminate between the likelihood of the two parses. Which rules are they? And what equation must the probabilities of these rules satisfy, in order to ensure that the tree that matches the most probable meaning is the more likely parse?

(f) Is the ambiguity in this sentence also reflected in its *dependency structure*? If so, give two alternative *labeled* dependency parses for the sentence and state which corresponds to the likely meaning. If not, explain why not.

Answer:

1R 2a) Two unambiguous phrases are:

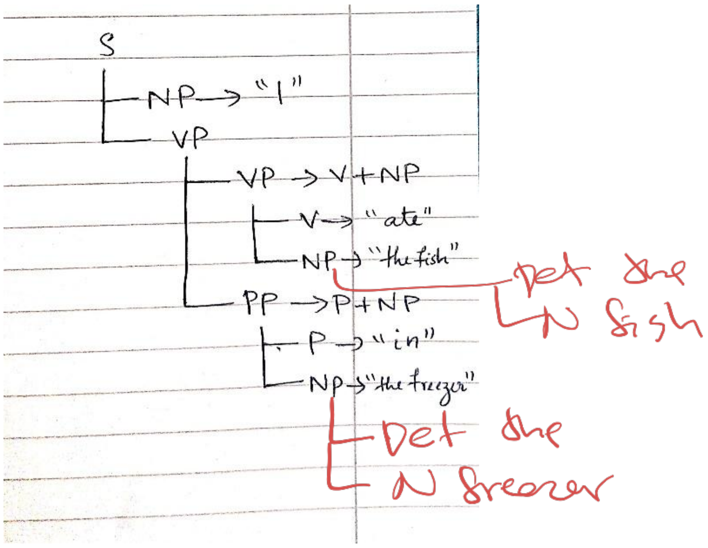
The sentence "I ate the fish in the freezer" has two meanings due to prepositional phrase (PP) attachment ambiguity.

One unambiguous phrase is "I was in the freezer when I ate the fish". Here, the phrase "in the freezer" modifies the verb phrase, implying that the action of eating happened inside the freezer**.**

The second unambiguous phrase is "I ate the fish that was in the freezer." Here the phrase "in the freezer" modifies the noun "fish," meaning the fish was located in the freezer, not the person. The ambiguity comes from whether the prepositional phrase "in the freezer" is describing where the action took place or which fish was eaten.

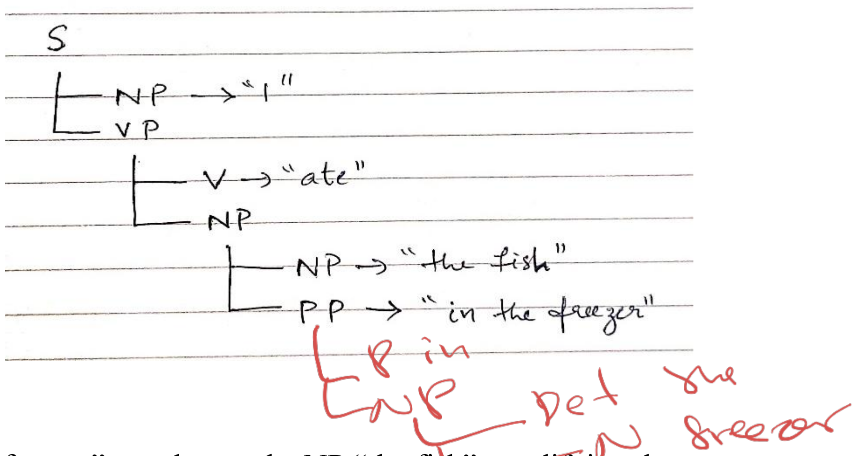
2b)

- For the sentence “I was in the freezer when I ate the fish” the syntax tree is defined below



Here the PP (“in the freezer”) attaches to the VP (“ate the fish”), modifying the action of eating.

- For the sentence “I ate the fish that was in the freezer” the syntax tree is defined below



Here the PP “in the freezer” attaches to the NP “the fish”, modifying the noun.

The abbreviations for the syntax structure are :

S – sentence

NP – Noun phrase

VP – Verb phrase

V – Verb

PP – Prepositional phrase

P – Preposition

2c) This sentence illustrates a case of attachment ambiguity, which occurs when it's unclear where a particular phrase connects within the sentence structure. Specifically, the ambiguity lies in whether the prepositional phrase (PP) “in the freezer” is linked to the verb phrase (VP) or the noun phrase (NP). If it attaches to the verb phrase, it implies that the action eating took place inside the freezer. On the other hand, if it attaches to the noun phrase, it suggests that the fish being eaten was located in the freezer. This is distinct from coordination ambiguity, as there is no compound or coordinated elements involved in the sentence.

2d) S → NP VP

VP → V NP

VP → VP PP

NP → Det N

NP → NP PP

PP → P NP

Det → "the"

N → "fish" | "freezer"

V → "ate"

P → "in"

NP → "I"

These grammar rules are designed to permit both interpretations of the sentence. The rule VP → VP PP allows a prepositional phrase (PP) to attach to a verb phrase (VP), meaning the prepositional phrase modifies the action. On the other hand, the rule NP → NP PP permits the prepositional phrase to attach to a noun phrase (NP), so it modifies the noun instead. This flexibility in attachment is what leads to the ambiguity in interpretation.

ok 2e) To convert the grammar into a Probabilistic Context-Free Grammar (PCFG), we assign probabilities to each rule. However, only a few rules influence the ambiguity in the sentence. Specifically, the rules VP → VP PP and NP → NP PP control whether the prepositional phrase “in the freezer” attaches to the verb or the noun.

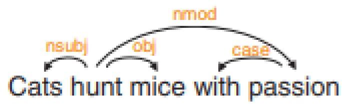
Let’s denote the probability of VP → VP PP as P1 and NP → NP PP as P2. If we want to favour the reading where the PP modifies the noun (i.e., the fish was in the freezer), then the overall probability of that parse must be higher than the alternative. This means the combined probability of the rules leading to the noun-attached structure must outweigh those leading to the verb-attached structure. Assuming other rule probabilities are the same in both cases, this results in the condition is P2 > P1.

ok 2f) Yes, this ambiguity is also present in the dependency structure, as the head of the prepositional phrase "in" can attach to different parts of the sentence. In Dependency Parse 1, where the prepositional phrase modifies the verb (VP attachment), the preposition "in" depends on the verb "ate", indicating the location where the eating occurred while in Dependency Parse 2, where the prepositional phrase modifies the noun (NP attachment), the preposition "in" depends on the noun "fish", directly modifying it and specifying that the fish was located in the freezer. Typically, the second interpretation, where the fish is in the freezer, is more probable in everyday situations, making the NP attachment (Tree 2) the more likely reading.

where are the parses?

Exercise 3 : Dependency Parsing: Arc Standard (4+1=5 Points)

You are given the following sentence and its labeled dependency parse tree:



(a) The lecture introduced the Arc-Standard algorithm for dependency parsing. Parse the following sentence using Arc Standard. At each step of the algorithm, add the content of the stack and the buffer in the corresponding columns in the table below. Chose the next transition (shift, left, right) based on the given reference parse in the same way as when constructing training data for an oracle.

	Transition	Stack	Buffer
0	init →	`root`	`Cats hunt mice with passion`
1	shift →	`root Cats`	`hunt mice with passion`
2	left →		
3			
4			
5			
6			
7			
8			
9			

(b) Arc Standard can only parse projective dependency structures. Inform yourself about and briefly explain the difference between projective and non-projective dependency structures.

Answer: + not algorithm from lecture (−1P)

2.5P

(a) We apply the Arc-Standard algorithm to parse the sentence “Cats hunt mice with passion.” This algorithm uses a stack (σ), a buffer (β), and a set of dependency relations (A). At each step, we perform one of the transitions: SHIFT, LEFT-ARC, or RIGHT-ARC, following the structure of the given dependency tree.

Step	Transition	Stack (σ)	Buffer (β)	Relations (A)
0	init →	[root]	[Cats, hunt, mice, with, passion]	–
1	SHIFT →	[root, Cats]	[hunt, mice, with, passion]	–
2	SHIFT →	[root, Cats, hunt]	[mice, with, passion]	–
3	LEFT-ARC (nsubj) →	[root, hunt]	[mice, with, passion]	A = {(hunt → Cats)}
4	SHIFT →	[root, hunt, mice]	[with, passion]	A = previous
5	~~SHIFT~~ → R	[root, hunt, ~~mice, with~~]	with [passion]	A = previous

6	SHIFT →	[root, hunt, ~~mice~~, with, ~~passion~~]	[] passion	A = previous
7	~~LEFT-ARC (case)~~ → S	[root, hunt, ~~mice~~ with, passion]	[]	A ∪ {(passion → with)}
8	LEFT-ARC (nmod) →	[root, hunt, passion]	[]	A ∪ {(mice → passion)}
9	RIGHT-ARC (obj) →	[root, hunt]	[]	A ∪ {(hunt → mice)}
10	RIGHT-ARC (root) →	[root]	[]	A ∪ {(root → hunt)}

(b) The Arc-Standard algorithm only supports projective dependency trees. A projective structure is one where all dependency arcs can be drawn above the sentence without crossing. This means that each dependent is directly reachable from its head through a continuous span of the sentence. In contrast, non-projective structures contain crossing arcs, which are common in languages with flexible word order. Arc-Standard cannot parse these because it builds trees in a strictly nested, non-crossing manner.