Lecture 1: Introduction to NLP

Natural Language Processing or NLP is a ~~ma~~ multidisciplinary field combining math linguistics and computer science, the goal is to get computer to do useful things with natural language data.

Common NLP tasks and applications:

→ Translation

→ Summarization

→ Question answering

→ speech recognition

→ classification

→ assisted writing

some reasons why NLP is hard:

1. Content
2. Inflections
3. Content
4. misspelled or misused words.
5. physical gesture.

Fundamentals of NLP

→ Preprocessing:

→ tokenization
→ simplification technique
→ tagging and
→ simple rules-based approaches.

→ Besic vectorization:

- → turning text to numbers.
- → measuring similarity between docs.

→ Modelling overview:

- → types of machine learning
- → algorithms
- → vs. models
- → evaluation

→ first steps into classification:

- → classifying text using Navie Bayes.
- → evaluation with precision and recall

→ Topic modeling:

- → Automatically finding topics in docs using latent dirichlet allocation.

Deep learning for NLP:

→ word vectors: capturing word meaning the Concept of embedding.

→ Recurrent Nural Networks:
Capturing sequence information and generditing ~~learnin~~ language.

→ Neural Networks:
what ~~a~~ they are, how they work and details around training

→ ~~Sequence2 sequence~~
seq2seq and attention: training a neural neawork to transform one sequenl to another.

→ Transformation: The dominant mainstream architecture today. Pretraining and transfer learning

Processing:

1. tokenization: The process of segmenting our documents into tokens.

~~Hea~~ Code:

```
// insurting ~~librad~~ libraries

()!Pip install -U spacy == 3.*

() !Python -m spacy info

() import spacy

()! python -m spacy download en_cone_web-sm

() nlp = spacy.load('en-cone-web-sm')

() type (nlp)

//sample sentense testing

 s = "I eat rice"
doc = nlp(s)
Print ([t.text for t in doc]
```

```
//output:
["I", "eat", "rice"
```

Basic programming

case folding, stop word removal, stemming, lemmatization.

case folding

sentense: "Mr. Cook went into the kitchen to cook dinner"

without cf:

{Cook, dinner, into, kitchen, mr, the to, went, cook}

with cf

{Cook, dinner, into, kitchen, mr, the, to, went}

Code:

```
Print([t.lower_ for t in doc])

//to skip for first word.
Print([t.lower_ if not t.is_sent_start else
        t for t in doc])
```

Stop word removal:

stop words → {the, a, of, an, this, that}

Code:

```
Print (nlp.Defaults.stop_words)
Print (len(nlp.Defaults.stop_words))

print([t for t in doc if not t.is_stop])
```

Stemming

removing word suffixes or prefixes.

Banking }
Banks } Bank

Lematization

Reduce a word down to its lemma or dictionary form.

Did
Done
Doing

Do

Code: `[(t.text, t.lemma_) for t in doc]`

week-2 // Advance processing

Part of speech (POS) Tagging:

{noun, verb, adjective, ...}

Example:

"John watched an old movie at the cinema."

Prop.N Verb Det Noun ADP DET ADJ ~~pun~~ Noun Punc

// Code

```
→ [(t.text, t.pose_) for t in doc]

→ [(t.text, t.tag_) for t in doc]
```

Named Identity Recognition (NER)

{a person, a location, an organization .....}

Named entity: anything that can be reffend by a proper ~~noun~~ name. They often have a proper Noun (PROPN) pos tag.

Example:

Person → PER

Location → LOC

Geopolitical Entity & → GPE

Organization → ORG

Usefulness:

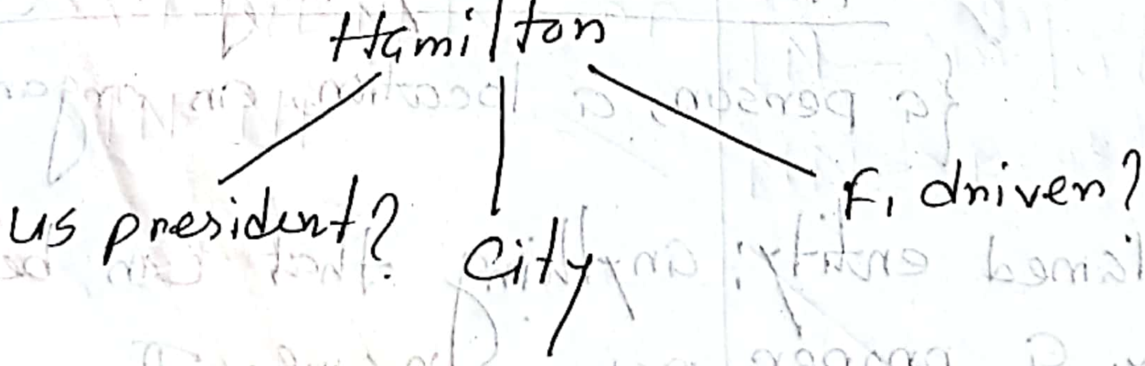
→ Organizing / categorizing corpus

→ Question answering

→ Critical in information extraction.

Challange:

An entity can speak multiple tokens.



Code:

```
doc = nlp(s)    //s is a sentense or string
[(t.tent, t.ent_type_) for t in doc)

// entity non zero check
[(t.tent, t.ent_type_) for t in doc if
                    t.ent_type != 0]

//iterating through ents.
[(ent.tent, ent.label) for ent in doc.ents)
```

Parsing

Determining the syntactic structure of a sentence.

type:

→ Constituency parcing

→ Dependency parcing

Constituency parsing using CFG:

NP – Noun phrase
VP → verb phrase
PP → Prepositional phrase
NN → Noun
PPP → personal pronoun
NNP → proper Noun
VB → verb (Base form
DT → Determiner
IN → Preposition

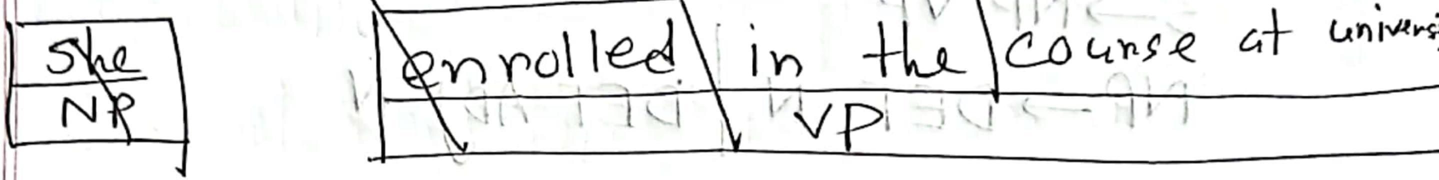
Production Rules

S → NP VP
NP → PRP | NNP | DT NN
VP → VB | VB NP | VP PP
PP → IN NP

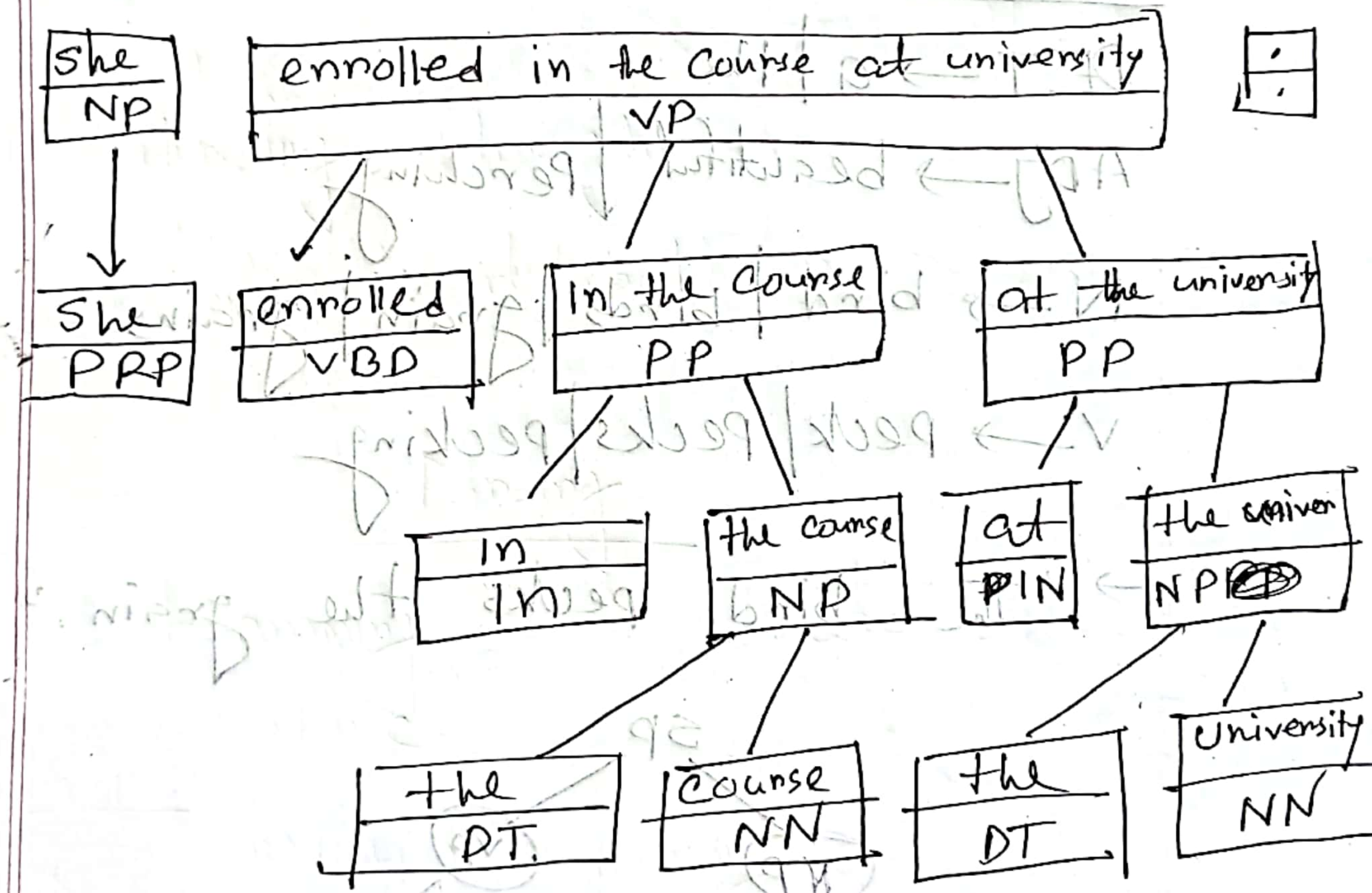
Lenicon

DT → the | a | this | that
PRP → I | she | he
IN → in | at
NN → book | hotel | room.

" She enrolled in th course at university",



"She enrolled in the Course at university."



Rule

S→NP VP

NP→DET N | DET ADJ N

VP → V NP

Lenicon

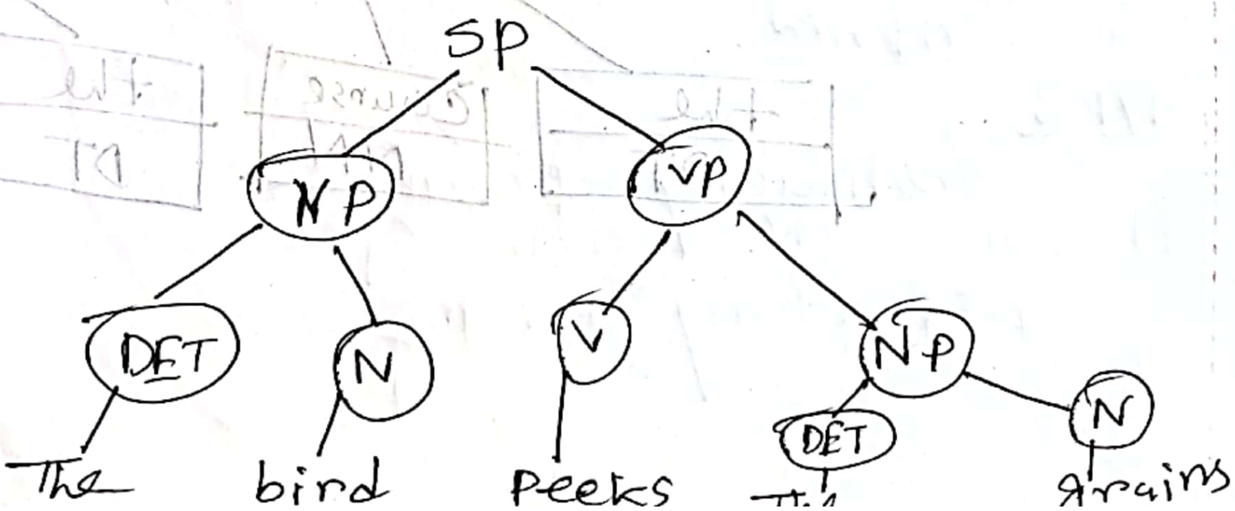
DET → a | the

ADJ → beautiful | Perching

N → bird | birds | grain | grains

V → peck | pecks | pecking

→ “The bird pecks the grain.”

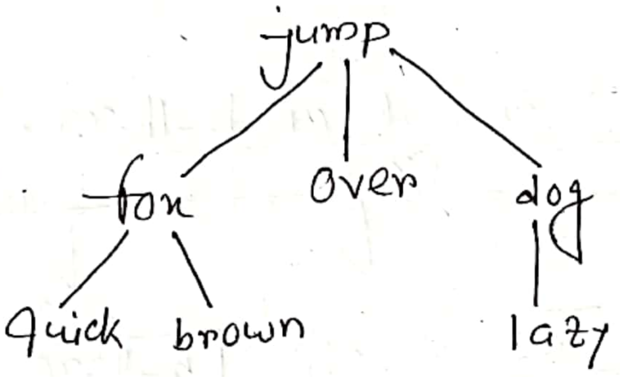


Parsing

→ Determine the cymetric structure of sentense.

“She enrolled

“The quick brown fox jump over the lazy dog”


jump
fox
over
dog
quick
brown
lazy