TEXAS
The University of Texas at Austin

I320D – Topics in Human Centered Data Science

Text Mining and NLP Essentials

Week 6: Syntax Analysis – II, Parsing, Information Extraction

Dr. Abhijit Mishra

Before we start …

- Assignment 3: Code speed up / memory efficiency issues
- **Suggestion 1:**
  - Use *max_features* in CountVectorizer to cap the number of vocabulary items.
  - Start with a small number (say *max_features = 100).* Build an inaccurate but working solution. Then increase the *max_features* value
- **Suggestion 2:**
  - **Optimize preprocesing.** E.g., lemmatize only once
  - Retain a cache of already lemmatized words. Look up. If not found, lemmatize, else reuse
- **Suggestion 3:**
  - Distance computation: use a native implementation using “for loop”. Provided in the latest announcement
- **Suggestion 4:**
  - Delete large variables after use (unless you need them again)

```
distance = compute_euclidean (X, Y)
sorted_outcome = sort_and_rank( processed_tweets, distance )
del distance
```

Week 5 : Recap

- **Lecture:**
  - Representing Syntax
  - Part of Speech Tagging
  - Chunking
- **Lab:**
  - Leveraging Syntax Analysis + RegEx for pattern extraction

Week 5: Recap

- **Syntax Analysis**
  - "Syntax analysis, also known as ***parsing***, is a crucial step in NLP that involves the "analysis of the grammatical structure of a sentence or text in order to understand its syntactic relationships and hierarchies."
- **Two Kinds:**
  - Shallow parsing
  - Deep parsing

Shallow Parsing Tasks

- Part of Speech Tagging
- Noun Phrases / Verb Phrases Chunking
- Named Entity Identification
- Multiword Detection

Part of Speech Tagging

- A kind of shallow parsing is that involves assigning a specific part of speech to each word in a sentence or text.
- **Objective:**
  - **Input:** A sequence of tokens $w_1, w_2, \ldots, w_N$ constituting a sentence S
  - **Output:** For each word w_i , assign a part-of-speech tag t_i from a predefined set of tags (e.g., noun, verb, adjective, adverb, etc.)**, also known as Tagset.**

PoS Examples

```
# Example 1
sentence_1 = ["<START>", "The", "cat", "is", "sleeping.", "<END>"]
tags_1 = ["<START>", "DT", "NN", "VBZ", "VBG", "<END>"]
# Example 2
sentence_2 = ["<START>", "She", "sells", "seashells", "by", "the", "seashore.", "<END>"]
tags_2 = ["<START>", "PRP", "VBZ", "NNS", "IN", "DT", "NN", "<END>"]
# Example 3
sentence_3 = ["<START>", "The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog.", "<END>"]
tags_3 = ["<START>", "DT", "JJ", "JJ", "NN", "VBZ", "IN", "DT", "JJ", "NN", "<END>"]
```

Why PoS Tagging?

- A crucial component in higher layers of NLP processing
  - Constituency and Dependency Parsing
  - Named Entity Identification
- Useful in pattern extraction
  - **Example:** Healthcare
    - Identifying medical conditions mentioned in patient records, such as "*diabetes mellitus type 2” (POS pattern: NN NN NN CD)*
  - **Example:** Finance
    - Recognizing financial terms in news articles, such as "stock market", "interest rate", or "bond yield", by identifying noun phrases (NP) with specific patterns like ”NN + NN" or ”JJ + noun".

Tagset

- Predefined set of tags or labels used to annotate words in a corpus or dataset with their corresponding parts of speech or grammatical categories
- Tags are often abbreviated in 2-3 capital letters
- Example :
  - Penn tagset for English
  - https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

PoS Tagging : Machine Learning Objective

$$\mathbf{T}^* = \mathbf{argmax_T}\ \boldsymbol{p}(\mathbf{T}|\mathbf{W})$$
$$= \mathbf{argmax_T}\ p(\mathrm{t}_1, \mathrm{t}_2, \ldots \mathrm{t}_\mathrm{N}|\mathrm{w}_1, \mathrm{w}_2, \ldots \mathrm{w}_\mathrm{N})$$

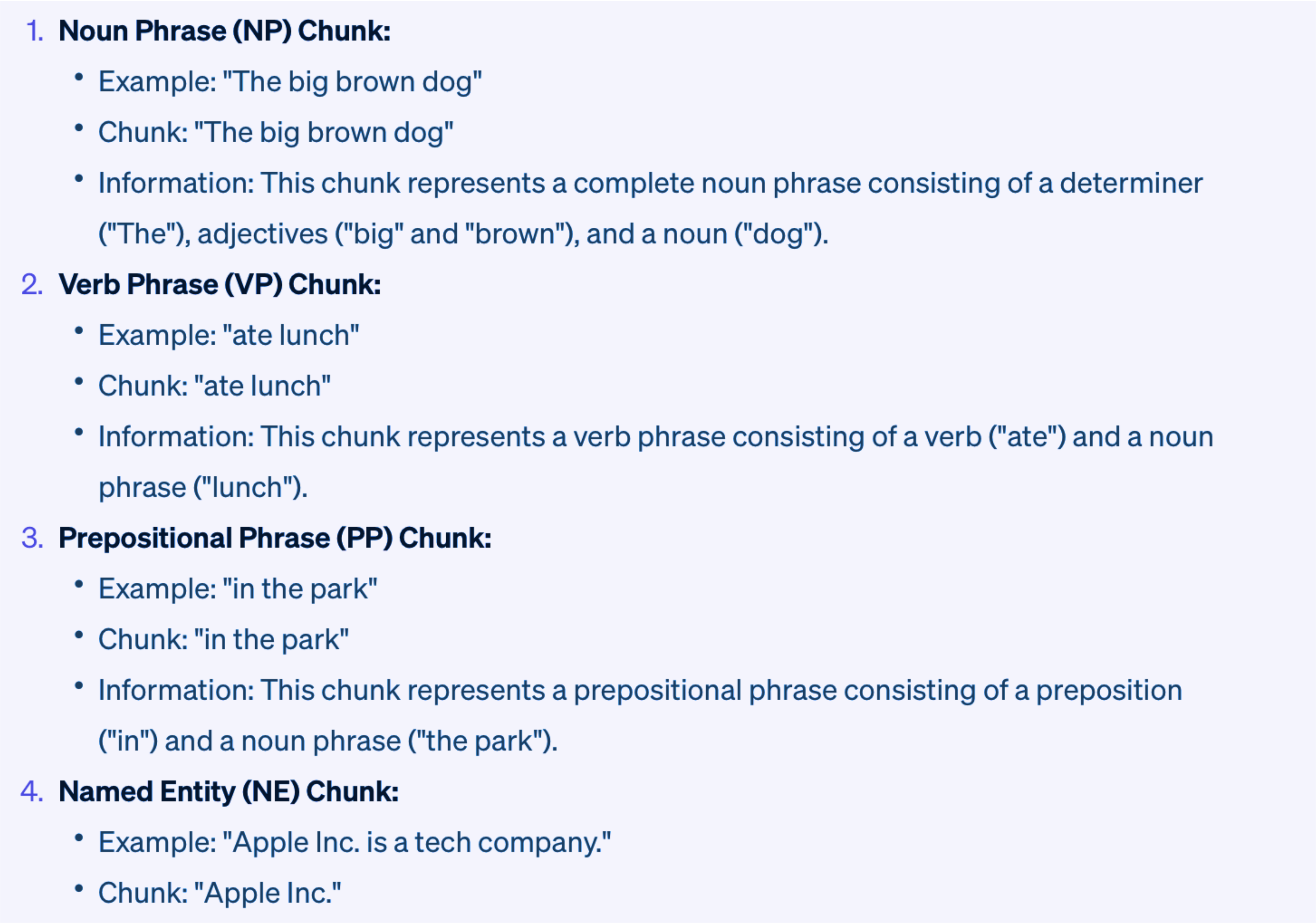
- Solved by approximated solutions :Hidden Markov Models, Conditional Random Fields, Recurrent Neural Networks , Transformers
- Also known as sequence labeling problem

Training Data for PoS taggers

- Used for determining the conditional probability values / estimating parameters of PoS tagging models
- A tagged dataset – or **POS Annotated Corpus**
- **Example POS Corpus:**
    - Penn Treebank Tagset (PTB Tagset)
    - Tweet tagset(CMU)
    - Brown Corpus Tagset
    - Google Universal POS Tagset
    - Indic POS Corpus

Recap: Chunking

- Chunking, involves identifying and grouping adjacent words or tokens in a sentence into meaningful units, often based on their grammatical structure or semantic roles.



1. **Noun Phrase (NP) Chunk:**
   - Example: "The big brown dog"
   - Chunk: "The big brown dog"
   - Information: This chunk represents a complete noun phrase consisting of a determiner ("The"), adjectives ("big" and "brown"), and a noun ("dog").
2. **Verb Phrase (VP) Chunk:**
   - Example: "ate lunch"
   - Chunk: "ate lunch"
   - Information: This chunk represents a verb phrase consisting of a verb ("ate") and a noun phrase ("lunch").
3. **Prepositional Phrase (PP) Chunk:**
   - Example: "in the park"
   - Chunk: "in the park"
   - Information: This chunk represents a prepositional phrase consisting of a preposition ("in") and a noun phrase ("the park").
4. **Named Entity (NE) Chunk:**
   - Example: "Apple Inc. is a tech company."
   - Chunk: "Apple Inc."

Rule Based Chunking

- Define a POS RegEx based grammar rule

  r"NP: {<DT>?<JJ>*<NN.*>} ”

- Obtain part of speech tags for the input sentence first

  The_DT quick_JJ brown_JJ fox_NN jumps_VBZ over_IN the_DT lazy_JJ dog_NN ._.

- Extract token sequences that conform to the pattern.
  - The quick brown fox
  - the lazy dog

ML based Chunking

- Similar to PoS Taggeing, chunking involves solving sequence labeling problems.
- Approaches include, Supervised Machine Learning, Deep Learning and Transfer Learning based approaches.
- Tags are often specified by the BIO tagging scheme.

BIO tagging scheme for chunking

- **B: Beginning** - Indicates the first token of a chunk.
- **I: Inside** - Indicates a token inside a chunk (following the first token).
- **O: Outside** - Indicates a token that is not part of any chunk.

Example

- Sentence: "The quick brown fox jumps over the lazy dog.”
- Say, we are interested in Noun Chunking

The" - B-NP

"quick" - I-NP

"brown" - I-NP

"fox" - I-NP

"jumps" - O

"over" - O

"the" - B-NP

"lazy" - I-NP

"dog" - I-NP

So far in I320D – Text Mining and NLP

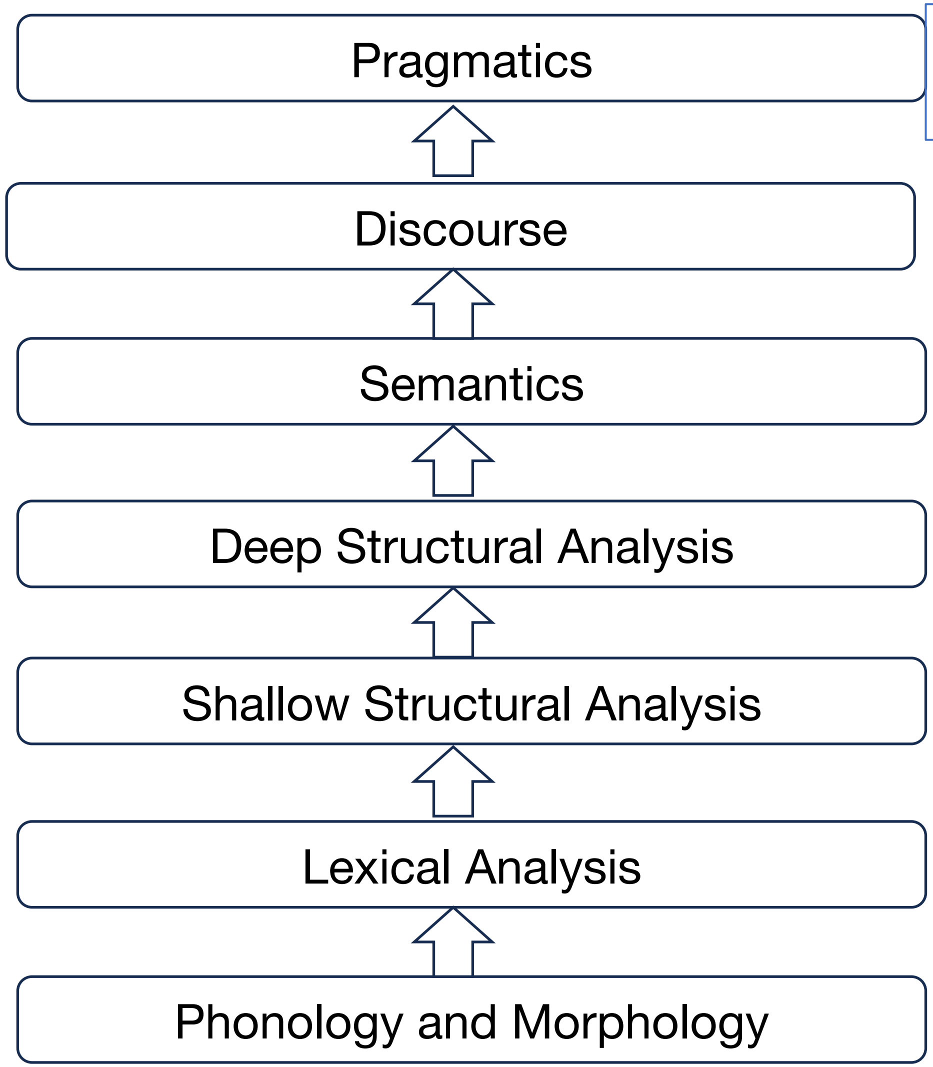
- W1. Language and Ambiguity
- W2. Basics of Text Data and Linguistic Concepts
- W3. Text Preprocessing Techniques
- W4. Lexical Analysis
- W5. Syntax Analysis
- W6. Information Extraction

W7. Machine Learning Methods for NLP
W8. Unsupervised ML and Topic Modeling Basics
W10-W11. Deep learning for NLP
W12. NLP Applications
W13. Small and Large Language Models and Prompt Engineering Basics
W14. Knowledge Networks
W15. Evaluation Metrics

Week 6: Roadmap

- Deep Parsing
  - Constituency and Dependency Parsing
- Information Extraction
  - Named Entity Identification

Recap : Layered View of NLP



Pragmatics

Discourse

Semantics

Deep Structural Analysis

Shallow Structural Analysis

Lexical Analysis

Phonology and Morphology

Context beyond presented text (world knowledge

Context from multiple sentences / documents

Context within to a sentence

Deep Syntax, Parses, Dependency between phrases

Syntax, Part-of-speech, Expressions

Word meaning without context

Words and surface forms

Parsing

Sentences are linear structures

But there is a hierarchy- a tree hidden behind the linear structure

There are constituents and branches

Remember Verb Phrase Extraction in Last Tutorial?

- If you closely monitor the examples given in the assignment statement, a verb phrase can subsume a noun phrase (plus some additional POS).

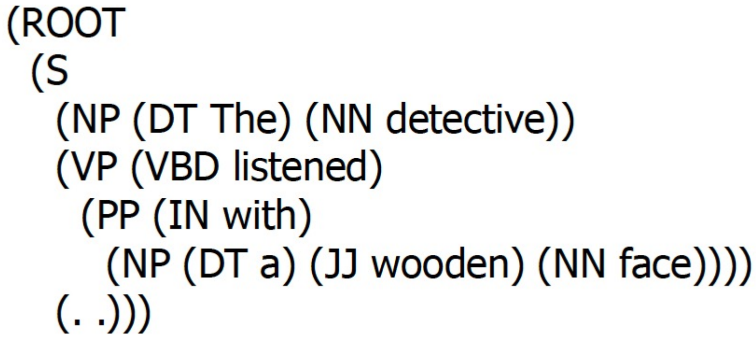
- So, it could have the following patterns (there could be many more, just showing some common ones)
  - VP => VERB + NP
  - VP => VERB + PREPOSITION+ NP
- ***Phrase are called constituents. Extracting phrase representation is Constituency Parsing***

Hypothetical Example : Metaphor Detection

- The detective listened to her tales with a wooden face.
- She was fairly certain that life was a fashion show.
- The typical teenage boy’s room is a disaster area.
- The children were roses grown in concrete gardens.
- Waves of spam emails inundated his inbox.

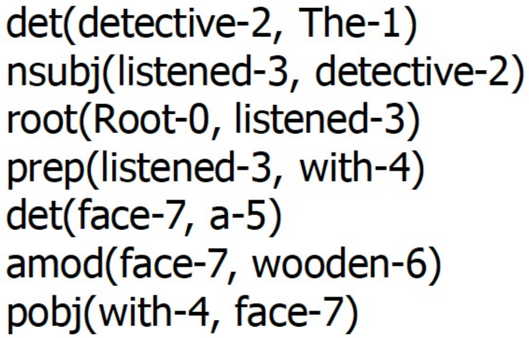
Parsing the sentence, “The detective…”

Constituency Parsing



```
(ROOT
  (S
    (NP (DT The) (NN detective))
    (VP (VBD listened)
      (PP (IN with)
        (NP (DT a) (JJ wooden) (NN face))))
    (. .)))
```

Dependency Parsing



```
det(detective-2, The-1)
nsubj(listened-3, detective-2)
root(Root-0, listened-3)
prep(listened-3, with-4)
det(face-7, a-5)
amod(face-7, wooden-6)
pobj(with-4, face-7)
```

Example: Sentiment Analysis + Metaphor

- A well-designed, but not overly stylized, interface makes IMDB easy to navigate, although you may find yourself going down plenty of rabbit holes as one page of facts leads to another, and another, and so on

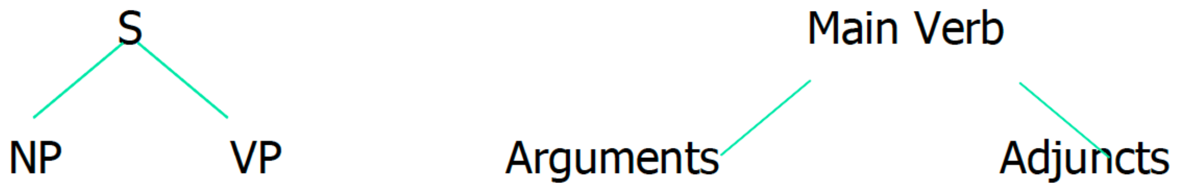
- Extremely challenging

“A well-designed, but not overly stylized, interface makes IMDB easy to navigate”

A/DT
well-designed/JJ
,/,
but/CC
not/RB
overly/RB
stylized/JJ
,/,
interface/NN
makes/VBZ
IMDB/NNP
easy/JJ
to/TO
navigate/VB
./.

root(ROOT-0, makes-10)
nsubj(makes-10, interface-9)
det(interface-9, A-1)
amod(interface-9, well-designed-2)
cc(well-designed-2, but-4)
neg(stylized-7, not-5)
advmod(stylized-7, overly-6)
conj(well-designed-2, stylized-7)
nsubj(easy-12, IMDB-11)
xcomp(makes-10, easy-12)
aux(navigate-14, to-13)
xcomp(easy-12, navigate-14)

Two kinds of parse representations: Constituency Vs. Dependency



S

NP VP

Main Verb

Arguments Adjuncts

- What is more important?
  - Noun or Verb
  - Sanskrit Tradition (Dhatujamah - धातुजमाह
    i.e. everything is derived from verbal root)

Dependency Parsing

- Dependency approach is suitable for free word-order language

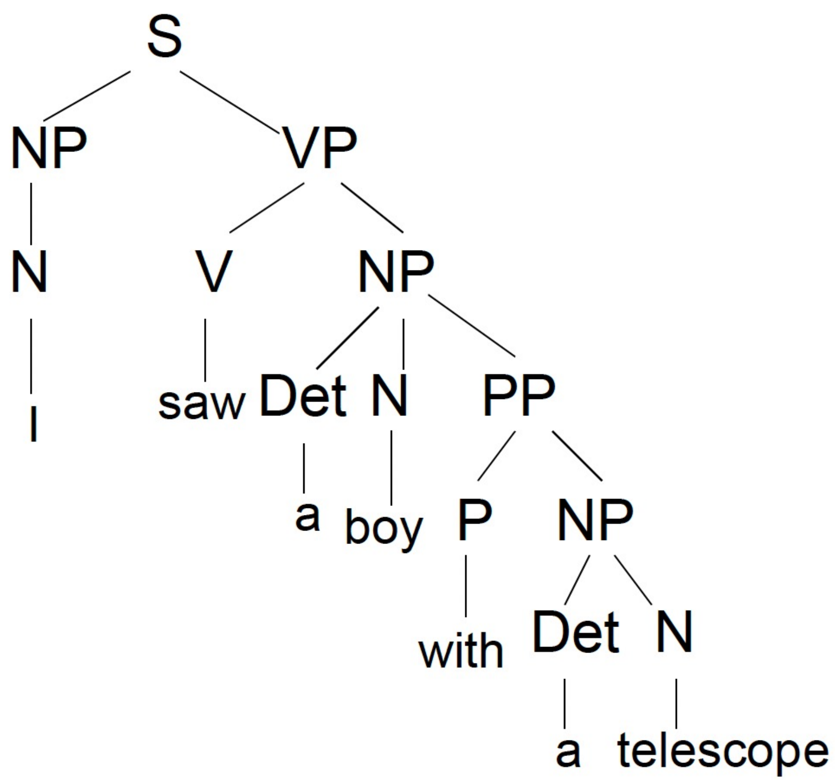
- Example : Hindi
  - राम ने शाम को देखा (Ram ne Shyam ko dekha)
  - शाम को राम ने देखा (Shyam ko Ram ne dekha)

- One step closer to **Semantics**

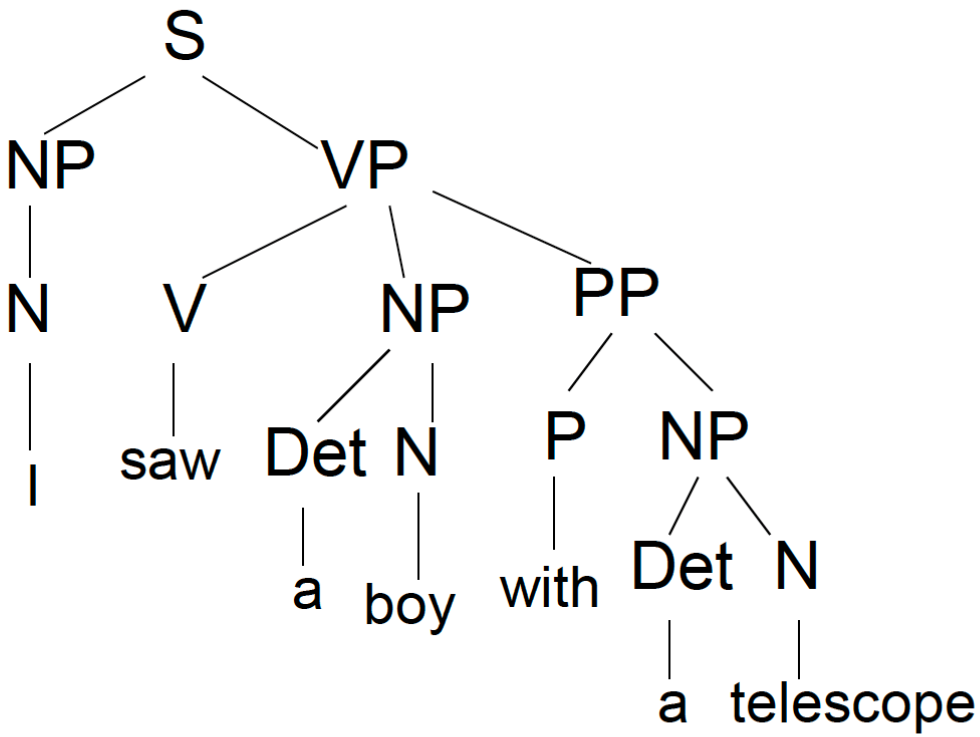
Parse Tree

- Within a sub-tree entities bind together more than they do with entities outside the sub-tree
- Example:
  - I saw the boy with a telescope

Constituency Parse Tree Candidates

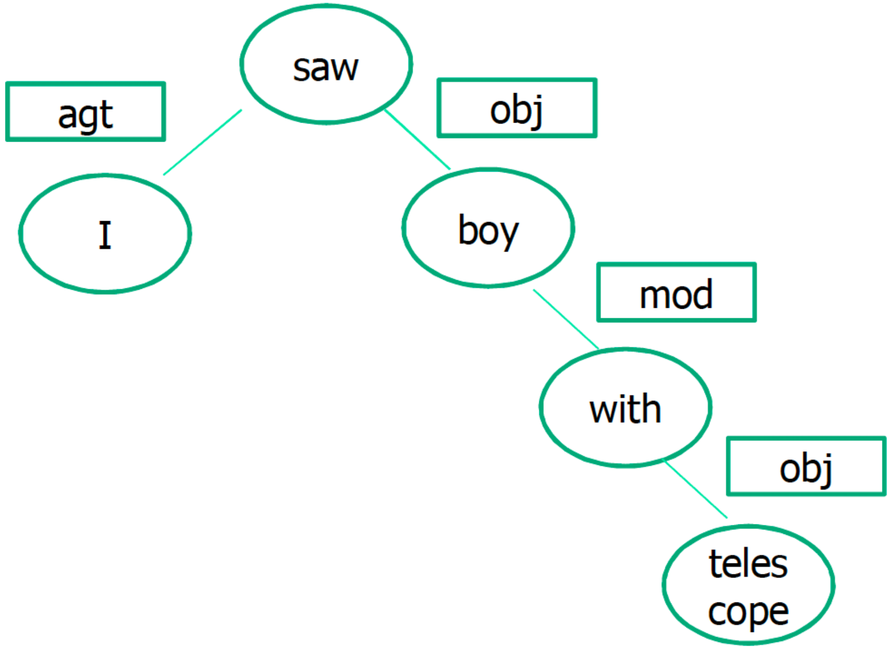


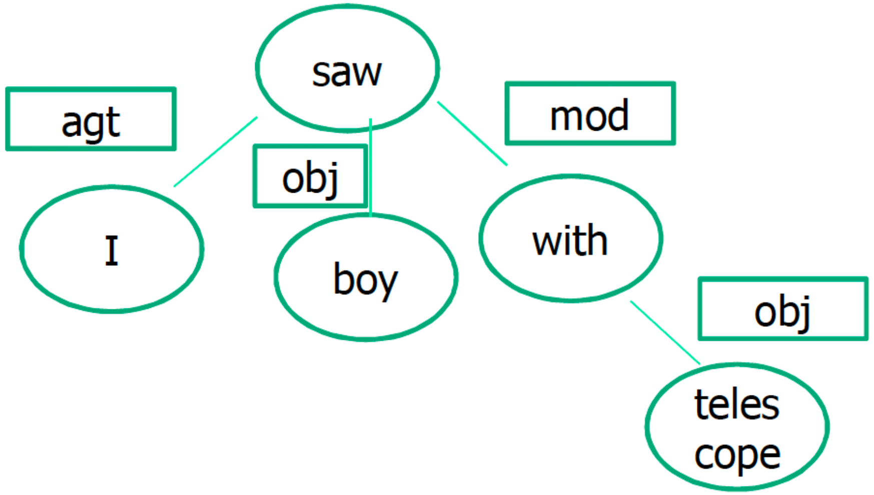
Candidate 1



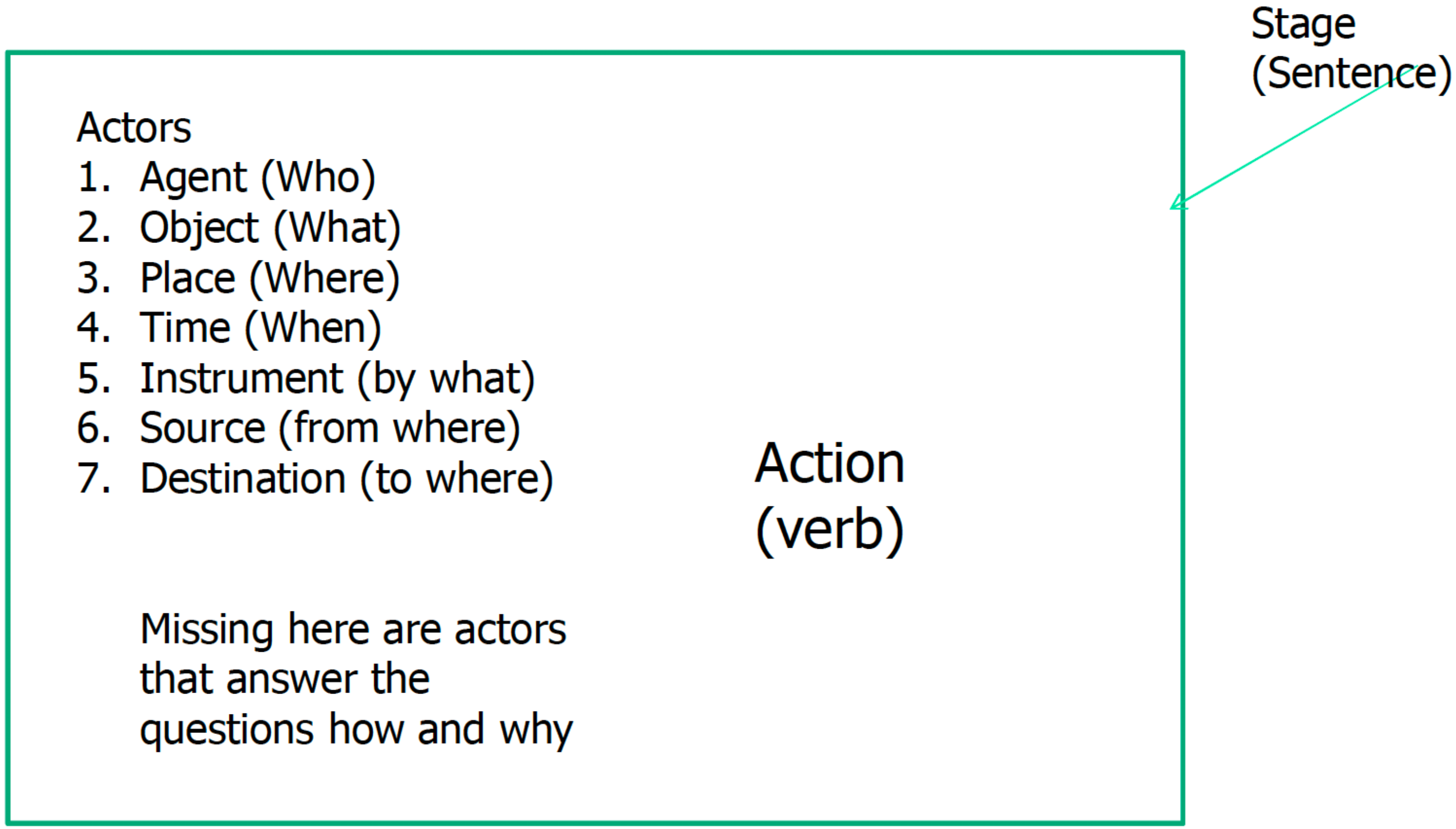
Candidate 2

Dependency Candidates





Verb Centric View of a Sentence



Stage
(Sentence)

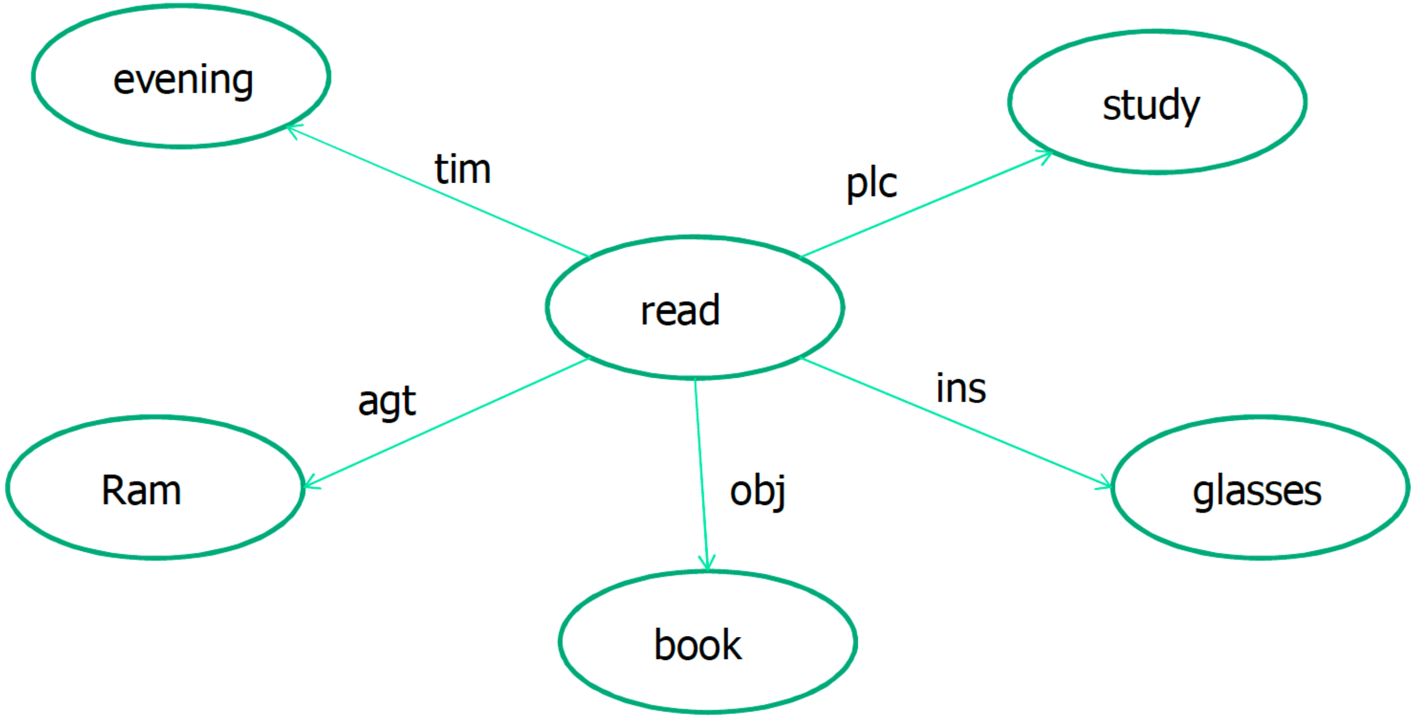
Actors

1. Agent (Who)
2. Object (What)
3. Place (Where)
4. Time (When)
5. Instrument (by what)
6. Source (from where)
7. Destination (to where)

Action
(verb)

Missing here are actors
that answer the
questions how and why

- Ram reads a book with his glasses in the evening in his study.



- The labels on the arcs are semantic roles and the task is Semantic Role Labeling.

Importance of Words - Dependency

- Bill shot at the President
  - Emphasizes Bill – agent
- The president was shot at by Bill
  - Emphasizes the President – object
- The President was shot at
  - Emphasizes shooting – action

Parsing Models

- Constituency
  - Probabilistic parsing (similar MLE formulation as POS tagging)
  - Grammar based parsing
- Dependency
  - Probabilistic parsing (similar MLE formulation as POS tagging)
  - Recurrent neural network based solutions
  - Universal Dependency Treebank – a dataset containing manually annotated trees
  - Tagset and Roles: https://universaldependencies.org

Named Entity Recognition – Another Sequence Labeling Task

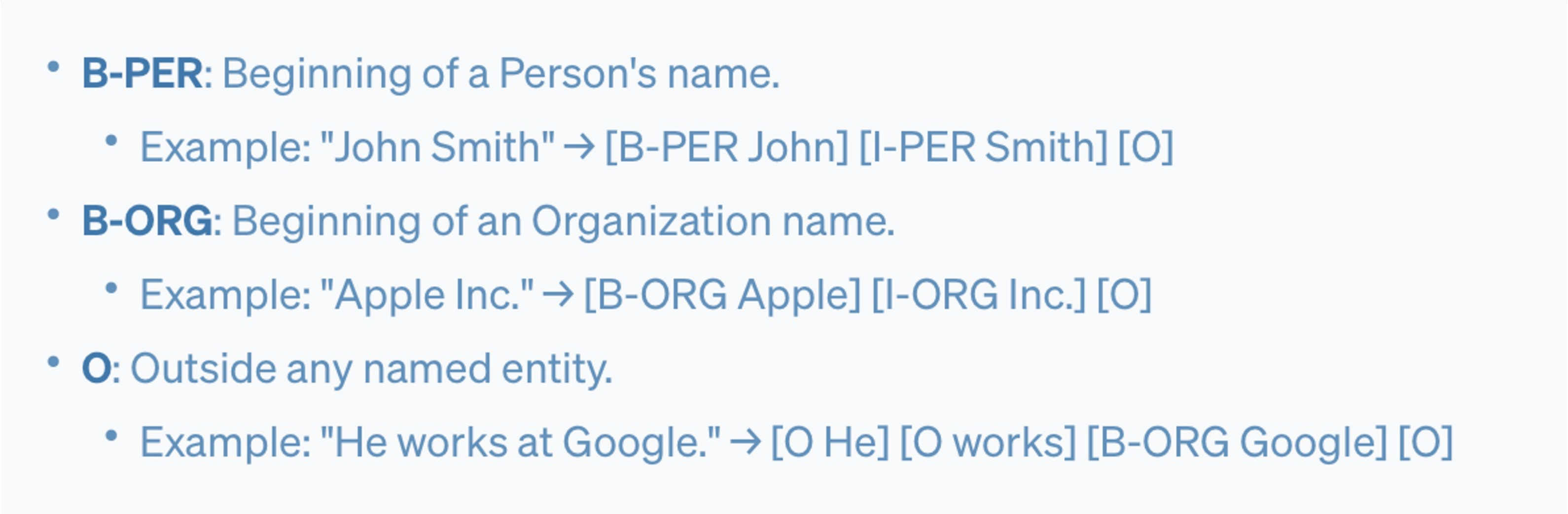
- Identifies and classifies entities in text.
- Common entities: Person, Organization, Date, Location.
- Helps extract structured information from unstructured text.

NER Applications

- Information retrieval: Improve search engines.
- Named entity linking: Connect entities to databases.
- Sentiment analysis: Identify entity sentiments.
- Chatbots: Enhance conversational AI understanding.
- Legal: Assist in contract analysis and compliance.

Named Entity Recognition (NER) with BIO Format

- **BIO** stands for **B**eginning, **I**nside, and **O**utside.
- A common encoding scheme for NER annotations.



- **B-PER**: Beginning of a Person's name.
  - Example: "John Smith" → [B-PER John] [I-PER Smith] [O]
- **B-ORG**: Beginning of an Organization name.
  - Example: "Apple Inc." → [B-ORG Apple] [I-ORG Inc.] [O]
- **O**: Outside any named entity.
  - Example: "He works at Google." → [O He] [O works] [B-ORG Google] [O]

- NER models: Sequence labeling models similar to POS tagging used

Putting Everything Together – Real World Scenario

- **Scenario:** You are a data analyst at a hospital, and your task is to extract vital patient information from medical reports to create a comprehensive database for clinical research.

Putting Everything Together – Real World Scenario

Your data (hypothetical)

- **Patient:** John Doe
- **Age:** 45
- **Date of Admission:** September 15, 2023
- **Diagnosis:** Mr. Doe presented with complaints of chest pain and shortness of breath. Physical examination revealed elevated blood pressure and irregular heartbeat.
- **Treatment:** He was prescribed medication (Aspirin 81mg, Lisinopril 10mg) to manage hypertension and referred for further cardiac evaluation.
- **Follow-up:** The patient is scheduled for an echocardiogram on September 20, 2023.
- **Attending Physician:** Dr. Smith

What POS Tagging will yield?

Patient: ["John" (PROPN) "Doe" (PROPN)]

Age: ["45" (NUM)]

Date of Admission: ["September" (PROPN) "15" (NUM), "2023" (NUM)]

Diagnosis: ["Mr." (PROPN) "Doe" (PROPN) "presented" (VERB) "with" (ADP) "complaints" (NOUN) of (ADP) "chest" (NOUN) "pain" (NOUN) and (CCONJ) "shortness" (NOUN) of (ADP) "breath" (NOUN). ...

Chunking ?

Patient: ["John Doe"]

Age: ["45"]

Date of Admission: ["September 15, 2023"]

Diagnosis: ["Mr. Doe", "complaints", "chest pain", "shortness of breath", "blood pressure", "irregular heartbeat"]

Treatment: ["medication (Aspirin 81mg, Lisinopril 10mg)", "hypertension", "further cardiac evaluation"]

Follow-up: ["echocardiogram", "September 20, 2023"]

Attending Physician: ["Dr. Smith"]

Named Entity Extraction

- Patient: [("John Doe", "PERSON")]
- Age: [("45", "AGE")]
- Date of Admission: [("September 15, 2023", "DATE")]

- Diagnosis: [("Mr. Doe", "PERSON"), ("chest pain", "SYMPTOM"), ("shortness of breath", "SYMPTOM"), ("blood pressure", "SYMPTOM"), ("irregular heartbeat", "SYMPTOM")]
- Treatment: [("Aspirin 81mg", "MEDICATION"), ("Lisinopril 10mg", "MEDICATION"), ("hypertension", "CONDITION"), ("further cardiac evaluation", "PROCEDURE")]
- Follow-up: [("echocardiogram", "PROCEDURE"), ("September 20, 2023", "DATE")]
- Attending Physician: [("Dr. Smith", "PERSON")]

Combining the outputs – Database Entry

{ "**Patient**": "John Doe", "Age": "45", "Date of Admission": "September 15, 2023",

"**Diagnosis**": ["chest pain", "shortness of breath", "blood pressure", "irregular heartbeat"],

"**Treatment**": ["Aspirin 81mg", "Lisinopril 10mg", "hypertension", "further cardiac evaluation”],

"**Follow-up**": ["echocardiogram", "September 20, 2023"],

"**Attending Physician**": "Dr. Smith"

}

Next Class

- **Practicum:**
  - Parsing and Named Entity Identification Examples
  - Extracting information tuples from raw text