CSCE 771: Computer Processing of Natural Language
Lecture 6: Shallow Parsing, Dependency Parsing, Project Review

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

5TH SEPTEMBER, 2024

Carolinian Creed: "I will practice personal and academic integrity."

Acknowledgement: Used materials by Profs. Mausam, Jurafsky & Martin, Stanford NLP

Organization of Lecture 6

- Opening Segment
  - Review of Last Lecture
- Main Lecture
- Concluding Segment
  - About Next Lecture – Lecture 7

Main Section
- Project: complete reviewing of topics
- Shallow Parsing
- Dependency Parsing

Recap of Lecture 5

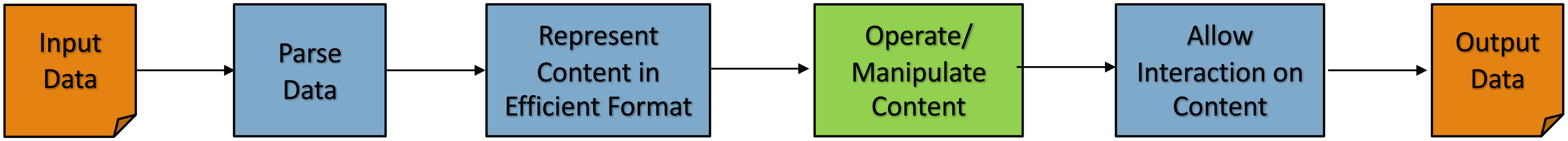
- We discussed the paper - ”Contextual Word Representations: Putting Words into Computers”, by Noah Smith, CACM June 2020”
- We looked at parsing
  - Roles it plays: verifying , generating, recognizing
  - Many types of parsing: shallow parsing for quick NLP tasks, phrase structure parsing, dependency parsing
- Started reviewing projects and states

Announcements

- Quiz 1 - next class, in-person and using paper and pen
  - No makeup (best of 3 from 4 quizzes)
  - Will cover concepts discussed in class

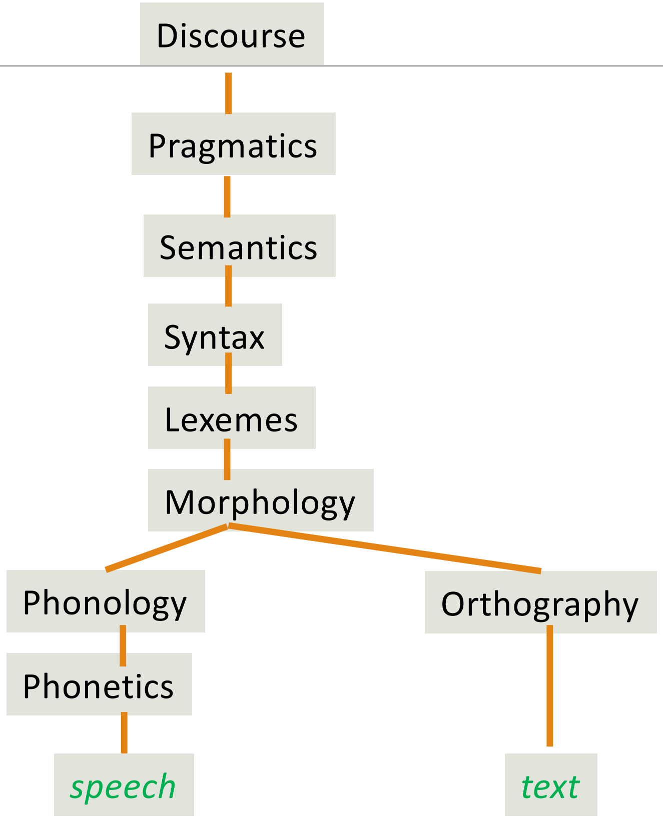
Main Lecture

Review: Parsing


Input Data
Parse Data
Represent Content in Efficient Format
Operate/ Manipulate Content
Allow Interaction on Content
Output Data

Parsing

Levels of Linguistic Studies



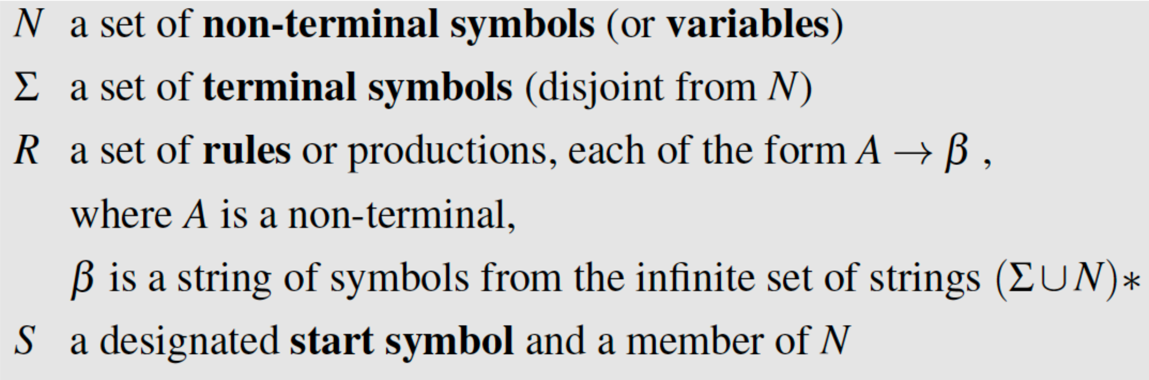
- **Discourse:** study of group of sentences
- **Pragmatics:** how context contributes to meaning of sentences
- **Semantics:** meaning of words and combinations of words
- **Syntax:** rules for combining and using words/ phonemes.
- **Lexemes:** a set of words that are related through inflection (fly: verb, fly: noun)
- **Morphology**—rules that govern morphemes - the minimal meaningful units of language (lemmas and affixes)
- **Orthography**: convention for writing a language. E.g., spelling
- **Phonology:** organization of speech sound (i.e., phoneme)
- **Phonetics**: study of how sound is made and received

Why Parsing

- Recognizing legal inputs from illegal
- Usage of parse representation - parse tree
  - Grammar checking
  - Semantic analysis
  - Machine translation
  - Question answering
  - Information extraction
  - Speech recognition
  - …

Adapted from material by Robert C. Berwick

Background: Context Free Grammar (CFG)



N a set of **non-terminal symbols** (or **variables**)

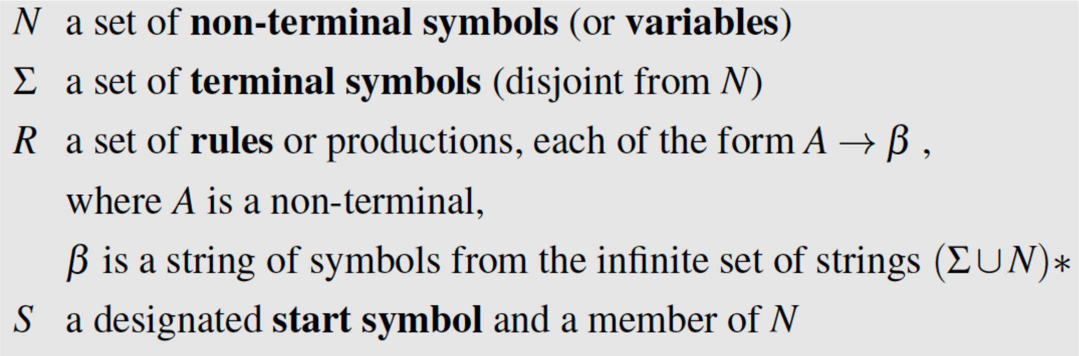
Σ a set of **terminal symbols** (disjoint from N)

R a set of **rules** or productions, each of the form $A \rightarrow \beta$,
where A is a non-terminal,
β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$

S a designated **start symbol** and a member of N

From Jurafsky & Martin

Review: Parsing - CFG



N a set of **non-terminal symbols** (or **variables**)

Σ a set of **terminal symbols** (disjoint from N)

R a set of **rules** or productions, each of the form $A \rightarrow \beta$,
where A is a non-terminal,
β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$

S a designated **start symbol** and a member of N

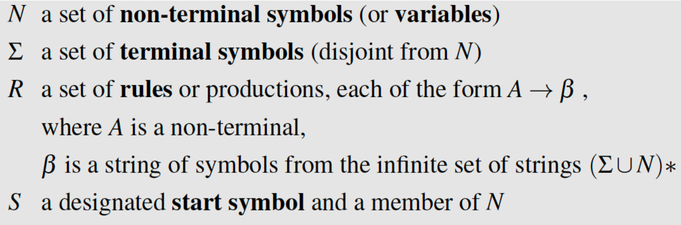
Example CFG:

- N = {S, NP, VP, }
- Σ = {he, she, walks, sleeps}
- R = {
  - S -> NP, VP
  - NP -> he
  - NP -> she
  - VP -> walks
  - VP -> sleeps

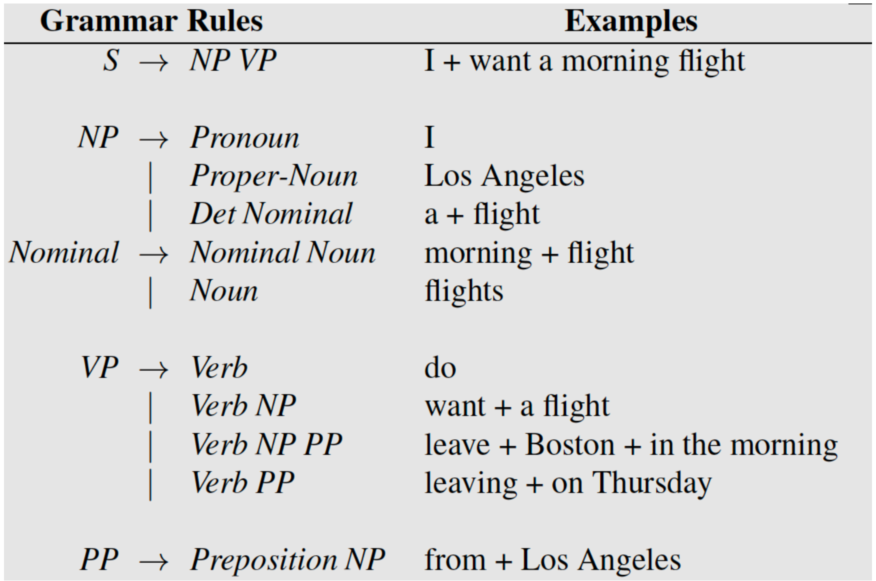
  }
- S = S

Questions: *which strings are in the language of example CFG*
(a) she sleeps (b) walks sheeps (c) sleeps he (d) she walks (e) he and she walks

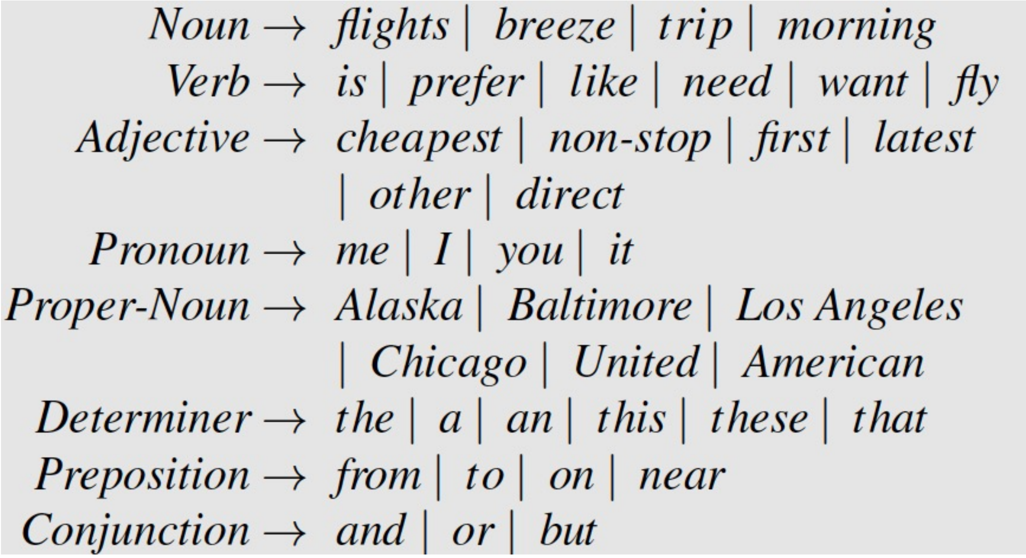
Simple Example Using CFGs



N a set of **non-terminal symbols** (or **variables**)
Σ a set of **terminal symbols** (disjoint from N)
R a set of **rules** or productions, each of the form $A \rightarrow \beta$,
where A is a non-terminal,
β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$
S a designated **start symbol** and a member of N



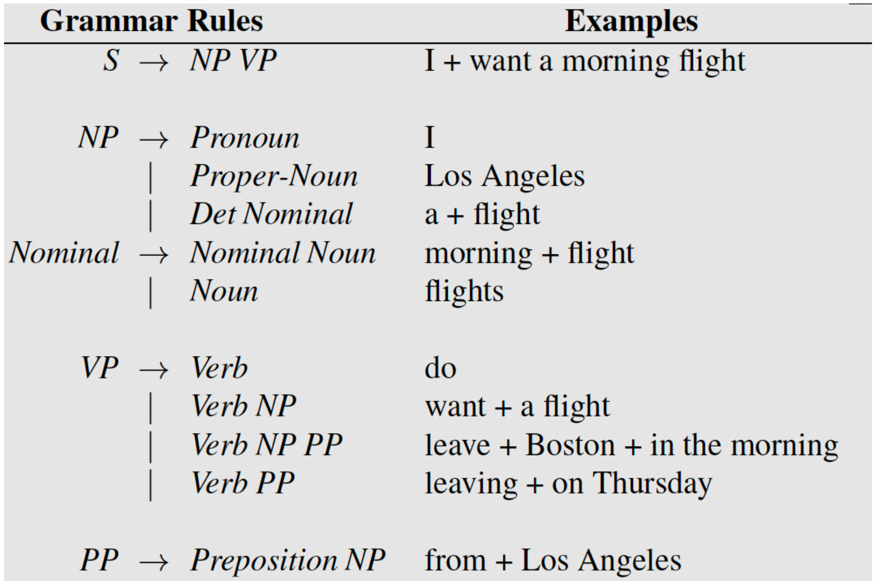
Grammar Rules	Examples
$S \rightarrow NP\ VP$	I + want a morning flight
$NP \rightarrow Pronoun$	I
$\mid$ *Proper-Noun*	Los Angeles
$\mid$ *Det Nominal*	a + flight
$Nominal \rightarrow Nominal\ Noun$	morning + flight
$\mid$ *Noun*	flights
$VP \rightarrow Verb$	do
$\mid$ *Verb NP*	want + a flight
$\mid$ *Verb NP PP*	leave + Boston + in the morning
$\mid$ *Verb PP*	leaving + on Thursday
$PP \rightarrow Preposition\ NP$	from + Los Angeles



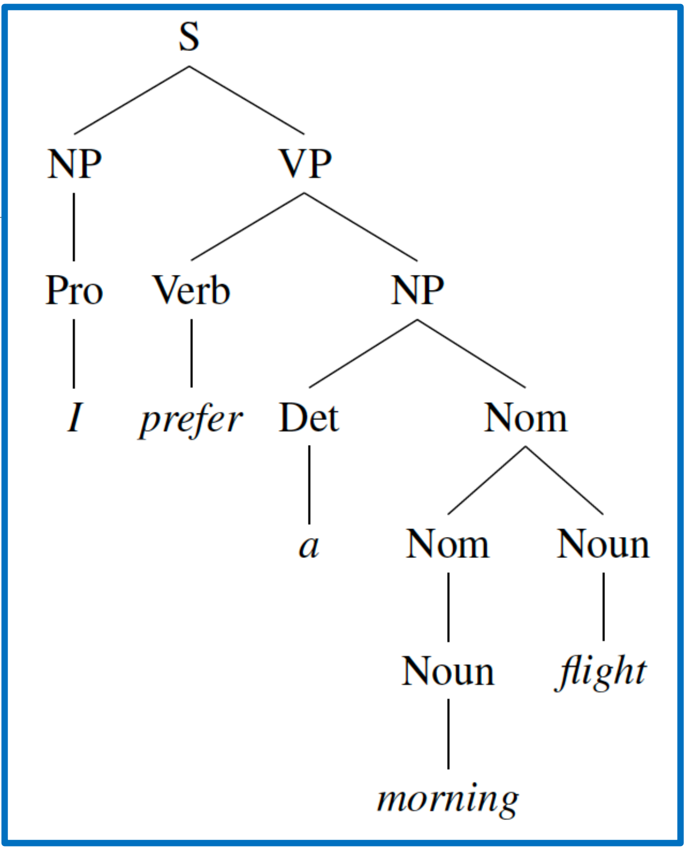
Noun → *flights* | *breeze* | *trip* | *morning*
Verb → *is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective → *cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun → *me* | *I* | *you* | *it*
Proper-Noun → *Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner → *the* | *a* | *an* | *this* | *these* | *that*
Preposition → *from* | *to* | *on* | *near*
Conjunction → *and* | *or* | *but*

From Jurafsky & Martin

An Example Using CFGs



Grammar Rules	Examples
S → *NP VP*	I + want a morning flight
NP → *Pronoun*	I
| *Proper-Noun*	Los Angeles
| *Det Nominal*	a + flight
Nominal → *Nominal Noun*	morning + flight
| *Noun*	flights
VP → *Verb*	do
| *Verb NP*	want + a flight
| *Verb NP PP*	leave + Boston + in the morning
| *Verb PP*	leaving + on Thursday
PP → *Preposition NP*	from + Los Angeles

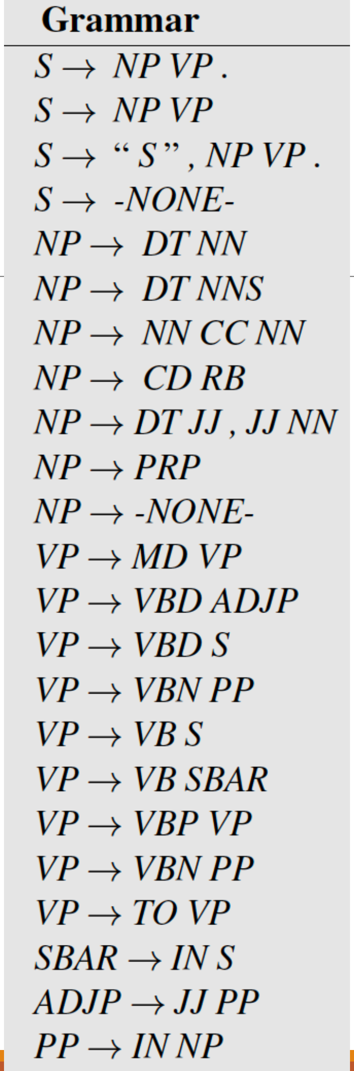


From Jurafsky & Martin

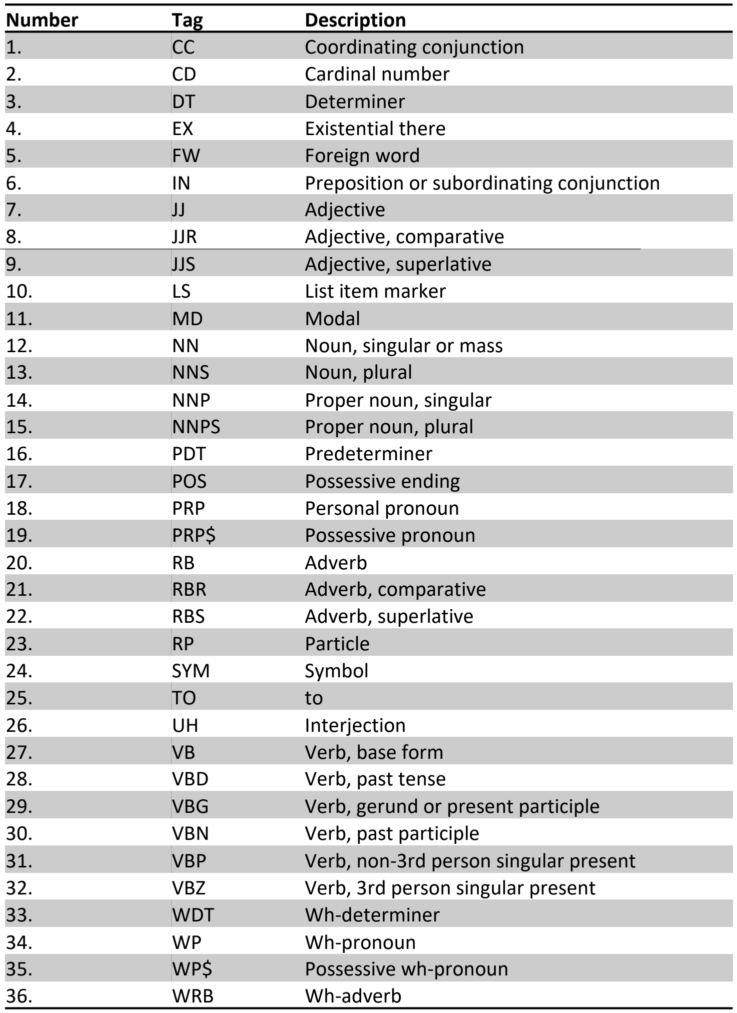
$[_S [_{NP} [_{Pro}\ \text{I}]] [_{VP} [_V\ \text{prefer}] [_{NP} [_{Det}\ \text{a}] [_{Nom} [_N\ \text{morning}] [_{Nom} [_N\ \text{flight}]]]]]]$

Bracketed Notation

Example: Larger English CFG



Grammar
$S \rightarrow NP\ VP\ .$
$S \rightarrow NP\ VP$
$S \rightarrow$ “ S ” , $NP\ VP\ .$
$S \rightarrow$ -NONE-
$NP \rightarrow DT\ NN$
$NP \rightarrow DT\ NNS$
$NP \rightarrow NN\ CC\ NN$
$NP \rightarrow CD\ RB$
$NP \rightarrow DT\ JJ\ ,\ JJ\ NN$
$NP \rightarrow PRP$
$NP \rightarrow$ -NONE-
$VP \rightarrow MD\ VP$
$VP \rightarrow VBD\ ADJP$
$VP \rightarrow VBD\ S$
$VP \rightarrow VBN\ PP$
$VP \rightarrow VB\ S$
$VP \rightarrow VB\ SBAR$
$VP \rightarrow VBP\ VP$
$VP \rightarrow VBN\ PP$
$VP \rightarrow TO\ VP$
$SBAR \rightarrow IN\ S$
$ADJP \rightarrow JJ\ PP$
$PP \rightarrow IN\ NP$



Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Table Source:
https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Interpretation of Parsing Rules

- generation (production): S → NP VP
- parsing (comprehension): S ← NP VP
- verification (checking): S = NP VP
- CFGs are <u>declarative</u> – tell us *what* the well-formed structures & strings are
- Parsers are <u>procedural</u> – tell us *how* to compute the structure(s) for a given string

From Robert C. Berwick

Types of Parsing

- **Phrase structure** / **Constituency Parsing**: find phrases and their recursive structure.

Constituency - groups of words behaving as single units, or constituents.

  - **Shallow Parsing/ Chunking**: identify the flat, non-overlapping segments of a sentence: noun phrases, verb phrases, adjective phrases, and prepositional phrases.

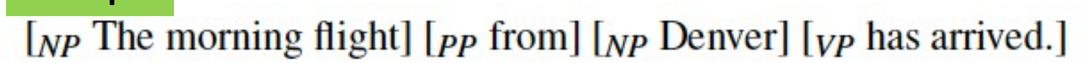
- **Dependency Parsing**: find relations in sentences

- **Probabilistic Parsing**: given a sentence X, predict the most probable parse tree Y

Chunking

- Chunking - process of identifying and classifying the flat, non-overlapping segments of a sentence that constitute the basic **non-recursive phrases** corresponding to the major content-word parts-of-speech:
  - noun phrases
  - verb phrases
  - adjective phrases, and
  - prepositional phrases

Example



[$_{NP}$ The morning flight] [$_{PP}$ from] [$_{NP}$ Denver] [$_{VP}$ has arrived.]

- Two operations in this type of parsing:
  - segmenting - finding the non-overlapping extents of the chunks and
  - labeling - assigning the correct tag to the discovered chunks
- Some words may not be part of any chunk

From Jurafsky & Martin

Shallow Parsing/ Chunking

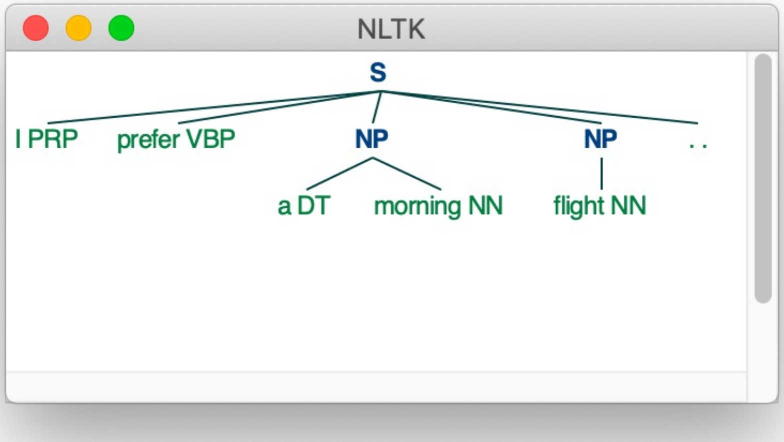
```
data = "I prefer a morning flight."

# Prepare data
tokens = nltk.word_tokenize(data)
tag = nltk.pos_tag(tokens)

# Grammar to use
grammar = "NP: {<DT>?<JJ>*<NN>}"
cp  =nltk.RegexpParser(grammar)

# Parse based on regex
result = cp.parse(tag)
print(result)
```

```
(S I/PRP prefer/VBP (NP a/DT morning/NN) (NP flight/NN) ./.)
```



Code and Examples

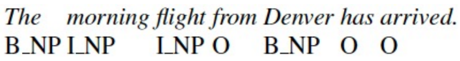
- Sample code – https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l6-l7-parsing/Chunking%20-%20syntax%20exploration.ipynb

- Advanced examples – https://www.nltk.org/book/ch07.html

IOB notation

- Chunking - IOB tagging
  - B - beginning of each chunk type
  - I - inside of each chunk type
  - O - one for tokens outside (O) any chunk
- Total: (2N + 1) tags for N chunk types

Example



The	*morning*	*flight*	*from*	*Denver*	*has*	*arrived.*
B_NP	I_NP	I_NP	O	B_NP	O	O

From Jurafsky & Martin

Code and Examples

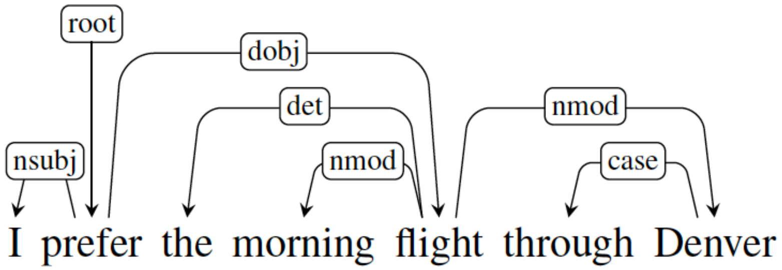
Sample 1 (nltk): https://www.geeksforgeeks.org/nlp-iob-tags/

Sample 2 (nltk): https://github.com/japerk/nltk3-cookbook/blob/master/chapter3.py

Sample 3(Spacy): https://github.com/biplav-s/course-nl-f24/blob/main/sample-code/l6-parsing/parsing%20spacy-iob.ipynb

Dependency Parsing

- **Meaning** depends on
  - Words (lemmas) in a sentence
  - Their directed binary grammatical relations with other words (and not on CFGs)
- **Notation**: Labeled arcs are from heads to dependents



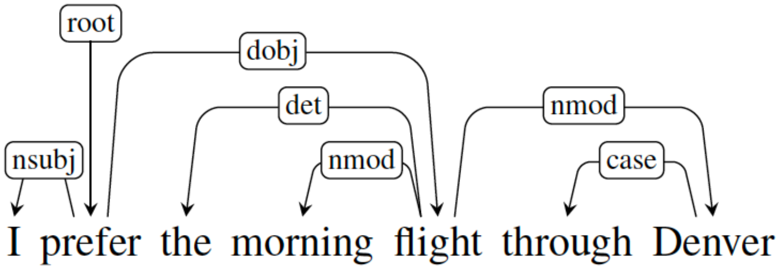
No node corresponding to phrasal constituents or lexical categories in the dependency parse

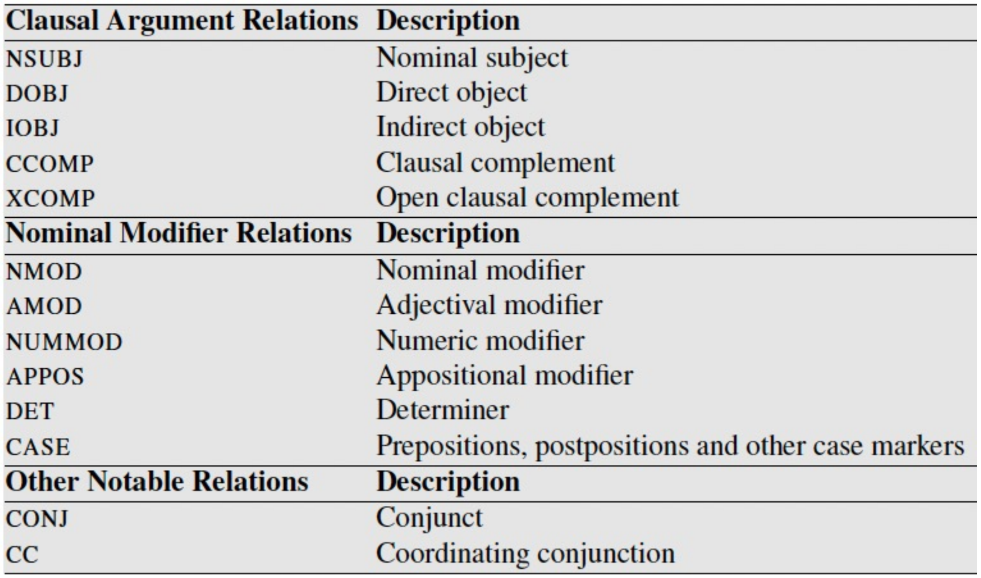
Dependency Conditions

1. There is a single designated root node that has no incoming arcs.
2. With the exception of the root node, each vertex has exactly one incoming arc.
3. There is a unique path from the root node to each vertex in V.

From Jurafsky & Martin

Dependency Parsing



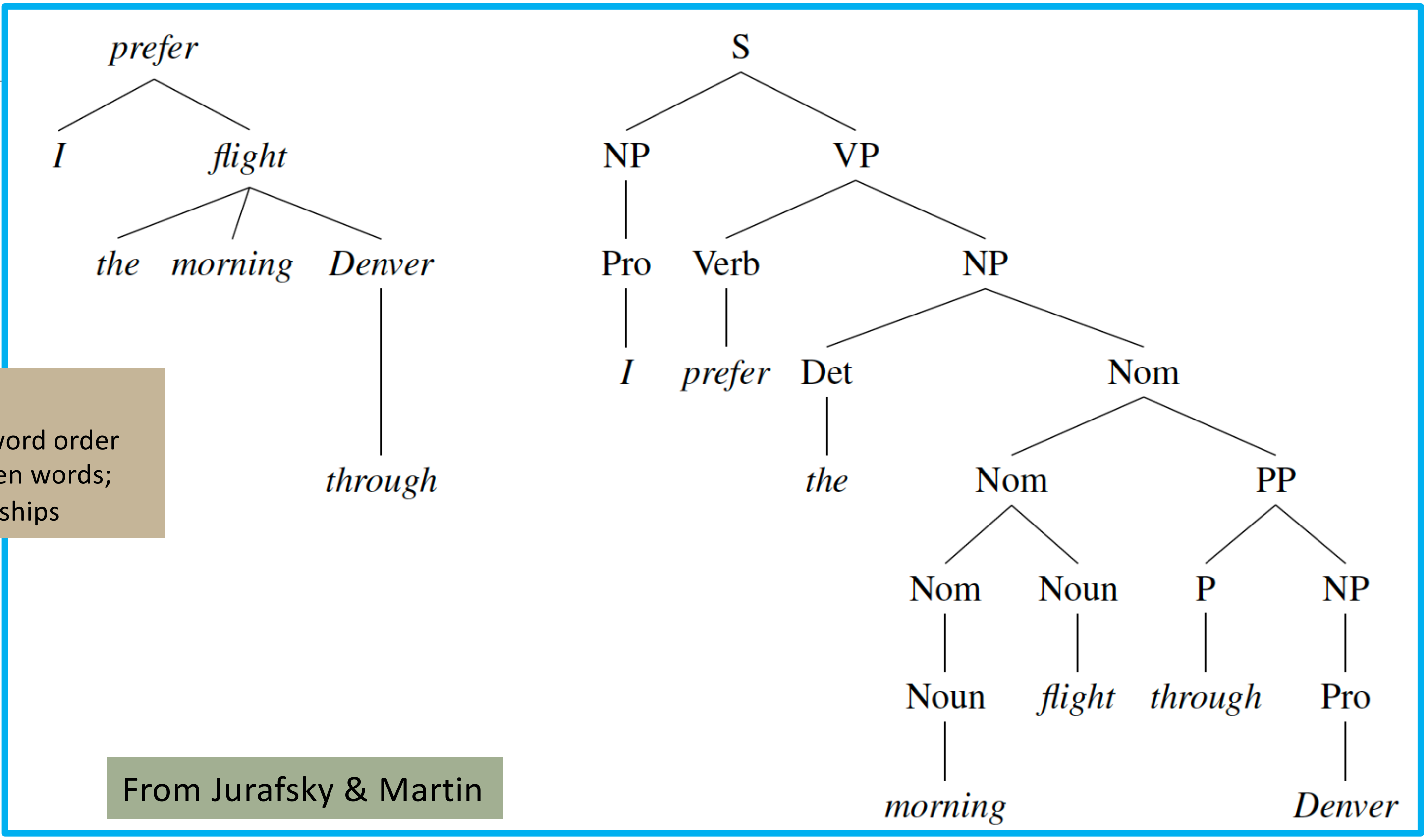


Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Edge: role that the dependent plays with respect to its head. Examples: subject, direct object and indirect object.

From Jurafsky & Martin

Comparison: Dependency and Phrase Structure

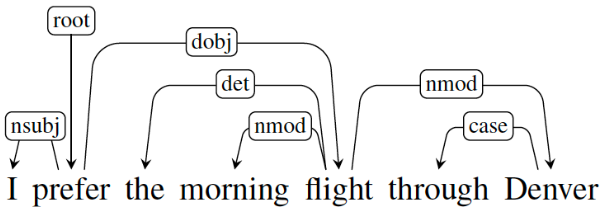


From Jurafsky & Martin

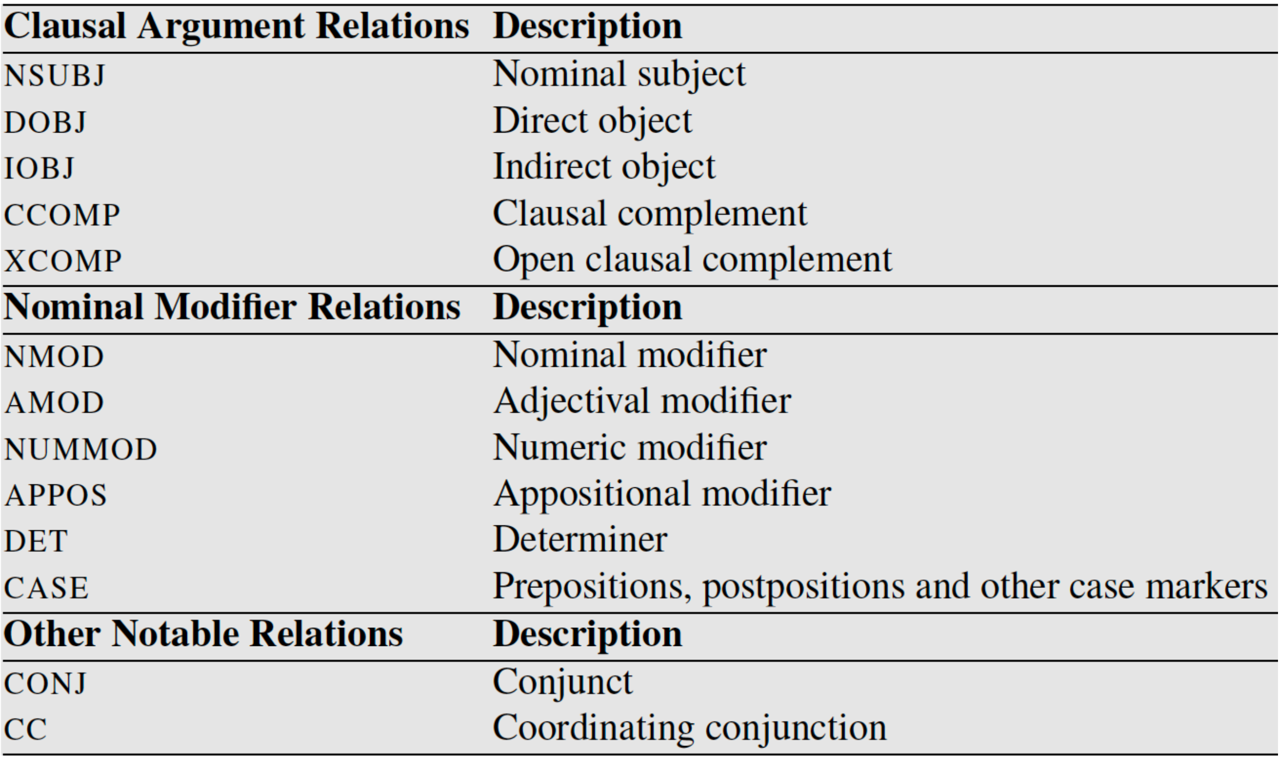
Dependency:

- Useful in languages with free word order
- Highlights relationships between words; useful is tasks needing relationships

Example Dependency Relationships



From Jurafsky & Martin



Clausal Argument Relations	**Description**
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Examples of Parsing with Spacy

- Sample code – https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l6-l7-parsing/parsing%20spacy.ipynb

See GitHub

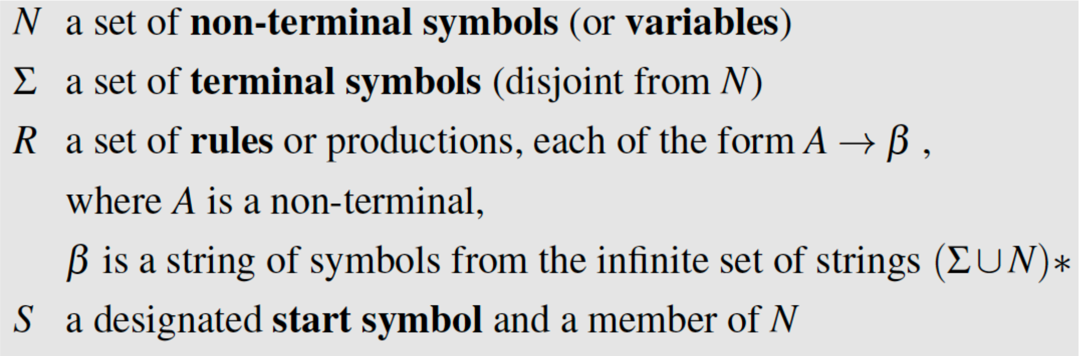
About Grammar Forms

- **Strong equivalent grammars**: Two grammars are strongly equivalent if they generate the same set of strings and if they assign the same phrase structure to each sentence
- **Weakly equivalent grammars**: Two grammars are weakly equivalent if they generate the same set of strings but do not assign the same phrase structure to each sentence.
- **Chomsky Normal Form**: a grammar which is
  - ϵ-free and
  - each production is either of the form A -> B C or A -> a.

- ***Any context-free grammar can be converted into a weakly equivalent Chomsky normal form grammar***
  - A -> B C D becomes
  - A -> B X and X -> C D

From Jurafsky & Martin

Review: Parsing - CFG



N a set of **non-terminal symbols** (or **variables**)

Σ a set of **terminal symbols** (disjoint from N)

R a set of **rules** or productions, each of the form $A \rightarrow \beta$,

where A is a non-terminal,

β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$

S a designated **start symbol** and a member of N

Example CFG:

- N = {S, NP, VP, }
- Σ = {he, she, walks, sleeps}
- R = {
  - S -> NP, VP
  - NP -> he
  - NP -> she
  - VP -> walks
  - VP -> sleeps

  }
- S = S

Questions: *which strings are in the language of example CFG*

(a) she sleeps (b) walks sheeps (c) sleeps he (d) she walks (e) he and she walks

Parsing Perspective

- **Question**: Is parsing of a sentence unique ?

Example 1: “*Book the dinner flight*”

- Book the flight which has dinner
- Book the flight for dinner

Example 2: *“I made her duck”*

- I cooked duck (**sense**: **animal**) for her
- I cooked duck (**sense: animal**) belonging to her.
- I turned her into duck (**sense**: **animal**)
- I created duck (**sense: object**) for her
- I made her to lower her head or body (**sense: posture**).

From Jurafsky & Martin

Parsing Perspective

- **Question**: Is parsing of a sentence unique ?
- **Answer**: Not necessarily

Issue: Then, which one to return?

Solution: Given a sentence X, predict its parse tree Y

Lecture 6: Concluding Comments

- We reviewed projects
- We reviewed parsers
  - Shallow parsers
  - Dependency parsers

Concluding Segment

Discussion: Course Project

Theme: Analyze quality of official information available for elections in 2024 [in a state]

- Take information available from
  - Official site: State Election Commissions
  - Respected non-profits: League of Women Voters
- Analyze information
  - State-level: Analyze quality of questions, answers, answers-to-questions
  - Comparatively: above along all states (being done by students)
- Benchmark and report
  - Compare analysis with LLM
  - Prepare report

- Process and analyze using NLP
  - Extract entities
  - Assess quality – metrics
    - Content – *Englishness*
    - Content – *Domain* -- election
  - … other NLP tasks
  - Analyze and communicate overall

Major dates for project check

- Sep 10: written – project outline
- Oct 8: in class
- Oct 31: in class // LLM
- Dec 5: in class // Comparative

Review current states chosen by others

Project Discussion

1. Go to Google spreadsheet against your name
2. Enter the **state** you will focus on for course project

---

1. Create a private Github repository called “CSCE771-Fall2024-<studentname>-Repo”. Share with Instructor (biplav-s) and TA (vr25)
2. Create Google folder called “CSCE771-Fall2024-<studentname>-SharedInfo”. Share with Instructor (prof.biplav@gmail.com) and TA (rawtevipula25@gmail.com)
3. Create a Google doc in your Google repo called “Project Plan” and have the following by Friday (Aug 30, 2024)

Timeline

1. Title: Analyze quality of official information available for elections in 2024 in <state>
2. Data need:
   1. Official: state’s election commission
   2. LWV: https://www.vote411.org/
3. Methods:
4. Evaluation:
5. Milestones
   - Sep 10: written and feedback
   - Oct 8: in class
   - Oct 31: in class
   - Dec 5: in class

Obtaining Election Data

Here are a few things to do:
A) **Official data** backed by laws: state election commission
a) Find the state's election commission
b) Find the Q/As they provide. They may be as FAQs or on different web pages.
c) Collect the Q/A programmatically

B) **Secondary data** sources: non-profit
a) Find Q/As from Vote 411 which is supported by the non-profit: LWV.

For reference, for SC,

A) Official - https://scvotes.gov/voters/voter-faq/

B) Secondary - https://www.vote411.org/south-carolina

For extraction, one or more approaches:
- Manually annotating
- BeautifulSoup,
- Tika
- or other open source libraries.

Discussion: Course Project

- Expectations
  - Apply methods learned in class or of interest to a problem of interest
  - Be goal oriented: aim to finish, be proactive, be innovative
  - Do top-class work: code, writeup, presentation
- Typical pitfalls
  - Not detailing out the project, assuming data
  - Not spending enough time
- What will be awarded
  - Results and efforts (balance)
  - Challenge level of problem

Review current states chosen by others

Course Project – Deadlines and Penalty Rubric

- Penalty
  - Missing milestones: **[-10%]**
  - Maximum**:** **[-40%]**
- Bonus possible
  - if two or more states considered

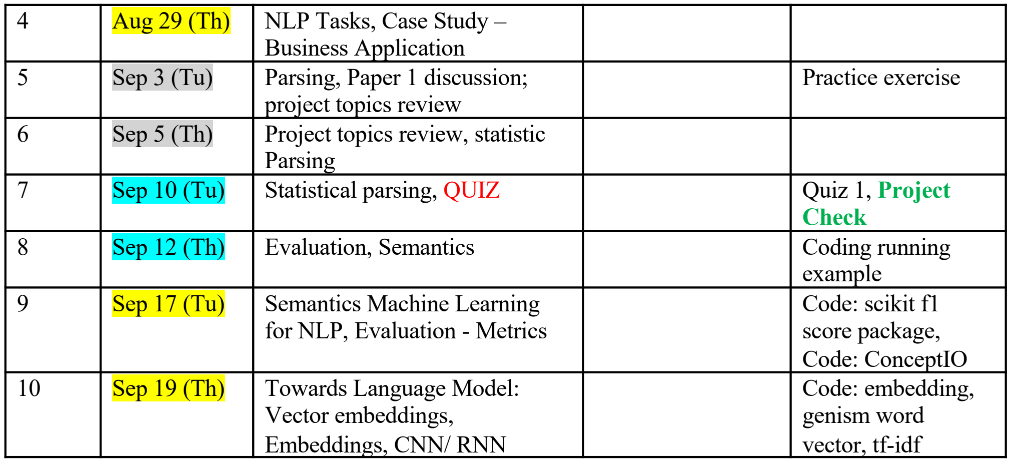
Timeline

1. Title: Analyze quality of official information available for elections in 2024 in <state>
2. Data need:
   1. Official: state’s election commission
   2. LWV: https://www.vote411.org/
3. Methods:
4. Evaluation:
5. Milestones
   - Sep 10: written and feedback
   - Oct 8: in class
   - Oct 31: in class
   - Dec 5: in class

About Next Lecture – Lecture 7

Lecture 7

- Statistical parsing
- QUIZ



4	Aug 29 (Th)	NLP Tasks, Case Study – Business Application		
5	Sep 3 (Tu)	Parsing, Paper 1 discussion; project topics review		Practice exercise
6	Sep 5 (Th)	Project topics review, statistic Parsing		
7	Sep 10 (Tu)	Statistical parsing, QUIZ		Quiz 1, **Project Check**
8	Sep 12 (Th)	Evaluation, Semantics		Coding running example
9	Sep 17 (Tu)	Semantics Machine Learning for NLP, Evaluation - Metrics		Code: scikit f1 score package, Code: ConceptIO
10	Sep 19 (Th)	Towards Language Model: Vector embeddings, Embeddings, CNN/ RNN		Code: embedding, genism word vector, tf-idf