NATURAL LANGUAGE PROCESSING

LESSON 3: N-GRAMS AND LANGUAGE MODELS

OUTLINE

- N-Grams
  - Language Models
  - N-Gram Models
  - Turkish Dictionary N-Grams

LANGUAGE MODELS

- Formal grammars like context free give a hard “binary” model of the legal sentences in a language: accept or reject.
- But for NLP, a probabilistic model of a language that gives a probability that a string is a member of a language is more useful.
- To specify a correct probability distribution, the probability of all sentences in a language must sum to 1.

USES OF LANGUAGE MODELS

- Speech recognition
  - “I ate a cherry” is a more likely sentence than “Eye eight uh Jerry”
- OCR & Handwriting recognition
  - Doctors are known to have bad handwriting. While pharmacists are reading this bad article, they model the similarity of the letter combination to the drug names on their hands.
- Machine translation
  - «Hanging the criminal» is translated as «killing the criminal», but «hanging the phone» is translated as «closing the phone».
- Context sensitive spelling correction
  - “Their are problems wit this sentence.”

COMPLETION PREDICTION

- A language model also supports predicting the completion of a sentence.
  - Please turn off your cell _____
  - Your program does not _____
- Predictive text input systems can guess what you are typing and give choices on how to complete it.

N-GRAMS

- The Markov assumption is the presumption that the future behavior of a dynamical system only depends on its recent history. In particular, in a kth-order Markov model, the next state only depends on the k most recent states, therefore an N-gram model is a (N-1)-order Markov model.
  - Unigram : P(phone)
  - Bigram : P(phone | cell)
  - Trigram : P(phone | your cell)

N-GRAMS MODELS

If we assume the sentence is as

$$w_1^n = w_1 \ldots w_n$$

Chain rule of probability

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)\ldots P(w_n \mid w_1^{n-1}) = \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

Bigram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-1})$$

N-gram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-N+1}^{k-1})$$

N-GRAMS MODELS

- Bigram counts for 7 of the words (out of 1.616 total word types) in Berkeley Restaurant Project Corpus of ~10.000 sentences.

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

N-GRAMS MODELS

P(<s> I want English food </s>)

= P(I | <s>) P(want | I) P(English | want)

P(food | English) P(</s> | food)

= .25 x .33 x .0011 x .5 x .68 = .000031

P(<s> I want Chinese food </s>)

= P(I | <s>) P(want | I) P(Chinese | want)

P(food | Chinese) P(</s> | food)

= .25 x .33 x .0065 x .52 x .68 = .00019

TURKISH DICTIONARY N-GRAMS

The numbers are calculated from the definition sentences in the Contemporary Dictionary of the Turkish Language Association.

2-Gram (Bigram)	**olma durumu**	
	-Hayırsız olma durumu	*4359*
	-Uçarı olma durumu	
	bir biçimde	
	-Çekimsere yakışır bir biçimde	*1196*
	-Tedbirsiz bir biçimde, tedbirsiz olarak	
	yaptığı iş	
	-Telgrafçının yaptığı iş	*907*
	-Kapıcının yaptığı iş	

TURKISH DICTIONARY N-GRAMS

Note that the trigram numbers drop dramatically compared to the bigram.

3-Gram (Trigram)	**işine konu olmak**	*416*
	-Başlama işine konu olmak	
	-Aktarma işine konu olmak	
	Bu renkte olan	*214*
	-Bu renkte olan	

TURKISH DICTIONARY N-GRAMS

The numbers dropped from Bigram to Trigram. Why do you think there are such large numbers for 4-gram?

4-Gram	**ihtimali veya imkânı bulunmak**	*1788*
	-Yavaşlama ihtimali veya imkânı bulunmak	
	-Tutulma ihtimali veya imkânı bulunmak	
	iline bağlı ilçelerden biri	*1061*
	-Adana iline bağlı ilçelerden biri	
	-Ankara iline bağlı ilçelerden biri	
	yapan veya satan kimse	*202*
	-Tatlı yapan veya satan kimse	
	-Yoğurt yapan veya satan kimse	

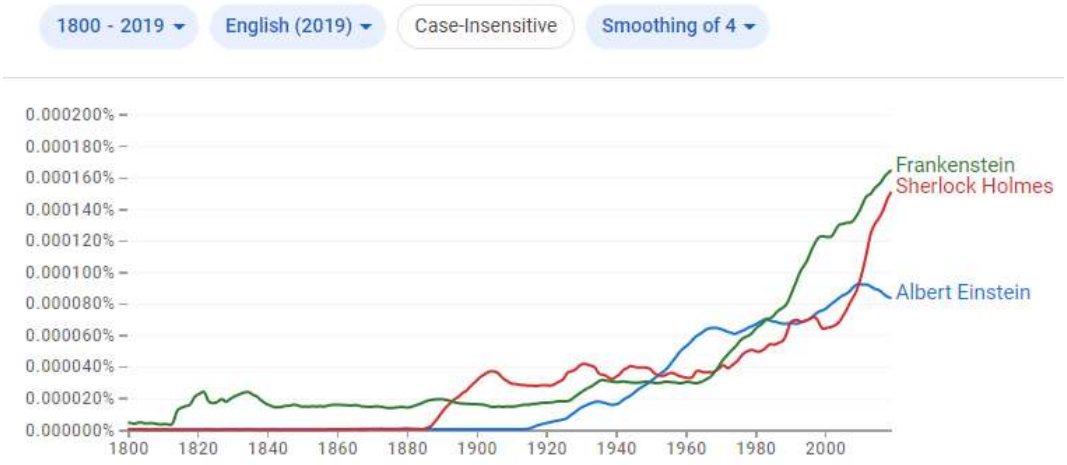
JUST GUESS

If we look appearance in English written books since 1800 for N-Grams: [‘Albert Einstein’, ‘Sherlock Holmes’, ‘Frankenstein’] what will be the graphic of these N-Grams?

- Sherlock Holmes first appears in 1887
- Frankenstein first published at 1818
- Albert Einstein published his paper about general relativity at 1916 and win Nobel Prize of Physics at 1921

https://books.google.com/ngrams

JUST GUESS



NATURAL LANGUAGE PROCESSING

LESSON 3: SYNTAX, PARSING, CONTEXT FREE LANGUAGE

OUTLINE

- Syntax
  - What is Syntax
  - Context-free grammar
  - Top Down Parsing

SYNTAX

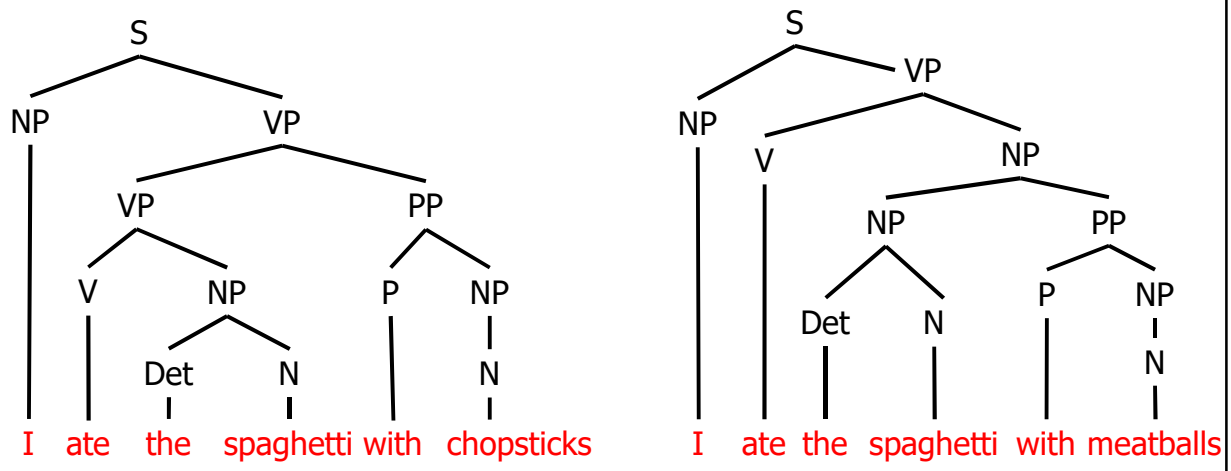
- Language is not a bag of words. It means, the order of the words is important. For this concept, we use Syntax term.
- In linguistics, syntax is the arrangement or order of words, determined by both the writer's style and grammar rules.
- Syntax can help people to guess unknown words by using its syntactic role. For example, you can guess the category of the unknown word in the following sentence.
  - Students will be really zealous for this class.
- Here, even if you don't know the meaning of "zealous", you can know it is an adjective

SYNTAX - WORD ORDER

Word order	English equivalent	Proportion of languages		Example languages
SOV	"She him loves."	45%		Proto-Indo-European, Sanskrit, Hindi, Ancient Greek, Latin, Japanese, Korean
SVO	"She loves him."	42%		English, French, Hausa, Indonesian, Malay, Mandarin, Russian
VSO	"Loves she him."	9%		Biblical Hebrew, Arabic, Irish, Filipino, Tuareg-Berber, Welsh
VOS	"Loves him she."	3%		Malagasy, Baure, Proto-Austronesian
OVS	"Him loves she."	1%		Apalaí, Hixkaryana
OSV	"Him she loves."	0%		Warao

Frequency distribution of word order in languages surveyed by Russell S. Tomlin in 1980s[10][11] (V · T · E)

SYNTACTIC PARSING



CONTEXT FREE GRAMMARS

In formal language theory, a context-free grammar is a formal grammar whose production rules are of the form

$$A \rightarrow \beta$$

where A is variable and β is terminal symbol.

Context-free grammars arise in linguistics where they are used to describe the structure of sentences and words in a natural language, and they were invented by the linguist Noam Chomsky.

CONTEXT FREE GRAMMARS

The task of the parsing is essentially to determine if and how the input can be derived from the start symbol of the grammar. This can be done in essentially two ways:

- Top-down parsing - Sentence is generated by recursively rewriting the variables from left to right until only terminal symbols remain.
- Bottom-up parsing - A parser starts to find the terminals in the sentence and by using the grammar, attempt to reach the start symbol.

CONTEXT FREE GRAMMAR

A SAMPLE GRAMMAR (Variables)

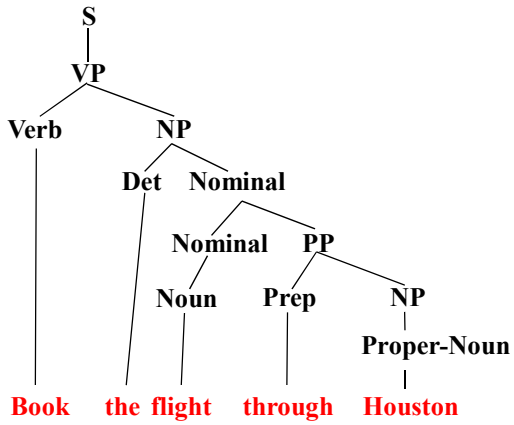
S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → VP PP
PP → Prep NP

A SAMPLE LEXICON (Terminals)

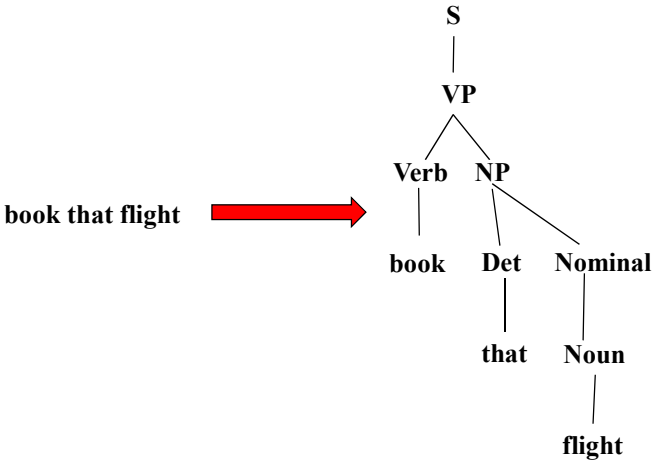
Det → the | a | that | this
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | he | she | me
Proper-Noun → Houston | NWA
Aux → does
Prep → from | to | on | near | through

Book the flight through Houston

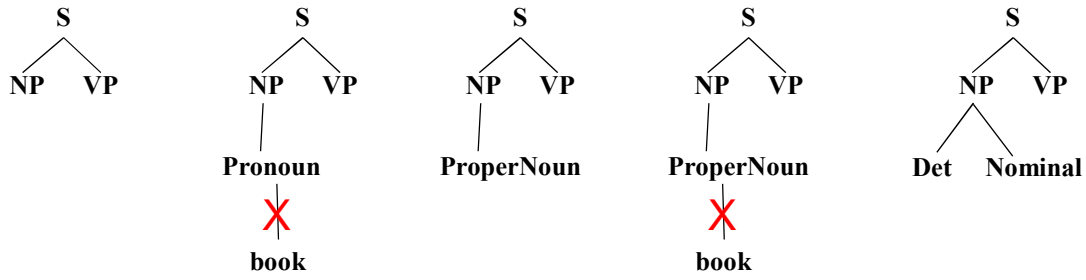
CONTEXT FREE GRAMMAR



TOP DOWN PARSING

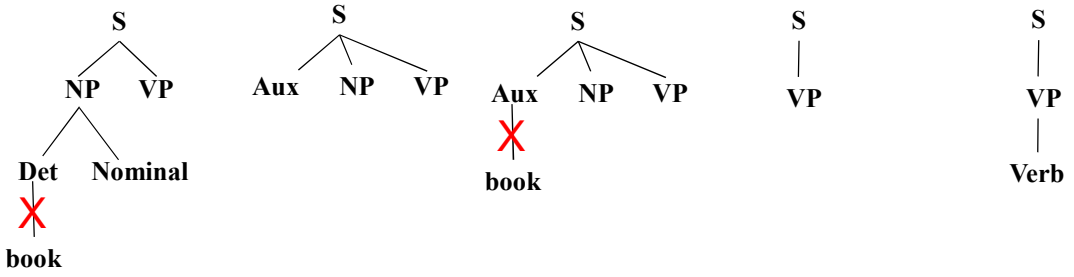


TOP DOWN PARSING



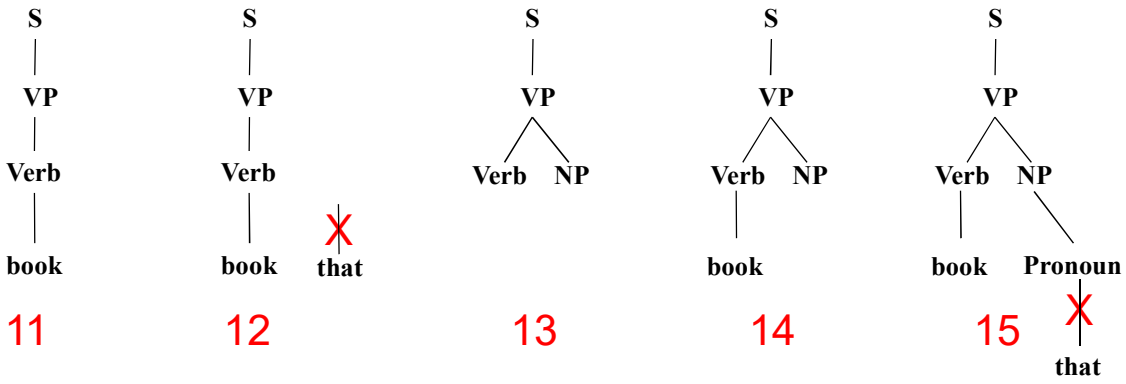
1 2 3 4 5

TOP DOWN PARSING

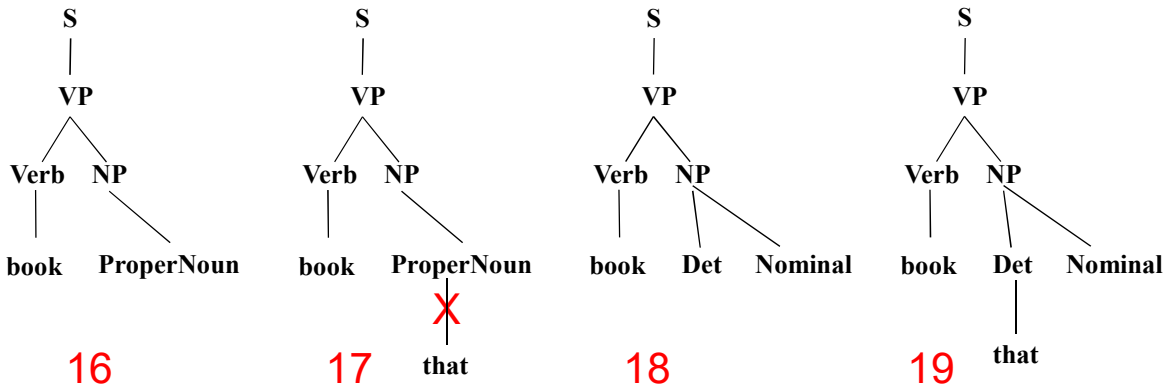


6 7 8 9 10

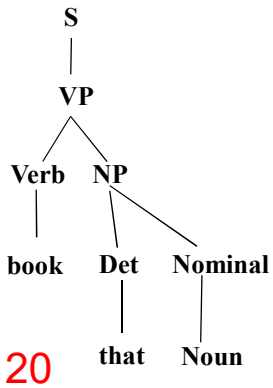
TOP DOWN PARSING



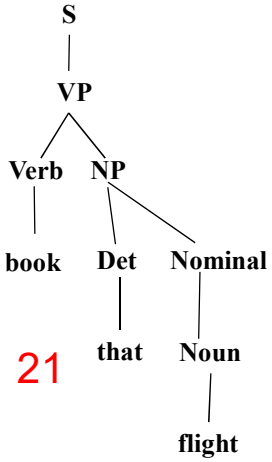
TOP DOWN PARSING



TOP DOWN PARSING

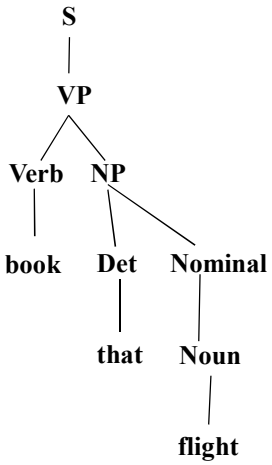


20



21

TOP DOWN PARSING



When you try to write this kind of a tree in a text box in a computer software, you can use infix parenthesis approach as follow:

S(VP(Verb(book)+NP(Det(that)+Nominal(Noun(flight)))))