NATURAL LANGUAGE PROCESSING

LESSON 2: TEXT NORMALIZATION, LEMMATIZATION, PARSING

OUTLINE

- Normalization
  - Tokenization
  - Morphology
- Removing Affixes
  - Stemming
  - Lemmatization
- Parsing
  - Parsing tree
  - Ambiguity

WHAT IS TEXT NORMALIZATION?

- Normalization is a process that transforms a list of words into a more uniform sequence. This is useful for preparing text for later processing. It reduces inflectional forms and sometimes derivationally related forms of a word to a common base form.
- Normalizing the text before storing or processing it allows separation of concerns as the input is guaranteed to be consistent before operations are performed on it.
- When we normalize text, we attempt to reduce its randomness, bringing it closer to a predefined standard. This helps us to reduce the amount of different information that the computer has to deal with, and therefore improves efficiency.

TEXT NORMALIZATION

Before starting the main operations on it, the raw text should be prepared by going through some processes. It is necessary to ensure that all texts go through the same cleaning and conversion processes. Thus, we can confidently begin the actual work we will do under NLP.

- All concerns such as "saving the characters as UTF8", "lefting Turkish characters in the text", "deleting punctuation" should be addressed.
- Similarly, auxiliary verbs used in English are considered unnecessary in many NLP processes and are therefore deleted in the normalization process.

TEXT NORMALIZATION

NLP tasks often need to do the following steps to normalize text:

- Tokenizing words in running text
- Standardization of word formats
- Segmenting sentences in running text

TOKENIZATION

There are two important concepts to know:

- **Lemma** is the dictionary form of any word in the raw text.

  kale (in Turkish) : castle (in English)

- **Token** is any word used in raw text. Tokens can be found in sentences in any form according to the affix structure of the language.

  kale (castle), kaleyi (the castle), kaleden (from the castle)

TOKEN – An English example

If we use a simple tokenizer on a English sentence

> Token is an individual occurrence of a symbol or string in speech or writing

All tokens of this sample sentence

> Token, is, an, individual, occurrence, of, a, symbol, or, string, in, speech, or, writing

TOKEN – A Turkish example (MWE)

If we use a multi word expression tokenizer on a Turkish sentence

> Türkiye Büyük Millet Meclisi ülkemizin parlamentosunun özel ismidir

Its tokens of this sample sentence are determined as follows

> Türkiye Büyük Millet Meclisi, ülkemizin, parlamentosunun, özel, ismidir

Look at the four words at the beginning of the sentence. They are captured as a multi word expression token.

TOKENS vs LEMMAS

T = tokens

|T| = the size of **T**

V = vocabulary (set of lemmas)

|V| = the size of **V**

How can we estimate the relationship, if any, between the size of T and the size of V?

TOKENS vs LEMMAS

In 1990, Church and Gale found that, in any text, the square root of the size of tokens is proportional to the size of vocabulary.

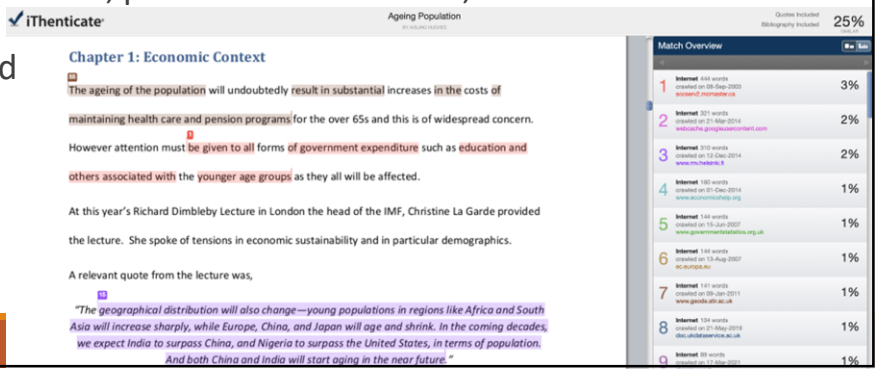
$$|V| > O(|T|^{½})$$

Here, |T| is the size of Tokens and |V| is the size of Vocabulary

	|T|	**|V|**
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884 000	31 thousand
Google N-grams	1 trillion	13 million

NORMALIZATION ISSUES

Let's say our goal is to measure lexical similarity by comparing two different texts. In this way, we can design a simple plagiarism (copying) detection system. In such a system, all characters in both texts are made lowercase, punctuation is deleted, inflectional and even derivational affixes are removed and comparison is made.



NORMALIZATION ISSUES

Öğrenciler'in defteri → Öğrenci? Öğrenciler? Öğrenciler'in?

Niçin → Ne için? Niçin?

Ural-Altay dil grubu → Ural-Altay? Ural Altay?

Birinci Dünya Savaşı → Simple token? MWE token?

However, it should be noted that there may be problems in word comparison due to some ambiguities. Therefore, all possible ambiguities should be solved.

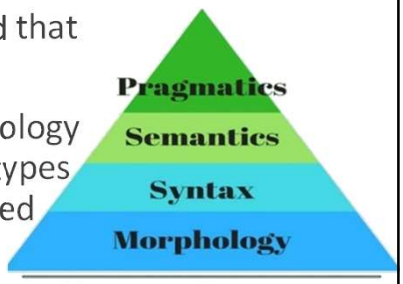
LEMMATIZATION

- **Lemmatization** aims to reduce inflections or variants to base form
  - *am, is, are → be*
  - *car, cars, car's, cars' → car*

    the boy's cars are different colors → the boy car be different color
- In Turkish, **Lemmatization** is a bit more difficult problem because of the agglutinative structure
- Before deep learning, it was used especially in machine translation
  - Turkish koşarız (we run), koşarsınız (you run) have the same lemma - koş (run)

MORPHOLOGY

- Morphology is a field of linguistics that studies the structure of words. It identifies how a word is produced through the use of morphemes.
- The morpheme is the smallest element of a word that has grammatical function and meaning.
- Inflectional Morphology and Derivational Morphology are the two types of morphology. Both of these types have their own significance in various areas related to the NLP.



NATURAL LANGUAGE PROCESSING PYRAMID

MORPHOLOGICAL ANALYSIS

- In inflected languages, words are formed through morphological processes such as affixation. For example, by adding the suffix '-s' to the verb 'to dance', we form the third person singular 'dances'.
- A morphological analyzer assigns the attributes of a given word by evaluating what morphological processes the form has undergone. If you give it the word 'zıplayacağım' in Turkish, it will tell you it is the first person, singular, simple future, indicative form of the verb "jump".

MORPHOLOGICAL PARSING

- It is the process of determining the morphemes from which a given word is constructed. Morphemes are the smallest meaningful words which cannot be divided further. Morphemes can be stem or affix. Stem are the root word whereas affix can be prefix, suffix or infix. For example,

  Unsuccessful → un success ful

  (prefix) (stem) (suffix)

- Order of words also decide the morphological parser. To design a morphological parser we require three things: lexicon, morphotactics and orthographic rules.

INFLECTIONAL MORPHOLOGY

- Inflectional morphology is the study of processes, including affixation and vowel change, that distinguish word forms in certain grammatical categories.
- The grammatical categories can be tense, case, voice, aspect, person, number, gender, mood, animacy, and definiteness.

DERIVATIONAL MORPHOLOGY

- Morphological derivation is the process of forming a new word from an existing word, often by adding a prefix or suffix. For example, "unhappy" and "happiness" derive from the stem "happy".
- It is differentiated from inflection, which is the modification of a word to form different grammatical categories without changing its core meaning: "determines", "determining", and "determined" are from the lemma "determine".

STEMMING vs. LEMMATIZATION

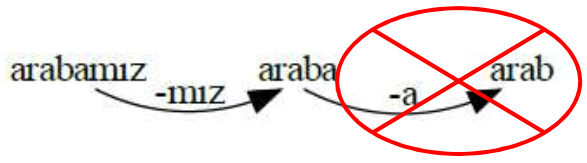
- The goal of both stemming and lemmatization is to reduce an inflectional form and sometimes a derivational form of a word to a common base form.
- **Stemming** usually refers to a heuristic process that chops off the ends of words in the hope of achieving the removal of derivational affixes.
- **Lemmatization** usually refers to reach the dictionary form of a word by removing its inflectional affixes.

STEMMING

- Stemming techniques aim to delete suffixes while reaching the root of the word, but sometimes delete suffixes incorrectly.

  e.g., ***boyama, boyalı, boyacı*** all reduced to ***boya***.

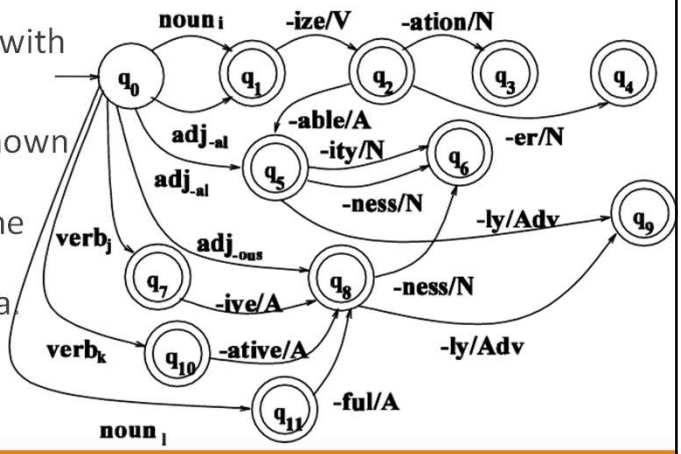
- But «arabamız» should not be resolved into the «arab»



LEMMATIZATION

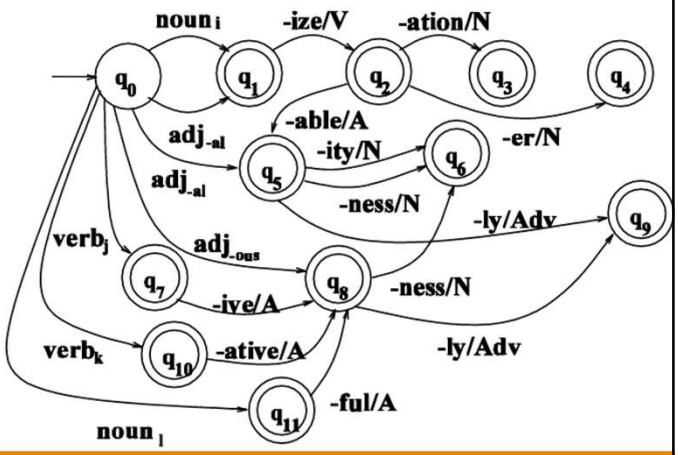
Similarly, lemmatization is also difficult, especially in languages with complex affixes such as Turkish.

The most successful methods known in the Turkish lemmatization literature are those that apply the morphosyntactic rules of the language in finite state automata.



LEMMATIZATION BY FSA

Suffix No	Suffix	Description for English language	Example
1	-lAr	Plural	gemi-ler
2	-(H)m	1st single person possessive	gemi-m
3	-(H)mHz	1st plural person possessive	gemi-miz
4	-(H)n	2nd single person possessive	gemi-n
5	-(H)nHz	2nd plural person possessive	gemi-niz
6	-(s)H	3rd single person possessive	gemi-si
7	-lArI	1st plural person possessive	gemi-leri
8	-(y)H	accusative case	gemi-yi
9	-nH	accusative case after possessive suffix added to lemma ends with vowel	gemi-si-ni
10	-(n)Hn	possessive	gemi-nin
11	-(y)A	dative case	gemi-ye
12	-nA	dative case after possessive suffix added to lemma ends with vowel	gemi-si-ne
13	-DA	preposition (in)	gemi-de
14	-nDA	preposition (in) possessive suffix added to lemma ends with vowel	gemi-si-nde
15	-DAn	preposition (from)	gemi-den
16	-nDAn	preposition (from) possessive suffix added to lemma ends with vowel	gemi-sin-den
17	-(y)lA	preposition (with)	gemi-yle
18	-ki	possessive	gemim-de-ki
19	-(n)cA	equative	gemim-ce

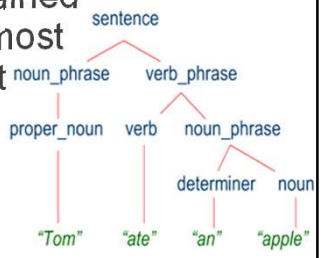


22

WHAT IS PARSING?

A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together and which words are the subject or object of a verb.

Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Their development was one of the biggest breakthroughs in natural language processing in the 1990s.



PARSING

S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP

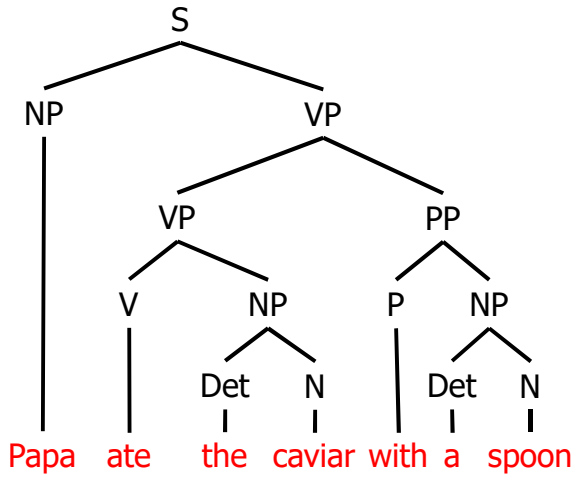
Let's say we have the following sentence in red color that is waiting to analyse.

Papa ate the caviar with a spoon

In order to analyze this sentence, we first need grammatical rules. Let these rules be given in the green window on the left.

PARSING

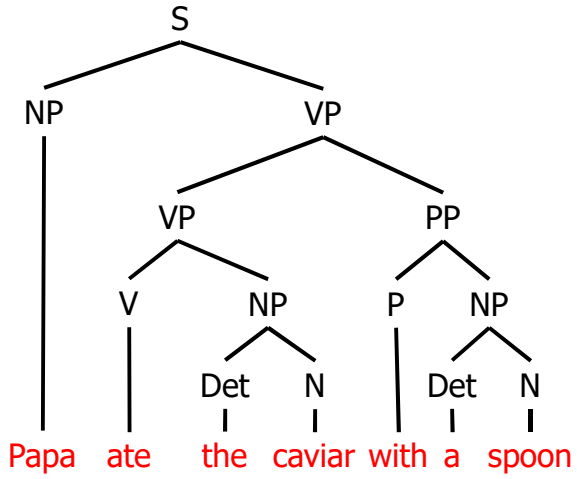
S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP



NP → Papa
N → caviar
N → spoon
V → spoon
V → ate
P → with
Det → the
Det → a

AMBIGUITY IN PARSING

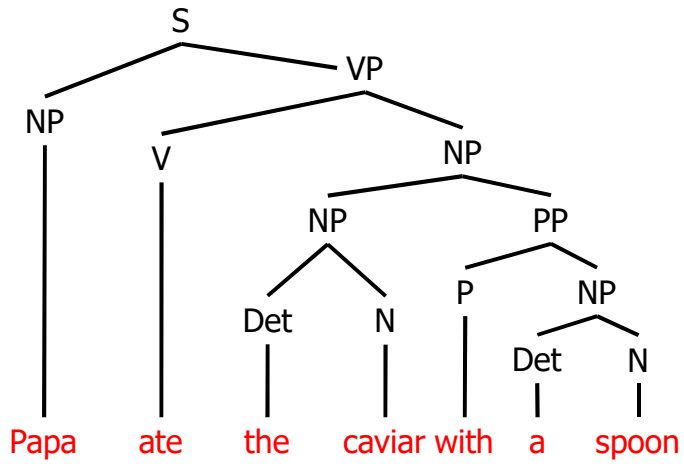
S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP



NP → Papa
N → caviar
N → spoon
V → spoon
V → ate
P → with
Det → the
Det → a

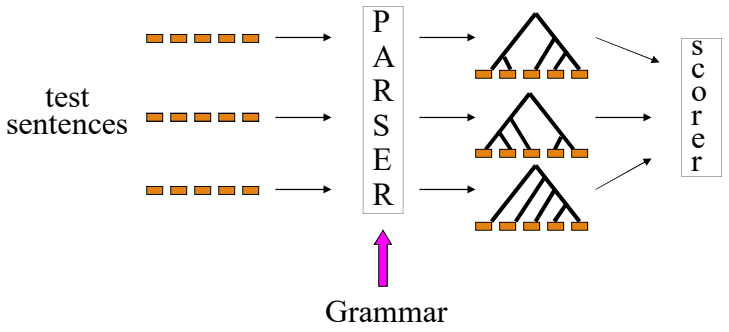
AMBIGUITY IN PARSING

S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP



NP → Papa
N → caviar
N → spoon
V → spoon
V → ate
P → with
Det → the
Det → a

THE PARSING PROBLEM



The source of success in such systems is that the one, which learns the cooccurrence, also learns the context.