# Lecture 2:
# Tokenization and Morphology

# Overview

1. Corpus
2. Word definition
3. Tokenization
4. Vocabulary

# General workflow of an NLP system

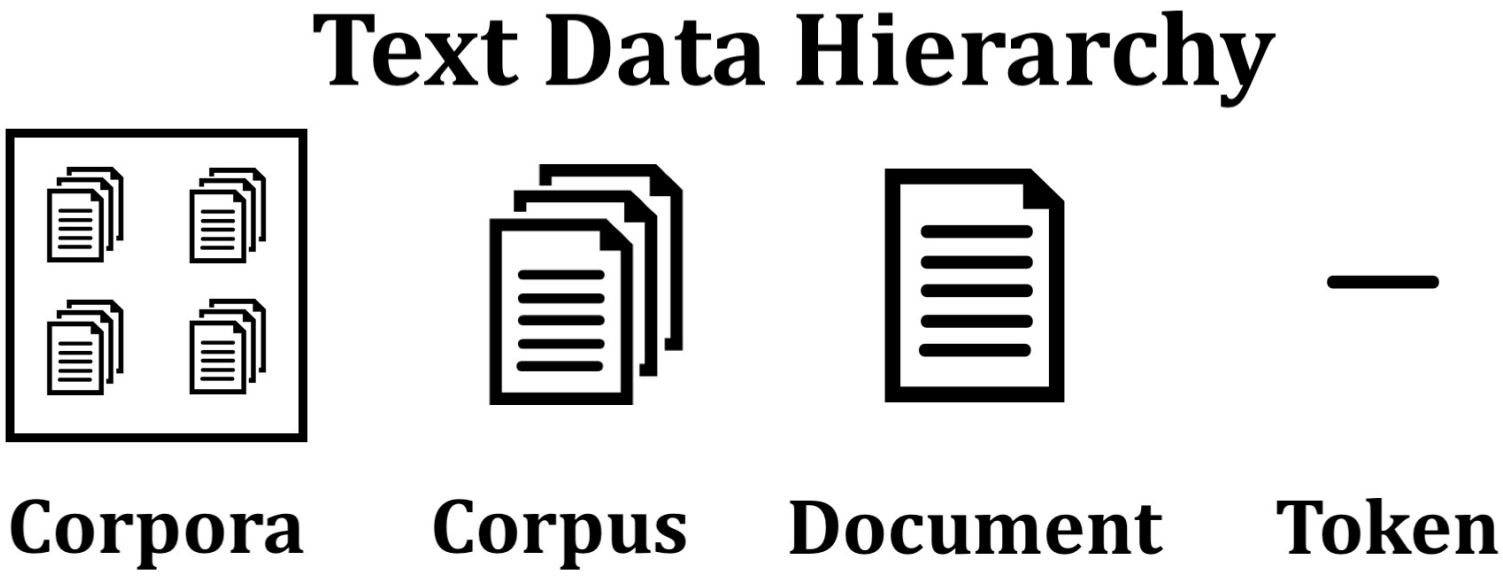
Process common to all if not most NLP task

**raw text corpus → tokenized text → corpus vocabulary → text representation → NLP task**

1. Raw text corpus: like data collection in any ML system
2. Tokenized: corpus reduced to unique tokens
3. Vocabulary: normalized tokens collecting of unique symbols of meaning
4. Text representation: encoding meaning into text
5. NLP task: machine translation, searching etc.

# Corpus in NLP

- An unbiased copy of text collected in a natural communicative setting for a specific purpose.
  - monolingual corpus: contain texts in a single language
  - multilingual corpus: text data in multiple languages



# Corpus design

- Though language is infinite, a corpus has to be finite in size.
  - should be a representative sample of the language under investigation.

- Corpus design considerations
  - Representativeness: findings from corpus can be generalized
  - Balance: cover a wide range of text categories
  - Sampling: related to representativeness ⇔ samples should cover variability in language
  - Corpus size: no scientific guidelines should be task dependent

# Examples of NLP corpora

- The Brown corpus (US English) :
  - a million words
- The Lancaster-Oslo-Bergen (LOB) corpus (British English)
  - about 1 million words
- The British National Corpus (BNC)
  - contains approx. 100 million words and includes 20 million words of spoken English
- The Bank of English corpus
  - 650 Million words.

# How do we identify words and sentences in a text?

***Word in English:*** any sequence of characters between whitespaces

- Text boundaries can be relative to..
  - Words, Phrase, Sentence, Paragraph
- Consider the following sentences

*Looking to get started with NLP? Here's the perfect first step.*

*[[Looking, to, get, started, with, NLP?], [Here, 's, the, perfect, first, step, .]]*

Easy to identify boundaries for languages like French and English

# How do we identify words in a text?

- This is good for first approximation but insufficient
  - What about compound words *New York-based, New York, ice scream*?
  - What about contractions like *doesn't, couldn't, 've, l'ensemble*?
    - Are these one or two words?

*Whitespaces* is not sufficient what about **Punctuations?**

- *doesn't* ⇔ *doesn, t*
- *l'ensemble* ⇔ *l?, L'? Le?*
- What about abbrevaitions like *D.C.*?
- What about names like *SARS-Cov-2, R2-D2*?

# How do we identify words in a text?

- What about languages without whitespaces (e.g. Chinese and Japanese)

**Chinese:** 我开始写小说 = **我** **开始** **写** 小说

*I* *start(ed)* *writing* *novel(s)*

- Further Japanese have multiple alphabets intermingled
- Languages where a single word can represent an entire sentence (e.g. Turkish)

# Finding boundaries

Finding word or sentence boundaries is not a trivial task.

- boundaries may interact with morphology
- and may include normalization
  - Normalization convert words to their base form (more on this later).

- Finding boundaries in text is referred to as tokenization or segmentation

# Finding boundaries

- How can we identify that this is two sentences?

  *Mr. Smith went to D.C. Ms. Johnson went to Chicago instead.*

  Challenge: punctuation marks in abbreviations (Mr., D.C, Ms,…)

- How many sentences are in this text?

  *"The San Francisco-based restaurant," they said, "doesn't charge $10".*

  - One sentence, even though "they said" appears in the middle of another sentence.

# Tokenization – identifying word boundaries

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into atomic units of meaning called ***tokens***.

The most common way of forming tokens is based on whitespace.

*What time is it?* ⇔ *[What, time, is, it?]*

- Main question?

*it = it. = it?*

- *Tokenization aims at handling word boundaries and should account for some variations in the text*

# What is a word?

Which words appear in this text?

- Most off the Actual text doesn't often consist of only dictionary entries
- Consider the sentence below

  *"Of course he wants to take the advanced course too. He already took two beginners' courses."*

  - *wants is a form of want*
  - *courses is a form of course*

# What is a word?

- Linguistic distinguish between two forms of words

    1. the **(surface) forms** that occur in text:
       *want, wants, beginners', took*

    2. the **lemmas** that are the uninflected forms of these words:
       *want, beginner, take*

- In NLP we often map words to lemmas (stems), but raw text consists of surface forms

# How many different words are there?

- Inflection creates different forms of the same word:
  - *Verbs: to be, being, I am, you are, he is, I was,*
  - *Nouns: one book, two books*
- Derivation creates different words from the same lemma:
  - *grace ⇒ disgrace ⇒ disgraceful ⇒ disgracefully*
- New words from compounding:
  - *cream ⇒ ice cream ⇒ ice cream cone ⇒ ice cream cone bakery*
- Word formation is productive:
  - New words are subject to all of these processes:
  - *Google ⇒ Googler, to google*

# Inflectional morphology

- Verbs:
  - Infinitive/present tense: walk, go
  - 3rd person singular present tense (s-form): walks, goes
  - Simple past: walked, went
  - Past participle (ed-form): walked, gone
  - Present participle (ing-form): walking, going
- Nouns:
  - Common nouns inflect for number:
    - singular (book) vs. plural (books)
  - Personal pronouns inflect for person, number, gender, case:

    I saw him; he saw me; you saw her; we saw them; they saw us.

# Derivational morphology

- Nominalization:
  - V + -ation: computerization
  - V+ -er: singer
- Negation:
  - un-: **unkind**, undo, unseen,...
  - mis-: mistake,...
- Adjectivization:
  - V+ -able: doable
  - N + -al: national

# Morphemes: stems, affixes

***dis-grace-ful-ly***

**prefix-stem-suffix-suffix**

- word forms consist of a ***stem*** + a number of *affixes* (*prefixes* or *suffixes*)
  - Exceptions: *Infixes* are inserted inside the stem
  - *Circumfixes* surround the stem

- Morphemes: the smallest (meaningful/grammatical) parts of words.
  - *Stems* are often free morphemes can occur by themselves as words.
  - *Affixes* (dis-, -ful, -ly) are usually bound morphemes.
    - Bound morphemes *have* to combine with others to form words.

# Morphemes and morphs

- The same information (plural, past tense, ...) is often expressed in different ways in the same language.
  - Some representations may be more common than others,  and exceptions may depend on specific words:
  - Most plural nouns: add ***-s*** to singular: book-book**s**
  - However others have: box-box**es**, fly-fl**ies**, child-child**ren**
  - Most past tense verbs add ***-ed*** to infinitive: walk-walk**ed**,
  - However others have: like-like**d**, leap-leap**t**
  - Such exceptions are called *irregular* word forms

# How do we handle spelling variants and typos?

The same word can be written in different ways (spelling variants, typos):

- with different **capitalizations**: ***fox, Fox, FOX***
- with different **abbreviation** or **hyphenation** styles:

  ***US-based, US based, U.S.-based, U.S. based***

  ***US-EU relations, U.S./E.U. relations, …***

- with **spelling variants** (e.g., regional variants of English):

  ***labor*** vs ***labour***

  ***neighbor vs neighbour***

- with **typos:** ***teh*** instead of ***the***

# How do we represent the structure of words?

**Words as atomic symbols**

- each word form its own symbol
- add generalization by mapping different forms of a word to the same atomic symbol
  1. **Normalization:** map all variants of the same word (form) to the same canonical variant (e.g. lowercase everything, normalize spellings, perhaps spell-check)
     - ***US-based, US based, U.S.-based, U.S. based*** ⇔ ***us-based***
     - ***labor , labour*** ⇔ ***labour***

# How do we represent the structure of words?

**2. Lemmatization**: maps each word to its lemma
a lemma maybe a word, (lemmatized text is no longer grammatical).

- Reduce inflections or variant forms to base form
  - ***am, are, is*** ⇔ ***be***
  - ***Car, cars, car's, cars'*** ⇔ ***car***

***The boy's cars are different colours*** ⇔ ***the boy car are be different colour***

- Lemmatization finds correct dictionary headwords but the resulting sentence must not necessarily be grammatically correct.

# How do we represent the structure of words?

**3. Stemming**: remove endings that differ among word forms (no guarantee that the resulting symbol is an actual word)

- Reduces words/terms to their stem i.e. crude chopping of affixes
- ***Automates, automatic, automation*** ⇔ ***automat***

*for example compressed and compression are both accepted as equivalent to compress*

*for exampl compress and compress are both accept as equival to compress*

# How do we represent the structure of a words?

**Represent the structure of each word**

- "books" => "book N pl" or "book V 3rd sg"

- This representation will require a morphological analyser

- The output of such representation is often
  - a lemma ("book") plus
  - morphological information ("N pl" i.e. plural noun)

# Handling unknown words

Many NLP systems assume a fixed vocabulary, but still have to handle **out-of-vocabulary (OOV)** words.

**1. The unknown – UNK token**

- Replace all rare words in your training vocabulary with an UNK token (UNK = "Unknown word").
  - Rare words are words with a frequency below a given threshold, e.g. 2, 3, or 5)

- Replace *all* unknown words during testing or in production (including rare training words) with the same UNK token

# Vocabulary

- The vocabulary is a holding area for processed text before it is transformed into some representation for the impending NLP task.

- The vocabulary of a language is the set of (unique) word types in the language corpus:

$$V = \{a, able \ldots., zebra\}$$

  - The tokens in a document include all occurrences of the word types
  - The frequency of a word (type) in a document is the number of occurrences (tokens) of that type.

# Vocabulary – how many words?

Consider the following sentence

*“they lay back on the San Francisco grass and looked at the stars and their”*

⇔

*[they, lay, back, on, the, San Francisco, grass, and, looked, at, their, stars]*

**Type:** and element of the vocabulary

- *How many types (13, 12, or 11?)*

**Token:** an instance of a type in a running text

- *How many tokens (15, 14 or 13)*

# Vocabulary – counting words

- How large is the vocabulary of the English language:

  *Vocabulary size = number of distinct word types*

  - Google n-gram corpus: 1 trillion tokens, 13M word types that appear 40+ times

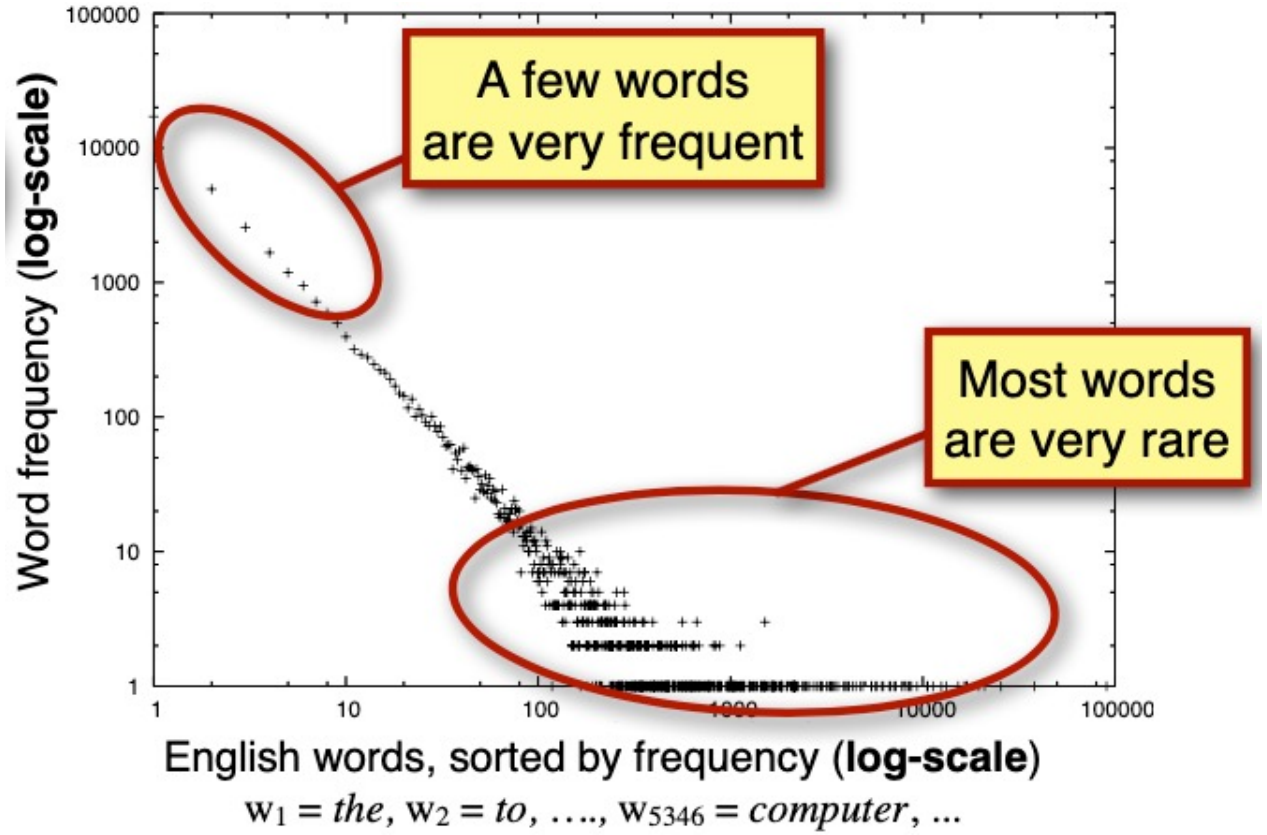
- For most corpus of text in the English language
  - close class words are very frequent  (the, be, to, of, and, a, in, that,...)
  - all open class words are very rare

**Word frequency**: the number of occurrences of a word type in a text (or in a collection of texts)

  - Biased towards open class words

# Vocabulary – Zipf's Law

Zipf's Law is a discrete probability distribution that tells you the probability of encountering a word in a given corpus.



# Vocabulary – Zipf's Law

The input is the **rank** of a word (in terms of is frequency) so you can use this distribution to ask questions like:

- What is the probability of encountering the most common word in a corpus with 100,000 words?
- What is the probability of encountering the 10th most common word in a corpus of 100,000 words?
- What is the probability of encountering the least common word in a corpus of 100,000 words?

# Vocabulary – Zipf's Law

Probability mass function in Zipf's law is defined as:

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}$$

where

- *K*: rank of the word we are interested in finding out the probability of appearing
- *N*: size of the vocabulary
- *s*: parameter of the probability distribution and is set to 1 in classic Zipf's law

# Implication of Zipf's Law

- Useful in identifying the very common words in a vocabulary
  - These are words seen very often that it's easily understood
  - These are words that help humans understand the structure and meaning of the text.
- Bias towards rare words
  - Words we haven't seen enough to easily know everything about them
  - They may occur with a meaning or an unseen part of speech
- Bias towards unknown words
  - Any text will contain a number of words that are **unknown** to us. We have *never* seen them before, but we still need to get at the structure (and meaning) of these texts.

# Dealing with biases in Zipf's Law

Every NLP systems need to be able to generalize to unseen samples.

There are two (complementary) approaches to generalization

- **Linguistics** provides us with insights about the rules and structures in language that we can exploit in the (symbolic) representations we use
  - e.g.: a finite set of grammar rules is enough to describe an infinite language

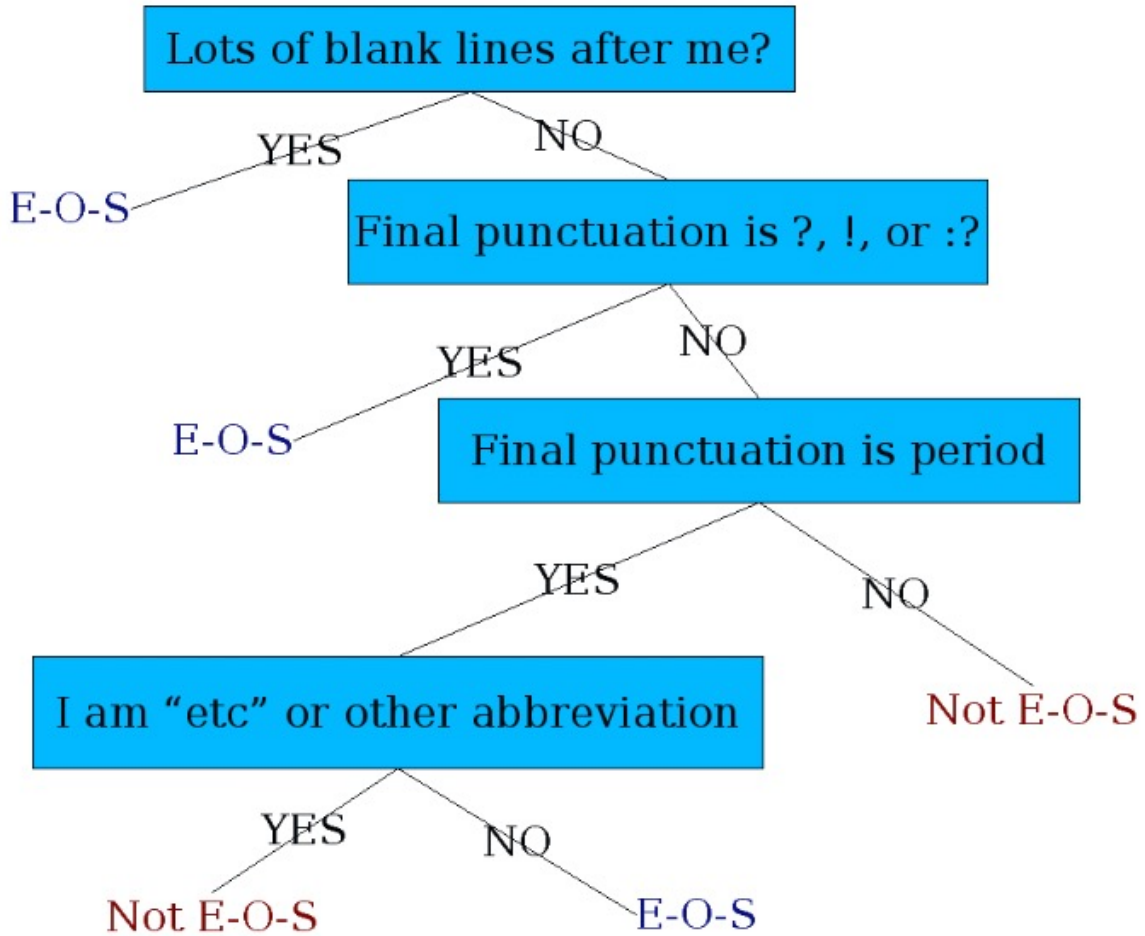
- **Machine Learning/Statistics** allows us to learn models from real data that often work well empirically on unseen data
  - e.g. most statistical or neural NLP

# Sentence segmentation

- !, ? are relatively unambiguous
- Period "." is quite ambiguous
  - Sentence boundary
  - Abbreviations like Inc. or Dr.
  - Numbers like .02% or 4.3

- Build a binary classifier
  - Looks at a "."
  - Decides EndOfSentence/NotEndOfSentence
  - Classifiers: hand-written rules, regular expressions, machine-learning

# Determining end-of-sentence – Decision Tree



# More sophisticated decision tree features

- Case of word with “.”: Upper, Lower, Cap, Number
- Case of word after “.”: Upper, Lower, Cap, Number

- Numeric features
  - Length of word with “.”
  - Probability (word with “.” occurs at end-of-s)
  - Probability(word after “.” occurs at beginning-of-s)