Lecture 2

Tokenization and Morphology

LECTURE 2: OVERVIEW

Focus of today lecture is on words and meaning

- What is a words
- What is the structure of a word
- How can we identify the structure of words?
- Identifying word boundaries

Key concepts

- Word forms vs lemmas
- Word tokens vs word types
- derivational morphology

CORPUS IN NLP

An unbiased copy of text collected in a natural communicative setting for a specific purpose.

- monolingual corpus
- multilingual corpus

Quality and quantity of the corpus greatly affects the quality of the NLP system.

Automatically constructed from existing resources

- Wikipedia, patent and legal text, …..

Manually constructed for domains which don’t have enough data.

CORPUS DESIGN

Should be a representative sample of the language under investigation.

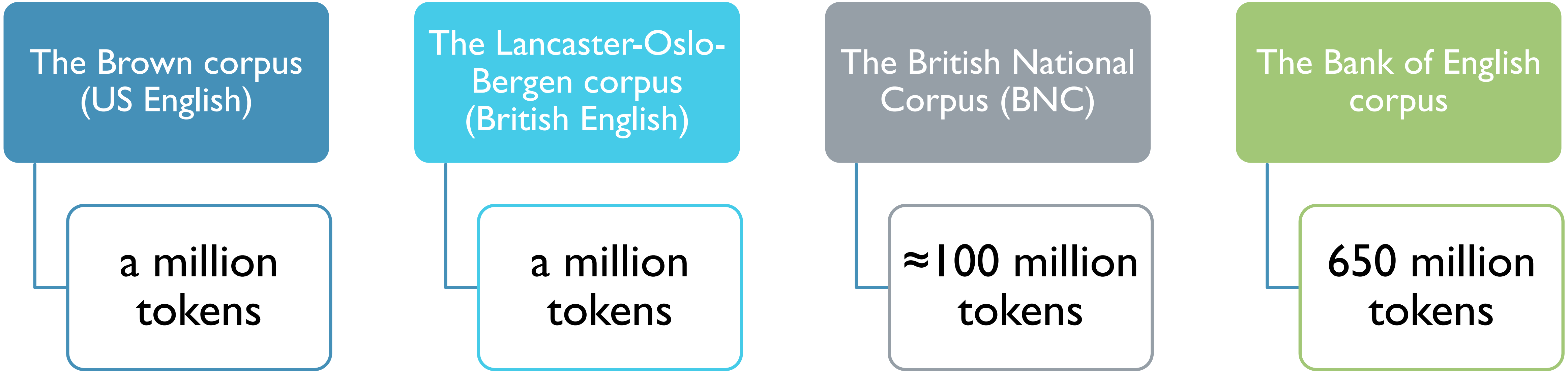
Representativeness: findings from corpus can be generalized.

Balance: cover a wide range of text categories.

Sampling: samples should cover variability in the language or text.

Corpus size: no scientific guidelines should be task dependent.

EXAMPLES OF NLP CORPORA



The Brown corpus (US English)

a million tokens

The Lancaster-Oslo-Bergen corpus (British English)

a million tokens

The British National Corpus (BNC)

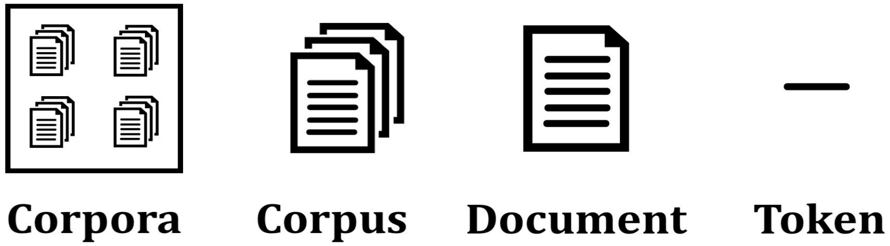
≈100 million tokens

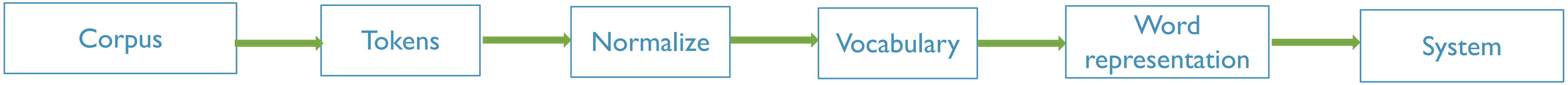
The Bank of English corpus

650 million tokens

GENERAL WORKFLOW OF AN NLP SYSTEM

Text Data Hierarchy





TOKENIZATION

The process of identifying boundaries in text.

Text boundaries are relative to:

- word, sentence boundaries, paragraph etc

Text is just a sequence of characters

The lecture today is about text boundaries. We will learn how to split text for NLP.

How do we split this text into words and sentences

[[The, lecture, today, is, about, text, boundaries, .],
[We , will, learn, how, to, split, text, for, NLP, .]]

TOKENIZATION – IDENTIFYING BOUNDARIES

- Tokenization - splitting a phrase, sentence, paragraph, or an entire text document into atomic units of meaning called tokens.
  - tokens can be words, phrases, sentence, paragraphs etc…

- Example: using whitespaces, we can form the following tokens from the sentence.

What time is it? <=> [What, time, is, it?] <=> [What, time, is, it, ?]

- Main question is how to handle it, should it be stored as it, it. or it?

HOW DO WE IDENTIFY BOUNDARIES IN TEXT?

This is good for first approximation but insufficient

- What about compound words:

  New York-based ⇔ [New York, based] or [New, York-based]
  ice scream or ice-scream ⇔ [ice, scream]
  website, web site ⇔ [web, site]

What about **Punctuations?**

- What about contractions like *couldn't, 've, he's?*

  he's ⇔ [he, s] [he, is], [he, has]

- English and French seems easy

  Italian: "dirglielo" = *dir + gli(e) + lo*
  tell + him + it

HOW DO WE IDENTIFY BOUNDARIES IN A TEXT?

- 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
- In Turkish there are single word that represent an entire sentence

nasilsin ⬄ *how are you?*

HOW DO WE IDENTIFY BOUNDARIES IN A TEXT?

How can we identify sentence boundaries

“Mr. and Mrs. Smith booked a hotel in D.C.”

Challenge: *punctuation marks in abbreviation (Mr., Mrs, D.C.), it is much harder to identify these cases*

How many sentences are in this text

They said: ”the San Francisco-based restaurant doesn’t charge $10”.

Answer: *just one sentence, even though there is a sentence separation after, they said*

WHAT IS A WORD?

Two words form exist

- **surface forms** that occur in text; *books, wants, beginners*.
- **lemmas** that are the uninflected or stem forms of words; *book, beginner, take*.

Inflection morphology creates different forms of the same word

Verbs

- Infinitive/present tense: walk, go
- 3rd person singular present tense (s-form): walks, goes
- Simple past: walked, went
- Present participle (ing-form): walking, going

Nouns

- Inflect for number: *book (singular) vs. books(plural)*
- Inflect for person, number gender; *I saw him; he saw me; you saw her; we saw them; they saw us*.

WHAT IS A WORD? - DERIVATIONAL MORPHOLOGY

Derivation creates different words from the same lemma:

- Nominalization:
  - *V + -ation: computer ⇔ computerization*
  - *V+ -er: sing ⇔ singer*
- Negation:
  - *un-: kind ⇔ unkind, do ⇔ undo*
  - *mis-: mistake, misplaced*
- Adjectivization:
  - *V+ -able: doable*
  - *N + -al: national*

WHAT IS A WORD - WORD FORMS

Words as atomic symbols

- each word form its own symbol
- add generalization by mapping different forms of a word to the same atomic symbol

Different words forms consist of a stem + affixes (prefixes or suffixes)

dis-grace-ful-ly
prefix-stem-suffix-suffix

HOW DO WE REPRESENT WORDS?

1. Normalization: map all variants of the same word (form) to the same canonical variant

- 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 WORDS?

2. Stemming:

- remove endings that differ among word forms
- no guarantee that the resulting symbol is an actual word)

Reduces words/terms to their stem (crude chopping of affixes)

Automates, automatic, automation ⇔ automat

- Examples:

Original: *for example, compressed and compression are both accepted as equivalent to compress.*

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

HOW DO WE REPRESENT WORDS?

3. Lemmatization:

- reduce inflections or variant forms to base form
- A lemma maybe a word, (lemmatized text is no longer grammatical).

am, are, is ⇔ *be*

Car, cars, car's, cars' ⇔ *car*

- Lemmatization finds correct dictionary headwords
- resulting sentence may not be be grammatically correct.

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

HOW DO WE REPRESENT WORDS?

4. Represent structure of each word

- Considers things like part of speech
- requires a morphological analyser (more on this later)

“books” => “book N pl” or “book V 3rd sg”

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

HOW DO WE REPRESENT WORDS?

Corpus: *The boy’s cars are different colours. The boy car.*

Normalize: *the boy car be different colour, the boy car*

unique words = {the, boy, car, be, different, colour}

Word frequency

number of tokens: 9

number of unique tokens (types): 6

VOCABULARY

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

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

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

V = {a, aardvark, …, zyzzva}

COUNTING WORDS

- How large is the vocabulary of the English language:

  Vocabulary size = number of distinct word types (forms)

  Google N-gram corpus: 1 trillion tokens, 13-million-word types that appear 40+

- For most corpus of text in the English language
  - close class words are very frequent (*the, be, to, of, and, a, in, that,...)*
    - *Referred to as stop words and often discarded*
  - most words (all open class) are very rare
  - Biased towards open class words

ZIPF'S LAW

- Zipf’s Law is a discrete probability distribution that tells you the probability of encountering a word in a given corpus.
- Captures the relationship between the probability of a word occurring in a corpus and its rank

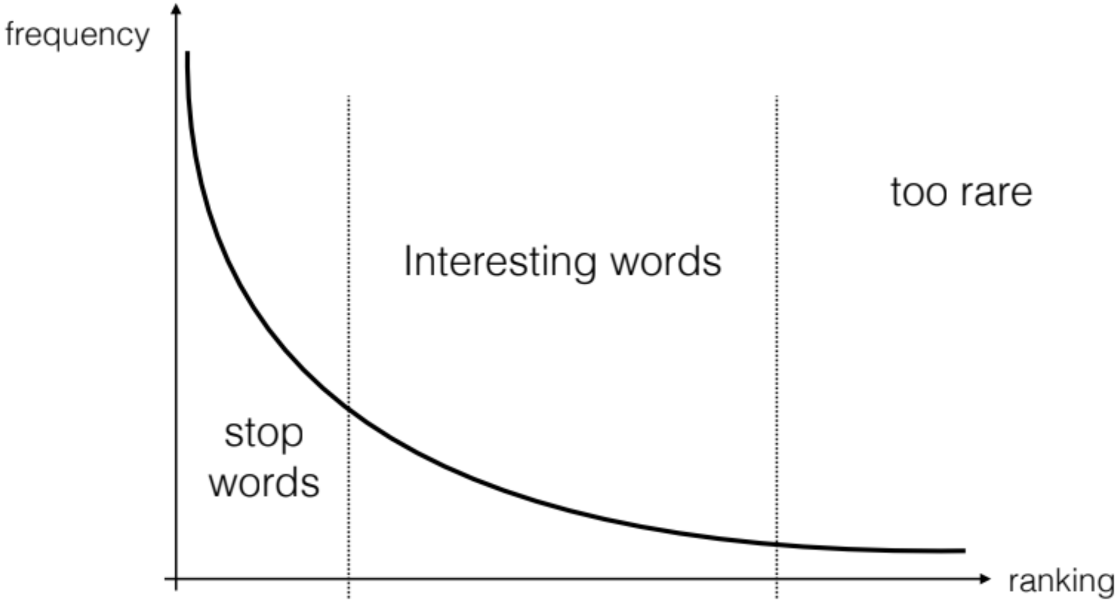
The probability mass function in Zipf’s Law is defined as:

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

- *k is the rank of the word we are interested in finding out the probability of appearing*
- *N is vocabulary size*
- *s is a parameter of the probability distribution in the classic of Zipf’s Law s = 1.*

ZIPF'S LAW: THE LONG TAIL

We can use Zipf's law to determine the probability of a word in a corpus, $\mathrm{r^{th}}$ most common word $\mathrm{w_r}$ has a probability $\mathrm{P(w_r)}\ \alpha\ \frac{1}{\mathrm{r}}$



https://www.ngcm.soton.ac.uk/2021/07/07/natural-language-processing-in-python/

IMPLICATIONS OF ZIPF'S LAW FOR NLP

Common words: Any text will contain several words that are very **common.** These are words that we have seen enough that we know (almost) everything about them. These words will help us get at the structure (and possibly meaning) of this text.

Rare words: Any text will contain several words that are **rare**. We know something about these words but haven't seen them often enough to know everything about them. They may occur with a meaning or a part of speech we haven't seen before.

Unknown words: Any text will contain 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.

IMPLICATIONS OF ZIPF'S LAW FOR NLP

Every system needs to be able to generalize from what they have seen to unseen events .

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 (and/or representations) from real data that often work well empirically on unseen data

e.g.: most statistical or neural NLP

SENTENCE SEGMENTATION

The staff was great. The receptionists were very helpful and answered all our questions. The room was clean and bright, and the room service was always on time. Will be coming back! Will I recommend N.Y. Hotel? Definitely!

Finding sentence boundaries that include fractions like .02 or 4.3 are difficult. It becomes more complicated when the last word is an abbreviation like Dr. or D.C.

!, ? are relatively unambiguous

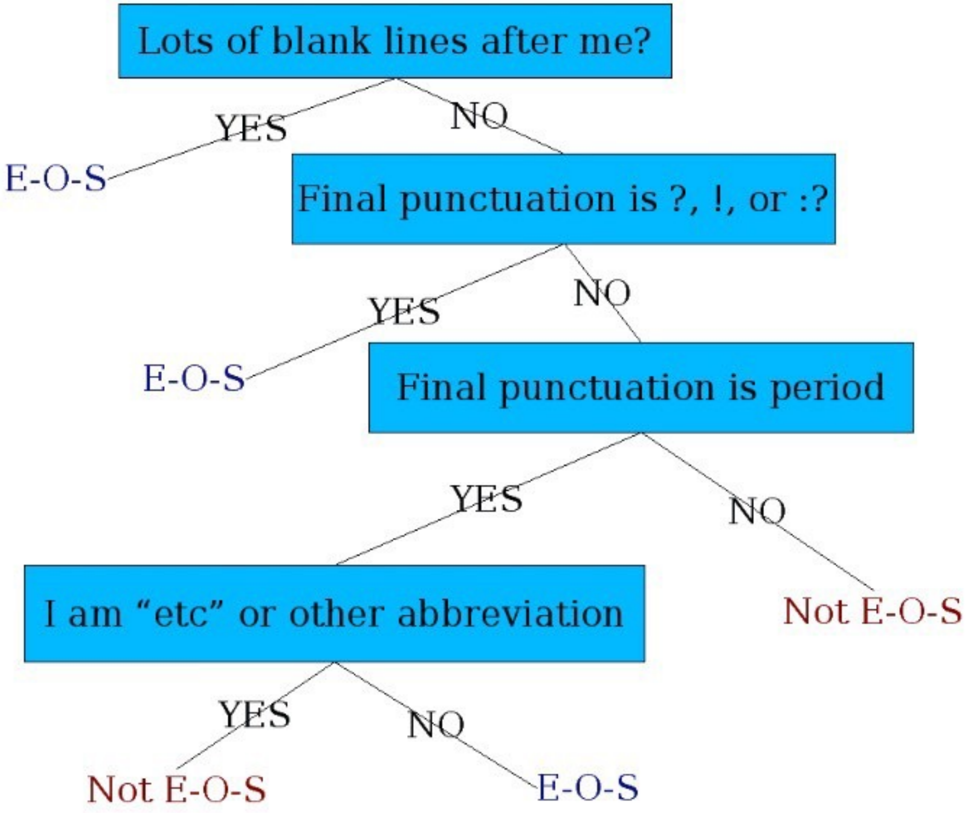
Period "." is quite ambiguous

Sentence boundary

Abbreviations like N.Y.

Numbers like .02% or 4.3

DETERMINING END-OF-SENTENCE – DECISION TREE


Lots of blank lines after me?
YES
NO
E-O-S
Final punctuation is ?, !, or :?
YES
NO
E-O-S
Final punctuation is period
YES
NO
I am “etc” or other abbreviation
Not E-O-S
YES
NO
Not E-O-S
E-O-S