TOKENIZATION

Tokenization → part of text normalization

1. Segmenting/tokenizing words
2. Normalizing words format
3. Segmenting sentences

filled pause

! do um main mainly business data processing → fragments

} how many words?

Titch's cat in the hat is different from other cats

} Cat vs Cats

Words

* Type : an element of the vocabulary
* Token : an instance of that type

* lemma : same stem, part of speech, rough word sense
  ↳ cat & cats is same lemma
* wordform : full inflected surface form
  ↳ cat & cats is different wordforms

Dan Jurafsky

How many words?

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

- **Type**: an element of the vocabulary.
- **Token**: an instance of that type in running text.

token → 15 tokens (or 14) "San", "Francisco" or "San Francisco"

Type → 13 types (or 12) "The" "They" "their" (or 11) "San Francisco"

In general,

N = number of tokens

V = Vocabulary = set of types,

with |V| is the size of

There are some issues in tokenization:

- use of ', "what're", "finland's", "isn't"
- use of -, "Hewlett-Packard",

the vocabulary

"state-of-the-art"

- "San Francisco", 1 or 2 tokens?
- m.p.h., PhD.
- language issues
  (german, french, japanese chinese, etc.)

NORMALIZATION

"Normalize"

* IR → indexed text & query terms must have same form
* asymmetric expansion
* reduce all letters to lowercase
  ↳ sometimes "Case" is helpful: sentiment analysis, MT, IE
  (US and us is important)

Lemmatize → reduce inflection/variant forms to base form

* am, are, is ⇒ be
* car, cars, car's, cars' ⇒ car

→ finding correct dictionary headword form

Looking at parts of words → morphology

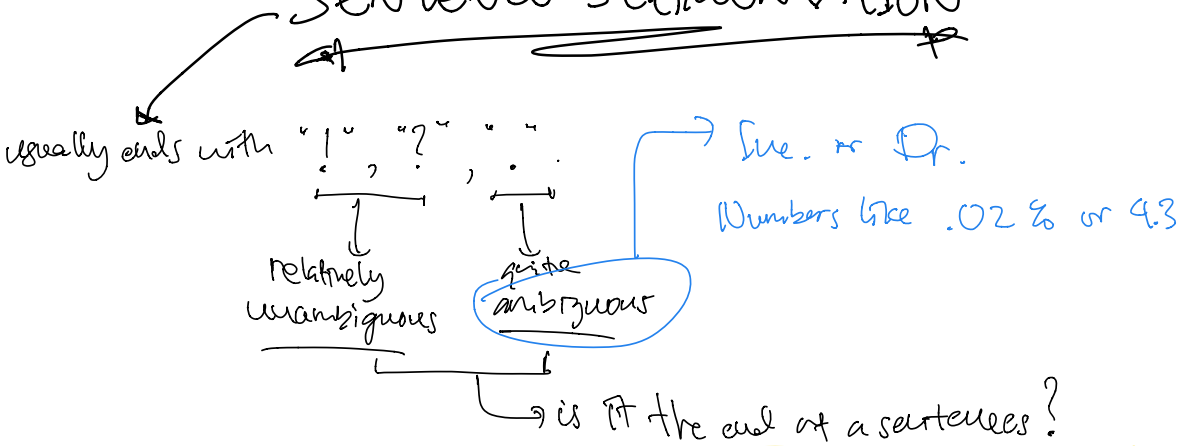
morphemes

↳ small meaning full units that make up words

↳ 2 kinds of morphemes:

- stems: core meaning-bearing units
- affixes: pieces that adhere to stems (often with grammatical function)

SENTENCE SEGMENTATION



How to detect that? → build a classifier that decides whether it's end of sentence / not end of sentence.

hand-written rules / regex / ML