Lecture 2:
Statistics of Natural Language

First-Year Project 4:
Natural Language Processing

Christian Hardmeier

4 May 2021

Talking about Language

Corpus

Corpus, n. – Plural: corpora

korpus substantiv, intetkøn

BØJNING -set eller (uofficielt) -et, -ser eller (uofficielt) -er, -serne eller (uofficielt) -erne

2. SPROG (elektronisk) samling af tekster der bruges til sproglige eller litterære undersøgelser

ORD I NÆRHEDEN database | tekstkorpus...vis mere

GRAMMATIK faglig, men uofficiel pl.-form: *korpora*

German: das Korpus – die Korpora
Swedish: en korpus – flera korpusar

- The *collection of texts under analysis*.
- Anything from one document to immense collections.

Layers of Linguistic Analysis

- Phonetics/Phonology – Orthography
- Morphology
- Syntax
- Semantics
- Discourse
- Pragmatics

Phonetics/Phonology – Orthography

- *Phonetics*: What sounds are there,
  and how are they produced?
- *Phonology*: How do languages use and classify sounds?
- *Orthography*: How is language written?

NLP fields:

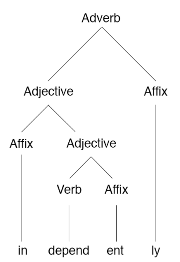
- Automatic Speech Recognition (ASR)
- Spellchecking
- Language identification

Morphology

- How are words formed, and what patterns do they follow?
- *Inflectional morphology*: Systematic patterns for certain word classes (Verbs: tense, person, mood, etc.;
  Nouns: number, case, etc.)
- *Derivational morphology*: Creating derived words
  - *Derivation*: displace → displacement
  - *Compounding*: snow + boots → snow boots

NLP fields:

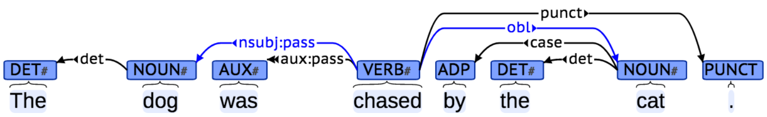
- Morphological analysis
- Lemmatising



https://commons.wikimedia.org/w/index.php?curid=58318220

Syntax

- How do words combine into sentences?
- Word order, grammatical agreement



NLP fields:

- Syntactic parsing
- Chunking

Semantics

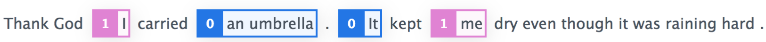
- What is the *meaning* of words?
- How do words combine to create meaning at a higher level?

NLP fields:

- Semantic role labelling
- Word sense disambiguation
- Textual entailment

Discourse

- Relations at the *text* level
- Cohesion/coherence: What ties texts together?
- Reference: What do linguistic expressions refer to?
- What is the relation between neighbouring sentences, paragraphs, etc.?



NLP fields:

- Coreference resolution
- Argument mining
- Discourse relation detection

Pragmatics

- Understanding speaker's *intentions*
- Language use in context

NLP fields:

- Dialogue systems
- All TweetEval tasks:
  Sentiment, stance, hate speech, etc.

What makes language processing difficult?

IT UNIVERSITY OF COPENHAGEN

Ambiguity

Lexical ambiguity: **Bank**

https://commons.wikimedia.org/wiki/File:European_Central_Bank_041107.jpg
https://commons.wikimedia.org/wiki/File:River_Erosion_-_geograph.org.uk_-_358650.jpg

IT UNIVERSITY OF COPENHAGEN

Ambiguity

Structural ambiguity

I saw the man in the park with the telescope.

- Who has the telescope?

Referential ambiguity

Anna has a little sister. She loves her very much.

- Who loves whom?

Vagueness

- Language is often *vague* or *underspecified*,
  and it would be unnatural to be totally precise in every situation.
- What is bigger?
  - A large dog,
  - or a small elephant?
- At what time does the afternoon end and the evening start?

Variation

- Languages
  - (even in one text: *code switching*)
- Register
  - Formal vs. informal
  - Written vs. spoken
- Domain
  - What *is* the text about?

World knowledge

- Humans use *world knowledge* to interpret language.
- Cues from *context* of language use:
  - Textual context
  - Audiovisual context
  - Situational context
  - Cultural context

World knowledge

– Well, what? He's not happy?
– He can't be, can he, if he's, you know, messing around.
– You gonna see him again?
– Do you think I should?
– Want me to be honest?
– No. No.
– When you gonna take that thing off?
– It's too tight. I've got to get it cut off.
– Mm.

- What is *that thing*?

Lantana (2001)

Statistics of
Natural Language Data

Descriptive statistics about language

Why look at descriptive statistics?

- Understand what data you have
- Uncover problems that would bite you later
  - Data mismatches
  - Tokenisation problems
  - Encoding errors
- Understand how similar or different your data is from other data sets
  - Domain adaptation

On a more general level:

- Understand specific challenges of natural language

NLP corpora can be large!

```
$ ls -lh
total 14G
-rw-r--r-- 1 user users 103M Apr 19 14:43 europarl-v10.en.gz
-rw-r--r-- 1 user users 368M Jan 14  2019 news.2010.en.gz
-rw-r--r-- 1 user users 715M Jan 14  2019 news.2011.en.gz
-rw-r--r-- 1 user users 724M Jan 14  2019 news.2012.en.gz
-rw-r--r-- 1 user users 1.2G Jan 14  2019 news.2013.en.gz
-rw-r--r-- 1 user users 1.2G Jan 14  2019 news.2014.en.gz
-rw-r--r-- 1 user users 1.2G Jan 14  2019 news.2015.en.gz
-rw-r--r-- 1 user users 902M Jan 14  2019 news.2016.en.gz
-rw-r--r-- 1 user users 1.3G Jan 14  2019 news.2017.en.gz
-rw-r--r-- 1 user users 895M Jan 14  2019 news.2018.en.gz
-rw-r--r-- 1 user users 2.1G Feb 27  2020 news.2019.en.gz
-rw-r--r-- 1 user users 2.6G Feb 26 12:56 news.2020.en.gz
-rw-r--r-- 1 user users  40M Jan 13 18:16 news-commentary-v16.en.gz
```

Corpus size

- File size (`ls -lh`)
- Number of sentences (`wc -l`)
- Number of tokens (`wc -w`)

```
$ wc emoji/*_text.txt
  50000  596032 3705901 emoji/test_text.txt
  45000  531790 3353167 emoji/train_text.txt
   5000   56167  341079 emoji/val_text.txt
 100000 1183989 7400147 total
$ wc -l emoji/train_*.txt
 45000 emoji/train_labels.txt
 45000 emoji/train_text.txt
 90000 total
$ gzip -cd news.2010.en.gz | wc
 7272144 146644972 882477016
```


Types and tokens

- **Tokens**: “Running words”.
- **Types**: “Different tokens”.
- **Vocabulary** or **lexicon**: List of all different tokens occurring in the text.
- **Type/token ratio:**
  Ratio of number of types (vocabulary size) to number of tokens (text/corpus size).

Types and tokens

Star light , star bright ,	6
First star I see tonight ;	6
I wish I may , I wish I might ,	10
Have the wish I wish tonight .	7
	29

```
$ wc star-light.txt
       4       29      117 star-light.txt
```

Types and tokens

Star light , star bright ,	6
First star I see tonight ;	6
I wish I may , I wish I might ,	10
Have the wish I wish tonight .	7
	29

```
$ tr ' ' '\n' <star-light.txt | sort | uniq -c | sort -r
      6 I             1 may
      4 wish          1 light
      4 ,             1 bright
      2 tonight       1 Star
      2 star          1 Have
      1 the           1 First
      1 see           1 ;
      1 might         1 .
```


Types and tokens

Star light , star bright ,	6
First star I see tonight ;	6
I wish I may , I wish I might ,	10
Have the wish I wish tonight .	7
	29

- In Python, try `set()` or `collections.Counter` to produce vocabularies!
- Lowercasing makes a difference!

Types and tokens

Star light , star bright ,	6
First star I see tonight ;	6
I wish I may , I wish I might ,	10
Have the wish I wish tonight .	7
	29

- Text size: 29 tokens
- Vocabulary size: 16 tokens
- Type-token ratio: 16/29 = 0.551

Zipf's law

- **The frequency of a word is inversely proportional to its rank in the frequency table.**
- A few words at the top of the table have a very high frequency.
- Most other words have very low frequencies.
- Frequent words: Often function words (prepositions, auxiliary verbs, etc.)
- Just a few words make up the bulk of any text.
- But a large part of any text consists of rare words.

Log-log plot

- Zipf's law can be made visible in a plot of log(frequency) against log(rank).
- Let f be the frequency, r the rank and a a constant.
  We expect:

$$f = \frac{a}{r}$$
$$\log f = \log a - \log r$$

- In a log-log plot, we should see a straight descending line.

Plotting word frequencies

Consequences

- Some words are very frequent, but **most words are rare**.
- No matter how large your corpus already is,
  you will always see new, unknown words
  if you add more data.
- **Vocabularies grow quickly** and create challenges
  in terms of storage and processing speed.
- **Data sparseness** is pervasive in natural language processing.
- **Domain shift:** Models trained for one domain
  may perform very poorly for another.

Preprocessing trade-offs

- Preprocessing is about making data sparseness manageable.
- Tokenisation creates categories that you can hope to find again,
  so you can generalise from earlier experience.
  - Don't make data sparseness artifically worse
    by keeping punctuation attached to words etc.
- Lowercasing or compound splitting can also help,
  especially for small data sets.

Modelling Language

Language Modelling

- *Language models* learn a probability distribution over sequences of words.

$$p(w_1, w_2, \ldots, w_N)$$

- Encode properties of a certain type of language.
- Distinguish plausible from less plausible word sequences.
- Generate plausible-sounding word sequences.
- Learn generic relations between words.
- Often used as components in other models:
  - ASR: Acoustic model + Language model
  - Statistical MT: Translation model + Language model

Chain rule of probability

Star	light	,	star	bright	,	first	star	I	see	tonight	!

$$\begin{aligned} p(w_1, w_2, \ldots, w_N) &= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_2, w_1) \cdot \\ &\quad p(w_4|w_3, w_2, w_1) \cdots p(w_N|w_{N-1}, \ldots, w_2, w_1) \\ &= p(w_1) \prod_{i=1}^{N} p(w_i|w_1, \ldots, w_{i-1}) \end{aligned}$$

- We need to estimate values for all these probabilities, for all possible instantiations of $w_1, \ldots, w_N$.
- That's a lot of parameters!

Independence assumptions

- Recall that natural language has strong *local* dependencies.
  - *the* strongly favours a following noun (or adjective)
  - After a full stop, we're likely to see a word starting with a capital letter.
- *Long-range dependencies* do exist and are important, but not as strong (and more difficult to model).
- We can make *independence assumptions* to simplify the model.

Markov assumption

- **Markov assumption:**
  Each element of the sequence depends only on the immediately preceding element
  and is *independent* of the previous history.

$$p(w_i|w_1,\ldots,w_{i-1}) \approx p(w_i|w_{i-1})$$

- **k-th order Markov assumption:**
  Each element of the sequence depends only on the k immediately preceding elements.

$$p(w_i|w_1,\ldots,w_{i-1}) \approx p(w_i|w_{i-k},\ldots,w_{i-1})$$

- Note: These are *approximations*!

2nd order Markov assumption

Star	light	,	star	bright	,	first	star	I	see	tonight	!

$$\begin{aligned} p(w_1,w_2,\ldots,w_N) &\approx p(w_1)\cdot p(w_2|w_1)\cdot p(w_3|w_2,w_1)\cdot \\ &\quad p(w_4|w_3,w_2)\cdots p(w_N|w_{N-2},w_{N-1}) \\ &= p(w_1)\prod_{i=1}^{N} p(w_i|w_{i-2},w_{i-1}) \end{aligned}$$

Model size

- Let V be the vocabulary size and
  N be the maximum sentence length.
- Each w_i can be any vocabulary item $\rightarrow V$ choices.
- For a model *without* independence assumptions,
  - we need to estimate $p(w_N|w_1,w_2,\ldots,w_{N-1})$.
  - up to V^N model parameters
- For a k-th order Markov model,
  - we need to estimate $p(w_{k+1}|w_1,\ldots,w_k)$.
  - up to V^{k+1} model parameters
- In a realistic language model,
  - $V \approx 10^4$ to 10^5
  - $N \approx 30$ to 80
  - $k \approx 2$ to 5

Sequence padding

- Is this a good complete sentence???

  Star light , star bright , first star I
- Is this a good start of a sentence???

  , star bright ,
- Add special symbols to mark the start and end of each sentence!
  - ⟨s⟩ or BOS for *beginning of sentence*
  - ⟨/s⟩ or EOS for *end of sentence*

  ⟨s⟩ Star light , star bright , first star I see tonight ! ⟨/s⟩

N-gram language model

- N-gram models are Markov models for language modelling.
- N-gram: sequence of n tokens in a text.
- 1-gram = unigram; 2-gram = bigram; 3-gram = trigram

⟨s⟩ Star light , star bright , first star I see tonight ! ⟨/s⟩
⟨s⟩ Star light
Star light ,
light , star
, star bright
star bright ,
bright , first
, first star
first star I
star I see
...

Parameter Estimation/ Model Training

IT UNIVERSITY OF COPENHAGEN

Maximum-likelihood estimation

Simplest method to estimate a conditional probability:
Count how often the target event occurs
in the context conditioned on.

$$p(w_3|w_1, w_2) = \frac{\text{count}(w_1 w_2 w_3)}{\text{count}(w_1 w_2 \bullet)}$$

Example:

⟨s⟩ Star light , star bright , first star I see tonight ! ⟨/s⟩

$$p(\text{bright}|\text{star}) = \frac{\text{count}(\text{star bright})}{\text{count}(\text{star} \bullet)} = \frac{1}{2} = 0.5$$

IT UNIVERSITY OF COPENHAGEN

Problems with maximum likelihood estimation

Any token that has not been seen in a particular context will have a count of 0, and therefore a probability of zero.

⟨s⟩ I wish I might have the wish I wish tonight ! ⟨/s⟩

$$p(\text{I}|\text{wish}) = \frac{\text{count}(\text{wish I})}{\text{count}(\text{wish} \bullet)} = \frac{2}{3} = 0.667$$

Now score this (assuming a bigram model):

I wish *you* might have the wish *you* wish tonight !

$$p(w_1, \ldots, w_N) = p(w_1) \prod_{i=1}^{N} p(w_i|w_{i-1})$$

IT UNIVERSITY OF COPENHAGEN

Problems with maximum likelihood estimation

Any token that has not been seen in a particular context will have a count of 0, and therefore a probability of zero.

⟨s⟩ I wish I might have the wish I wish tonight ! ⟨/s⟩

$$p(\text{I}|\text{wish}) = \frac{\text{count}(\text{wish I})}{\text{count}(\text{wish } \bullet)} = \frac{2}{3} = 0.667$$

Now score this (assuming a bigram model):

I wish *you* might have the wish *you* wish tonight !

$$p(\text{you}|\text{wish}) = \frac{\text{count}(\text{wish you})}{\text{count}(\text{wish } \bullet)} = \frac{0}{3} = 0$$

Problems with maximum likelihood estimation

- MLE *underestimates* the probability of n-grams not seen in the training data.
- MLE *overestimates* the probability of n-grams seen only a few times.

$$p(\text{the}|\text{have}) = \frac{\text{count}(\text{have the})}{\text{count}(\text{have } \bullet)} = \frac{1}{1} = 1$$

- We get good estimates of very frequent tokens.
- But Zipf's law says *most tokens are **not** frequent*!