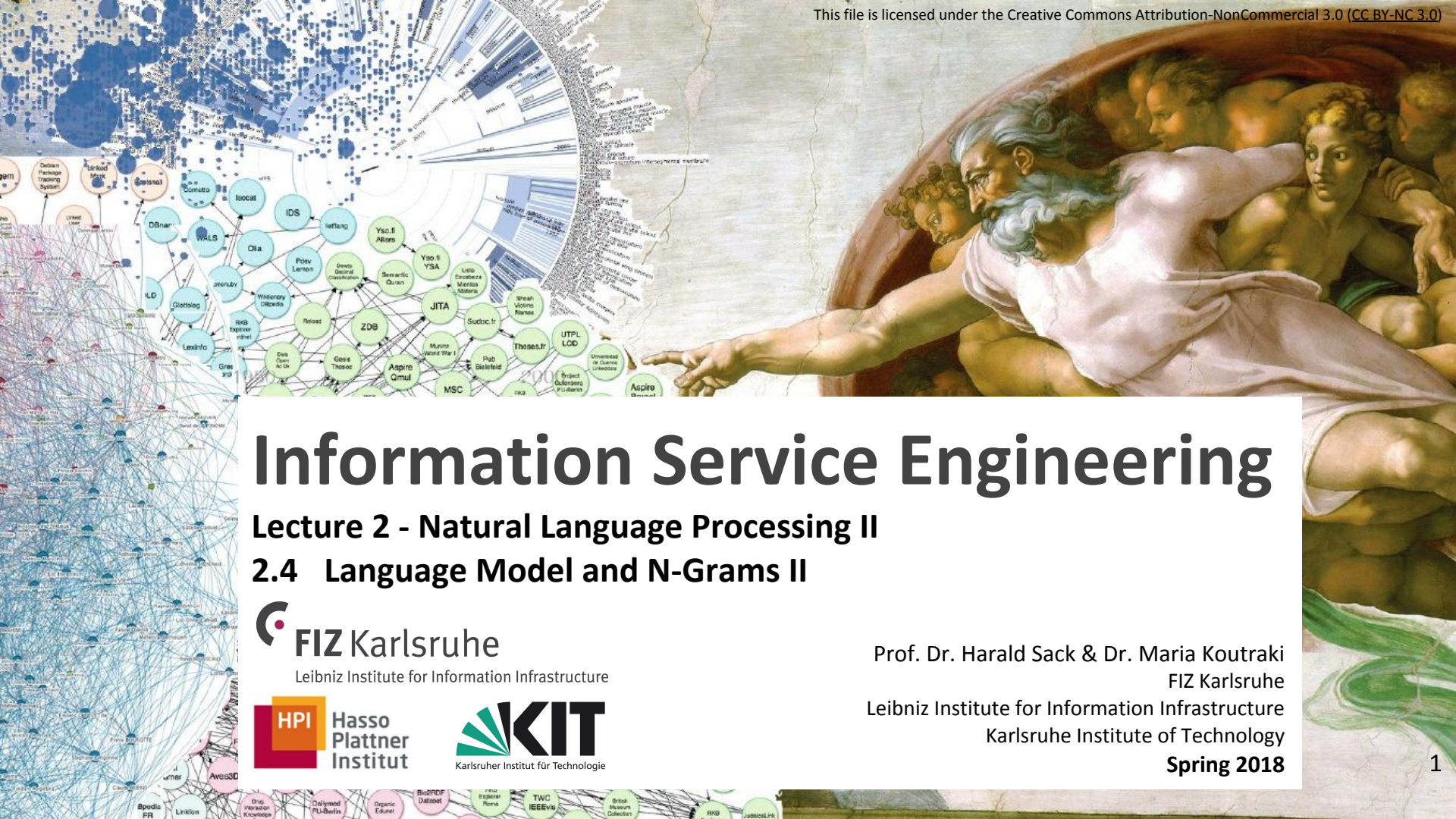




Information Service Engineering

Lecture 2 - Natural Language Processing II

2.4 Language Model and N-Grams II



FIZ Karlsruhe

Leibniz Institute for Information Infrastructure



Hasso
Plattner
Institut



Karlsruher Institut für Technologie

Prof. Dr. Harald Sack & Dr. Maria Koutraki
FIZ Karlsruhe

Leibniz Institute for Information Infrastructure
Karlsruhe Institute of Technology
Spring 2018

Information Service Engineering

Lecture 2: Natural Language Processing II

2.1 Finite State Automata

2.2 Finite State Transducers

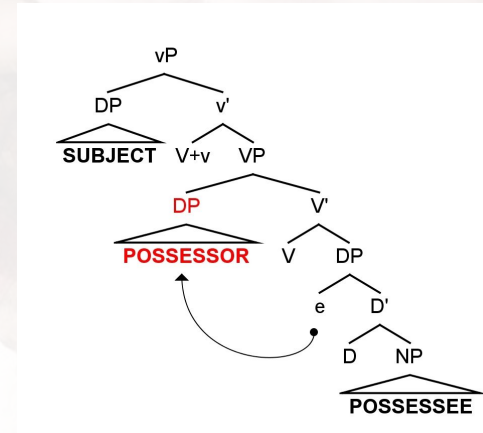
2.3 Language Model and N-Grams

2.4 Language Model and N-Grams II

2.5 Language Model and N-Grams III

2.6 Part-of-Speech Tagging

2.7 Part-of-Speech Tagging II



Conditional Probability

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

$$P(S) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdots P(w_n|w_1, \cdots, w_{n-1})$$

$$P(S) = \prod_{i=1}^n P(w_i|w_1, \cdots, w_{i-1})$$

Generalization of Bayes Theorem for modelling a sequence of words in a (natural) language

- $P(\text{to be or not}) = P(\text{to}) \cdot$
 $P(\text{be}|\text{to}) \cdot$
 $P(\text{or}|\text{to,be}) \cdot$
 $P(\text{not}|\text{to,be,or})$

Language Model

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

$$P(S) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdots P(w_n|w_1, \cdots, w_{n-1})$$

$$P(S) = \prod_{i=1}^n P(w_i|w_1, \cdots, w_{i-1})$$

- Complexity: $O(|V|^{n^*})$ V...Vocabulary, n^* ...maximum sentence length
- 475,000 main headwords in Webster's Third New International Dictionary
- Average English sentence length: 14.3 words
- A rough estimate: $O(475000^{14}) \sim 3.38 \cdot 10^{66}$ TB
- By applying an N-gram model, we make the model more compact: $O(475000^n)$

But, how do we determine the probability of words?

The Corpus as Basis of the Language Model

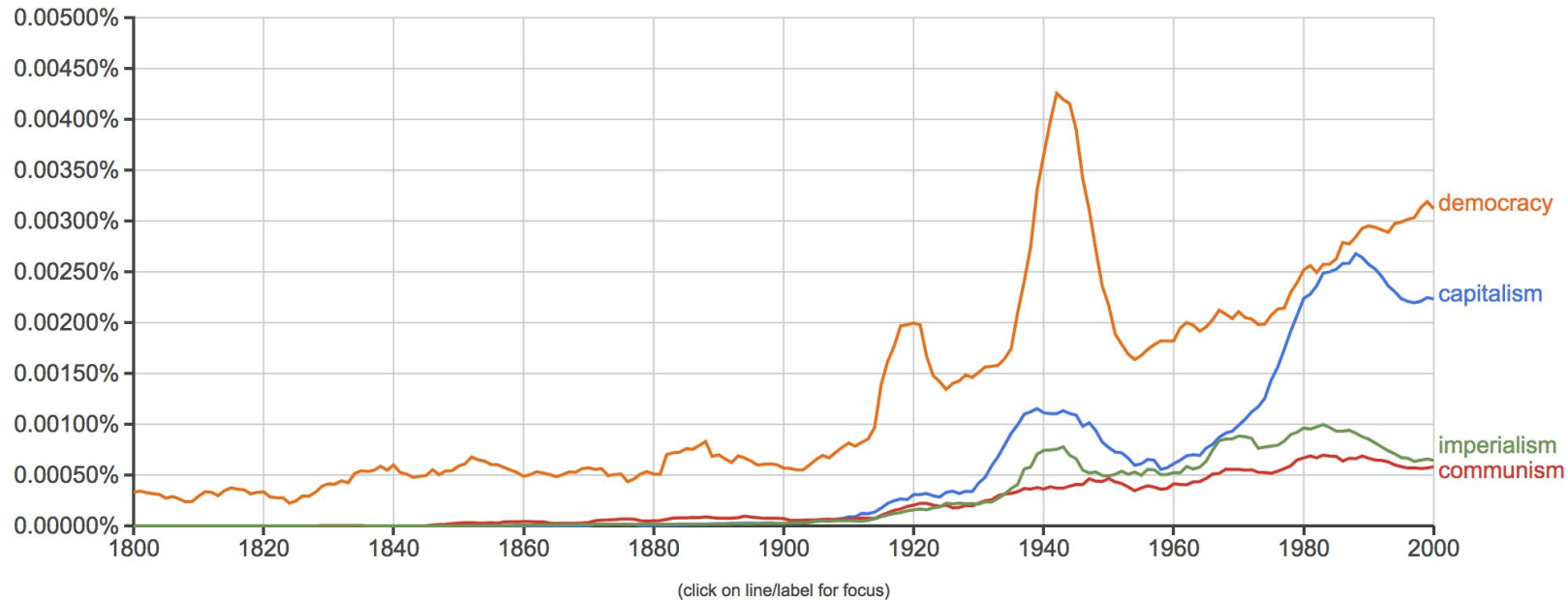
Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- **A corpus** is a computer-readable collection of text or speech
 - Corpus of Contemporary American English
 - 520m words, US, 1990-2015
 - The British National Corpus
 - 100m words, UK, 1991-1994
 - The International Corpus of English
 - 23 local corpora, 1m words each
 - **The Google N-gram Corpus**
 - N-grams from printed sources, 1500-2008, in English, Chinese, French, German, Hebrew, Italian, Russian, or Spanish

<http://www.corpusdata.org/>
<http://www.natcorp.ox.ac.uk/>
<https://books.google.com/ngrams>

Graph these comma-separated phrases: ☐ case-insensitive

between and from the corpus with smoothing of [Search lots of books](#)



Search in Google Books:

[1800 - 1936](#)
[1937 - 1987](#)
[1988 - 1991](#)
[1992 - 1996](#)
[1997 - 2000](#)
[capitalism](#)
[English](#)
[1800 - 1917](#)
[1918 - 1981](#)
[1982 - 1987](#)
[1988 - 1993](#)
[1994 - 2000](#)
[communism](#)
[English](#)

Word Occurrence

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- Probabilities are based on counting things
- A **language** consists of a set **V** of words (**Vocabulary**)
- A **word** can occur several times in a text
 - **Word Token**: each occurrence of words in text
 - **Word Type**: each unique occurrence of words in the text
- Example:
 - *"To be or not to be."*
 - # Word Tokens: 6
 - # Word Types: 4

Excursion

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- Determine the number of **word tokens** and **word types** of William Shakespeare's "A Midsummernight's Dream"

1. Download the Shakespeare corpus at

<http://lexically.net/wordsmith/support/shakespeare.html>

2. Convert into plain text (only English characters):

```
tr -d -c '[:alpha:][:space:]'
```

3. Word tokenization

```
tr -s '[:space:]' '\n'
```

4. # of Word tokens: `wc -l`

```
# of Word types:      sort -u | wc -l
```

Word Frequency

Information Service Engineering - Lecture 2 Natural Language Process

Rank	Word	Count	Freq(%)
1	The	69970	6.8872
2	of	36410	3.5839
3	and	28854	2.8401
4	to	26154	2.5744
5	a	23363	2.2996
6	in	21345	2.1010
7	that	10594	1.0428
8	is	10102	0.9943
9	was	9815	0.9661
10	He	9542	0.9392
11	for	9489	0.9340
12	it	8760	0.8623
13	with	7290	0.7176
14	as	7251	0.7137
15	his	6996	0.6886
16	on	6742	0.6636
17	be	6376	0.6276
18	at	5377	0.5293
19	by	5307	0.5224
20	I	5180	0.5099

Shakespeare Example:
<https://goo.gl/aoAUeN>

Back to the N-Gram Model

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- The intuition of the **N-gram model** is that
 - instead of computing the probability of a word given its entire history,
 - we can **approximate** the history by just the last few words.
- For the **bigram model** we approximate the probability of a word given all its previous words by using **only the conditional probability of its preceding word (Markov assumption)**

$$P(w_n | w_1, \dots, w_{n-1}) \approx P(w_n | w_{n-1})$$

Markov Assumption

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- Using the **Markov Assumption** to compute the probability of a text sequence for the bigram model:

$$P(S) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$



$$P(S) = \prod_{i=1}^n P(w_i | w_{i-1})$$

Maximum Likelihood Estimation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- How to estimate N-gram probabilities?
- **Maximum Likelihood Estimation (MLE)**
 - a method of estimating the parameters of a statistical model given observations
 - by finding the parameter values that maximize the likelihood of making the observations given the parameters.
- The **MLE for the parameters of an N-gram model** is computed by **normalizing counts from a corpus**

$$P(w_n | w_{n-1}) = \frac{\#(w_{n-1} w_n)}{\sum_w \#(w_{n-1} w)} = \frac{\#(w_{n-1} w_n)}{\#w_{n-1}}$$

Markov Assumption and MLE

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

$$P(\text{to be or not}) = P(\text{not}|\text{to be or}) \cdot P(\text{or}|\text{to be}) \cdot P(\text{be}|\text{to}) \cdot P(\text{to})$$



Markov Assumption (Bigrams)

$$P(\text{to be or not}) = P(\text{to}) \cdot P(\text{be}|\text{to}) \cdot P(\text{or}|\text{be}) \cdot P(\text{not}|\text{or})$$

MLE



$$P(\text{not}|\text{or}) = \frac{\#or\ not}{\#or}$$

N-Gram Model

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- Unigram:
$$P(S) = \prod_{i=1}^n P(w_i)$$
- Bigram:
$$P(S) = \prod_{i=1}^n P(w_i | w_{i-1})$$
- Trigram:
$$P(S) = \prod_{i=1}^n P(w_i | w_{i-1}, w_{i-2})$$
- N-gram:
$$P(S) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$

N-Gram Model - Generating Shakespeare

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.4 Language Model and N-Grams II

- **Unigram:** To him swallowed confess hear both. Which. Of save on trail for are
ay device and rote life have
Hill he late speaks; or! a more to leg less first you enter.
- **Bigram:** Why dost stand forth thy canopy, forsooth; he is this palpable hit
the King Henry. Live king. Follow.
What means, sir. I confess she? then all sorts, he is trim, captain
- **Trigram:** Fly, and will rid me these news of price. Therefore the sadness of
parting, as they say,'tis done.
This shall forbid it should be branded, if renown made it empty.
- **4-gram:** King Henry. What! I will go seek the traitor Gloucester. Exeunt some
of the watch. A great banquet serv'd in; It cannot be but so.



Language Model and N-Grams III

Next Lecture...

Picture References:

- P.5: Shakespeare's Words, <https://www.flickr.com/photos/disowned/1158260369>
- P.17: Pieter Brueghel the Elder, The Parable of the Blind Leading the Blind (1568), [https://commons.wikimedia.org/wiki/File:Pieter Bruegel the Elder - The Parable of the Blind Leading the Blind - WGA3511.jpg](https://commons.wikimedia.org/wiki/File:Pieter_Bruegel_the_Elder_-_The_Parable_of_the_Blind_Leading_the_Blind_-_WGA3511.jpg)