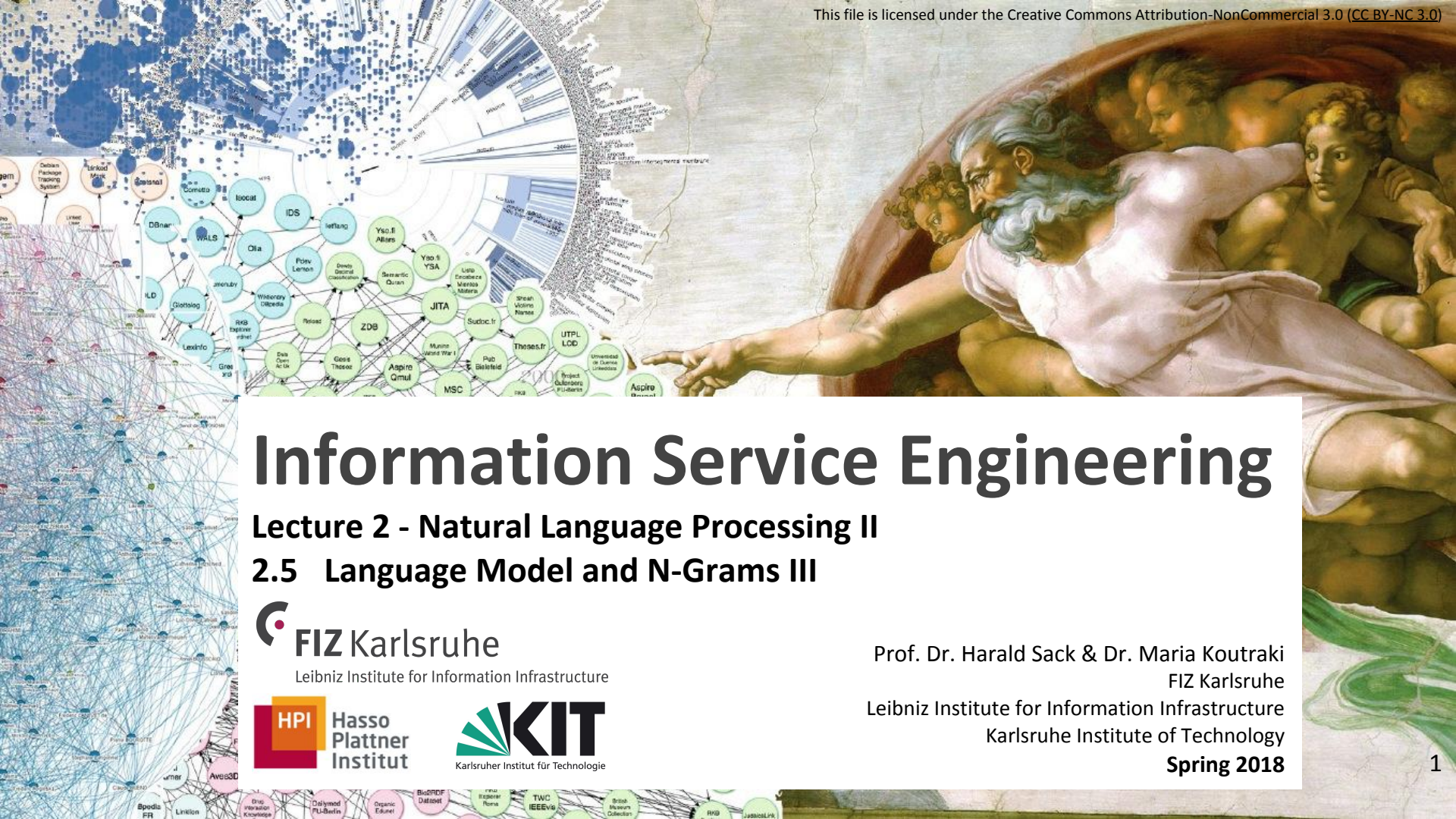




Information Service Engineering

Lecture 2 - Natural Language Processing II

2.5 Language Model and N-Grams III



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

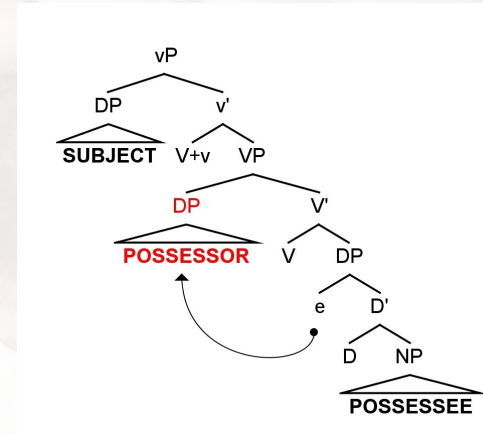
2.3 Language Model and N-Grams

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



Maximum Likelihood Estimation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **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**

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

Maximum Likelihood Estimation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **Example Corpus:**
 - `<s> I saw the boy </s>`
 - `<s> the man is working </s>`
 - `<s> I walked in the street </s>`
- **Vocabulary:**
 - $V = \{I, \text{saw}, \text{the}, \text{boy}, \text{man}, \text{is}, \text{working}, \text{walked}, \text{in}, \text{street}\}$

Maximum Likelihood Estimation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

1. Bracket each sentence by **special start and end symbols** $\langle s \rangle \dots \langle /s \rangle$:
 $\langle s \rangle$ I saw the boy $\langle /s \rangle$
(We only assign probabilities to strings ...)
2. Count the **frequency of each n-gram**:
 $\#(\langle s \rangle \text{ I}) = 1, \#(\text{I saw}) = 1, \dots$
3. **Normalize** to get the **probability**:

$$P(\text{the}|\text{saw}) = \frac{\# \text{saw the}}{\# \text{saw}}$$

This is the **relative frequency estimate**

Maximum Likelihood Estimation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **Example Corpus:**

- <s> I saw the boy </s>
- <s> the man is working </s>
- <s> I walked in the street </s>

boy	I	in	is	man	saw	street	the	walked	working	<s>	</s>
1	2	1	1	1	1	1	3	1	1	3	3

unigram counts

Maximum Likelihood Estimation

	boy	I	in	is	man	saw	street	the	walked	working	<s>	</s>
boy	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	1	0	0	1	0	0	0
in	0	0	0	0	0	0	0	1	0	0	0	0
is	0	0	0	0	0	0	0	0	0	1	0	0
man	0	0	0	1	0	0	0	0	0	0	0	0
saw	0	0	0	0	0	0	0	1	0	0	0	0
street	0	0	0	0	0	0	0	0	0	0	0	1
the	1	0	0	0	1	0	1	0	0	0	0	0
walked	0	0	1	0	0	0	0	0	0	0	0	0
working	0	0	0	0	0	0	0	0	0	0	0	1
<s>	0	2	0	0	0	0	0	1	0	0	0	0
</s>	0	0	0	0	0	0	0	0	0	0	0	0

bigram counts

Maximum Likelihood Estimation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **Estimation** of the Maximum Likelihood Estimation for a new sentence:
 - **<s> I saw the man**

$$\begin{aligned} P(S) &= P(I | < s >) \cdot P(\text{ saw } | I) \cdot P(\text{ the } | \text{ saw }) \cdot P(\text{ man } | \text{ the }) \\ &= \frac{\#(< s > I)}{\#(< s >)} \cdot \frac{\#(I \text{ saw})}{\#(I)} \cdot \frac{\#(\text{ saw the })}{\#(\text{ saw })} \cdot \frac{\#(\text{ the man })}{\#(\text{ the })} \\ &= \frac{2}{3} \cdot \frac{1}{2} \cdot \frac{1}{1} \cdot \frac{1}{3} = \frac{1}{9} \end{aligned}$$

MLE - Unknown Words

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- What if a word occurs that is not part of our vocabulary?
 - `<s> I saw the girl </s>`
- **Closed Vocabulary Assumption:**
The test set can contain only words from our vocabulary
- **Open Vocabulary Assumption:**
The test set can contain **unknown words (out of vocabulary words, OOV)** that are not part of our vocabulary

MLE - Open Vocabulary

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- In an **Open Vocabulary system** unknown words are modeled by adding a pseudo-word **<UNK>**
 1. **Choose** a (fixed) vocabulary
 2. **Convert** in the **training set** any word not in the vocabulary (**OOV word**) into **<UNK>**
 3. **Convert** in the **test set** any unknown word into **<UNK>**
 4. **Estimate** probabilities for **<UNK>** like for any regular vocabulary word
- **Alternatively**, don't choose a vocabulary, but turn every first occurrence of a word type in the **training set** into **<UNK>**

MLE - Evaluation

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- Divide the corpus to two parts:
 - **training set** and **test set**
- Build a **language model** from the **training set**
 - Compute word frequencies, etc..
- Estimate the **probability** of the **test set**
- Calculate the **average branching factor** of the test set

Average Branching Factor

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- The **branching factor** of a language is the **number of possible next words that can follow any word**
- A good language model should be able to minimize this number
 - i.e., give a higher probability to the words that occur in real texts
- The **average branching factor** is referred to as **Perplexity (PP)** and is the most common evaluation metric for N-gram language models
- **Perplexity** is a measurement of how well a probability distribution or probability model predicts a sample

Perplexity

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **Goal:** give higher probability to frequent texts
 - Therefore: minimize the perplexity of the frequent texts

$$P(S) = P(w_1, w_2, \dots, w_n)$$

$$PP(S) = P(w_1, w_2, \dots, w_n)^{-\frac{1}{n}} = \sqrt[n]{\frac{1}{P(w_1, w_2, \dots, w_n)}}$$

$$PP(S) = \sqrt[n]{\prod_{i=1}^n \frac{1}{P(w_i | w_1, w_2, \dots, w_{i-1})}}$$

Perplexity Examples

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- Wall Street Journal (19,979 word vocabulary)
 - Training set: 38 million word
 - Test set: 1.5 million words
- Perplexity:
 - Unigram: 962
 - Bigram: 170
 - Trigram: 109

Unknown N-Grams

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **Corpus:**

- $\langle s \rangle$ I saw the boy $\langle /s \rangle$
- $\langle s \rangle$ the man is working $\langle /s \rangle$
- $\langle s \rangle$ I walked in the street $\langle /s \rangle$

- **Test set:**

- $\langle s \rangle$ I saw the man in the street $\langle /s \rangle$

$$P(S) = P(I | \langle s \rangle) \cdot P(\text{saw} | I) \cdot P(\text{the} | \text{saw}) \cdot P(\text{man} | \text{the}) \cdot P(\text{in} | \text{man}) \cdot P(\text{the} | \text{in}) \cdot P(\text{street} | \text{the})$$

$$P(S) = \frac{\#(\langle s \rangle I)}{\#(\langle s \rangle)} \cdot \frac{\#(I \text{ saw})}{\#(I)} \cdot \frac{\#(\text{saw the})}{\#(\text{saw})} \cdot \frac{\#(\text{the man})}{\#(\text{the})} \cdot \frac{\#(\text{man in})}{\#(\text{man})} \cdot \frac{\#(\text{in the})}{\#(\text{in})} \cdot \frac{\#(\text{the street})}{\#(\text{the})}$$

$$P(S) = \frac{2}{3} \cdot \frac{1}{2} \cdot \frac{1}{1} \cdot \frac{1}{3} \cdot \frac{0}{1} \cdot \frac{1}{1} \cdot \frac{1}{3} = 0$$

Unknown N-Grams

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

Example:

- Shakespeare corpus consists of $N=884,647$ word tokens and a vocabulary of $V=29,066$ word types
- Only 30,000 word types occurred (of possible 475.000 referenced by Webster's 3rd New International Dictionary)
 - Words not in the training data $\Rightarrow P(\text{unknown Word})=0$
- Only **0.04%** of all possible 2-grams occurred

Laplace Smoothing

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.5 Language Model and N-Grams III

- **Assign a small probability to all unknown N-grams** that do not occur in the train corpus
 - i.e. add 1

	boy	I	in	is	man	saw	street	the	walked	working
boy	1	1	1	1	1	1	1	1	1	1
I	1	1	1	1	1	2	1	1	2	1
in	1	1	1	1	1	1	1	2	1	1
is	1	1	1	1	1	1	1	1	1	2
man	1	1	1	2	1	1	1	1	1	1
saw	1	1	1	1	1	1	1	2	1	1
street	1	1	1	1	1	1	1	1	1	1
the	2	1	1	1	2	1	2	1	1	1
walked	1	1	2	1	1	1	1	1	1	1
working	1	1	1	1	1	1	1	1	1	1

$$P(w_i|w_{i-1}) = \frac{\#(w_{i-1}w_i)}{\#(w_{i-1})}$$



$$P(w_i|w_{i-1}) = \frac{\#(w_{i-1}w_i) + 1}{\#(w_{i-1}) + |V|}$$



Part-of-Speech Tagging

Next Lecture...

Picture References:

- P.19: Pieter Brueghel the Elder, The Blue Cloak, The Topsy Turvy World (1559),
[https://commons.wikimedia.org/wiki/File:Pieter Brueghel the Elder - The Dutch Proverbs - Google Art Project.jpg](https://commons.wikimedia.org/wiki/File:Pieter_Brueghel_the_Elder_-_The_Dutch_Proverbs_-_Google_Art_Project.jpg)