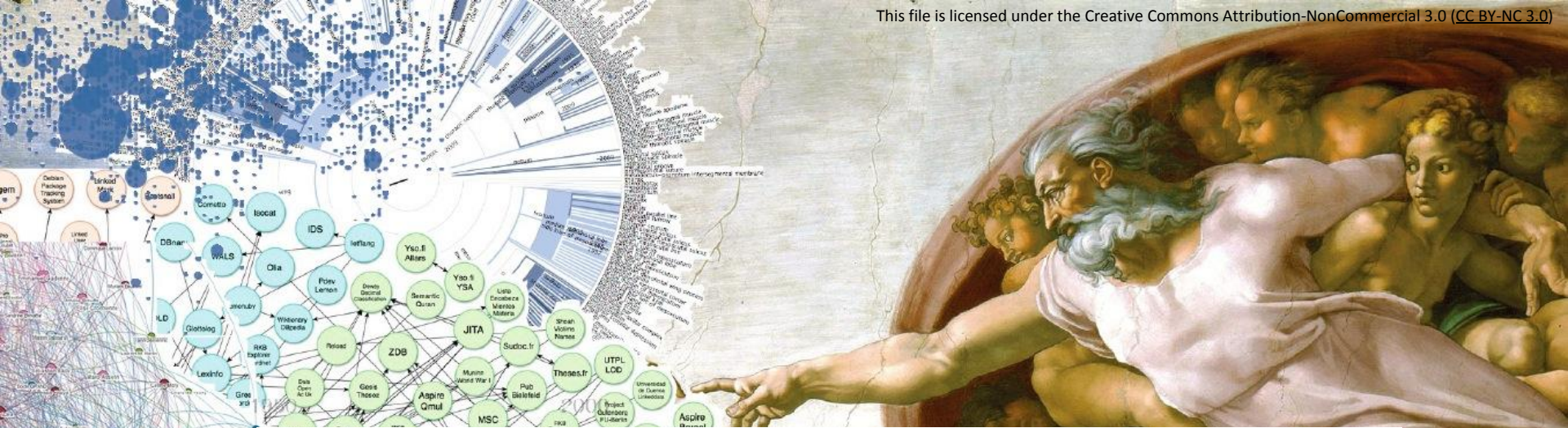




Information Service Engineering

Lecture 2 - Natural Language Processing II

2.3 Language Model and N-Grams



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

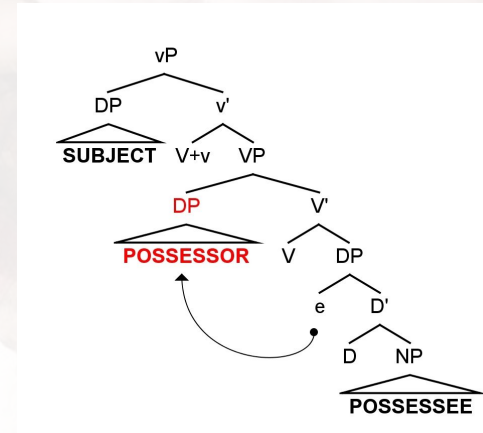
2.3 Language Model and N-Grams

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



Word Prediction

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- “To be or not to...”
- “The pen is mightier than the...”
- “You can’t judge a book by...”
- “Es irrt der Mensch, solange’ er”
- “Die Botschaft hör ich wohl, allein mir fehlt der ...”

Human Prediction

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- How do humans predict words?
 - **Domain knowledge**, as e.g.
red blood vs. red hat
 - **Syntactic knowledge**, as e.g.
The ... <adjective|noun>
 - **Lexical knowledge**, as e.g.
Baked potato vs. steak
- **Claim:** A useful part of the knowledge needed to allow Word Prediction can be captured using **simple statistical techniques**

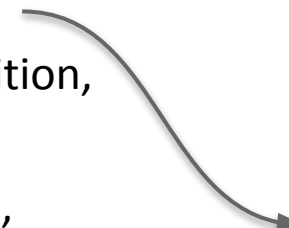
N-Gram Models

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- **Word Prediction** can be formalized with probabilistic **N-gram models**
 - **2-gram (bigram)**: (to, be), (be, or), (or, not), (not, to)
 - **3-gram (trigram)**: (to, be, or), (be, or, not), (or, not, to)
- An **N-gram** is an N-Token of words.
- In an N-gram model, the **last word w_n** depends only on the **previous n-1 words $(w_1, \dots, w_{n-1})$** (**Markov assumption**)
- and thus, the **last word w_n** will be computed **from the previous n-1 words $(w_1, \dots, w_{n-1})$**

N-Gram Models

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- Statistical models of word sequences are called **Language Models (LM)**
 - **N-grams** are used for
 - **speech recognition**,
 - handwriting recognition,
 - spelling correction,
 - machine translation,
 - authorship identification,
 - sentiment analysis,...
- 
- „Computers can recognize speech.“
„Computers can wreck a nice peach.“
 - „Give peace a chance.“
„Give peas a chance.“
 - „ice cream.“
„I scream.“
 - “Two birds are flying.”
“Two beards are flying.”

Basic Probability Theory

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- **Trial:**
 - Throwing a dice, predicting a word
- **Sample space Ω :**
 - The set of all possible outcomes
(all numbers in a lottery; all words in Shakespeare's plays)
- **Event $\omega \subseteq \Omega$:**
 - An actual outcome (a subset of Ω)
(predicting 'the', throwing a "3",...)

The Probability of Events

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- **Kolmogorov Axioms:**

1. Each event has a probability between 0 and 1.

$$0 \leq P(\omega \subseteq \Omega) \leq 1$$

The Probability of Events

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- **Kolmogorov Axioms:**

1. Each event has a probability between 0 and 1.

$$0 \leq P(\omega \subseteq \Omega) \leq 1$$

2. The null event has probability 0.

The probability that any event happens is 1.

$$P(\emptyset) = 0 \text{ and } P(\Omega) = 1$$

The Probability of Events

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

- **Kolmogorov Axioms:**

1. Each event has a probability between 0 and 1.

$$0 \leq P(\omega \subseteq \Omega) \leq 1$$

2. The null event has probability 0.

The probability that any event happens is 1.

$$P(\emptyset) = 0 \text{ and } P(\Omega) = 1$$

3. The probability of all disjoint events sums to 1

$$\sum_{\omega_i \subseteq \Omega} P(\omega_i) = 1 \quad \forall j \neq i : \omega_i \cap \omega_j = \emptyset, \bigcup_i \omega_i = \Omega$$

Conditional Probability

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Conditional Probability $P(B|A)$

that **event B** occurs under the assumption that **event A** has already occurred

Conditional Probability

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Conditional Probability $P(B|A)$

that **event B** occurs under the assumption that **event A** has already occurred

$$P(A, B) = P(A) \cdot P(B|A)$$

Bayes Formula

The probability that **event A occurs followed by event B** equals the probability that event A occurs and event B occurs under the assumption that event A has occurred

Conditional Probability

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Conditional Probability $P(B|A)$

that **event B** occurs under the assumption that **event A** has already occurred

$$P(A, B) = P(A) \cdot P(B|A)$$

Bayes Formula

The probability that **event A occurs followed by event B** equals the probability that event A occurs and event B occurs under the assumption that event A has occurred

$$P(A, B, C, D) = P(A) \cdot P(B|A) \cdot P(C|A, B) \cdot P(D|A, B, C)$$

Extension to 4 events

Conditional Probability

Information Service Engineering - Lecture 2 Natural Language Processing II - 2.3 Language Model and N-Grams

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

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

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 and N-Grams II

Next Lecture...

Picture References:

- P.16: Jan Brueghel the Elder, The Entry of the Animals into Noah's Ark (1613), https://commons.wikimedia.org/wiki/File:Jan_Brueghel_the_Elder_-_The_Entry_of_the_Animals_into_Noah%27s_Ark_-_Google_Art_Project.jpg