Natural language processing

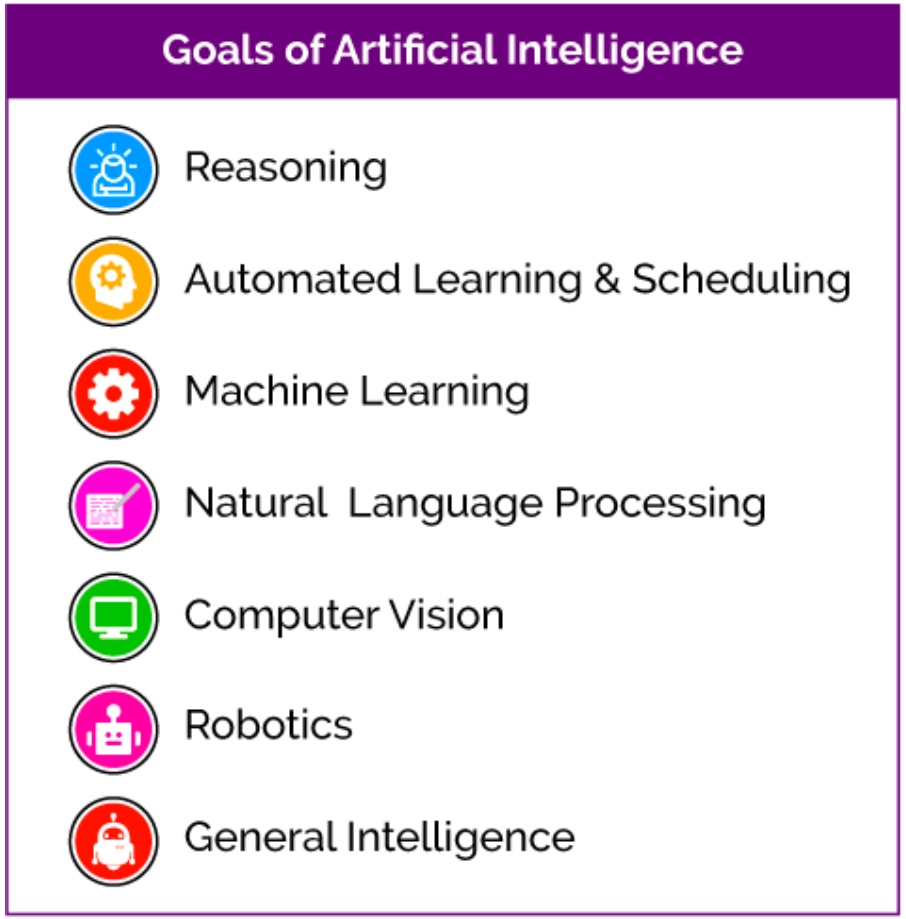
Guangyu Ge

Betreuer: Tobias 29.01.2018

GLIEDERUNG

- 1. Einführung
- 2. Anwendungsbereiche
- 3. Verfahren bei NLP
- 4. Zusammenfassung

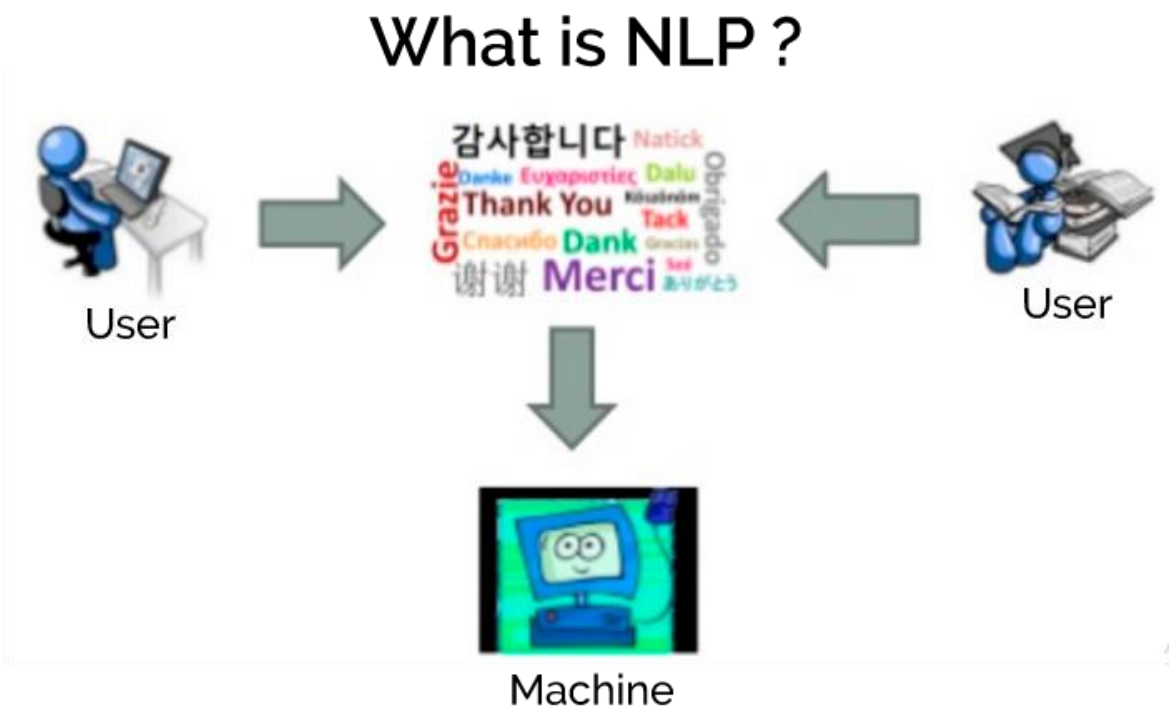
ZIELE VON KÜNSTLICHER INTELLIGENZ



Quelle: https://cdn-images-1.medium.com/max/800/0*Y707xutX3ZMVjkXX.png

DEFINITION

- **Natural-language processing** (**NLP**) is a field of computer science, artificial intelligence concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language data.



Quellle: https://content-static.upwork.com/blog/uploads/sites/3/2017/06/27091504/image-54.png

DIE GESCHICHTE

- 1950: Erster Konzept: Turing-Test.
- 1954: das Georgetown-Experiment, über 60 Sätze von Russisch ins Englisch übersetzt.
- 1960: SHRDLU, erst Sprache System.
- 1966: ELIZA, von Joseph Wiesenbaum entworfen, um "Personal-Center-Behandlung" zu simulieren.
- 1970: Programme zu entwerfen, "konzeptionelle Ontologien„

  Chat-Roboter.

KOMPLEXITÄT BEI NLP

Natural Language Understanding

- Syntaktische Ambiguität
- Semantische Ambiguität
- Anaphorische Mehrdeutigkeit

Natural Language Generation

- Textplanung
- Satzplanung
- Realisierung

GLIEDERUNG

- 1. Einführung
- 2. **Anwendungsbereiche**
- 3. Verfahren bei NLP
- 4. Zusammenfassung

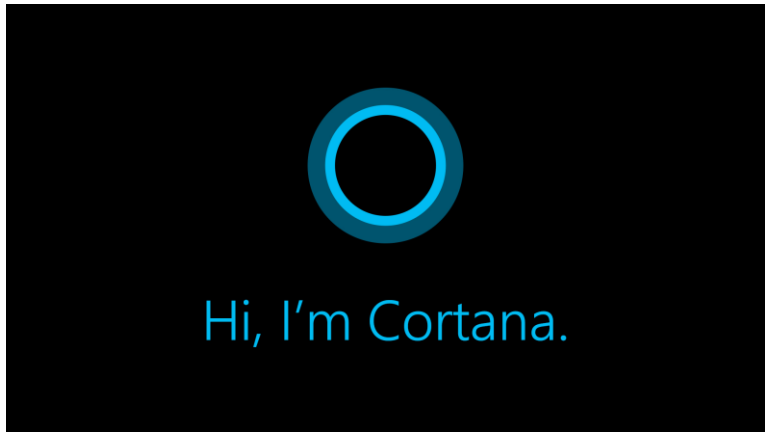
ANWENDUNGSBEREICHE

- 1980er- 1990er:
  - Textkomprimierung
  - Rechtschreibkorrektur
- 1990er:
  - automatische Klassifikation
  - automatische Indizierung

ANWENDUNGSBEREICHE

- Heutzutage:
  - automatisiertes Sprechen
  - automatisiertes Schreiben
  - automatische Zusammenfassung
  - maschinelle Übersetzung

BEKANNTE NLP BEISPIELE



Quelle:
http://www.cnet.de/wp-content/uploads/2015/01/google-translate-icon.png
https://cdn-images-1.medium.com/max/1600/1*6XGc6lTW_AdSgOuaI9ZPKg.png
http://logodatabases.com/wp-content/uploads/2012/03/alexa-logo-2012.jpg
https://searchengineland.com/figz/wp-content/seloads/2014/08/cortana-logo-1920.png

GLIEDERUNG

- 1. Einführung
- 2. Anwendungsbereiche
- **3. Verfahren bei NLP**
- 4. Zusammenfassung

N-GRAMM

Der Text wird dabei zerlegt, und jeweils aufeinanderfolgende Fragmente werden als N-Gramm zusammengefasst.

- N ist für beliebige Zahl, z.B n = 2, 3, 4,……
- Das Ziel ist Wort/ Satz in N-Teile zerlegen.
- Das Ergebnis der Zerlegung eines Textes in Fragmente.
- Die Fragmente:
  - Buchstaben,
  - Phoneme,
  - Wörter

2 BEISPIELE:

„Informationsverarbeitung“ unter 5-Gramm:

Infor, nform, forma, ormat, rmati, matio, ation, tions, ionsv, onsve, nsver, svera, verar, erarb, rarbe, arbei,rbeit, beitu, eitun, itung.

- Manchmal wird „Leerzeichen“ am Anfang ergänzt, um einfach zu suchen.

  _i, _in, _inf, _info, Infor, nform, forma, ……

- „Das ist nur ein Beispiel.“ unter Bigram:

  Das ist, ist nur, nur ein, ein Beispiel.

STATISITISCHER ANSATZ

Das ist nur ein Beispiel. unter Bigram:

Das ist, ist nur, nur ein, ein Beispiel.

Annahme: Wahrscheinlichkeit des Wortes hängt von den vorherigen Wörtern ab.

"P(ist) = P(ist | Das)

Analog:

$P(T)=P(W_1W_2W_3W_n)=P(W_1)P(W_2|W_1)P(W_3|W_1W_2)...P(W_n|W_1W_2...W_{n-1})$

Nachteile: Zu viel Paramente, zu wenig Relevante Daten

STATISITISCHER ANSATZ

Verbesserte Annahme: Die Wahrscheinlichkeit des jeweiligen Wortes hängt nur von der Wahrscheinlichkeit des vorherigen Wortes ab.

Markov:

$P(T) = P(W_1)P(W_2|W_1)P(W_3|W_2)\ldots P(W_n|W_{n-1})$

Nachteile: aber ungenauer bzw. zusätzliche Annahme!

STATISITISCHER ANSATZ

Angenommen ist Gesamtzahl der Wörter von bestimmtem Korpus 17,000.

Er	4627
möchte	2873
eine	4892
Reise	748
nach	482
Mond	303
kaufen	1042

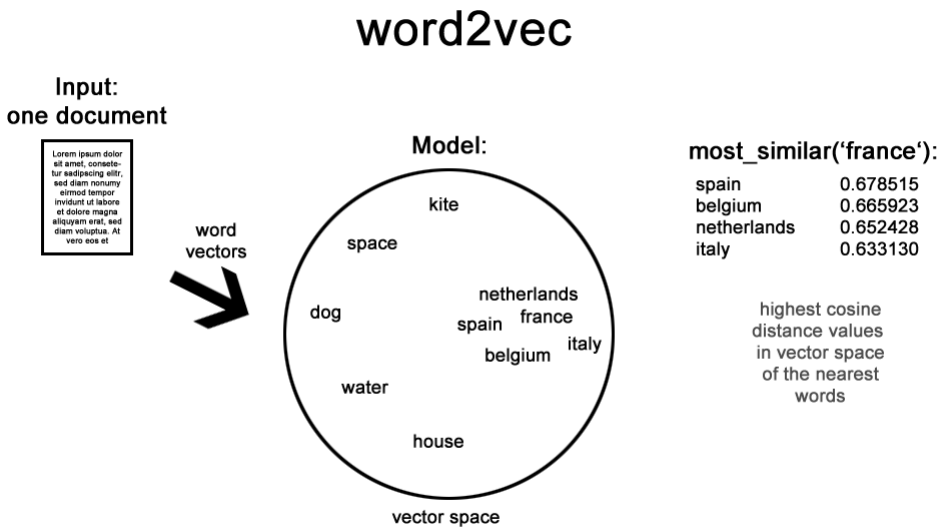
Das Ergebnis der Wahrscheinlichkeit:

	Er	möchte	eine	Reise	nach	Mond	kaufen
Er	8	1087	0	13	0	0	0
möchte	1560	0	786	0	475	8	1328
eine	0	1342	0	860	340	0	102
Reise	0	450	2	0	1048	2	52
nach	2	40	0	1380	0	120	1
Mond	9	2	17	0	0	0	1
Kaufen	0	1509	1327	0	650	0	0

$$P(T)= P(W_1)P(W_2|W_1)P(W_3|W_2)\ldots P(W_n|W_{n-1})$$

WORD2VEC

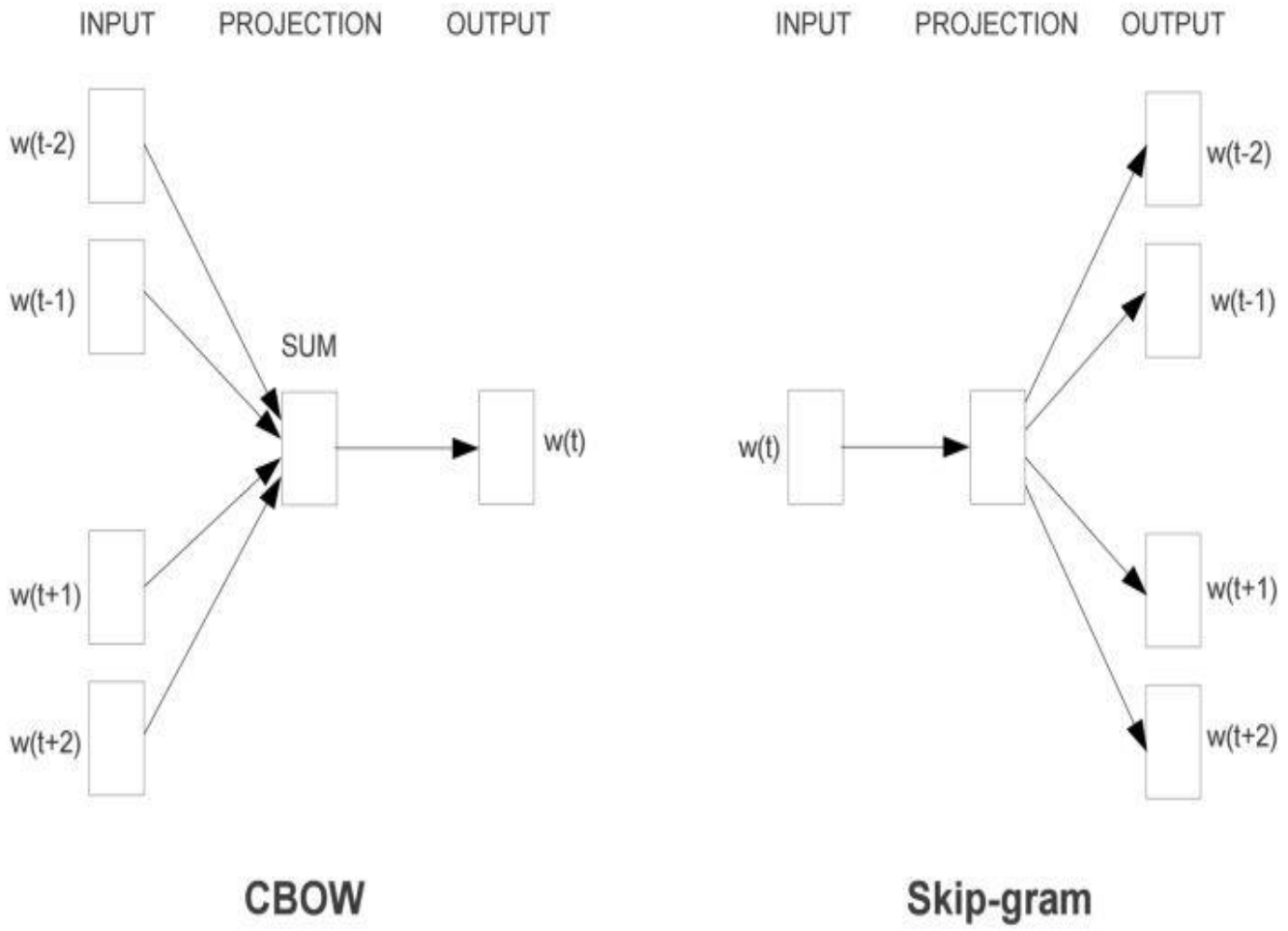
- Wort in numerische Form umgewandelt werden.
- Eine Abbildung von f (Wort) -> Vektor konstruieren
- Hier: f(x) ist Word2Vec



http://picload.org/image/paagcga/word2vec.png

WORD2VEC

Zwei bekannte Modelle:



Quelle: https://i.stack.imgur.com/O2YeO.png

STATISITISCHER ANSATZ

Angenommen:

Vokabular V = 2,000

Unter Bigram:

die Möglichkeit von N-Gramm ist 4,000,000.

Unter Trigramm:

die Möglichkeit von N-Gramm ist 8,000,000,000.

➢Bigram und Trigramm werden häufig verwendet.

SKIP-GRAMM

- eine Verallgemeinerung von N-Grammen.
- bei denen die Komponenten (typischerweise Wörter) im betrachteten Text nicht aufeinanderfolgen müssen, aber Lücken hinterlassen können, die übersprungen werden.
- Beispiel:

Das ist ein Test für das Experiment

SKIP-GRAMM

- eine Verallgemeinerung von N-Grammen.
- bei denen die Komponenten (typischerweise Wörter) im betrachteten Text nicht aufeinanderfolgen müssen, aber Lücken hinterlassen können, die übersprungen werden.
- Beispiel:

  Das ist ein Test für das Experiment. x = Test, y = Experiment

SKIP-GRAMM

- eine Verallgemeinerung von N-Grammen.
- bei denen die Komponenten (typischerweise Wörter) im betrachteten Text nicht aufeinanderfolgen müssen, aber Lücken hinterlassen können, die übersprungen werden.
- Beispiel:

  Das ist ein Test für das Experiment. x = Test, y = Experiment

  F(x)->y Berechnung der Wahrscheinlichkeit, dass „Experiment" im Umfeld ist, wenn „Test" gegeben ist.

SKIP-GRAMM

- eine Verallgemeinerung von N-Grammen.
- bei denen die Komponenten (typischerweise Wörter) im betrachteten Text nicht aufeinanderfolgen müssen, aber Lücken hinterlassen können, die übersprungen werden.
- Beispiel:

  Das ist ein Test für das Experiment. x = Test, y = Experiment

Wie soll man F(Test)-> Experiment berechnen?

VERFAHREN BEI WORT ZU VEKTOR

One-hot Representation:

- Jedes Wort entspricht einer Stelle in einem Vektor.
- Diese Stelle ist mit 1 gekennzeichnet, während alles anderen 0 sind.
- Eine nummerische ID in Hash-Tabelle gespeichert.

Beispiel:

Test: [000010000000…]

Experiment: [100000000000…]

Problem:

Das „Wortlücken“ –Phänomen sind isoliert.

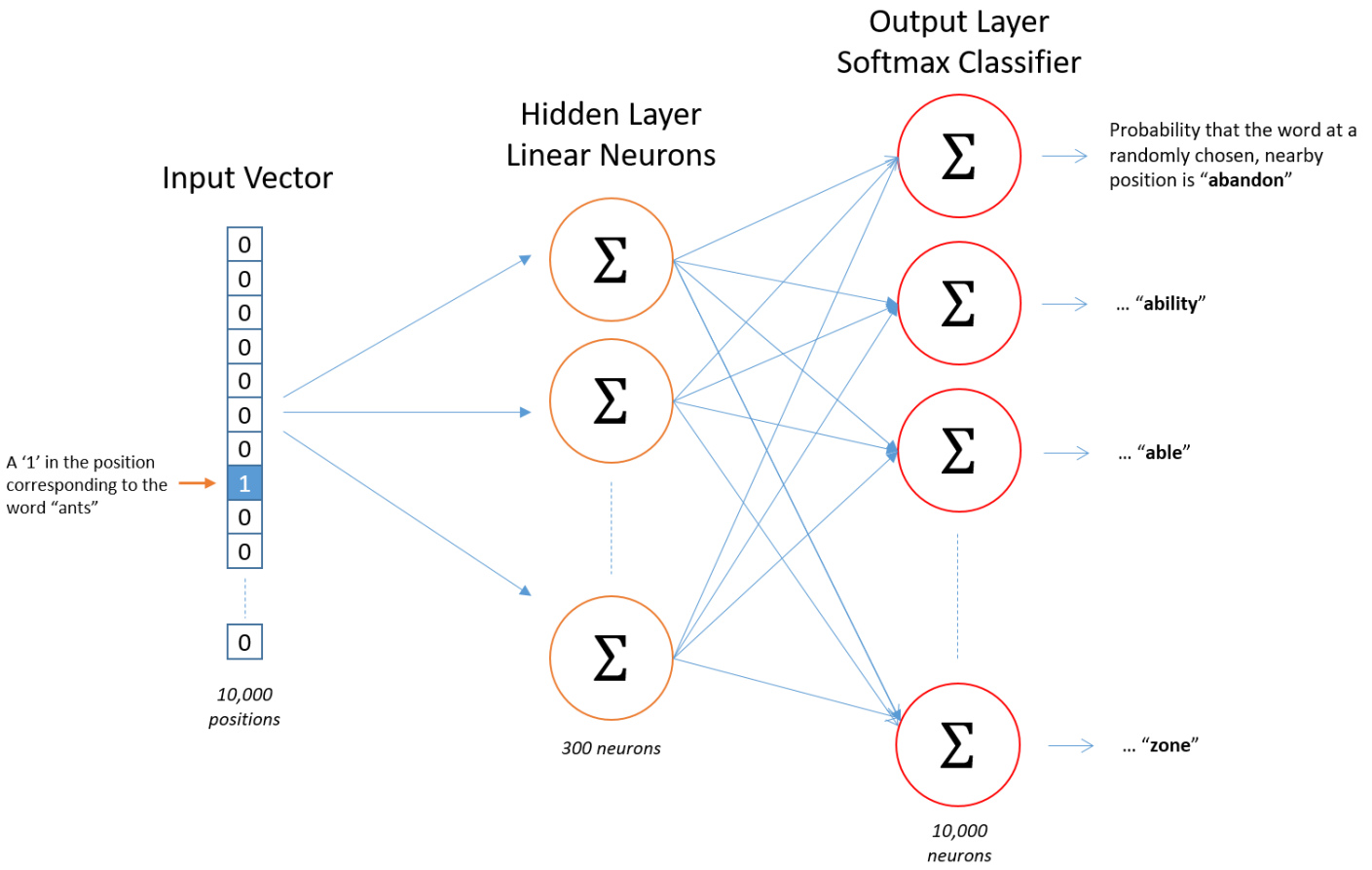
MODEL DATAILS

Annahme:

- Vokabular insgesamt: 10,000 Wörter.
- Input Wort z.B. x= Test, hat folgenden one-Hot Vektor:

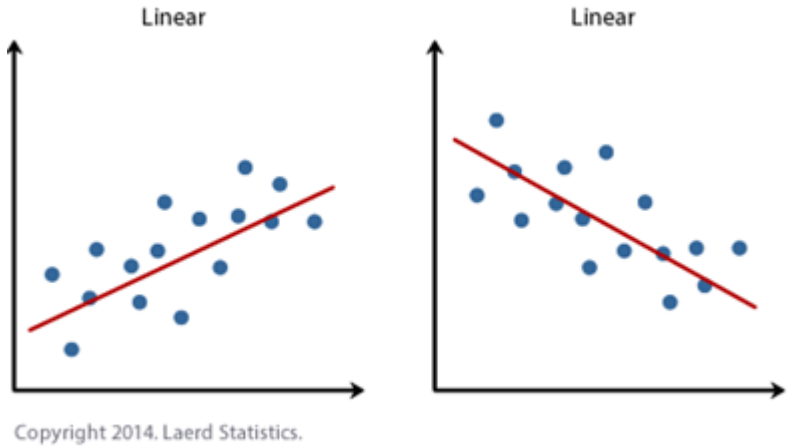
- Der Vektor hat 10,000 Komponenten.
- Output des Netzwerks beinhaltet die Wahrscheinlichkeit jedes Wortes in der Vokabular, die zufällig neben dem Wort ausgewählt worden ist.

ARCHITEKTUR VON SKIP-GRAM

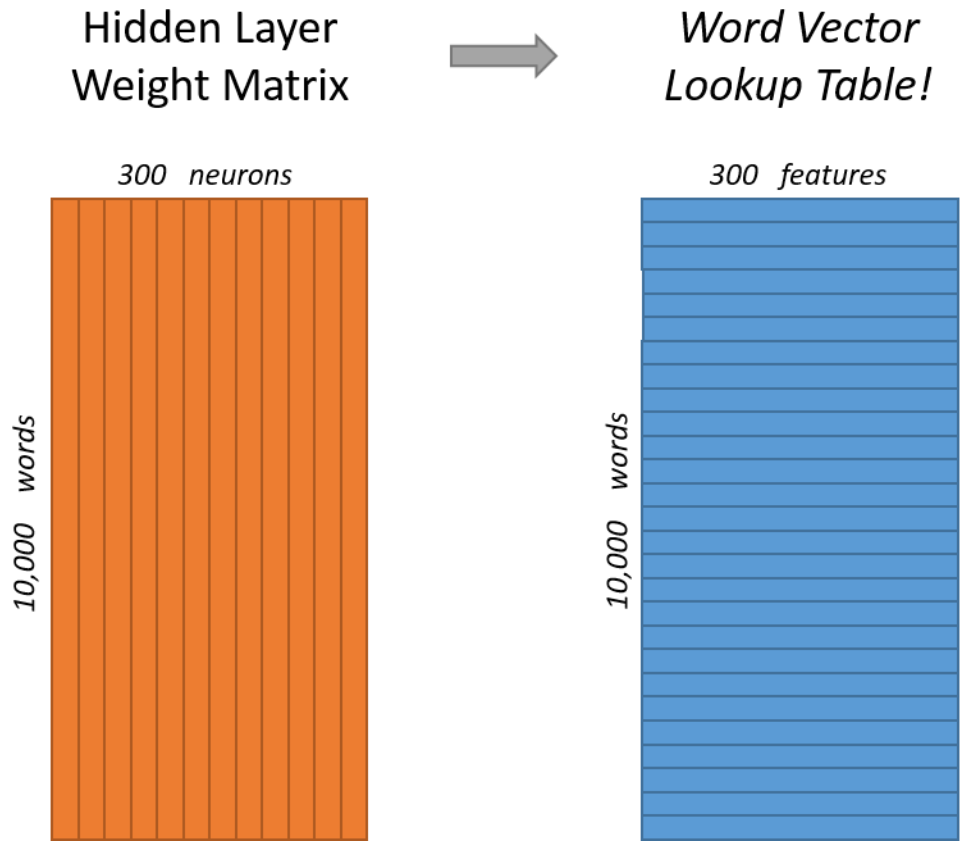


Quelle: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

MEHR DETAILS ZU HIDDEN LAYER



Quelle:
https://statistics.laerd.com/spss-tutorials/img/lr/linear-nonlinear-relationships.png



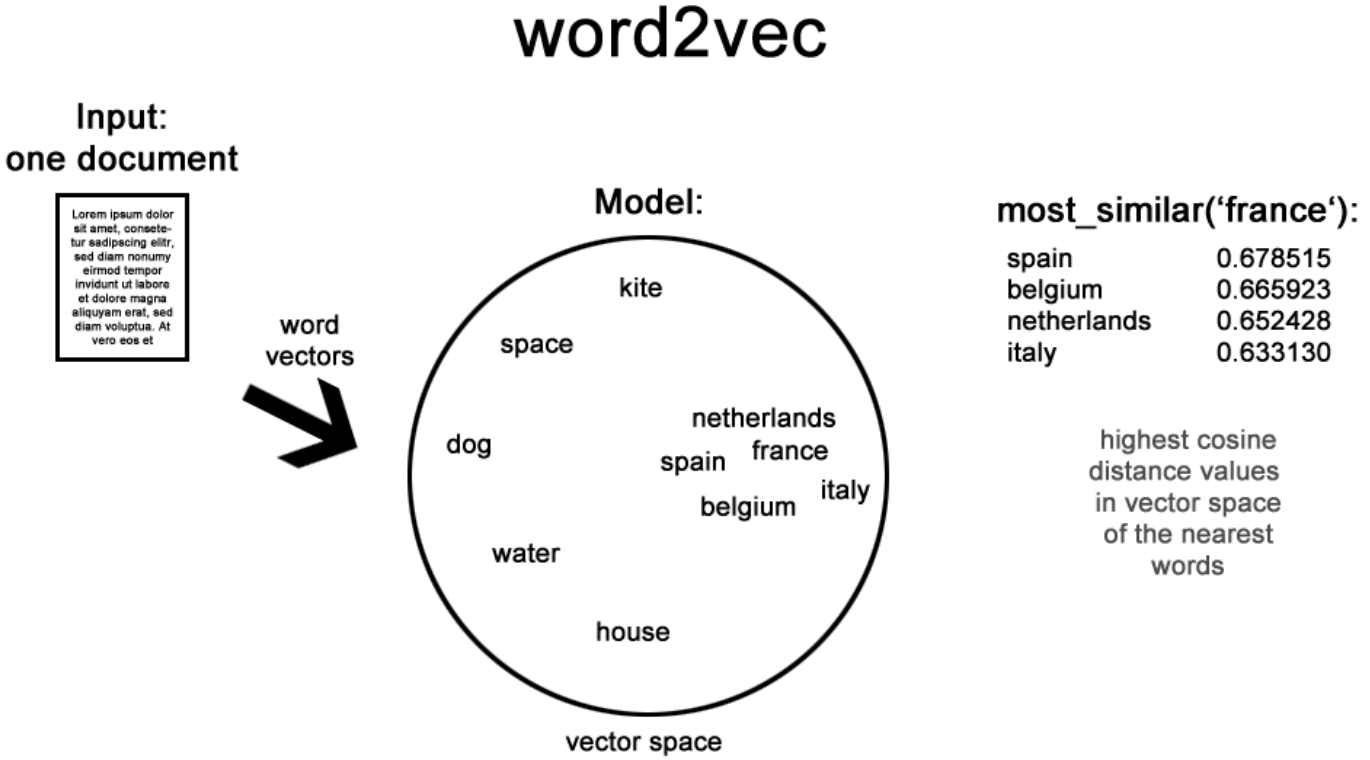
Quelle:
http://mccormickml.com/assets/word2vec/word2vec_weight_matrix_lookup_table.png

MEHR DETAILS ZUM HIDDEN LAYER

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

Quelle: http://mccormickml.com/assets/word2vec/matrix_mult_w_one_hot.png

ERGEBNIS



http://picload.org/image/paagcga/word2vec.png

DISTRIBUTED REPRESENTATION

- Durch die Modellierung wird jedes Wort mit seinem Kontext verknüpft und durch Training werden die Parameter optimiert. Eingaben werden nicht optimiert!
- Eine Verlustfunktion wird minimiert. z.B. Gradient Descent.
- Wird erhalten die Lookup Tabelle für unsere Wörter als Gewichte für den hidden Layer.
- [0.792, -.177, -0.107, 0.109, -0.542, ...].

ZUSAMMENFASSUNG

- Definition von NLP
- Anwendungsbereiche
- Mögliche Verfahren bei NLP:
  - N-gramm
  - Word2Vec

LITERATUR

- https://de.ryte.com/wiki/Natural_Language_Processing
- http://web.stanford.edu/class/cs224n/syllabus.html
- http://www.bigdataway.net/node/11026
- https://en.wikipedia.org/wiki/Natural-language_processing
- https://www.xenonstack.com/blog/overview-of-artificial-intelligence-and-role-of-natural-language-processing-in-big-data
- https://zhuanlan.zhihu.com/p/32829048
- https://www.zhihu.com/question/44832436
- http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
- https://code.google.com/archive/p/word2vec/