UNIVERSITÄT ZU KÖLN

Computational Linguistics: Big Picture, Part 2

Einführung in die Informationsverarbeitung

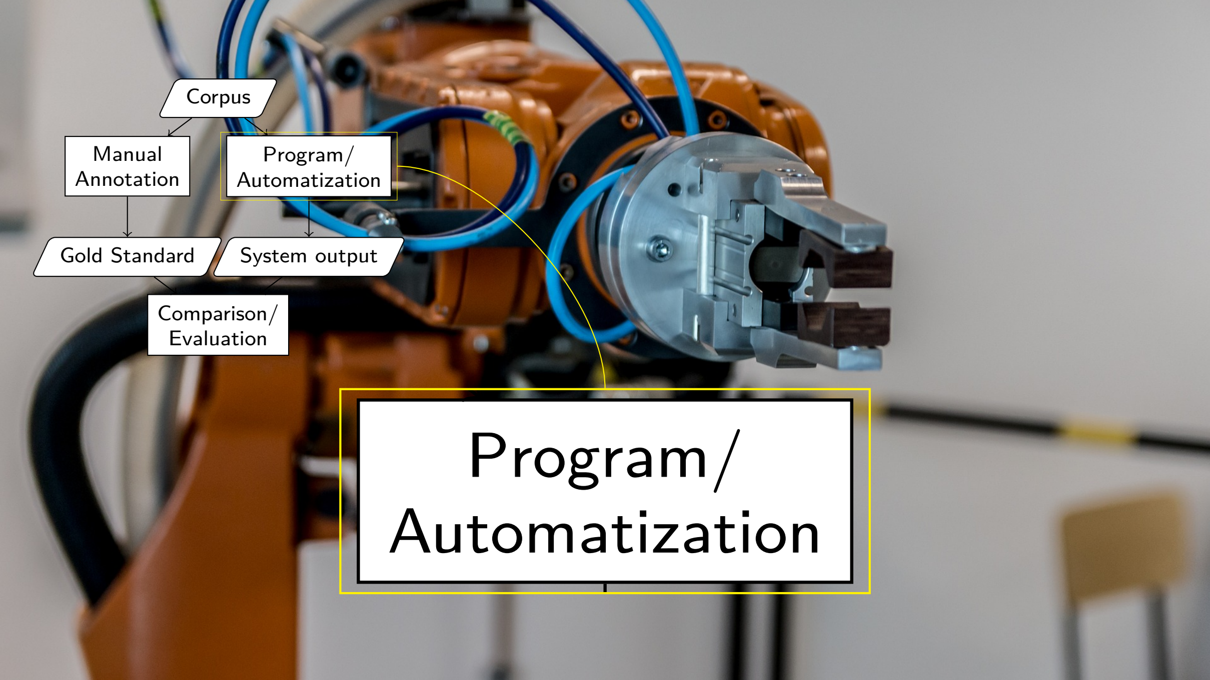
Nils Reiter

October 26, 2023

INSTITUT FÜR DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Section 1

Computational Linguistics


Corpus
Manual Annotation
Program/ Automatization
Gold Standard
System output
Comparison/ Evaluation
Program/ Automatization

Systems

= Programs, models, ...

- Predict annotations
- Idealy: The same annotations as a human (the ‘correct’ ones)

Systems

= Programs, models, ...

- Predict annotations
- Idealy: The same annotations as a human (the 'correct' ones)

Types

- Rule-based (popular 1950s – 2010s)
- Statistical (popular 2005 – 2015)
  - Now often called 'classical machine learning'
- Neural (popular since 2015)
  - Also often called 'deep learning'

Rule-based Systems

- Manually specified rules over certain criteria
  - e.g., HPSG grammar
- Criteria: Vocabulary from which rules are created
  - e.g., Noun: Every token, that starts with an upper case letter
  - e.g., Noun: Every token, that starts with an upper case letter and is not sentence initial

Supervised Systems

- Classification: Associate items with previously known categories
- Learn patterns from annotated data ($=$ training data)
- Relations between input (represented as a vector of feature values, X) and output (Y)
  - Can be an n-to-m relation, but mostly n-to-1 (i.e., we predict a single target category)

Features

- The properties of a item that is to be classified
- Classical machine learning
  - Manual coding of explicit, scientifically validated features: Feature extraction
  - “Translation” of the corpus into feature vectors
  - Feature engineering
  - Design and implementation of feature extractors
  - Linguistic features need to be determined somehow
    $\rightarrow$ Dependencies, modularization

Features

- The properties of a item that is to be classified
- Classical machine learning
  - Manual coding of explicit, scientifically validated features: Feature extraction
  - "Translation" of the corpus into feature vectors
  - Feature engineering
  - Design and implementation of feature extractors
  - Linguistic features need to be determined somehow
    $\rightarrow$ Dependencies, modularization
- Deep learning
  - Embeddings used as features
    - A word is mapped onto an n-dimensional vector, which is then put into the ML system
    - Vector dimensions = features
    - But not so helpful anymore

Example: Parts of Speech

Features	Data type
Case	Binary
Length	> 0

Table: Features

Token	Case	L.
Der	u	3
Hund	u	4
bellt	l	5
.	?	1
Die	u	3
Katze	u	5
schnurrt	l	8
.	?	1

Table: Feature extraction

Example: Parts of Speech

Feature	Data type
Case	Binary
Length	> 0
Sentence initial	Binary

Table: Features

Token	Case	L.	S. initial
Der	u	3	Y
Hund	u	4	N
bellt	l	5	N
.	Jein	?	N
Die	u	3	Y
Katze	u	5	N
schnurrt	l	8	N
.	?	1	N

Table: Feature extraction

Example: Parts of Speech

Feature	Data type
Case	Binary
Length	> 0
Sentence initial	Binary

Tabl Features

Introduces dependency!

Token	Case	L.	S. initial
Der	u	3	Y
Hund	u	4	N
bellt	l	5	N
.	Jein	?	N
Die	u	3	Y
Katze	u	5	N
schnurrt	l	8	N
.	?	1	N

Table: Feature extraction

Embeddings

- Classical machine learning
  - Instance is represented by a feature vector
  - Features are humanly interpretable properties of the instance

Embeddings

- Classical machine learning
  - Instance is represented by a feature vector
  - Features are humanly interpretable properties of the instance
- Embeddings
  - Instance is represented by a vector in an n-dimensional space
  - Mapping of instance to vector is learned, i.e., part of the training process

Embeddings

- Classical machine learning
  - Instance is represented by a feature vector
  - Features are humanly interpretable properties of the instance
- Embeddings
  - Instance is represented by a vector in an n-dimensional space
  - Mapping of instance to vector is learned, i.e., part of the training process

Example (Vector for “köln”)

0.0539 -0.0030 0.0203 -0.1084 -0.0099 0.0705 -0.0546 -0.0433 -0.0096 0.0561 -0.0095 0.0280 0.1726 0.0190 0.0369 0.0217 -0.0002 -0.0309 0.0347 -0.0749
-0.0202 0.0151 -0.0195 0.0001 0.0232 0.0243 -0.0170 -0.0090 -0.0108 -0.0943 0.0376 0.1118 -0.0324 0.0148 -0.0033 0.0537 -0.0681 -0.0733 -0.0201 -0.0329
0.1242 0.0324 -0.0744 -0.0149 -0.0047 -0.0484 -0.0483 0.0481 0.0107 0.0101 -0.0704 0.0500 0.0112 -0.0227 0.0499 -0.0259 -0.0441 0.0712 -0.0157 -0.1271
0.0407 -0.0495 -0.0359 0.0202 0.0024 0.0764 0.0196 0.0267 -0.0117 0.0026 0.0171 -0.0121 -0.1374 -0.0370 0.0247 -0.0113 -0.0094 0.0322 -0.0347 -0.0866 0.0042
-0.0014 0.0067 0.0591 0.0009 0.0085 0.0310 0.0479 -0.0511 0.0198 -0.0886 -0.0274 -0.1364 0.0322 -0.1638 -0.0689 0.0016 -0.1039 0.0059 0.0757 -0.0034 0.1013
-0.0034 -0.0065 -0.0468 0.1577 -0.0065 -0.0478 -0.0004 0.0682 0.0045 -0.0607 -0.0590 0.0343 0.0036 -0.1014 -0.0136 -0.0063 0.0801 0.0360 0.0579 -0.0039
0.0975 0.0500 -0.0558 -0.0095 0.0057 -0.0246 0.1070 -0.0186 0.0669 -0.0781 -0.0569 -0.1286 -0.0834 0.0106 -0.0672 -0.0205 0.0613 0.0290 -0.0545 -0.0481
-0.0882 -0.0489 0.0622 -0.0730 -0.0192 -0.0415 -0.0287 0.0218 -0.0427 -0.0046 0.0255 -0.1164 0.0077 -0.0546 -0.0786 0.0000 -0.0456 0.0943 0.0157 -0.0117
-0.0441 -0.0015 -0.0556 -0.0508 0.0088 0.0418 0.0030 -0.1450 -0.0663 0.0800 0.0172 -0.0289 0.1178 -0.0973 0.0888 0.0637 -0.0295 0.0212 0.0100 -0.0860 0.0035
0.0730 0.0425 -0.0080 0.0885 -0.0166 -0.0765 0.0004 -0.0118 0.0138 -0.0093 -0.0606 -0.0447 -0.0746 0.0131 -0.0447 -0.0763 0.0032 0.1181 0.0542 0.0431
-0.0273 0.0547 0.0135 0.0006 -0.0241 -0.0418 0.0278 -0.0821 -0.0572 -0.0039 0.0214 -0.0196 0.0449 -0.0286 0.0204 0.0681 -0.0901 -0.0266 -0.0287 -0.0874
0.0797 -0.0784 -0.0920 0.0380 0.0411 0.0859 0.0369 0.0595 0.0446 0.0363 -0.0353 -0.0044 -0.0061 0.1134 0.1420 -0.0026 -0.0013 0.0033 0.0508 0.0096 -0.0757
0.0085 -0.0099 -0.0384 0.0218 -0.0259 -0.0112 -0.0212 0.0273 0.0532 -0.0278 -0.0634 0.0317 -0.0022 0.0882 -0.0240 0.0031 -0.0370 0.0747 -0.0097 -0.0315
0.0405 0.0124 -0.1416 -0.0768 0.0363 -0.1248 -0.0134 0.0702 -0.0905 -0.0387 0.0683 -0.0784 0.0886 0.0640 0.0611 -0.0199 -0.0447 -0.1331 -0.1247 0.0540
0.0499 -0.0212 -0.0544 -0.1161 -0.0729 0.0894 0.0532 0.0164 -0.0039 -0.0108 -0.0248 -0.1021 -0.0549 -0.0318 0.0309 -0.0691

Machine Learning

Language Modeling

- One of the oldest NLP tasks
  - Long before predictive typing on smart phones became a thing
  - Long before "large language models" became a thing
- Language model (LM) predicts the next word, given previous words (history)
- Formally: $p(\text{word}|\text{history})$

Beispiel

Maria hat an der Universität zu Köln ______.

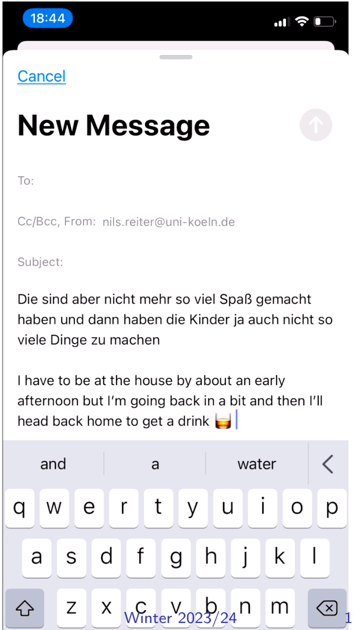
Machine Learning

Language Modeling

- One of the oldest NLP tasks
  - Long before predictive typing on smart phones became a thing
  - Long before "large language models" became a thing
- Language model (LM) predicts the next word, given previous words (history)
- Formally: $p(\text{word}|\text{history})$

Beispiel

Maria hat an der Universität zu Köln ______.



Machine Learning

From Language Models to Large Language Models

$$\boxed{p(w_n|w_1, w_2, \ldots)}$$

Machine Learning

From Language Models to Large Language Models

$$\boxed{p(w_n | w_1, w_2, \ldots)} \longrightarrow \boxed{p\left(w_n \middle| \begin{bmatrix} w_1 \\ \text{wortart}_1 \\ \vdots \end{bmatrix}, \begin{bmatrix} w_2 \\ \text{wortart}_2 \\ \vdots \end{bmatrix}, \ldots\right)}$$

Linguistische Merkmale des Kontextes

Machine Learning

From Language Models to Large Language Models

$$\boxed{p(w_n | \color{red}{w_1}, w_2, \ldots)}$$

$$p\left(w_n \middle| \color{red}{\begin{bmatrix} w_1 \\ \text{wortart}_1 \\ \vdots \end{bmatrix}}, \begin{bmatrix} w_2 \\ \text{wortart}_2 \\ \vdots \end{bmatrix}, \cdots \right)$$

Linguistische Merkmale des Kontextes

$$p\left(w_n \middle| \color{red}{\begin{bmatrix} 0.105 \\ -0.145 \\ 0.742 \\ \vdots \end{bmatrix}}, \begin{bmatrix} -0.427 \\ 0.864 \\ -0.232 \\ \vdots \end{bmatrix}, \cdots \right)$$

Kontextmerkmale aus Daten ‘gelernt’
(Mikolov et al., 2013)

Machine Learning

From Language Models to Large Language Models

$$p(w_n | {\color{red}w_1}, w_2, \dots)$$

$$p\left(w_n \middle| {\color{red}\begin{bmatrix} w_1 \\ \text{wortart}_1 \\ \vdots \end{bmatrix}}, \begin{bmatrix} w_2 \\ \text{wortart}_2 \\ \vdots \end{bmatrix}, \dots \right)$$

Linguistische Merkmale des Kontextes

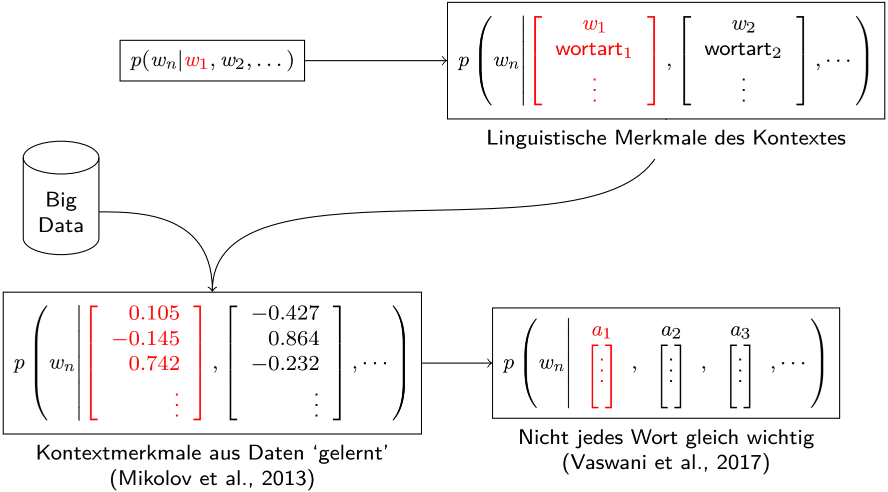
Big Data

$$p\left(w_n \middle| {\color{red}\begin{bmatrix} 0.105 \\ -0.145 \\ 0.742 \\ \vdots \end{bmatrix}}, \begin{bmatrix} -0.427 \\ 0.864 \\ -0.232 \\ \vdots \end{bmatrix}, \dots \right)$$

Kontextmerkmale aus Daten ‘gelernt’
(Mikolov et al., 2013)

Machine Learning

From Language Models to Large Language Models



Machine Learning

From Language Models to Large Language Models

$$p(w_n | {\color{red}w_1}, w_2, \ldots)$$

$$p\left(w_n \middle| {\color{red}\begin{bmatrix} w_1 \\ \text{wortart}_1 \\ \vdots \end{bmatrix}}, \begin{bmatrix} w_2 \\ \text{wortart}_2 \\ \vdots \end{bmatrix}, \ldots\right)$$

Linguistische Merkmale des Kontextes

Big Data

‘neural revolution’

$$p\left(w_n \middle| {\color{red}\begin{bmatrix} 0.105 \\ -0.145 \\ 0.742 \\ \vdots \end{bmatrix}}, \begin{bmatrix} -0.427 \\ 0.864 \\ -0.232 \\ \vdots \end{bmatrix}, \ldots\right)$$

Kontextmerkmale aus Daten ‘gelernt’
(Mikolov et al., 2013)

$$p\left(w_n \middle| \overset{\color{red}a_1}{\color{red}\begin{bmatrix} \vdots \end{bmatrix}}, \overset{a_2}{\begin{bmatrix} \vdots \end{bmatrix}}, \overset{a_3}{\begin{bmatrix} \vdots \end{bmatrix}}, \ldots\right)$$

Nicht jedes Wort gleich wichtig
(Vaswani et al., 2017)

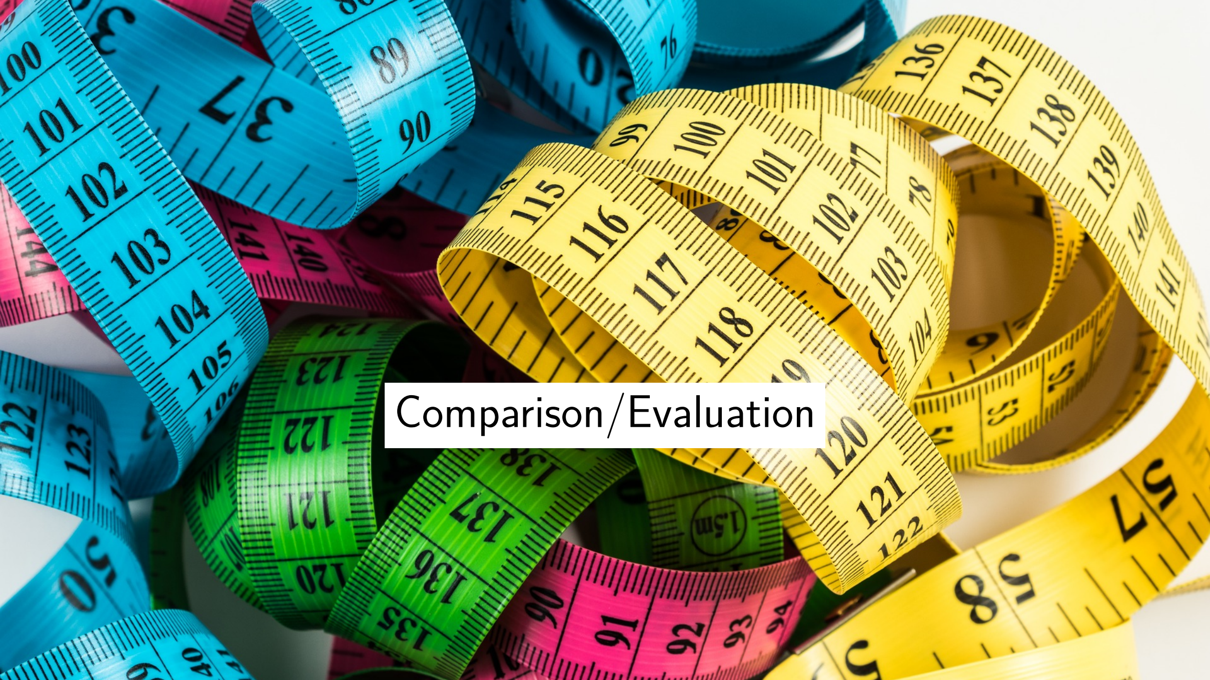
Multiple Ways of Using a Large Language Model

1. Let it generate free text ("prompting")
    - Impressive for lay persons and investors
    - Difficult to evaluate exactly: How to measure if a text is "correct" or "good"?
    - Terms are not defined explicitly, and (presumably) used with their every-day meaning (which is vague)
        - Challenge for sciences with specialised vocabulary

Multiple Ways of Using a Large Language Model

1. Let it generate free text ("prompting")
   - Impressive for lay persons and investors
   - Difficult to evaluate exactly: How to measure if a text is "correct" or "good"?
   - Terms are not defined explicitly, and (presumably) used with their every-day meaning (which is vague)
     - Challenge for sciences with specialised vocabulary
2. Fine-tune it to a specific task and let it do classification
   - Requires explicitly labeled training data
   - More labor-intensive
   - Model outputs probabilities for classes, straightforward evaluation
     - E.g., what's the percentage of correct predictions

Comparison / Evaluation



Evaluation

Intrinsic

- Compare the automatically produced annotations with the gold standard
- Can be quantified (similar to inter-annotator agreement)
- Accuracy:
  $X\%$ of the items are classified as they are in the gold standard
- Other metrics: *precision*, *recall*, *f-score*

Evaluation

Intrinsic

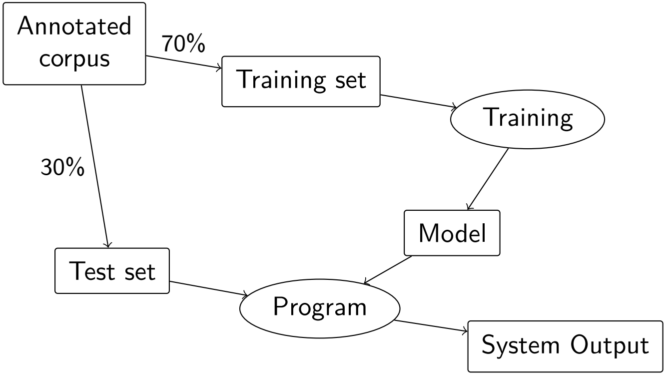
- Compare the automatically produced annotations with the gold standard
- Can be quantified (similar to inter-annotator agreement)
- Accuracy:
  $X\%$ of the items are classified as they are in the gold standard
- Other metrics: *precision*, *recall*, *f-score*

Extrinsic

- Use of the program in another program that can be evaluated
  - *downstream tasks*
  - e.g., use of a PoS tagger in a machine translation system

Intrinsic Evaluation

- Goal: Predict the quality on new data
- The program cannot have seen the data, so that it's a realistic test



Summary

Summary

- Experimental science
  - Real-world applications, from smart assistants over machine translation to text generation
- Annotations as coded, machine-readable ‘nature’
  - Annotation decisions based on annotation guidelines
  - Inter-annotator agreement for validation
- Programs to automatically detect/categorize linguistic phenomena
  - Rule-based: Manually crafted rules between input and output
  - Classical machine learning: Computer learns rules based on manually crafted features
  - Deep learning: Computer also learns its own input representation
- Evaluation: Knowing how well something works
  - Evaluation scores
  - Training and test split

References I

Mikolov, Tomáš/Ilya Sutskever/Kai Chen/Greg S Corrado/Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges/L. Bottou/M. Welling/Z. Ghahramani/K. Q. Weinberger. Curran Associates, Inc., pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Vaswani, Ashish/Noam Shazeer/Niki Parmar/Jakob Uszkoreit/Llion Jones/Aidan N. Gomez/Lukasz Kaiser/Illia Polosukhin (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].