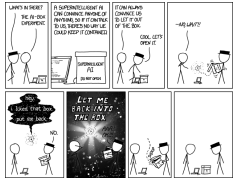


Advanced Natural Language Processing & Large Language Models

Felix Neubürger

2025

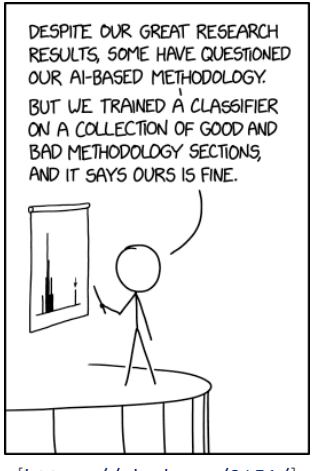
Fachhochschule Südwestfalen, Ingenieurs- & Wirtschaftswissenschaften

Inhalte der Vorlesung

- Wie funktioniert Natural Language Processing
- Sprachdarstellung zum Rechnen
- Attentionmechanismus
- Transformerarchitektur
- von BERT zu DeepSeek-v3
- Wie es weitergehen kann
- Nutzungsmöglichkeiten: RAG, Agentensysteme
- AI Safety und Ethik

Ziele der Vorlesung - Welche Fragen sollen beantwortet werden?

- Was sind die Grundlagen von Natural Language Processing (NLP)?
- Wie funktionieren Attention-Mechanismen und warum sind sie wichtig?
- Was ist die Transformer-Architektur und wie unterscheidet sie sich von anderen Ansätzen?
- Wie werden Sprachmodelle wie BERT und GPT trainiert und genutzt?
- Welche Herausforderungen und ethischen Fragen gibt es bei der Nutzung von LLMs?
- Welche praktischen Anwendungen und Zukunftsperspektiven gibt es für LLMs?



[https://xkcd.com/2451/]

Format der Vorlesung - Wie sollen diese Fragen beantwortet werden?

- Theroretischer Teil mit Folien
- Praktischer Teil in Gruppen an einem Projekt
- Gruppengröße 2 oder 3 Personen
- Einzelarbeit möglich wenn eigenes Thema vorhanden
- Abgabe der Ausarbeitung einen Tag vor der Veranstaltung in der Blockwoche
- Vorstellung der Projektergebnisse in der Blockwoche
- Gewichtung der Bewertung Projektausarbeitung (50%) und Vortrag (50%)

[https://xkcd.com/1425/]

Wie funktioniert Natural Language Processing

- Definition und Ziele des NLP
- Herausforderungen bei der maschinellen Sprachverarbeitung
- Anwendungen von NLP in der Praxis

Definition und Ziele des NLP

- NLP steht für Natural Language Processing, die Verarbeitung natürlicher Sprache durch Computer.
- Ziel: Maschinen ermöglichen, menschliche Sprache zu verstehen, zu interpretieren und zu generieren.
- Anwendungen: Übersetzungen, Chatbots, Textanalyse, Sprachassistenten.

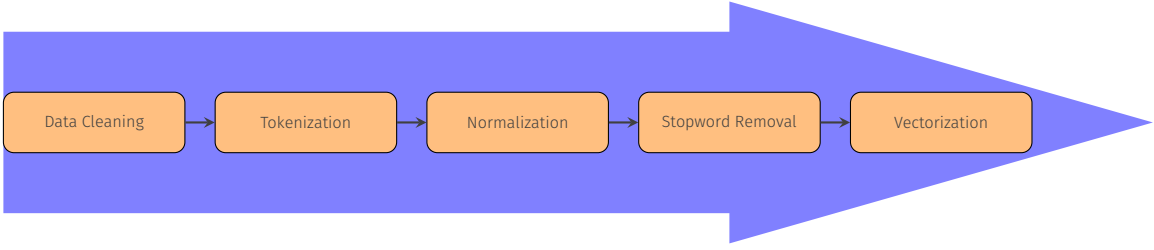
Herausforderungen bei der maschinellen Sprachverarbeitung

- Ambiguität: Mehrdeutigkeit in der Sprache.
- Kontextabhängigkeit: Bedeutung hängt vom Kontext ab.
- Umgang mit Synonymen und Homonymen.
- Verarbeitung großer Datenmengen und Rechenaufwand.

Anwendungen von NLP in der Praxis

- Sentiment-Analyse: Erkennung von Meinungen in Texten.
- Maschinelle Übersetzung: Automatische Übersetzung zwischen Sprachen.
- Sprachgesteuerte Assistenten: Siri, Alexa, Google Assistant.
- Textzusammenfassung: Automatische Erstellung von Textzusammenfassungen.

Text Preprocessing Pipeline



Data Cleaning

- **Definition:** Entfernen oder Korrigieren von fehlerhaften, unvollständigen oder irrelevanten Daten.
- **Schritte:**
  - Entfernen von Sonderzeichen, HTML-Tags und Emojis.
  - Korrektur von Rechtschreibfehlern.
  - Vereinheitlichung von Groß- und Kleinschreibung.
- **Ziel:** Verbesserung der Datenqualität für nachfolgende Verarbeitungsschritte.

Tokenization

- **Definition:** Zerlegung von Text in kleinere Einheiten (Tokens), z. B. Wörter oder Satzzeichen.
- **Arten:**
  - Wortbasierte Tokenization: "Das ist ein Satz." → ["Das", "ist", "ein", "Satz", "."]
  - Zeichenbasierte Tokenization: "Hallo" → ["H", "a", "l", "l", "o"]
  - Subwortbasierte Tokenization: "unbelievable" → ["un", "believ", "able"]
- **Herausforderungen:** Umgang mit zusammengesetzten Wörtern, Abkürzungen und Sonderzeichen.

Normalization

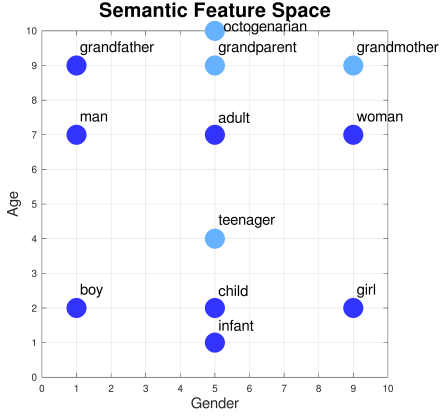
- **Definition:** Vereinheitlichung von Textdaten, um Konsistenz zu gewährleisten.
- **Schritte:**
  - Umwandlung in Kleinbuchstaben: ”Haus” → ”haus”.
  - Entfernen von Akzenten: ”café” → ”cafe”.
  - Stemming: Reduktion auf Wortstamm, z. B. ”running” → ”run”.
  - Lemmatization: Rückführung auf Grundform, z. B. ”better” → ”good”.
- **Ziel:** Reduktion der Variabilität in den Daten.

Stopword Removal

- **Definition:** Entfernen von häufig vorkommenden Wörtern, die wenig Bedeutung tragen (z. B. ”der”, ”und”, ”ist”).
- **Vorgehen:**
  - Verwendung einer vordefinierten Stopword-Liste (z. B. ”der”, ”die”, ”und”, ”ist”, ”ein”, ”zu”).
  - Anpassung der Liste an den spezifischen Anwendungsfall.
- **Vorteile:**
  - Reduktion der Datenmenge.
  - Verbesserung der Modellleistung durch Fokus auf relevante Wörter.
- **Herausforderung:** Manche Stopwords können je nach Kontext wichtig sein.

Sprachdarstellung zum Rechnen

- Wortvektoren und Einbettungen (Embeddings)
- One-Hot-Encoding vs. verteilte Repräsentationen
- Word2Vec, GloVe und andere Einbettungsmethoden



Wortvektoren und Einbettungen (Embeddings)

- Ziel: Repräsentation von Wörtern in einem kontinuierlichen Vektorraum.
- Mathematische Definition:
  - Gegeben eine Menge von Wörtern $W = \{w_1, w_2, \ldots, w_n\}$.
  - Eine Einbettung ist eine Funktion $f : W \to \mathbb{R}^d$, wobei d die Dimension des Vektorraums ist.
  - Beispiel: $f(w_i) = \mathbf{v}_i \in \mathbb{R}^d$.
- Vorteile:
  - Semantische Ähnlichkeit wird durch Nähe im Vektorraum dargestellt.
  - Reduktion der Dimensionalität im Vergleich zu One-Hot-Encoding.

One-Hot-Encoding vs. Verteilte Repräsentationen

- **One-Hot-Encoding:**
  - Jedes Wort wird als Vektor mit einer einzigen Eins und sonst Nullen dargestellt.
  - Beispiel: Für $W = \{w_1, w_2, w_3\}$, w_2 wird als $[0, 1, 0]$ kodiert.
  - Nachteile: Hohe Dimensionalität, keine semantische Information.
- **Verteilte Repräsentationen:**
  - Nutzen kontinuierliche Vektorräume, um semantische Beziehungen darzustellen[1]..
  - Ermöglichen die Nutzung von Modellen wie Word2Vec und GloVe [2].
  - Wörter werden als dichte Vektoren in einem "niedrig"dimensionalen Raum dargestellt.
  - Semantisch ähnliche Wörter haben ähnliche Vektoren.
  - Beispiel: $f(w_1) = [0.2, 0.8]$, $f(w_2) = [0.3, 0.7]$.

[1] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[2] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Word2Vec, GloVe und andere Einbettungsmethoden

- **Word2Vec:**
  - Skip-Gram-Modell: Vorhersage des Kontexts basierend auf einem Zielwort[3].
  - CBOW-Modell: Vorhersage des Zielworts basierend auf dem Kontext[4].
- **GloVe (Global Vectors for Word Representation):**
  - Nutzt globale Wort-Kooccurenz-Matrizen[5].
  - Optimiert eine Zielfunktion, die Wortpaare und ihre Häufigkeiten berücksichtigt[6].
- **Andere Methoden:**
  - FastText: Berücksichtigt Subwortinformationen.
  - BERT-Embeddings: Kontextabhängige Einbettungen.

[3] Das Skip-Gram-Modell versucht, für ein gegebenes Zielwort die umgebenden Kontextwörter vorherzusagen.

[4] Das Continuous Bag of Words (CBOW)-Modell sagt ein Zielwort basierend auf den umgebenden Kontextwörtern vorher. Es ist effizienter als das Skip-Gram-Modell, aber weniger präzise bei seltenen Wörtern.

[5] GloVe basiert auf der Idee, dass die globale Häufigkeit von Wortpaaren in einem Korpus genutzt werden kann, um semantische Beziehungen zwischen Wörtern zu modellieren.

[6] Die Zielfunktion von GloVe minimiert den Unterschied zwischen der inneren Produktdarstellung von Wortvektoren und der logarithmierten Häufigkeit von Wortpaaren.

Neuartige Embeddings (Teil 1)

- **Kontextabhängige Embeddings:**
  - Modelle wie BERT[7], GPT[8] und T5[9] generieren Embeddings, die den Kontext eines Wortes berücksichtigen.
  - Beispiel: Das Wort ”Bank” hat unterschiedliche Embeddings in den Sätzen ”Ich sitze auf der Bank” und ”Ich gehe zur Bank”.
- **Sentence Embeddings:**
  - Repräsentieren ganze Sätze statt einzelner Wörter.
  - Modelle wie Sentence-BERT (SBERT)[10] ermöglichen semantische Suche und Textähnlichkeitsbewertung.

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[8] Brown, T. et al. (2020). Language Models are Few-Shot Learners. https://arxiv.org/abs/2005.14165

[9] Raffel, C. et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://arxiv.org/abs/1910.10683

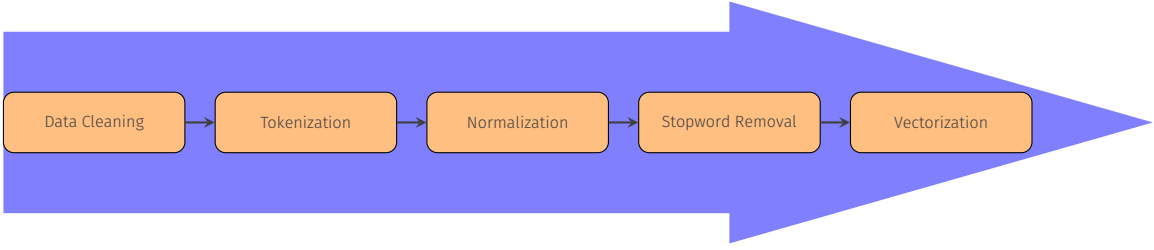
[10] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://arxiv.org/abs/1908.10084

Neuartige Embeddings (Teil 2)

- **Multimodale Embeddings:**
  - Kombinieren Informationen aus verschiedenen Modalitäten wie Text, Bild und Audio.
  - Beispiel: CLIP (Contrastive Language–Image Pretraining)[11] von OpenAI.
- **Graphbasierte Embeddings:**
  - Repräsentieren Wörter als Knoten in einem Graphen, wobei Kanten Beziehungen zwischen Wörtern darstellen.
  - Beispiel: Node2Vec[12] und GraphSAGE[13].
- **Adapter-basierte Embeddings:**
  - Ermöglichen die Anpassung vortrainierter Modelle an spezifische Aufgaben durch leichte Modifikationen.
  - Reduzieren den Speicherbedarf im Vergleich zu vollständigem Fine-Tuning[14].

[11] Radford, A. et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. https://arxiv.org/abs/2103.00020
[12] Grover, A., & Leskovec, J. (2016). node2vec: Scalable Feature Learning for Networks. https://arxiv.org/abs/1607.00653
[13] Hamilton, W. et al. (2017). Inductive Representation Learning on Large Graphs. https://arxiv.org/abs/1706.02216
[14] Houlsby, N. et al. (2019). Parameter-Efficient Transfer Learning for NLP. https://arxiv.org/abs/1902.00751

Text Preprocessing Pipeline



Attention-Mechanismus

- Motivation für Attention in Sequenzmodellen
- Funktionsweise des Attention-Mechanismus
- Unterschied zwischen Self-Attention und Cross-Attention

Motivation für Attention in Sequenzmodellen

- Problem: In langen Sequenzen verlieren Modelle wie RNNs und LSTMs den Überblick über frühere Informationen.
- Lösung: Der Attention-Mechanismus ermöglicht es, gezielt auf relevante Teile der Eingabesequenz zu fokussieren.
- Beispiel: Bei der Übersetzung eines Satzes kann Attention bestimmen, welches Wort im Quelltext für ein bestimmtes Wort im Zieltext wichtig ist.

Funktionsweise des Attention-Mechanismus

- **Gegeben:** Eine Eingabesequenz mit n Elementen $\{x_1, x_2, \ldots, x_n\}$.
- **Ziel:** Berechnung einer gewichteten Summe der Eingaben, wobei die Gewichte die Relevanz jedes Elements darstellen.
- **Schritte:**
  1. **Berechnung der Scores:**

$$e_{ij} = \text{score}(h_i, h_j), \quad \text{wobei } h_i \text{ und } h_j \text{ die Hidden States sind.}$$

  2. **Normalisierung der Scores:**

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \quad \text{(Softmax)}.$$

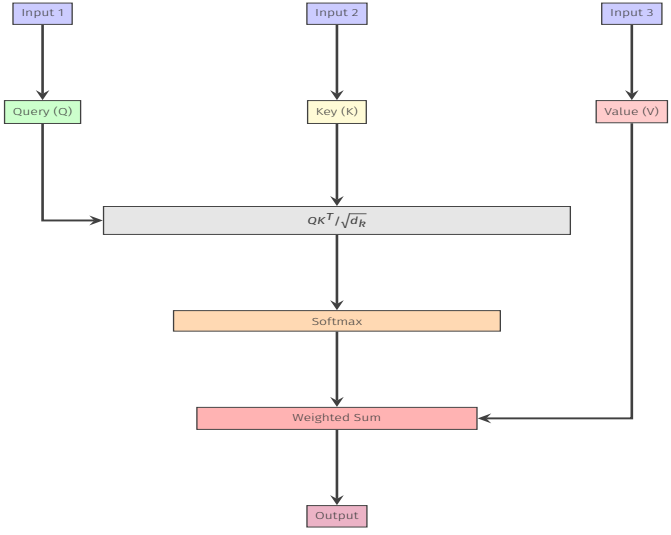
  3. **Gewichtete Summe:**

$$z_i = \sum_{j=1}^{n} \alpha_{ij} h_j.$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK}{\sqrt{d_k}}\right)V \tag{1}$$

Hierbei sind Q (Query), K (Key) und V (Value) Matrizen, und d_k ist die Dimension der Keys.

Funktionsweise des Attention-Mechanismus



Self-Attention vs. Cross-Attention

Self-Attention:

- Jeder Token in der Sequenz bezieht sich auf alle anderen Tokens in derselben Sequenz.
- Beispiel: Kontextualisierung eines Wortes in einem Satz.

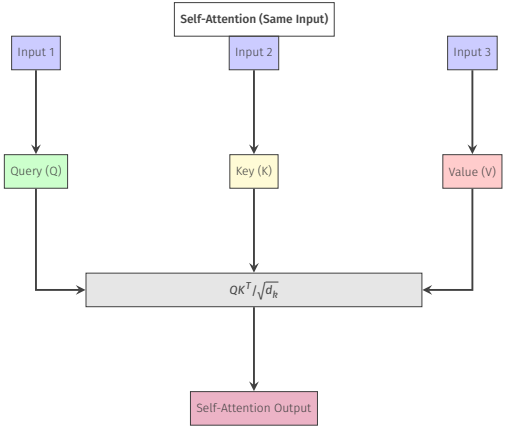
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK}{\sqrt{d_k}}\right)V, \quad Q = K = V$$

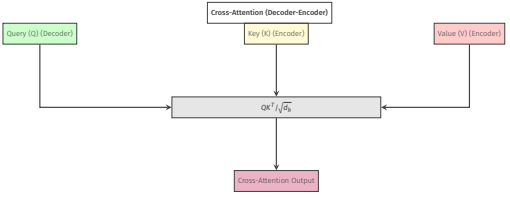
Cross-Attention:

- Tokens in einer Sequenz beziehen sich auf Tokens in einer anderen Sequenz.
- Beispiel: Übersetzung, bei der der Zieltext auf den Quelltext achtet.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK}{\sqrt{d_k}}\right)V, \quad Q \neq K = V$$

Self-Attention vs. Cross-Attention





Visualisierung der Attention-Matrix

- Die Attention-Matrix zeigt die Gewichte α_{ij}, die die Relevanz von Token j für Token i darstellen.
- Beispiel: Bei der Übersetzung eines Satzes zeigt die Matrix, welche Wörter im Quelltext für ein bestimmtes Wort im Zieltext wichtig sind.

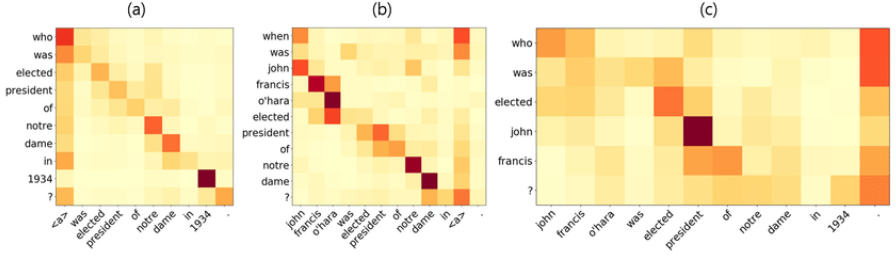


Abbildung: Beispiel einer Attention-Matrix. Kim, Yanghoon & Hwanhee, Lee & Shin, Joongbo & Jung, Kyomin. (2018). Improving Neural Question Generation using Answer Separation. 10.48550/arXiv.1809.02393.

Transformer-Architektur

- Überblick über die Transformer-Architektur
- Encoder-Decoder-Struktur
- Vorteile gegenüber rekurrenten Netzwerken

Von BERT zu DeepSeek-v3

- Einführung in BERT und seine Architektur
- Weiterentwicklungen: GPT, RoBERTa, T5
- Überblick über DeepSeek-v3 und seine Besonderheiten

Wie es weitergehen kann

- Aktuelle Forschungstrends im NLP
- Herausforderungen und offene Fragen
- Ethische Überlegungen und gesellschaftliche Auswirkungen

Nutzungsmöglichkeiten: RAG, Agentensysteme

- Retrieval-Augmented Generation (RAG) und seine Anwendungen
- Entwicklung und Einsatz von Agentensystemen im NLP
- Kombination von LLMs mit externem Wissen

AI Safety und Ethik

- Bedeutung von Sicherheit und Ethik in der KI-Entwicklung
- Risiken und Herausforderungen bei der Nutzung von LLMs
- Ansätze zur Förderung von verantwortungsvoller KI

Bedeutung von Sicherheit und Ethik

- **Sicherheit:** Verhindern von Fehlverhalten oder schädlichem Verhalten durch KI-Systeme.
- **Ethik:** Sicherstellen, dass KI-Systeme fair, transparent und respektvoll gegenüber menschlichen Werten sind.
- **Gesellschaftliche Auswirkungen:** Einfluss von KI auf Arbeitsplätze, Privatsphäre und soziale Gerechtigkeit.

Risiken und Herausforderungen

- **Bias und Diskriminierung:** LLMs können Vorurteile aus Trainingsdaten übernehmen.
- **Desinformation:** Generierung von falschen oder irreführenden Inhalten.
- **Missbrauch:** Einsatz von LLMs für schädliche Zwecke wie Phishing oder Propaganda.
- **Black-Box-Problem:** Mangel an Transparenz und Nachvollziehbarkeit in KI-Modellen.

Ansätze zur Förderung von verantwortungsvoller KI

- **Regulierung und Richtlinien:** Entwicklung von Standards und Gesetzen für den Einsatz von KI.
- **Technische Lösungen:**
  - Bias-Detektion und -Korrektur.
  - Explainable AI (XAI) zur Verbesserung der Transparenz.
- **Bildung und Bewusstsein:** Förderung von Wissen über KI-Sicherheit und Ethik in der Gesellschaft.
- **Zusammenarbeit:** Interdisziplinäre Ansätze zwischen Technik, Recht und Philosophie.

LLM Standardwerke

- **Build LLMs from Scratch** (Raschka)
  - `https://github.com/rasbt/LLMs-from-scratch`
  - Praktische Implementierung in PyTorch
- **Transformers for NLP** (Rothman)
  - ISBN 978-1803247335
  - BERT/GPT Anwendungen

LLM Forschungsarbeiten

- **Attention Is All You Need** (2017)
  - `https://arxiv.org/abs/1706.03762`
  - Transformer-Architektur
- **BERT Paper** (Devlin 2019)
  - `https://arxiv.org/abs/1810.04805`
  - Bidirektionale Pretraining
- **GPT-3 Paper** (Brown 2020)
  - `https://arxiv.org/abs/2005.14165`
  - Few-Shot Learning

LLM Praktische Ressourcen

- **Hugging Face Transformers**
  - `https://github.com/huggingface/transformers`
- **LangChain**
  - `https://python.langchain.com/`
  - LLM Orchestrierung
- **LLaMA & LlamaIndex**
  - `https://github.com/facebookresearch/llama`
  - Open-Weight Modelle

References I