Llama2 – Architektur und Experimente

Monica Riedler

Profilierungsmodul Computerlinguistik I

Dr. Robert Zangenfeind

20.12.2023

Übersicht

1. Einleitung
2. Llama2: Architektur und Training
   - Pretraining
   - Fine-tuning
3. Llama2: Experimente
   - Morphologie: Bestimmung von Wortstrukturen
   - Syntax: Dependenzanalyse
4. Fazit

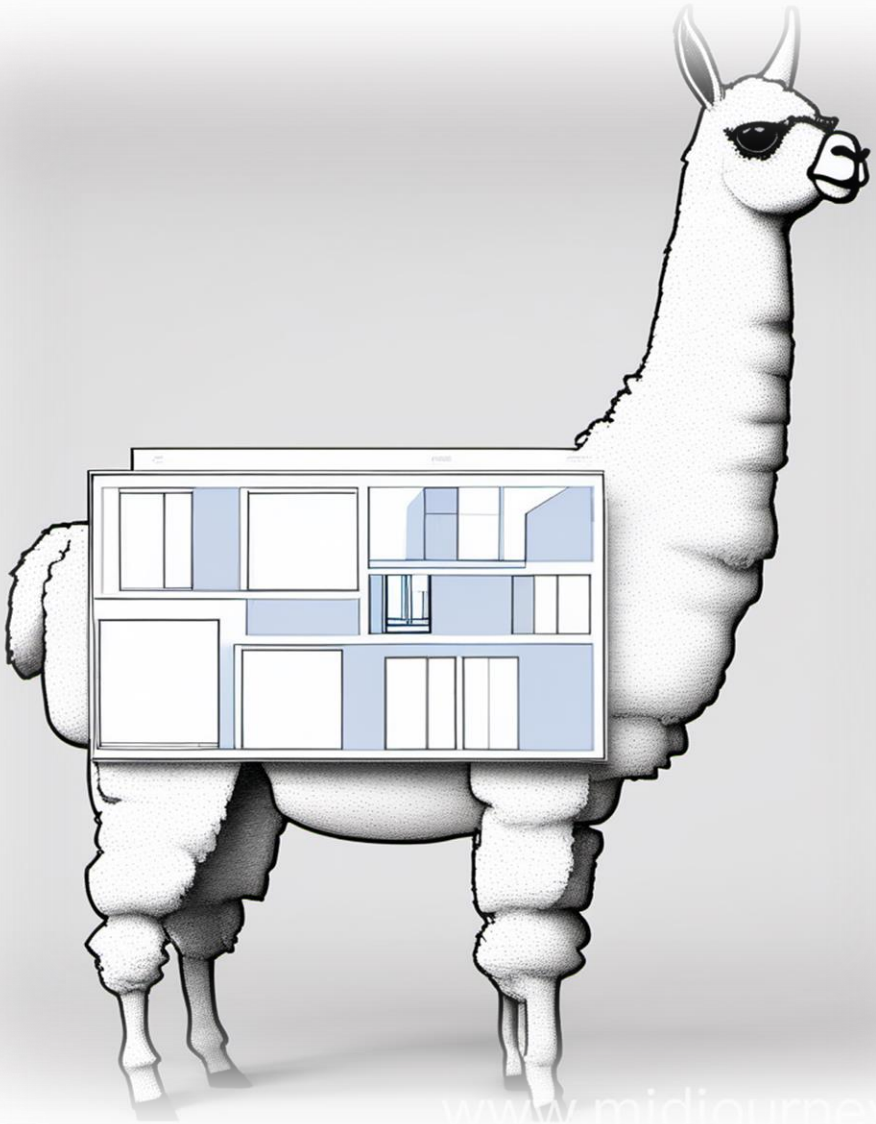
Einleitung

Was ist Llama2?

- Familie von großen Sprachmodellen in verschiedenen Größen und Varianten:
  - **Base-Varianten** (nur Pretraining)
  - **Chat-Varianten** (Fine-Tuning für Chat-Anwendungen)
  - Jeweils mit 7, 13 und 70 Milliarden Parameter
- **Aufgabe:** Vorhersage des nächsten Wortes anhand des vorhergehenden Kontexts zur Generierung von Texten.

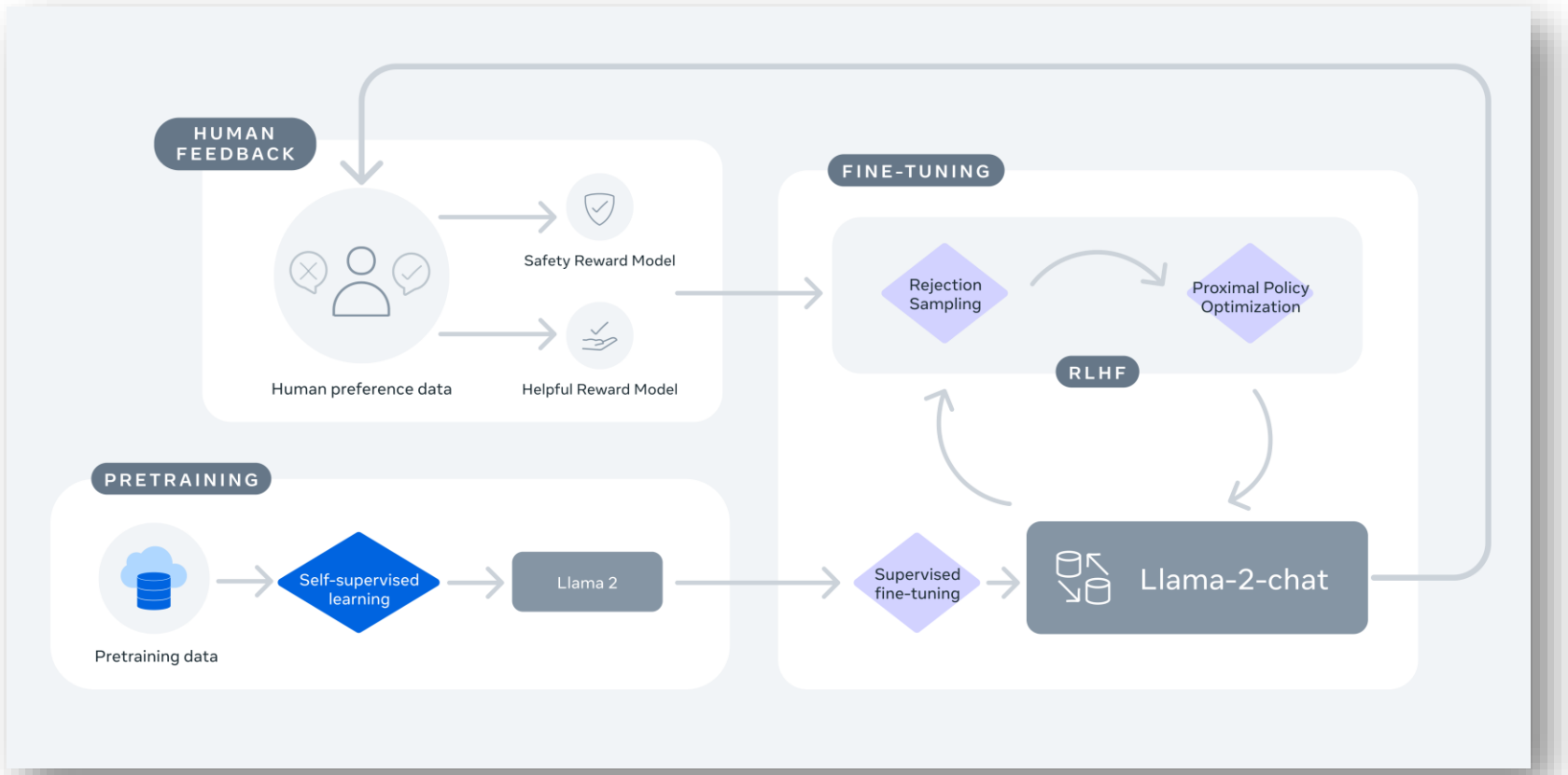
Motivation:

- **Aufstieg der Großen Sprachmodelle:** Große Sprachmodelle sind in den Bereichen der natürlichen Sprachverarbeitung und künstlichen Intelligenz stark im Einsatz.
- **Universelle Modelle:** Ein aktueller Trend besteht darin, ein Modell für alle Aufgaben zu nutzen.
- **Herausforderung für LLama2:** Die Frage, ob große Sprachmodelle wie LLama2 auf domänenspezifische linguistische Aufgaben übertragbar sind, steht im Fokus.



Llama2 – Architektur und Training

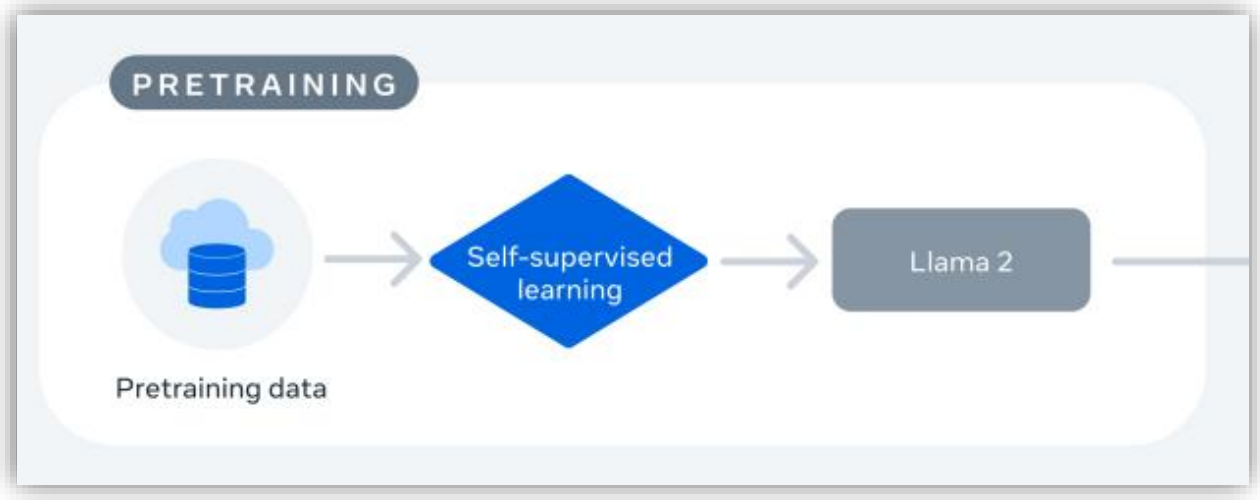
Llama2: Architektur und Training



Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Pretraining

- Llama2 (Base-Varianten) werden durch selbstüberwachtes Lernen auf einer großen Menge ungelabelter Daten trainiert (Next Word Prediction).



Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Pretraining

Verwendete Daten:

- Mischung aus verschiedenen öffentlich zugänglichen Online-Datenquellen
- 40% mehr Daten als bei Llama1
- 2 Billionen Tokens
- Trainingsdaten bestehen hauptsächlich aus englischen Texten

Language	Percent	Language	Percent
en	89.70%	uk	0.07%
unknown	8.38%	ko	0.06%
de	0.17%	ca	0.04%
fr	0.16%	sr	0.04%
sv	0.15%	id	0.03%
zh	0.13%	cs	0.03%
es	0.13%	fi	0.03%
ru	0.13%	hu	0.03%
nl	0.12%	no	0.03%
it	0.11%	ro	0.03%
ja	0.10%	bg	0.02%
pl	0.09%	da	0.02%
pt	0.09%	sl	0.01%
vi	0.08%	hr	0.01%

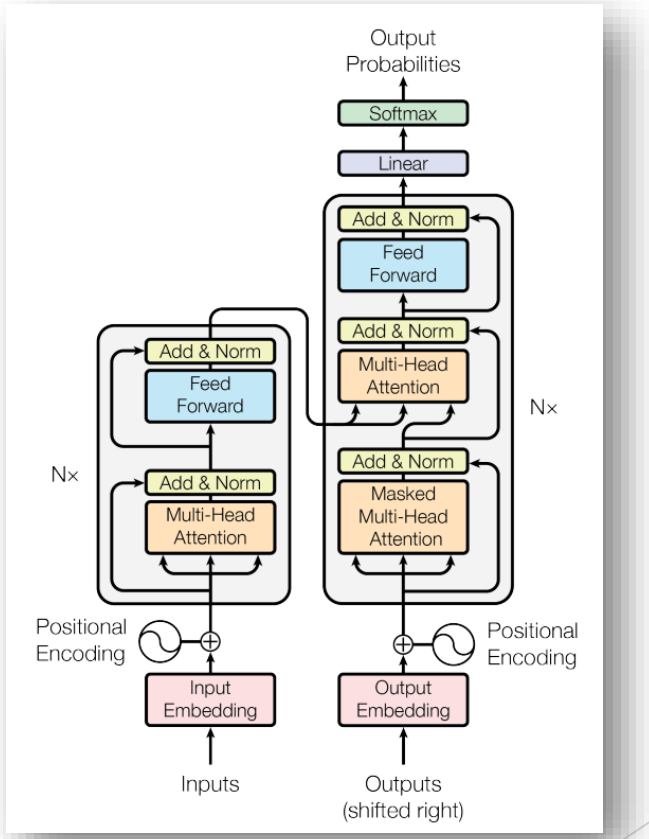
Sprachverteilung den in Trainingsdaten

Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Pretraining

Architektur:

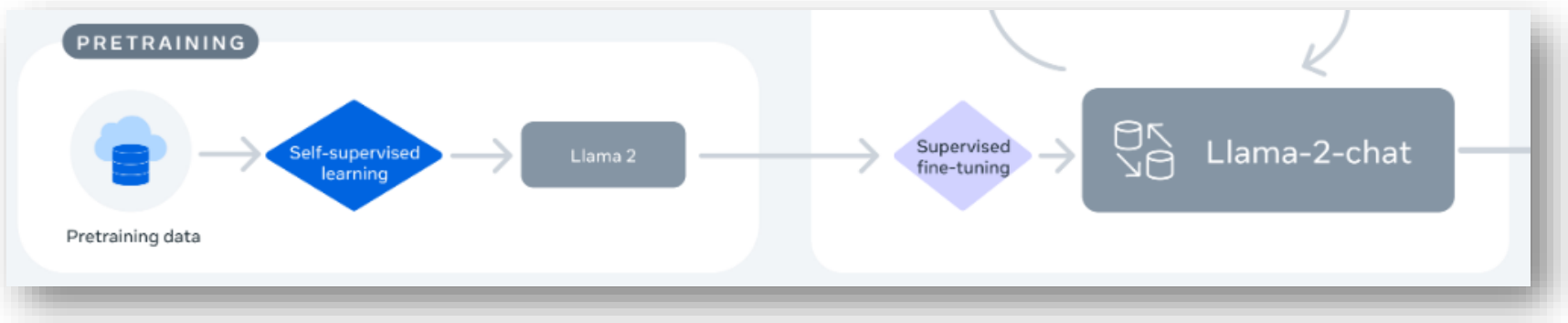
- 3 Varianten in unterschiedlichen Größen (7, 13 und 70 Milliarden Parameter)
- Basierend auf **Transformer-Architektur** + verschiedene Anpassungen für verbesserte Performance:
  - Normalisierung mit RMSNorm
  - SwiGLU Aktivierungsfunktion
  - Grouped Query Attention (GQA)



Quelle: Vaswani, Ashish et al. "Attention is All you Need." Neural Information Processing Systems (2017).

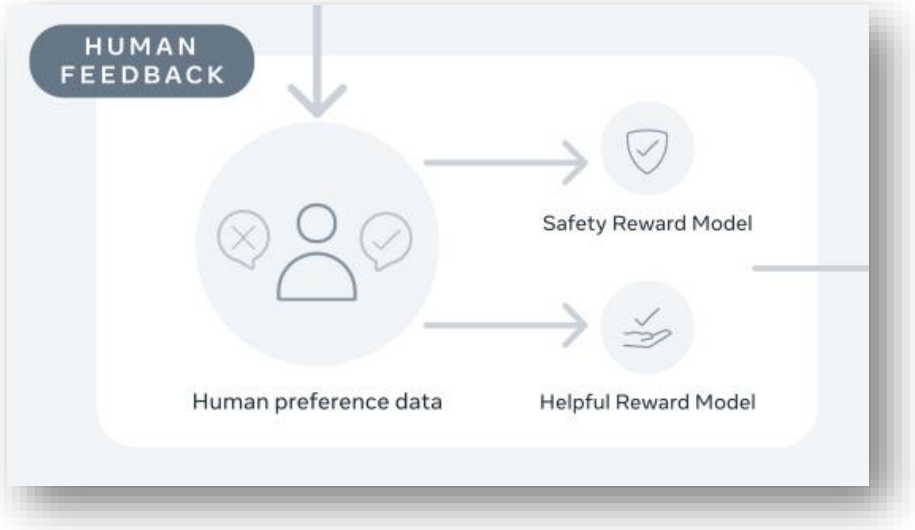
Fine-Tuning

- Eine erste Version von Llama2-Chat entsteht durch überwachtes Fine-Tuning auf gelabelten Daten, die aus einem Prompt und der entsprechenden Antwort bestehen.



Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Fine-Tuning

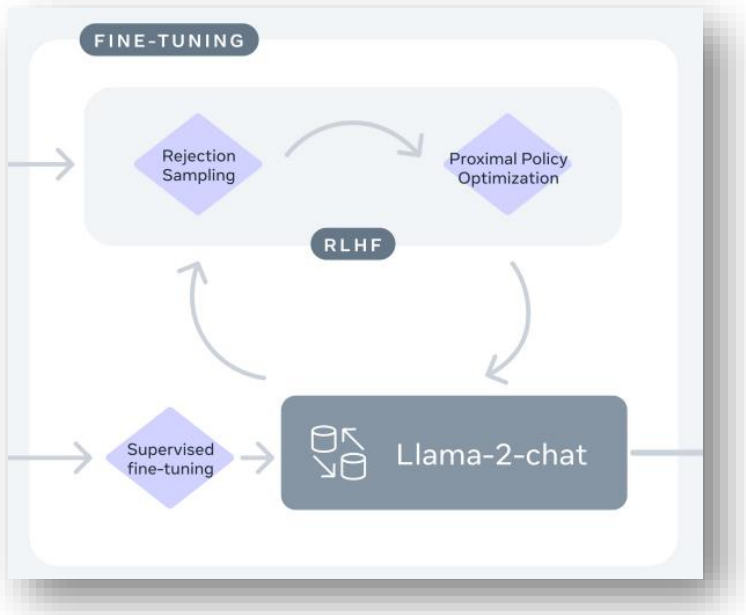


Reinforcement Learning with Human Feedback (RLHF):

- RLHF ist ein Verfahren zur iterativen Anpassung des Modellverhaltens an **menschliche Präferenzen** und das **Befolgen von Anweisungen**.
- Dazu werden Daten gesammelt, die menschliche Präferenzen repräsentieren, indem **Annotatoren** angeben, welche von **zwei Modellausgaben sie bevorzugen**.
- Besondere Aufmerksamkeit wird dabei auf **hilfreiche** und **sichere Ausgaben** gelegt (engl. *helpfulness* und *safety*)
- Dieses menschliche Feedback wird verwendet, um zwei **„Belohnungsmodelle“** (engl. *reward models*) zu trainieren, die Muster in den Präferenzen der Annotatoren lernen und diese in den Entscheidungsprozess des Modells einfließen lassen.

Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Fine-Tuning

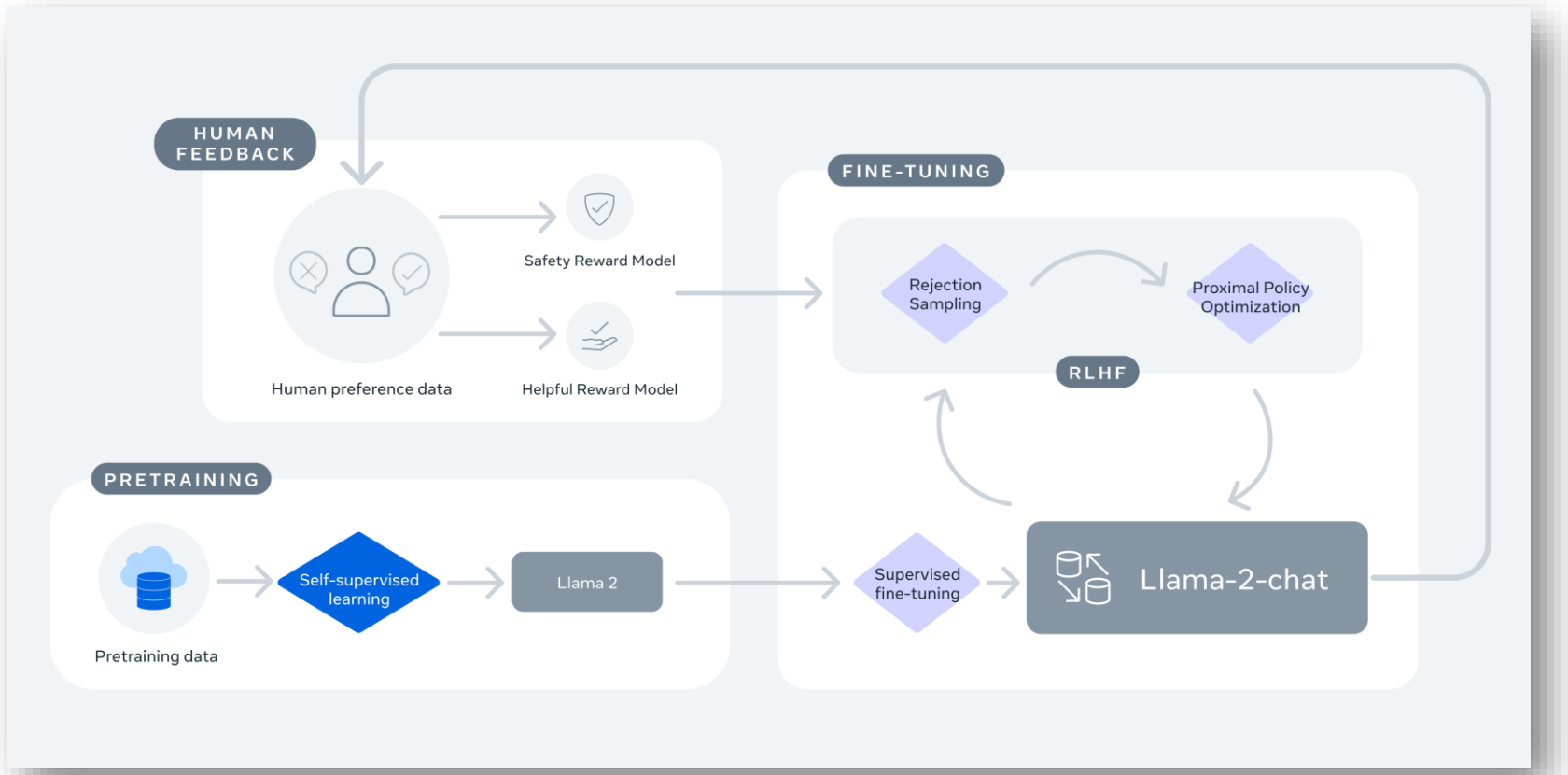


Reinforcement Learning with Human Feedback (RLHF):

- Die Belohnungsmodelle berechnen aus dem Prompt und der Ausgabe von Llama2-Chat einen **Score**, der die Qualität der Ausgabe bewertet.
- **Rejection Sampling:** K Ausgaben des Modells werden gesampelt und der vielversprechendste Kandidat wird aufgrund des Scores der Belohnungsmodelle ausgewählt. Die ausgewählten Ausgaben bilden den neuen Goldstandard für das weitere Fine-Tuning.
- **Proximal Policy Optimization:** Das Belohnungsmodell wird als Belohnungsfunktion verwendet. Das Modell wird so angepasst, dass es eine möglichst große Belohnung erhält.

Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Llama2: Architektur und Training



Quelle: Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ *arXiv preprint arXiv:2307.09288* (2023).

Fine-Tuning

Übersicht mit Modellgrößen und verwendete Daten

MODEL SIZE (PARAMETERS)	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	Model architecture:	Data collection for helpfulness and safety:
13B	Pretraining Tokens: 2 Trillion	Supervised fine-tuning: Over 100,000
70B	Context Length: 4096	Human Preferences: Over 1,000,000

Quelle: https://ai.meta.com/llama/

Llama2: Vor- und Nachteile

- **Vorteile:**
  - Als großes Sprachmodell bietet Llama2 ein breites Anwendungsspektrum.
  - Llama2 ist in verschiedenen Größen und Varianten verfügbar.
  - Die Open-Source-Zugänglichkeit von Llama2 fördert die kollaborative Entwicklung und Transparenz.

- **Nachteile:**
  - Aktuell beschränkt sich LLama2 fast ausschließlich auf die englische Sprache.
  - Die Nutzung von LLama2 erfordert dedizierte Hardware, insbesondere GPUs.
  - Die Ausgabe des Modells ist nicht immer leicht zu interpretieren und zu steuern.

Llama2 - Experimente

Morphologie: Wortstrukturen

- **Ziel:** Bestimmung von Wortstrukturen
- Vorgehen:
  - **Llama2-Eingabe:**
    - Prompt mit Aufgabenbeschreibung, Beispiel im gewünschten Format und zu analysierendes Wort (siehe nächste Folie)
  - **Llama2-Ausgabe:** Analyse in Klammernotation
  - **Visualisierung** der Llama2-Ausgabe mit Online-Tool (http://mshang.ca/syntree/)
  - **Vergleich** mit gewünschter Ausgabe

Morphologie: Wortstrukturen

- **Llama2-Eingabe:**

  Prompt bestehend aus:

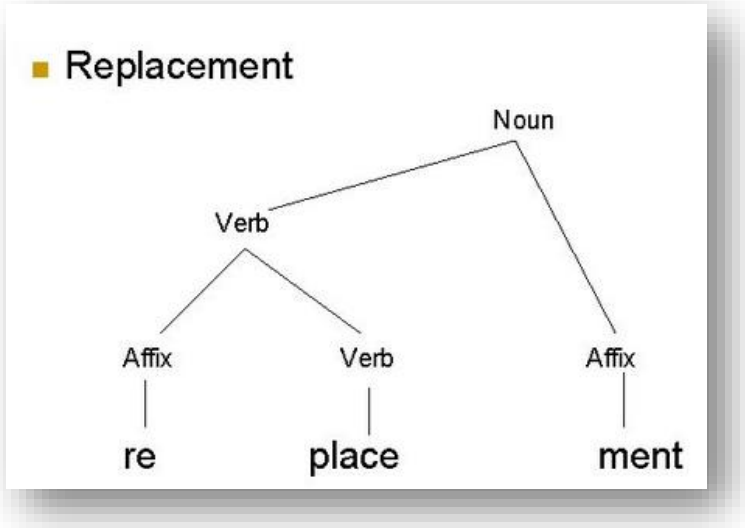
  1. Aufgabenbeschreibung
  2. Eine Beispielanalyse im gewünschten Format (One-Shot-Learning)
  3. Weitere Hinweise
  4. Zu analysierendes Wort

1. Generate a morphology tree in square bracket notation based on the following example:
2. independently ->
   [adverb [adjective [prefix [in]] [adjective [verb [depend]] [suffix [ent]]]] [suffix [ly]]]
3. The tree contains the moprhemes as well as their word forms.
   Now it's your turn to generate the morphology tree (use the same format as the example and give no additional explanations):
4. unlockable ->

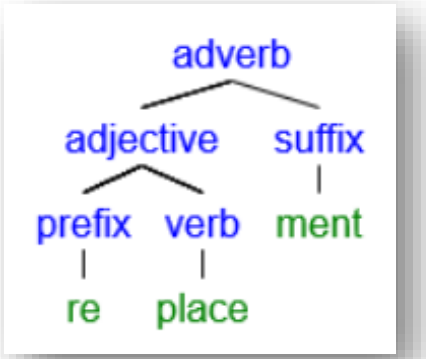
Morphologie: Wortstrukturen

- Wort: „Replacement“

- **Referenz:**



- **Llama2:**



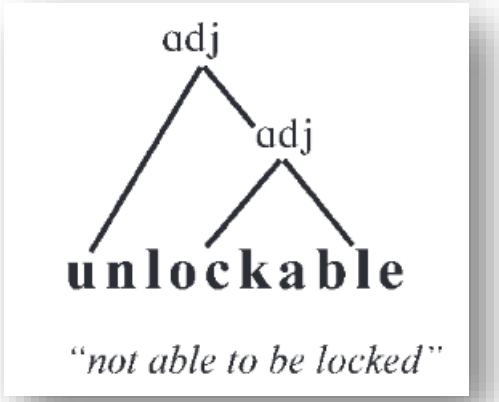
LLama2-Ausgabe:
[adverb [adjective [prefix [re]] [verb [place]]] [suffix [ment]]]

Referenz Quelle: https://slidetodoc.com/morphology-class-6-lesson-3-content-words-x/

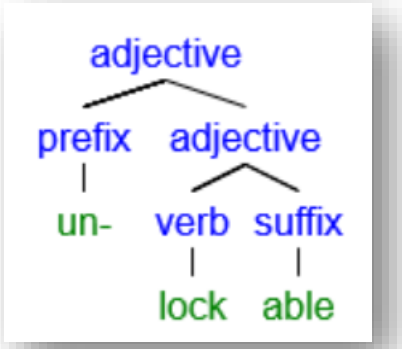
Morphologie: Wortstrukturen

- Wort: „Unlockable“

- **Referenz:**



- **Llama2:**



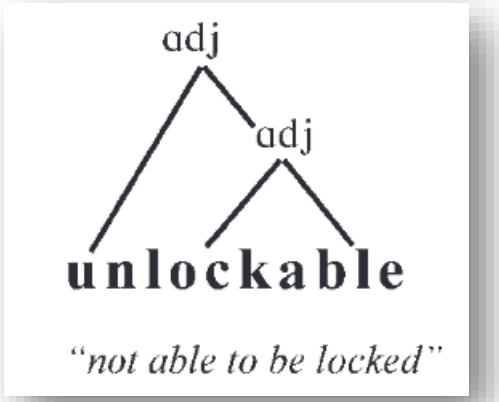
LLama2-Ausgabe:
[adjective [prefix [un-]] [adjective [verb [lock]] [suffix [able]]]]

Referenz Quelle: https://www.tandfonline.com/doi/abs/10.1080/01690960444000232

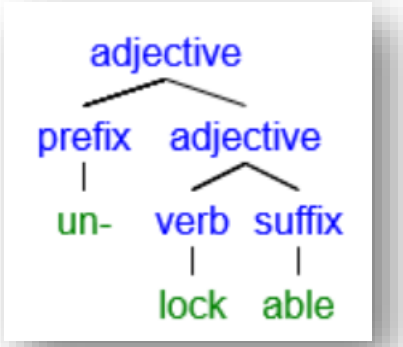
Morphologie: Wortstrukturen

- Problem: „Unlockable“ hat zwei mögliche Analysen:

- **Referenz:**



- **Llama2**:



Referenz Quelle: https://www.tandfonline.com/doi/abs/10.1080/01690960444000232

Syntax: Dependenzanalyse

- **Ziel:** Generierung von syntaktischen Dependenzanalysen
- Vorgehen:
  - **Llama2-Eingabe:**

    Prompt mit Aufgabenbeschreibung, Beispielanalysen im gewünschten Format und zu analysierender Satz (siehe nächste Folie)
  - **Llama2-Ausgabe:** Dependenzanalyse in tabellarischer Form
  - **Verarbeitung und Visualisierung** der Ausgabe mit Python
  - **Vergleich** zwischen Llama2-Ausgabe und von Spacy erzeugter Ausgabe

  (Code zur Verarbeitung und Visualisierung der Ausgaben von Llama2 und Spacy im Anhang)

Syntax: Dependenzanalyse

- **Llama2-Eingabe:**

  Prompt bestehend aus:

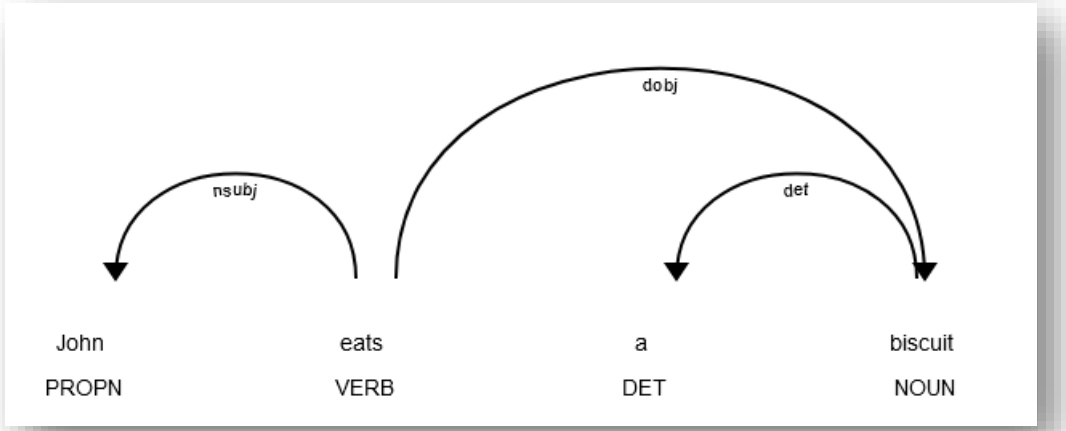
  1. Aufgabenbeschreibung
  2. Zwei Beispielanalysen im gewünschten Format (Few-Shot-Learning)
  3. Weitere Hinweise
  4. Zu analysierender Satz

1.
```
Analyze the sentence based on the following examples:
```
2.
```
Example 1:

Sentence: "I ate a sandwich"
Analysis: data = {
"Token": ["I", "ate", "a", "sandwich"],
"POS Tag": ["PRON", "VERB", "DET", "NOUN"],
"Dependency": ["nsubj", "ROOT", "det", "dobj"],
"Head": [1, None, 3, 1],
}

Example 2:

Sentence: "The dog chases the cat"

Analysis: data = {
"Token": ["The", "dog", "chases", "the", "cat"],
"POS Tag": ["DET", "NOUN", "VERB", "DET", "NOUN"],
"Dependency": ["det", "nsubj", "ROOT", "det", "dobj"],
"Head": [1, 2, None, 4, 2],
}

Use the same format.

Sentence: "The mouse runs"
```

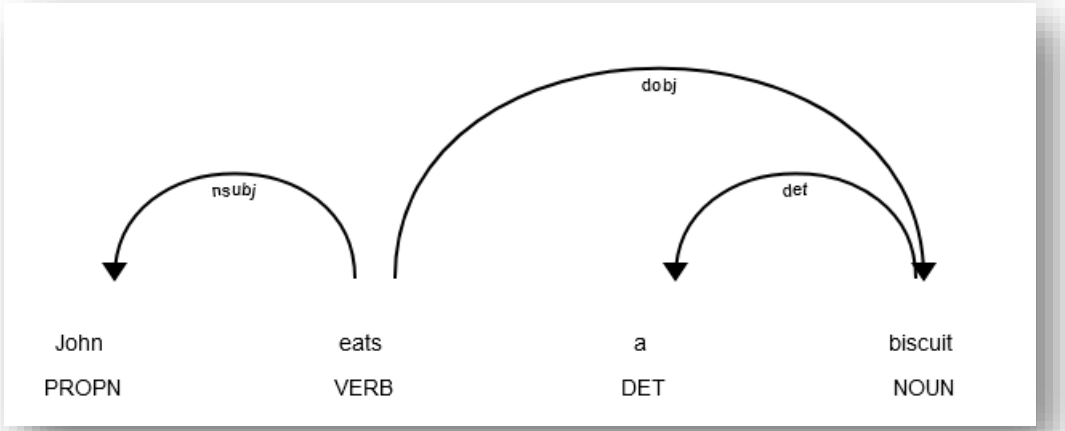
Syntax: Dependenzanalyse

- Satz: „John eats a biscuit“

- **Spacy:**

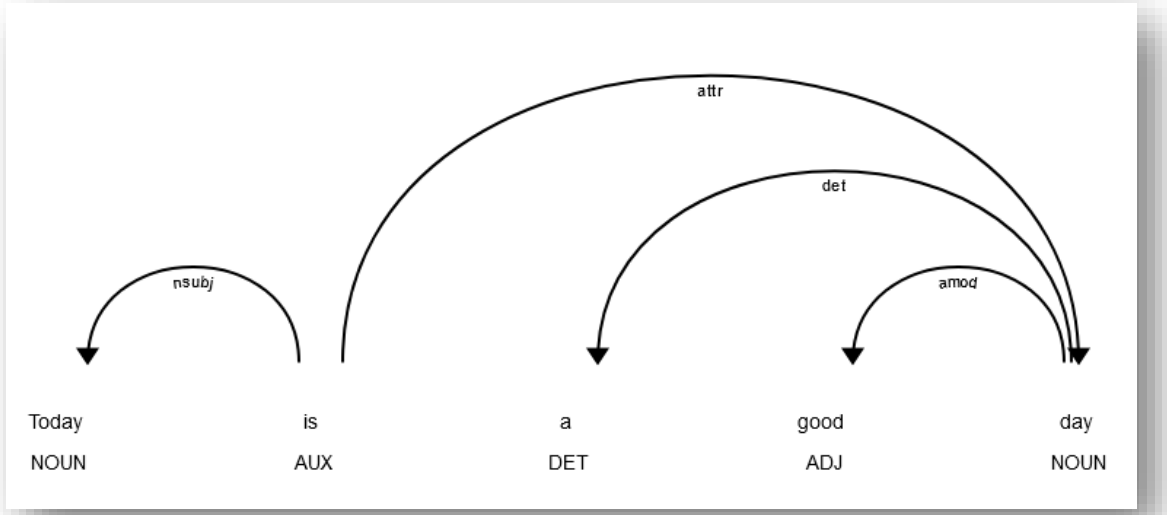


- **Llama2:**

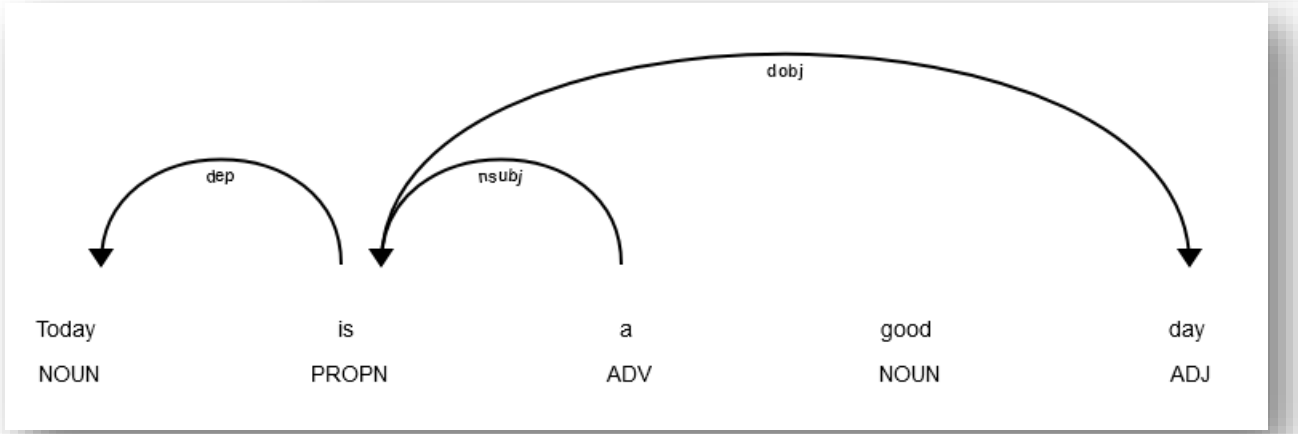


Syntax: Dependenzanalyse

- Satz: „Today is a good day“
  - **Spacy:**



  - **Llama2**:



Fazit

- Bestimmung von Wortstrukturen:
  - Grundlegendes linguistisches Wissen scheint vorhanden zu sein, jedoch sind die Ergebnisse nicht immer zuverlässig
  - Besonders Problematisch bei mehreren möglichen Analysen
- Syntaktische Dependenzanalyse:
  - Nur für sehr einfache Sätze möglich
  - Für komplexere Sätze ist Llama2 nicht ohne weiteres zu gebrauchen
- Mögliche Verbesserungen:
  - Prompt-Engineering
  - Weiteres Training (sehr rechen- und zeitintensiv)
  - PEFT (Parameter Efficient Fine-Tuning)

Literatur

- Touvron, Hugo, et al. „Llama 2: Open foundation and fine-tuned chat models.“ arXiv preprint arXiv:2307.09288 (2023).
- De Almeida et al. „Changing morphological structures: The effect of sentence context on the interpretation of structurally ambiguous English trimorphemic words.“ Language and Cognitive Processes - LANG COGNITIVE PROCESS. 20. 10.1080/01690960444000232. (2005).
- Vaswani, Ashish et al. “Attention is All you Need.” Neural Information Processing Systems (2017).
- Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).
- Llama2 Code: https://github.com/facebookresearch/llama
- Llama2 7B Online-Tool: https://huggingface.co/spaces/huggingface-projects/llama-2-7b-chat
- Llama2 70B Online-Tool: https://stablediffusion.fr/llama2
- Visualisierung der Wortstrukturen: http://mshang.ca/syntree/
- Erzeugung der Dependenzanalysen: https://spacy.io/
- Lama-Bilder generiert mit: https://www.midjourneyai.ai/de (Folien 1, 4, 15, 27)

Fragen?

Anhang: Code für syntaktische Dependenzanalyse

- Unten: Dependenzanalyse mit Spacy-Modell
- Rechts: Dependenzanalyse mit Verarbeitung der Ausgabe von Llama2

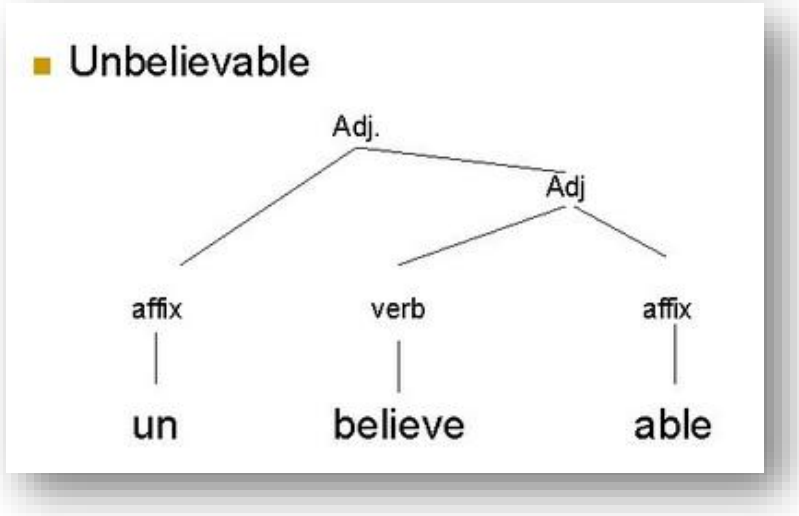
```python
import spacy
from spacy import displacy

# Load the SpaCy English model
nlp = spacy.load("en_core_web_sm")

# Process the sentence
sentence = "Today is a good day"
doc = nlp(sentence)

svg = displacy.render(doc, style="dep", jupyter=True)
```

```python
import spacy
from spacy import displacy
from IPython.display import SVG, display

data = {
    "Token": ["Today", "is", "a", "good", "day"],
    "POS Tag": ["NOUN", "PROPN", "ADV", "NOUN", "ADJ"],
    "Dependency": ["dep", "nsubj", "root", "dep", "dobj"],
    "Head": [1, 2, None, 3, 1],
}

# Create a spaCy Doc object
nlp = spacy.blank("en")
doc = nlp(" ".join(data["Token"]))

# Create spaCy tokens
for i, (token, pos, dep, head) in enumerate(zip(data["Token"], data["POS Tag"],
                                                data["Dependency"], data["Head"])):
      doc[i].pos_ = pos
      doc[i].dep_ = dep
      if head:
        doc[i].head = doc[head]

# Visualize with displacy
svg = displacy.render(doc, style="dep", page=True)
display(SVG(svg))
```

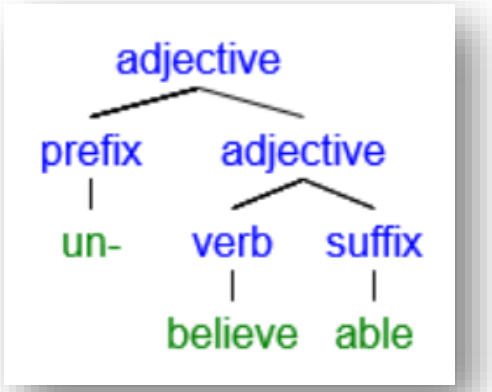
Anhang: Weitere Experimente zur Analyse von Wortstrukturen

- Wort: „Unbelievable“

- **Referenz:**



- **Llama2:**



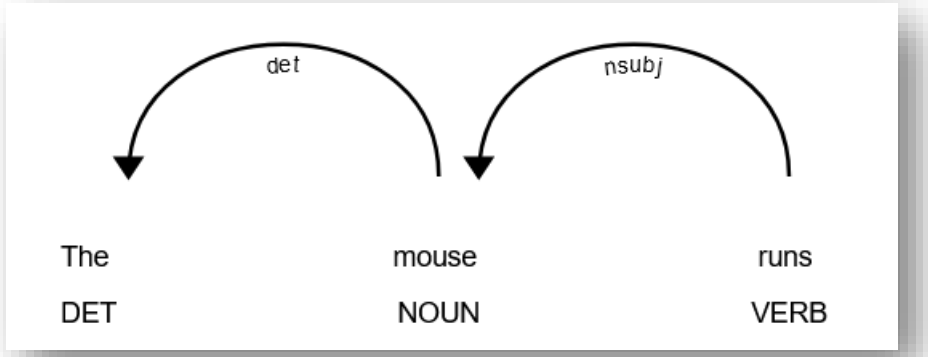
LLama2-Ausgabe:
[adjective [prefix [un-]] [adjective [verb [believe]] [suffix [able]]]]

Referenz Quelle: https://slidetodoc.com/morphology-class-6-lesson-3-content-words-x/

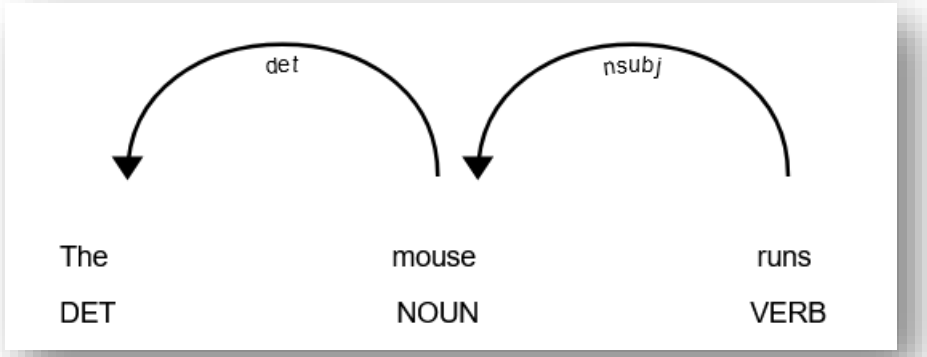
Anhang: Weitere Experimente zur Dependenzanalyse

- Satz: „The mouse runs“

- **Spacy:**



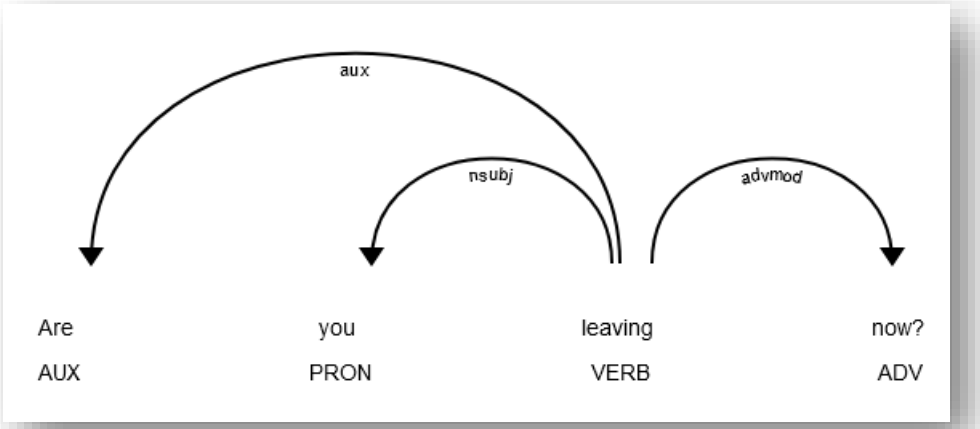
- **Llama2:**



Anhang: Weitere Experimente zur Dependenzanalyse

- Satz: „Are you leaving now?“

- **Spacy:**



- **Llama2:**