Large Language Models in Theorie und Praxis with Python

Timo Baumann

OTH REGENSBURG

Tutorial bei der StatWo 09/2024

Ist ChatGPT,
sind Large Language Models intelligent?

OTH REGENSBURG

Was tut ein Language Model?

- berechnet Wahrscheinlichkeiten für Wortfolgen:

  P(“Heute ist der Himmel blau”)

klassische Anwendungen von Language Models

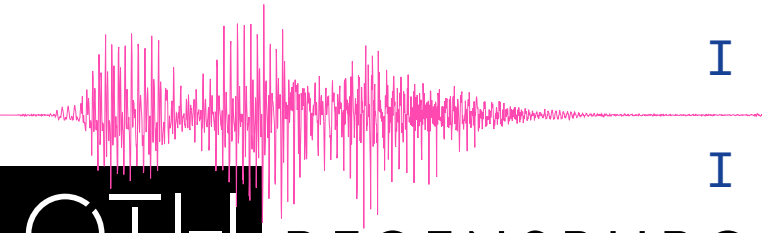
- als Teil maschineller Übersetzung: quellentreu und *flüssig*

ich drücke die Daumen

I'm crossing my fingers

:

I'm pressing the thumbs



- als Teil von Spracherkennung

I helped Apple wreck a nice beach

I helped Apple recognize speech

OTH REGENSBURG

Was tut ein Language Model?

- berechnet Wahrscheinlichkeiten für Wortfolgen:

  P("Heute ist der Himmel blau")

  = P(1. Wort ist "Heute"
  ∧ 2. Wort ist "ist"
  ∧ 3. Wort ist "der"
  ∧ ...
  ∧ 5. Wort ist "blau")

- Kettenregel für Wahrscheinlichkeiten:

  = P(1. Wort ist "Heute")
  × P(2. Wort ist "ist" |
  1. Wort war "Heute")
  × ...
  × P(5. Wort ist "blau" |
  die Wörter davor waren
  "Heute ist der Himmel")

- für Wortfolge $\mathbf{W} = w1 \ldots wn$
  $P(\mathbf{W}) = \Pi k \in 1..n \; P(w_k / w_{1..k-1})$

Entscheidungen mit beschränktem Kontext

- $P(\mathbf{W}) = \Pi k \in 1..n\ P(_k|w_{1..k-1})$
- Markov-Annahme: $P(w_k|w_{1..k-1}) \approx P(w_k|w_{k-(N-1)..k-1})$
  - ferne Vergangenheit ist unwichtig
  - N-1 Tokens genügen um das N'te vorherzusagen
- zunächst N-Gramme: kleine N (3-5 Wörter)
- dann RNNs/LSTMs: beliebig große N
- heute Transformer: große (aber beschränkte) N (2000, 4000, 8000, ...)

Was tut ein Language Model?

- berechnet Wahrscheinlichkeiten für Wortfolgen
  insbesondere für das nächste Wort

- gegeben einen Kontext berechnet es die
  Wahrscheinlichkeitsverteilung über
  alle möglichen nächsten Wörter

Heute ist der Himmel

Kontext

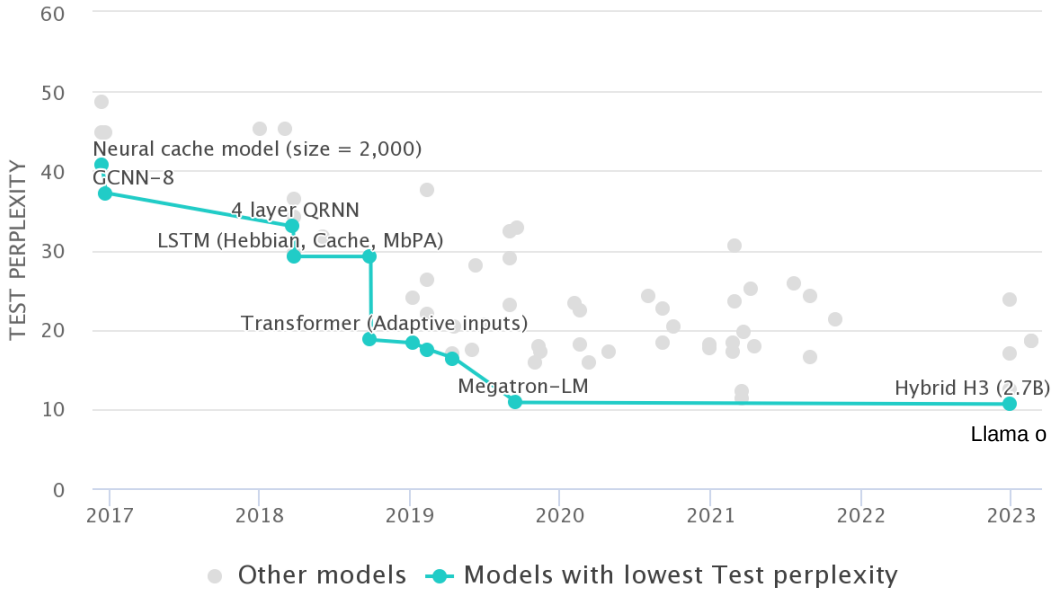
- Training mittels bereits bestehender
  Texte (Zeitung, Bücher, Internet, ...)

nächstes Wort

blau
grau
bewölkt
⋮
nicht
grün
ganz
⋮
und
fliegt
Frankfurt
Wassermelone

Language Models wurden schon immer immer besser

- schnellere Computer
- mehr Speicher
- mehr Trainingstext



- Kriterium Entropie: informationstheoretisches Maß, wie gut ungesehene Texte vorhergesagt werden

OTH REGENSBURG

https://paperswithcode.com/sota/language-modelling-on-wikitext-103

Language Models werden seit über 100 Jahren entwickelt

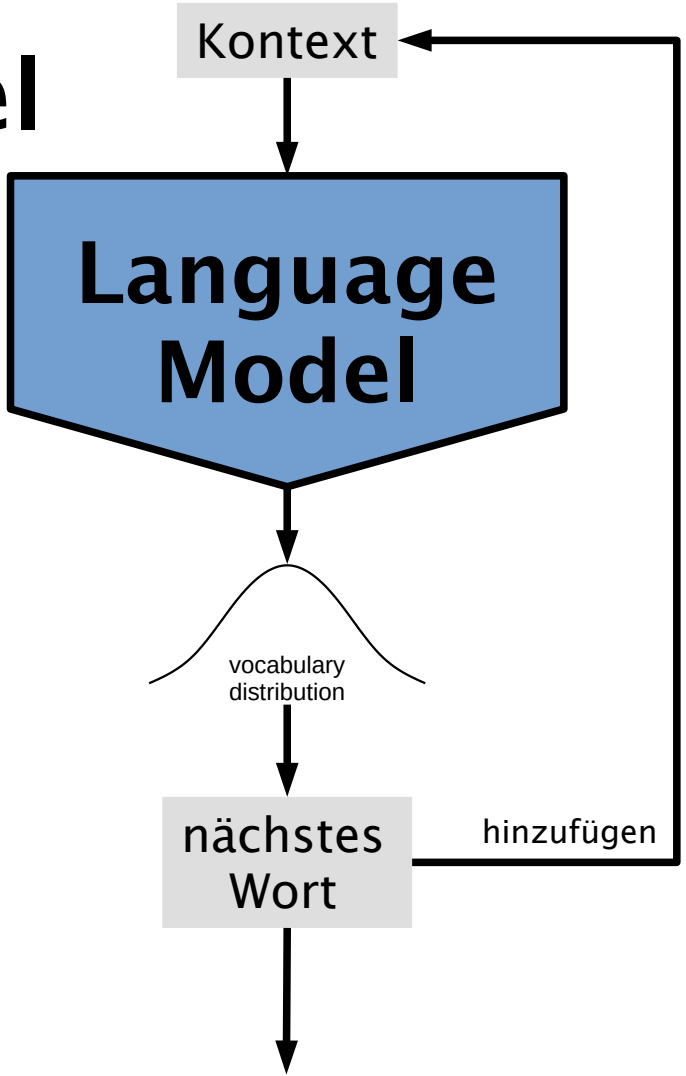
- Markov (1913) – Folge von Vokalen/ Konsonanten in Pushkin Novelle
- Shannon (1948)'s Word Game
- seit 1975 in Spracherkennungssystemen...

- und seit 2023 in aller Munde – warum?

Textgenerierung mit einem Language Model

- Startkontext (ggfs. leer)
- Wahrscheinlichkeitsverteilung über alle möglichen nächsten Wörter errechnen
- das wahrscheinlichste* als nächstes Wort wählen
- dieses zum Kontext hinzufügen
- weiter generieren (bis das Spezialwort </ende> generiert wird)

* alternativ: nach Wahrscheinlichkeiten gewichtete Auswahl



Auf einmal kamen schlaue Antworten (=zero-shot learning)

- `Heute ist der Himmel` → `blau`
- `In Regensburg liegt die Steinerne` → `Brücke`

- wiederholte Anwendung (“autoregressiv”) auf Basis vorangegangener Ergebnisse
  - `der UN-Sicherheitsrat tagt in` → `New`
  - `der UN-Sicherheitsrat tagt in New` → `York`
  - `der UN-Sicherheitsrat tagt in New York` → `</ende>`

- ausführlichere Kontexte sind geeignet sinnvolle Antworten zu **generieren**:
  - `Q: Erkläre mir die allgemeine Relativitätstheorie.`
    `A:` → `Die allgemeine Relativitätstheorie (ART) ist eine physikalische ...`
  - `Wer hat die nochmal erfunden?` → `ART wurde von Albert Einstein entwickelt.`
    (hier ist also die vorangegangene Frage und Antwort Teil des Kontextes!)

Emergente Intelligenz?

aktuelle Language Models:

- kennen (praktisch) alle Bü
- kennen (praktisch) alle Or
- kennen (praktisch) alle W

- werden einmal trainiert
  (haben also jeweils einen

haben alles Wissen, das öf

- Faktenwissen
- prozedurales Wissen (“wenn dies dann das”)

meine Meinung: Intelligenz = Wissen + Verstehen

Definition “Kipppunkt”:

quantitativ etwas besser,
qualitativ ein Durchbruch

vorher zu schlecht,
jetzt gut genug!

Zu den Tasten!

www.timobaumann.de/work/Main/StatWoLLMs

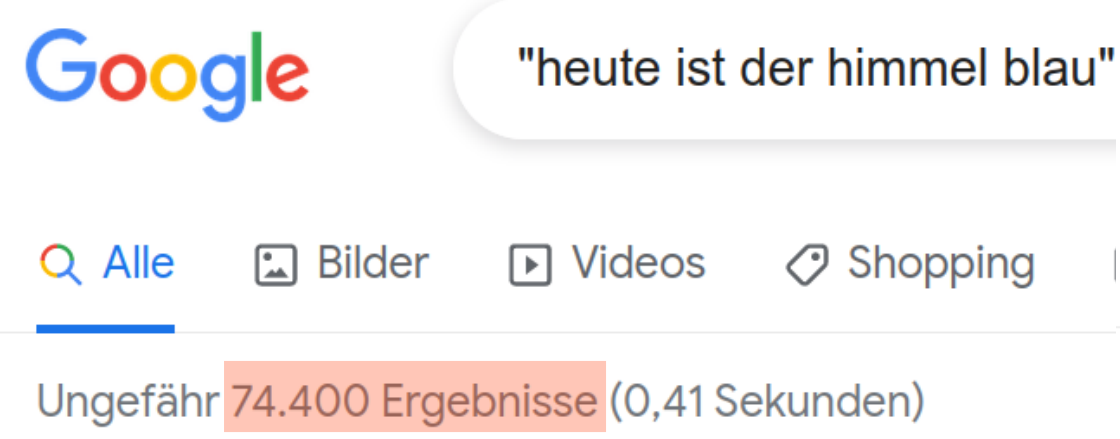
Wie funktionieren denn nun (Large) Language Models?

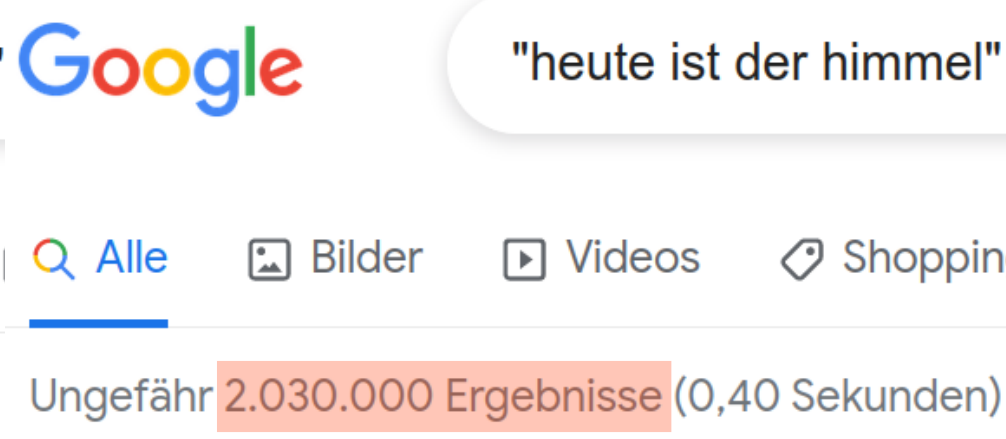
N-Gramme (die ersten 100 Jahre)

- Grundidee: “Heute ist der Himmel rot” → 7 mal im ganzen Internet

$$P(\text{blau} \mid \text{Heute ist der Himmel}) \approx 3{,}7\%$$

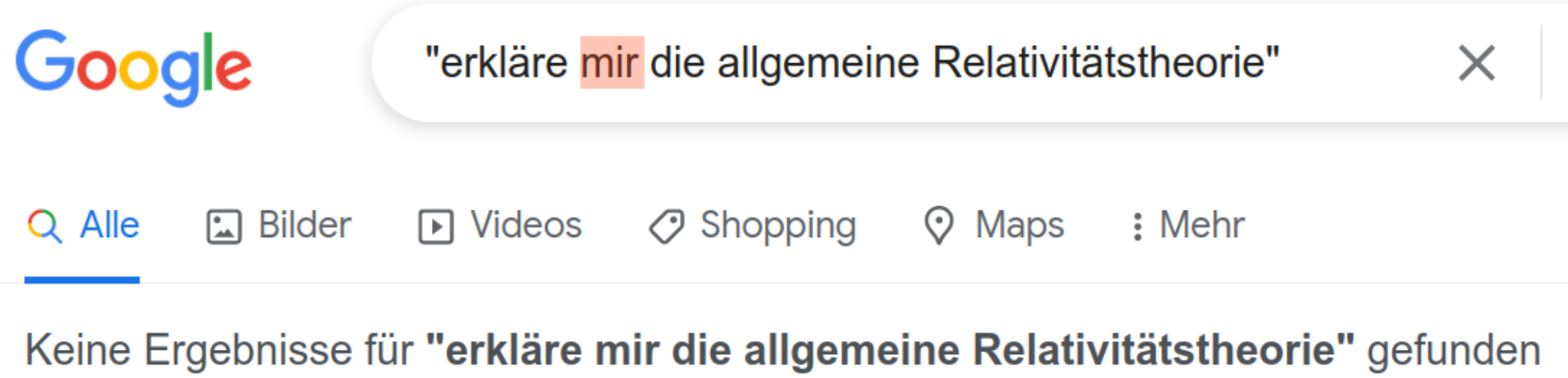
$$= \frac{\text{Count}(\text{Heute ist der Himmel blau})}{\text{Count}(\text{Heute ist der Himmel \_\_\_\_})}$$





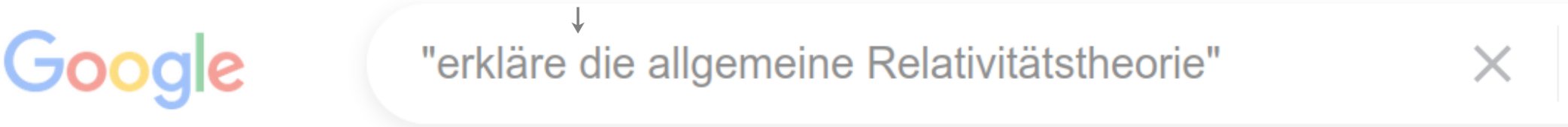
N-Gramme und Datenspärlichkeit

- Auszählen ist nur für kurze Kontexte nützlich (weil lange Kontexte zu selten vorkommen)



ohne “mir”

↓



Ungefähr 1 Ergebnisse

7170 × "Erklärung der allgemeinen Relativitätstheorie"
714 × "allgemeine Relativitätstheorie erklärt"
1530 × "allgemeine Relativitätstheorie für Babys" (!)

Astrotreff

https://www.astrotreff.de › forum › 279876-ki-im-astr...

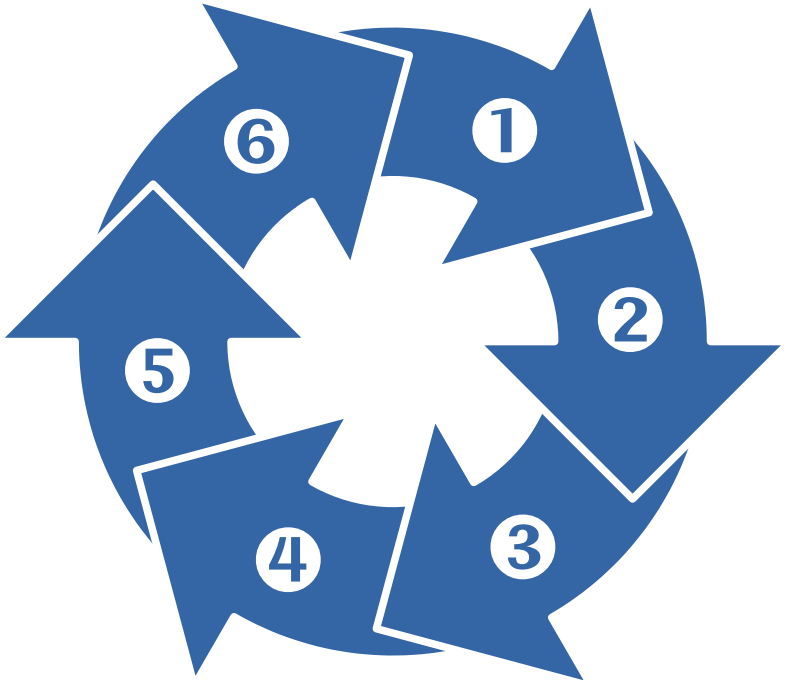
KI im Astrotreff - ein Meilenstein! - Seite 13

16.03.2023 — **erkläre die Allgemeine Relativitätstheorie**. Die Allgemeine Relativitätstheorie ist eine Theorie, die sich mit der Gravitation und der ...

Neuronale Netze

- ein Modell berechnet, welche Wörter als nächstes wie wahrscheinlich sind
  - Parameter bestimmen das Ergebnis der Berechnung
- damit wird es möglich, *ähnliche* Kontexte ähnlich zu bewerten
  - also ähnlich nächste Wörter vorzuschlagen

Training eines Neuronalen Netzes (=Parameter richtig einstellen)



1. mit bestehenden Parametern erste Ergebnisse berechnen für einige Trainingsdaten

   P(blau | Heute ist der Himmel) = 1%

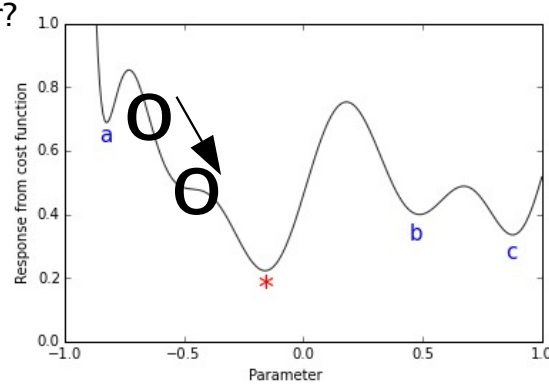
   “Heute ist der Himmel blau”

   “Heute ist der Himmel Wassermelone”

2. Fehler berechnen

   Fehler je nach Trainingsdatum entweder 99% oder 1%

3. uns interessiert die “Richtung” des Fehlers, insb. in welche “Richtung” wird er kleiner? (Das geht über Differenzialrechnung!)



4. Einfluss jedes Parameters auf die Fehlerrichtung berechnen

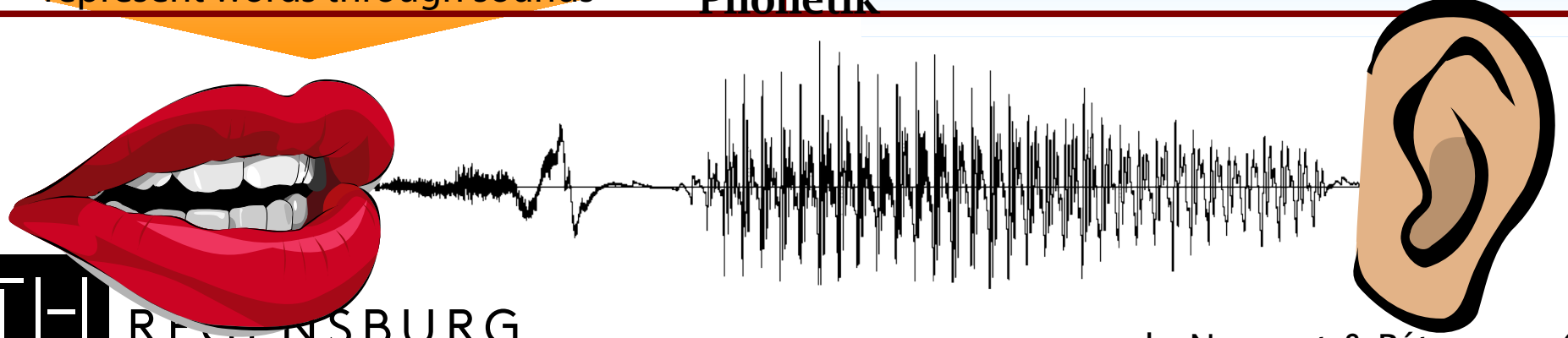
5. alle Parameter ein bisschen in Richtung kleinere Fehler anpassen

6. wiederholen (bis Zeit/Geld alle oder Ergebnisse gut genug)

OTH REGENSBURG

Sprachliche Kommunikationskette

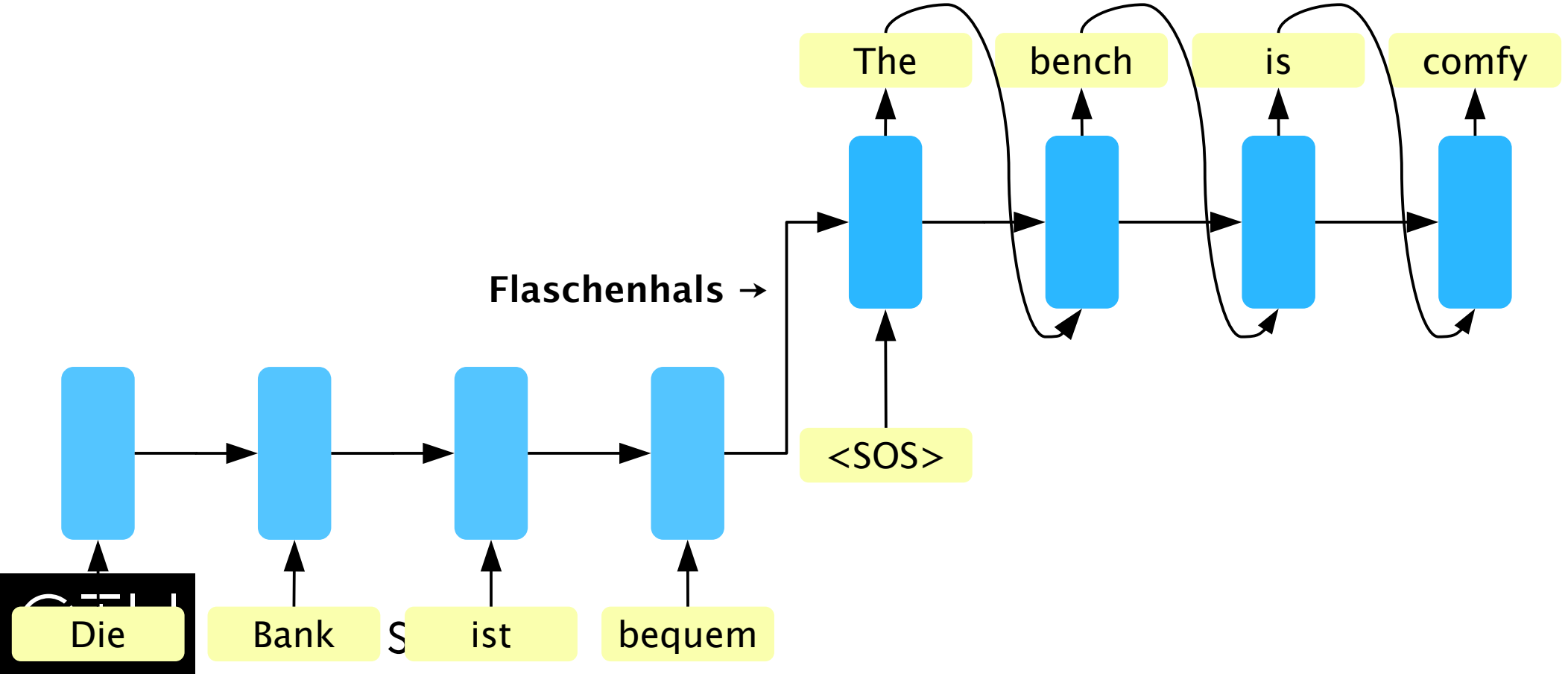
find message that describes idea	**Pragmatik**	recover idea described by message
determine structure to convey meaning	**Semantik/ Lexikologie**	determine meaning of structure
sequentialize structure to word stream	**Syntax/ Morphologie**	recover structure of sequence
represent words through sounds	**Phonologie/ Phonetik**	recombine sounds to words



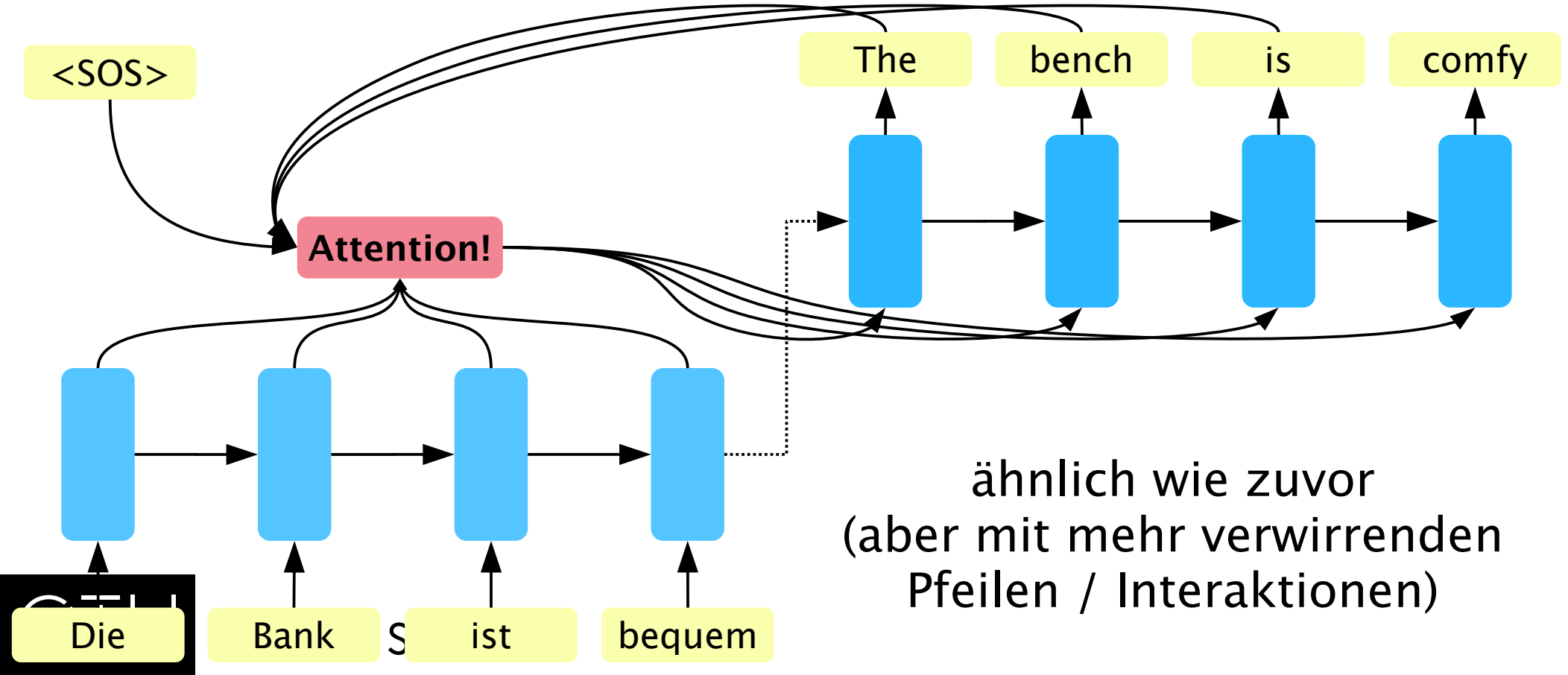
OTH REGENSBURG

nach: Neppert & Pétursson (1996)

neuronales Encoder-Decoder-Netzwerk (psycholinguistisch halbwegs plausibel)

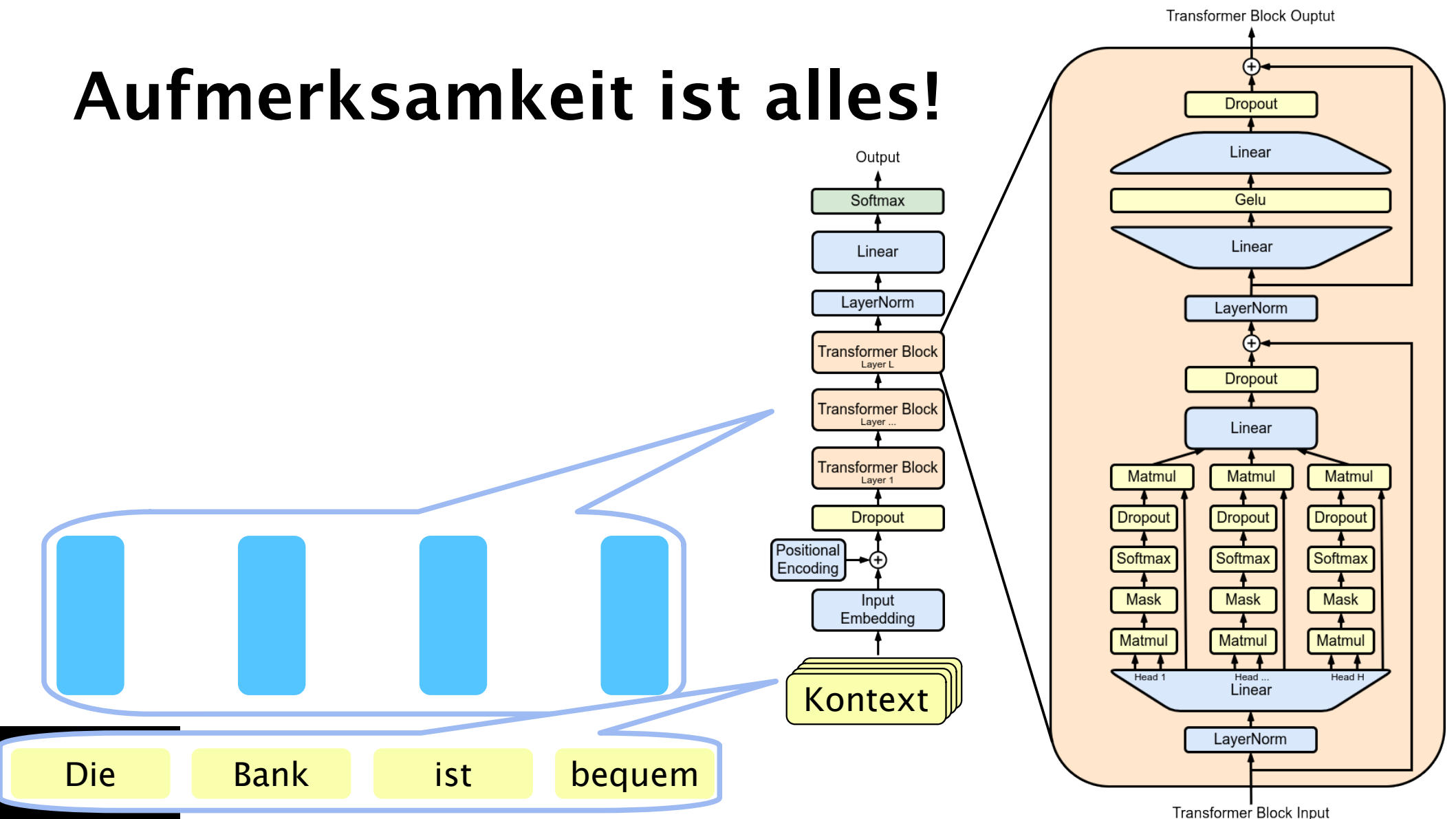


Encoder-Decoder-Netzwerk mit Aufmerksamkeitsmodellierung



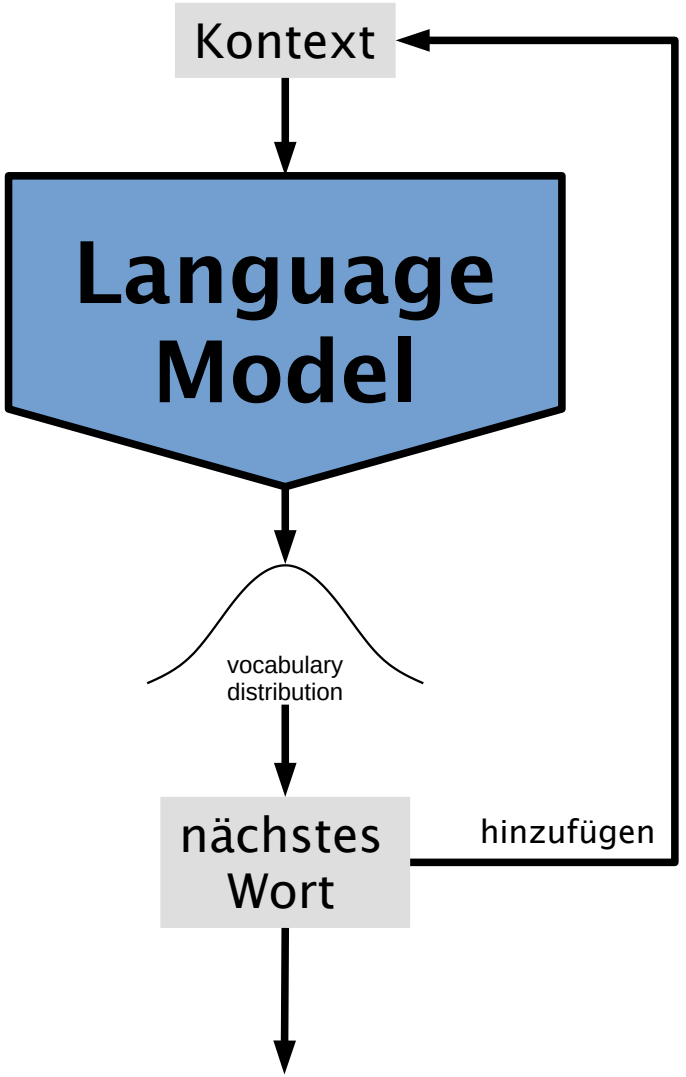
ähnlich wie zuvor
(aber mit mehr verwirrenden
Pfeilen / Interaktionen)

Aufmerksamkeit ist alles!



Generierung mit GPT-Modellen

- ganzen Kontext mittels mehrstufiger Aufmerksamkeit analysieren
  - Rechendauer unabhängig von Kontextlänge
- 1 Ausgabetoken generieren
  - und zum Kontext hinzufügen



Wort-Vektoren: *Idee*

- es gibt *sehr* viele unterschiedliche Wörter
  - Text mit 1 Milliarde Wörtern → fast 1 Million unterschiedliche Wörter (die meisten nur einmal)
- Wörter können verschieden aber ähnlich sein:
  - `ähnlich` und `änhlich` zum Beispiel
- wenn man mit Ähnlichkeiten rechnen könnte:
  - `Königin` − `Frau` $\stackrel{?}{=}$ `König` − `Mann` $\stackrel{?}{=}$ `royal`

Tokens statt Wörter

- seltene Wörter aus Teilen zusammensetzen:
  - Regensburg = Reg + ens + burg
- statistisches Verfahren zur Distillation der Tokens aus Zeichenfolgen
  - keine linguistische Analyse (Regen + s + burg)
  - keine besondere Betrachtung von Leer- oder Satzzeichen (alles Teil der Tokens)

https://platform.openai.com/tokenizer

(Teil-)Wort-Vektoren: *Embeddings*

- Einbettung jedes Wortes in einen Vektorrraum mit z.B. “nur” 300 Dimensionen
- Optimierung der Plazierung, sodass ähnliche Tokens an ähnlichen Stellen, unterschiedliche Tokens an unterschiedlichen Orten im Raum landen

Netzwerk-Architektur für LLMs

- optimiert auf schnelles, parallelisierbares Training mit großen Datenmengen
  - auf Kosten der psycholinguistischen Plausibilität
- optimiert auf große Kontextfenster, sodass viel Material zum “Verständnis” beitragen kann
- verarbeitet nicht Wort-für-Wort sondern Abstraktionsebene-für-Abstraktionsebene
  - Menschen zum Beispiel würden ja eher Wort-für-Wort verarbeiten...
  - bei der Generierung wird jeweils wieder komplett neu berechnet (auch implausibel)
- Zusammenhang zwischen Wörtern wird ausschließlich über Aufmerksamkeitssteuerung und ihre Position zueinander hergestellt

Aktuelle LLMs (I)

- nutzen Texte mit Milliarden oder Wörtern
- haben Milliarden Parameter
- benötigen zum Training tausende GPUs und trainieren dann viele Wochen
- jedes Neu-Training kostet viele Mio€

Aktuelle LLMs (II)

- können relativ einfach adaptiert werden:
  - Anpassung der Parameter mittels wenig, domänenspezifischer Daten (und mäßiger Trainingsdauer)
- können menschliches Feedback über die Qualität von Antworten integrieren
  - noch einfachere und spezifischere Anpassung als durch domänenspezifischer Daten

vom LLM zum ChatBot

- Textvervollständigung macht noch keinen ChatBot!
  - Anfrage:
    ```
    Explain the moon landing to a 6 year old in a few
    sentences.
    ```
  - Generierte Ausgabe (LLM):
    ```
    Explain the theory of gravity to a 6 year old.
    Explain the theory of relativity to a 6 year old in a
    few sentences.
    Explain the big bang theory to a 6 year old.
    Explain evolution to a 6 year old.
    ```

Take-aways

- Language Models berechnen das wahrscheinlich nächste Wort gegeben einen Kontext von vorangegangenen Wörtern
  - wiederholt man dies, können sie Text generieren

- Language Models lernen nicht dazu, während Sie sie benutzen
  - das wäre zu teuer; stattdessen werden vergangene Interaktionen zum Kontext hinzugefügt
  - möglicherweise lernen Sie dazu, während Sie sie benutzen?

- werden immer besser (in ihrer objektiv messbaren Leistung)
  - sind für viele Anwendungen “gut genug”
  - wann die nächste Leistungsstufe “emergiert” ist unmöglich vorherzusagen

Vielen Dank für Ihre Aufmerksamkeit!

timo.baumann@oth-regensburg.de

OTH REGENSBURG

ohne Wort-Vektoren

- Vokabular z.B. nur 100.000 häufigste Wörter
- naïve Vektordarstellung:
  - 100.000 Dimensionen
  - das k-te Wort im Vokabular hat an der k-ten Dimension den Wert 1 und in allen anderen 0
- keinerlei Ähnlichkeiten, jeder Abstand ist gleich

Lecture Plan: From Language Models to Assistants

1. **Zero-Shot (ZS) and Few-Shot (FS) In-Context Learning**
   - \+ No finetuning needed, prompt engineering (e.g. CoT) can improve performance
   - – Limits to what you can fit in context
   - – Complex tasks will probably need gradient steps
2. **Instruction finetuning**
   - \+ Simple and straightforward, generalize to unseen tasks
   - – Collecting demonstrations for so many tasks is expensive
   - – Mismatch between LM objective and human preferences
3. **Reinforcement Learning from Human Feedback (RLHF)**
   - \+ Directly model preferences (cf. language modeling), generalize beyond labeled data
   - – RL is very tricky to get right
   - – Human preferences are fallible; *models* of human preferences even more so
4. **What's next?**