

Neural Netspeak - Exploring the Performance of Transformer Models as Idiomatic Writing Assistants

Betreut von: Matti Wiegmann, Martin Potthast

Fabian Thies

18. September 2020

- 1 Motivation
- 2 Ausgangssituation
- 3 Neural Netspeak
- 4 Erste Ergebnisse
- 5 Ausblick

- 1 Motivation
- 2 Ausgangssituation
- 3 Neural Netspeak
- 4 Erste Ergebnisse
- 5 Ausblick

Netspeak Ein Wort ergibt das andere.

Englisch

Deutsch



i x ↺

how to ? this	Das ? findet ein Wort.
see ... works	Die ... finden mehrere Wörter.
it's [great well]	Die [] vergleichen Alternativen.
and knows #much	Das # findet ähnliche Wörter.
{ more show me }	Die { } prüfen die Reihenfolge.
m...d ? g?p	Das Leerzeichen ist wichtig.

Abbildung 1: Netspeak User Interface

Ausgangspunkt: Netspeak (<https://netspeak.org>)

- Schreibassistent in Form einer Suchmaschine
- Anfragen in eigener Query-Sprache
- Basiert auf indizierten Google N-Grammen ($N \leq 5$)
- Länge der Anfragen u. Ergebnisse auf 5 Wörter beschränkt
- Ergebnisse absteigend sortiert nach Häufigkeit der N-Gramme im Datensatz
- Häufigkeit ist Maß für Gebräuchlichkeit im idiomatischen Kontext

Probleme:

- Wörter in Query müssen in selber Reihenfolge auch als N-Gramm vorliegen
 - Beschränkung auf max. 5 Wörter
- ⇒ Je länger die Anfrage, desto unwahrscheinlicher sind Treffer

Idee: Nutzen von Sprachmodellen

- sind kompositional
- längere Input-Sequenzen möglich
- allerdings: Query-Preprocessing nötig

Zusammensetzung meines Themas:

- **Neural Netspeak:** Version eines Netspeak Backends auf Grundlage eines Sprachmodells
- **Exploring the Performance of:** Implementieren und Testen des Backends, sowie Auswertung der Ergebnisse
- **Transformer Models:** Sprachmodelle mit Encoder-Decoder Struktur
- **as Idiomatic:** Arbeiten mit Text in idiomatischer (von Menschen gesprochener / geschriebener) Sprache
- **Writing Assistants:** Am Beispiel des Schreibassistenten Netspeak

Ziel meiner Arbeit:

- Nützlichkeit eines Language Models zur Verarbeitung Netspeak-Anfragen erforschen
- Implementierung eines neuen Backends für Netspeak, das als Drop-In-Replacement verwendet werden kann
- Testen, Ergebnisse auswerten und Netspeak gegenüberstellen

- 1 Motivation
- 2 Ausgangssituation**
- 3 Neural Netspeak
- 4 Erste Ergebnisse
- 5 Ausblick

Netspeak versteht folgende Tokens in Anfragen:

Token	Beschreibung
?	Ein Zeichen in einem Wort, oder ein ganzes Wort
...	Mehrere Zeichen in einem Wort, oder mehrere ganze Wörter
#	Findet Synonyme für das auf das Token folgende Wort
[]	Bewertet die alternativen Wörter zwischen den Klammern
{ }	Wortreihenfolge

Tabelle 1: Netspeak query language tokens

Beispiele für valide Netspeak-Anfragen:

Token	Beispiel	Bestes Ergebnis
?	how to ? this	how to use this
...	see ... works	see how it works
#	and knows #much	and knows a lot
[]	it's [great well]	it's great
{ }	{ more show me }	show me more

Tabelle 2: Netspeak query examples



Abbildung 2: BERT

Bildquelle: <https://content1.promiflash.de/article-images/gallery1024/bert-aus-der-sesamstrasse.jpg>

- Sehr gutes language model
- Trainiert für zwei Aufgaben auf großen Datensätzen
 - 1 Masked word prediction
 - 2 Next sentence prediction
- Keine Verwendung von Recurrence
- Anpassung durch fine-tuning möglich (z.B. durch Hinzufügen eines Klassifikationslayers)

- 1 Motivation
- 2 Ausgangssituation
- 3 Neural Netspeak**
- 4 Erste Ergebnisse
- 5 Ausblick

Neural Netspeak - Übersicht

- Python-backend abgekapselt in einem docker-Container
- Anfragen kommen am Protobuf-Server an
- Verarbeitung der Queries mit BERT
- Liste von Ergebnissen von Protobuf-Server zurückgegeben

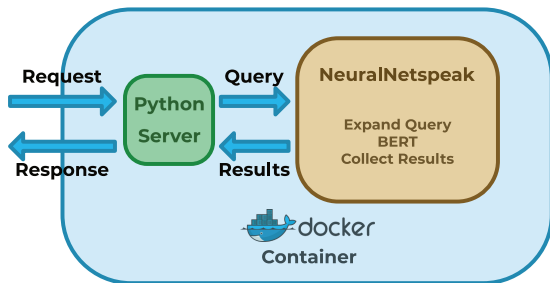


Abbildung 3: Neural Netspeak Architektur

- 1 Konvertierung der Netspeak-Query in Inputs, die von BERT verarbeitet werden können
- 2 Bewertung und Mask-Prediction mit BERT
- 3 Sammeln der Ergebnisse und von BERT zugewiesenen Scores
- 4 Retrieval und Verarbeitung von Synonymen, ggf. erneutes Scoring durch BERT
- 5 Rückgabe der nach Score sortierten Ergebnisse

how to ? this
↓
how to [MASK] this

Abbildung 4: Single mask operator

see ... works



see [MASK] [MASK] works,
see [MASK] [MASK] [MASK] works

Abbildung 5: Multi mask operator

balloon r?se into the sky



balloon rose into the sky,

balloon rise into the sky,

balloon ruse into the sky

Abbildung 6: Single in-word mask operator, multi mask in-word operator analog

and knows #much



and knows much

Abbildung 7: Synonyms operator

it's [great well]



it's great,
it's well

Abbildung 8: Alternatives operator

{ more show me }



more show me,

more me show,

show more me,

show me more,

...

Abbildung 9: Order operator

Neural Netspeak - Query Mask Prediction

Input: how to [MASK] this

Output:

- | | | |
|-----|---------------------|--------------------|
| 1: | how to do this | ⇒ 33.78 (100.00 %) |
| 2: | how to explain this | ⇒ 31.77 (94.03 %) |
| 3: | how to say this | ⇒ 31.07 (91.98 %) |
| 4: | how to solve this | ⇒ 30.74 (90.99 %) |
| 5: | how to fix this | ⇒ 30.69 (90.84 %) |
| 6: | how to use this | ⇒ 30.67 (90.77 %) |
| 7: | how to stop this | ⇒ 30.56 (90.45 %) |
| 8: | how to play this | ⇒ 30.25 (89.54 %) |
| 9: | how to handle this | ⇒ 30.13 (89.20 %) |
| 10: | how to put this | ⇒ 30.01 (88.83 %) |
| 11: | how to write this | ⇒ 29.85 (88.35 %) |

...

Input: and knows #much

Output:

- | | | |
|----|------------------------|--------------------|
| 1: | and knows much | ⇒ 19.01 (100.00 %) |
| 2: | and knows often | ⇒ 18.28 (96.15 %) |
| 3: | and knows a great deal | ⇒ 16.66 (87.63 %) |

Score wird in 3 Schritten berechnet:

- 1 Berechnung eines Base-Scores
- 2 Verrechnung der Mask-Scores
- 3 Verrechnung der Scores der Synonyme

Base-Score für eine Query q :

$$\text{score}_{\text{base}}(q) = \sum_{w \in F \subseteq q} \text{score}_{\text{word}}(w),$$

wobei $\text{score}_{\text{word}}(w)$ den Score eines Wortes w durch BERT berechnet, und $F \subseteq q$ die Menge der finalen / statischen Wörter aus q ist.

Score für eine Kombination aus Wörtern C_j , die für die Gruppe j von [MASK] Tokens aus q von BERT predicted wurden:

$$\text{score}_{\text{mask}}(C_j) = \frac{1}{|C_j|} \cdot \sum_{w \in C_j} \text{score}_{\text{word}}(w)$$

Score für ein Synonym S_i , das aus mehreren Wörtern bestehen kann und für das i -te Wort in q gefunden wurde:

$$\text{score}_{\text{syn}}(S_i) = \frac{1}{|S_i|} \cdot \sum_{w \in S_i} \text{score}_{\text{word}}(w)$$

Score für ein Ergebnis r der Query q :

$$\text{score}(q \mid M_r, S_r) = \text{score}_{\text{base}}(q) + \sum_{M_i \in M_r} \text{score}_{\text{mask}}(M_i) + \sum_{S_i \in S_r} \text{score}_{\text{syn}}(S_i),$$

wobei M_r je eine Kombination von durch BERT predicteten Wörtern für jede [MASK]-Token-Gruppe in q enthält und S_r aus jeweils einem Synonym für alle mit dem #-Operator gekennzeichneten Wörter besteht.

Demo

- Evaluierung auf Grundlage von Datensätzen aus verschiedenen Genres, welche auch die Nutzungsgebiete von Netspeak abbilden
 - ① British National Corpus
 - ② Europarl Corpus
 - ③ The New York Times Corpus
 - ④ Wikipedia Featured Articles
- Prämisse: Texte sind linguistisch sehr gut; Autoren haben jedes Wort gezielt gewählt, um die beste Formulierung zu verwenden

- Automatisierte Query-Generierung aus Sätzen der verschiedenen Datensätze
- Generierung von kurzen und langen Queries

Satz:	Easy writing is still damned hard reading.
Kurz:	{ is writing still } damned
Lang:	Easy { is writing still } damned hard reading
Ergebnis:	writing is still damned

Tabelle 3: Beispiel für einen Query-Datensatz

Bewertungskriterien:

- 1 Recall at 10, 20 und 100
- 2 Güte der Ergebnisse (durchschn. Rang der gesuchten Ergebnisse)
- 3 Qualitative Auswertung ausgewählter Queries

Baseline ist dabei Netspeak

- 1 Motivation
- 2 Ausgangssituation
- 3 Neural Netspeak
- 4 Erste Ergebnisse**
- 5 Ausblick

- Bereits alle vier Datensätze getestet
- Unterscheidung zwischen Ergebnissen nur von kurzen Anfragen und Ergebnissen nur von langen Anfragen
- Recall: ca. 150% Verbesserung
- Durchschnittlicher Rang: 30% Verbesserung
- Durchschnittliche Dauer pro Query: 0,19 Sekunden

Recall	Netspeak	NN Short	NN Full
Recall at 100	27.810 (27,8 %)		
Recall at 20	27.300 (27,3 %)		
Recall at 10	26.977 (27,0 %)		

Tabelle 4: Recall von insgesamt 99.944 Anfragen. NN = Neural Netspeak

Recall	Netspeak	NN Short	NN Full
Recall at 100	27.810 (27,8 %)	69.491 (69,5 %)	
Recall at 20	27.300 (27,3 %)	69.037 (69,1 %)	
Recall at 10	26.977 (27,0 %)	67.799 (67,8 %)	

Tabelle 4: Recall von insgesamt 99.944 Anfragen. NN = Neural Netspeak

Recall	Netspeak	NN Short	NN Full
Recall at 100	27.810 (27,8 %)	69.491 (69,5 %)	76.789 (76,8 %)
Recall at 20	27.300 (27,3 %)	69.037 (69,1 %)	75.279 (75,3 %)
Recall at 10	26.977 (27,0 %)	67.799 (67,8 %)	74.152 (74,2 %)

Tabelle 4: Recall von insgesamt 99.944 Anfragen. NN = Neural Netspeak

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total			
Single Mask			
Single IW-M.			
Multi Mask			
Multi IW-M.			
Synonyms			
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask			
Single IW-M.			
Multi Mask			
Multi IW-M.			
Synonyms			
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.			
Multi Mask			
Multi IW-M.			
Synonyms			
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.	0,04	0,28	0,06
Multi Mask			
Multi IW-M.			
Synonyms			
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.	0,04	0,28	0,06
Multi Mask	—	43,38	28,38
Multi IW-M.			
Synonyms			
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.	0,04	0,28	0,06
Multi Mask	—	43,38	28,38
Multi IW-M.	—	—	—
Synonyms			
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.	0,04	0,28	0,06
Multi Mask	—	43,38	28,38
Multi IW-M.	—	—	—
Synonyms	0,22	0,21	0,05
Order			
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.	0,04	0,28	0,06
Multi Mask	—	43,38	28,38
Multi IW-M.	—	—	—
Synonyms	0,22	0,21	0,05
Order	0,12	0,58	0,12
Alternatives			

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

Erste Ergebnisse - Average Rank

Operator	Netspeak	NN Short	NN Full
Total	1.27	0,92	1,33
Single Mask	6,78	4,21	1,95
Single IW-M.	0,04	0,28	0,06
Multi Mask	—	43,38	28,38
Multi IW-M.	—	—	—
Synonyms	0,22	0,21	0,05
Order	0,12	0,58	0,12
Alternatives	0,06	0,22	0,06

Tabelle 5: Durchschnittliche Platzierung des erwarteten Ergebnisses, nullindiziert

- 1 Motivation
- 2 Ausgangssituation
- 3 Neural Netspeak
- 4 Erste Ergebnisse
- 5 Ausblick**

- Integration von Neural Netspeak in die bestehende Netspeak-Weboberfläche
- Weitere Auswertung

- Entwicklung neuer Query-Operatoren
- Testen verschiedener Sprachmodelle (SpanBERT, RoBERTa, XMLnet, GTP-2, ...)

Vielen Dank für Ihre
Aufmerksamkeit.