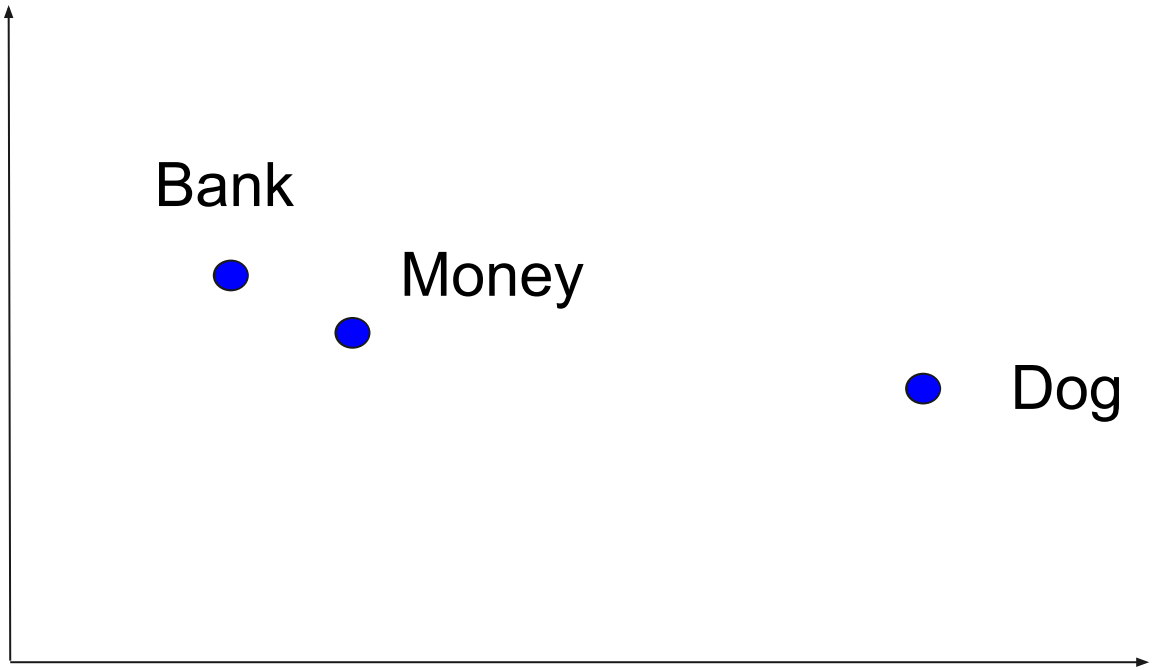
From Contextualized to Static Word Embeddings

Hannes Hansen, M.Sc. Informatik
Betreuer: Niklas Deckers

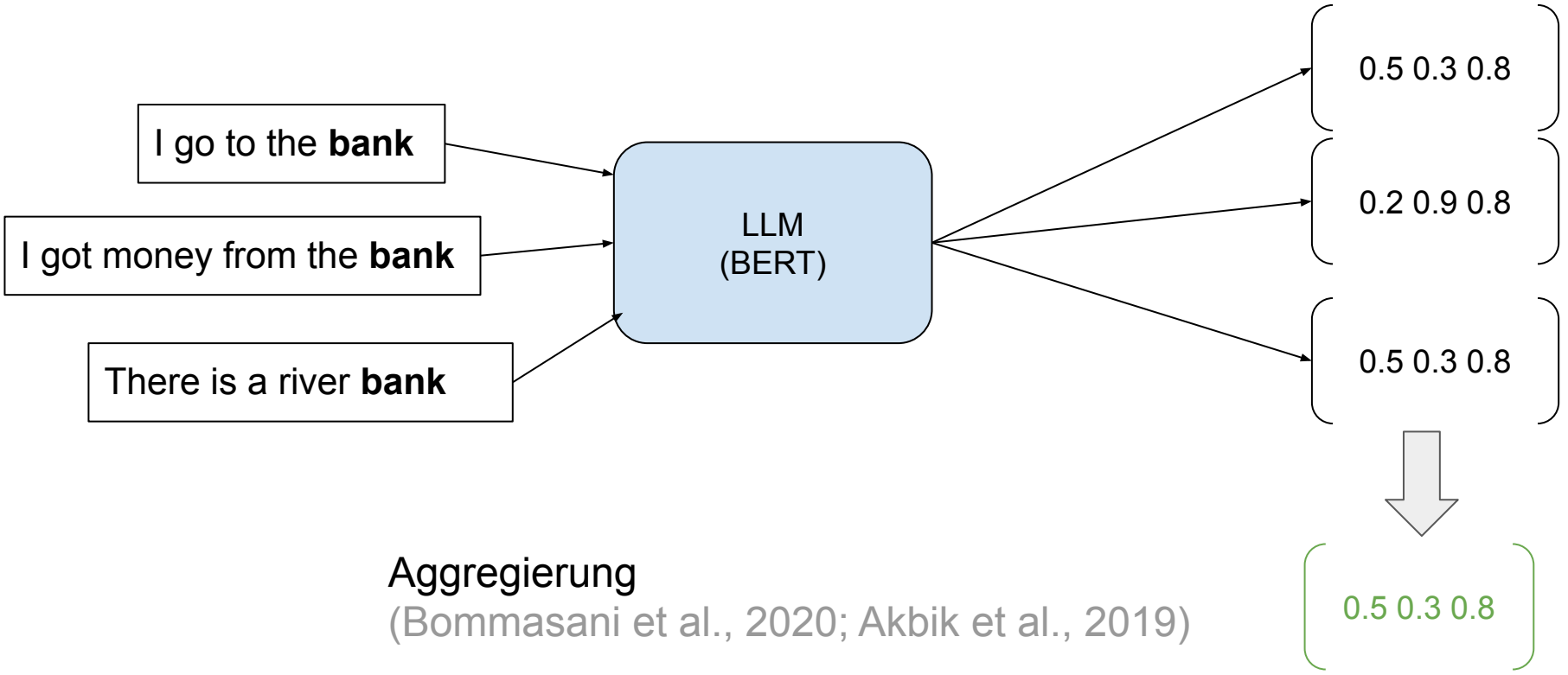
Word Embedding

Numerische Repräsentation eines Wortes, die die semantische Bedeutung widerspiegelt
Beispiele: Word2Vec, GloVe



Large Language Models

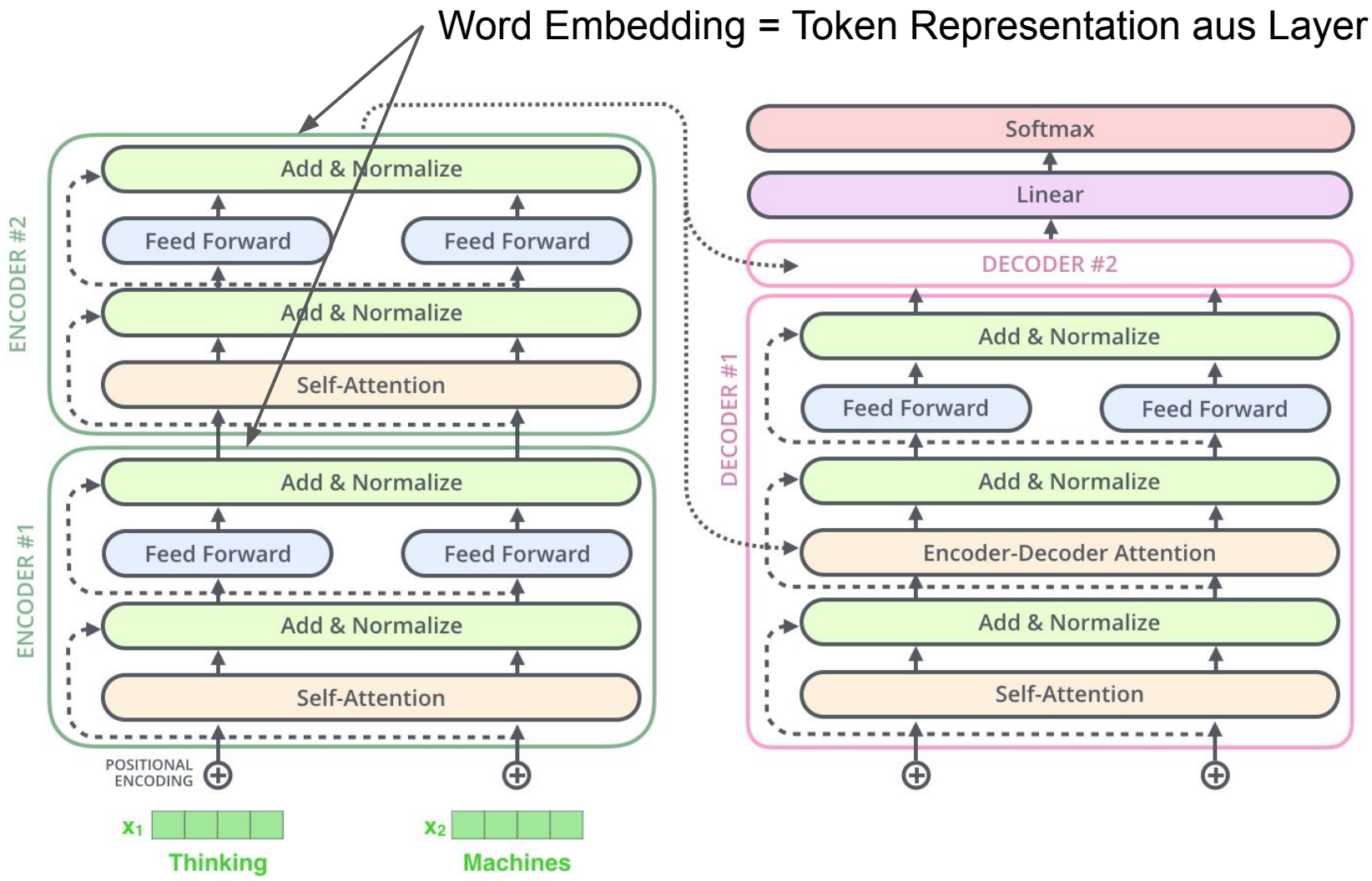
- Transformer basierte Sprachmodelle wie BERT und GPT-2 (Wang et.al, 2018)
- abhängig vom Kontext! -> kontextualisierte Word Embeddings



Aggregierung
(Bommasani et al., 2020; Akbik et al., 2019)

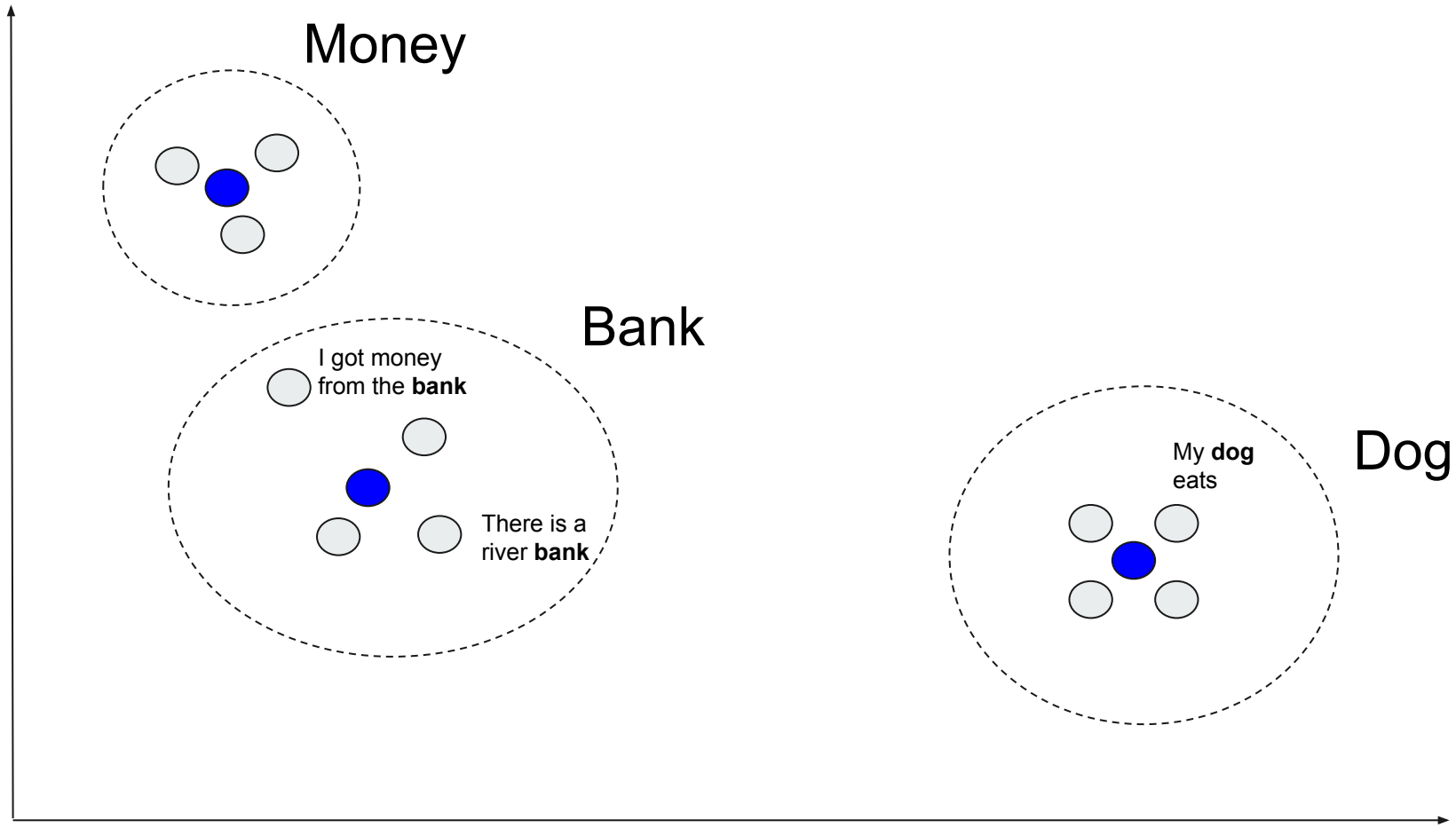
Mittelwert Aggregation über
Kontexte und Sub Tokens

Transformer



Quelle: https://jalammar.github.io/illustrated-transformer/

Aggregierung

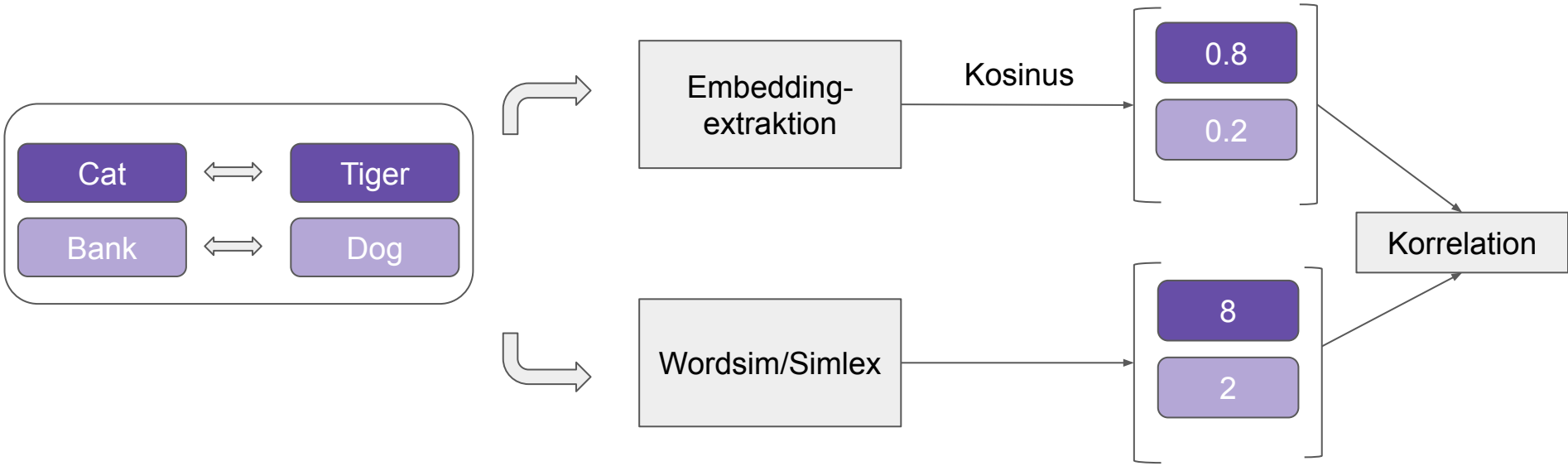


Warum die Kontextualisierung entfernen?

es gibt nicht immer Kontext
und
die Erzeugung ist rechenaufwendig

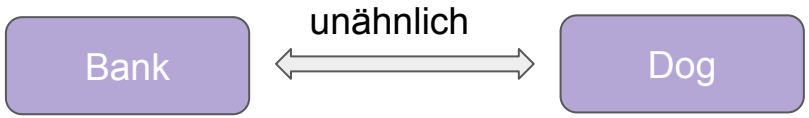
Word Similarity

- Ähnlichkeiten zwischen Wörtern basierend auf menschlichen Einschätzungen
  - Wordsim-353 (Wertebereich 1-10, 353 Paare; Hill et. al, 2015)
  - Simlex-999 (Wertebereich 1-10, 999 Paare; Agirre et. al, 2009)
- Korrelation zwischen Ähnlichkeitsvektoren
- Kosinusähnlichkeit als Ähnlichkeitsmetrik für Word Embeddings

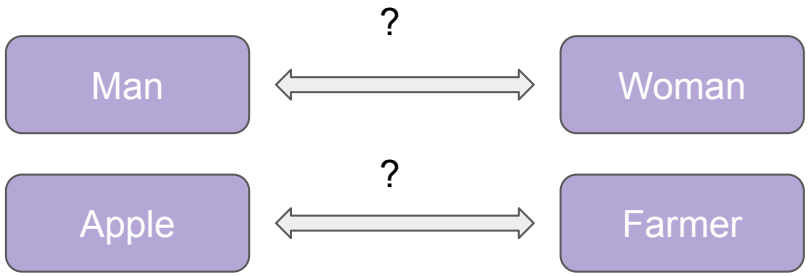


Ähnlichkeit

- klassische Wortähnlichkeit wird durch paarweise Einschätzung durch Menschen erhoben



- Problem: unklar, welche Eigenschaft/Dimension für die Ähnlichkeit entscheidend ist

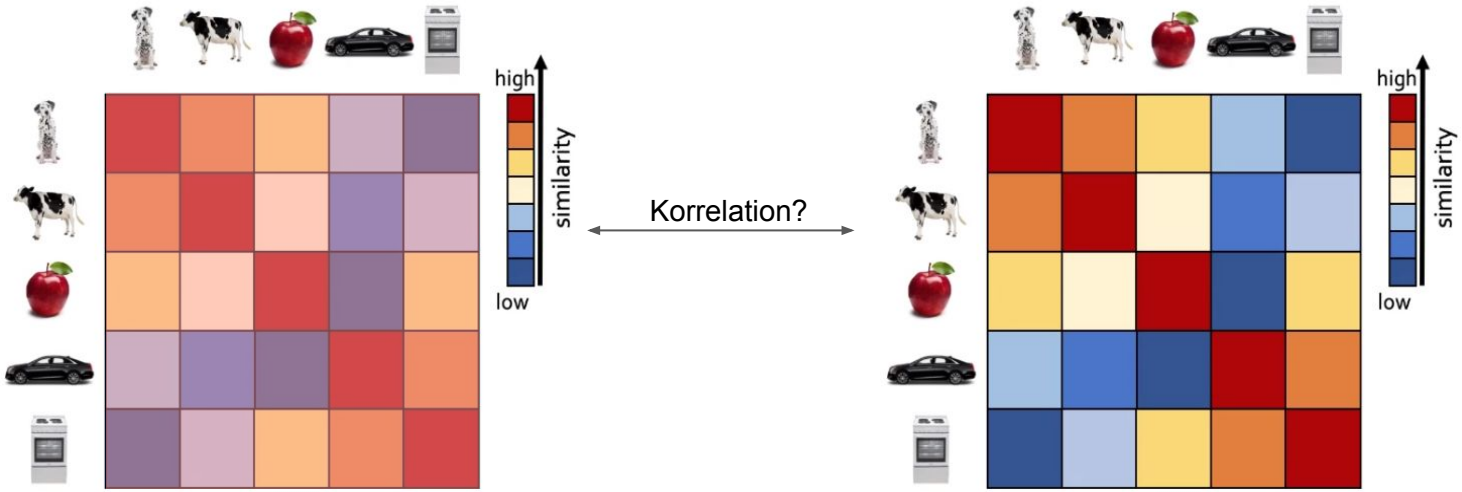


Dimensions-basierte Ähnlichkeit

- THINGS dataset (Hebart et. al. 2019)
  - 1854 Begriffe (für häufig sowie selten vorkommende Objekte)
- 49 interpretierbare Dimensionen wie ***animal-related, round-shaped, valuable*** (Hebart et. al. 2020)
  - basierend auf menschlichen Ähnlichkeitseinschätzungen

	animal-related	**valuable**
Dog	2.36	0.16
Money	0.001	0.97

Dimensions-basierte Ähnlichkeit



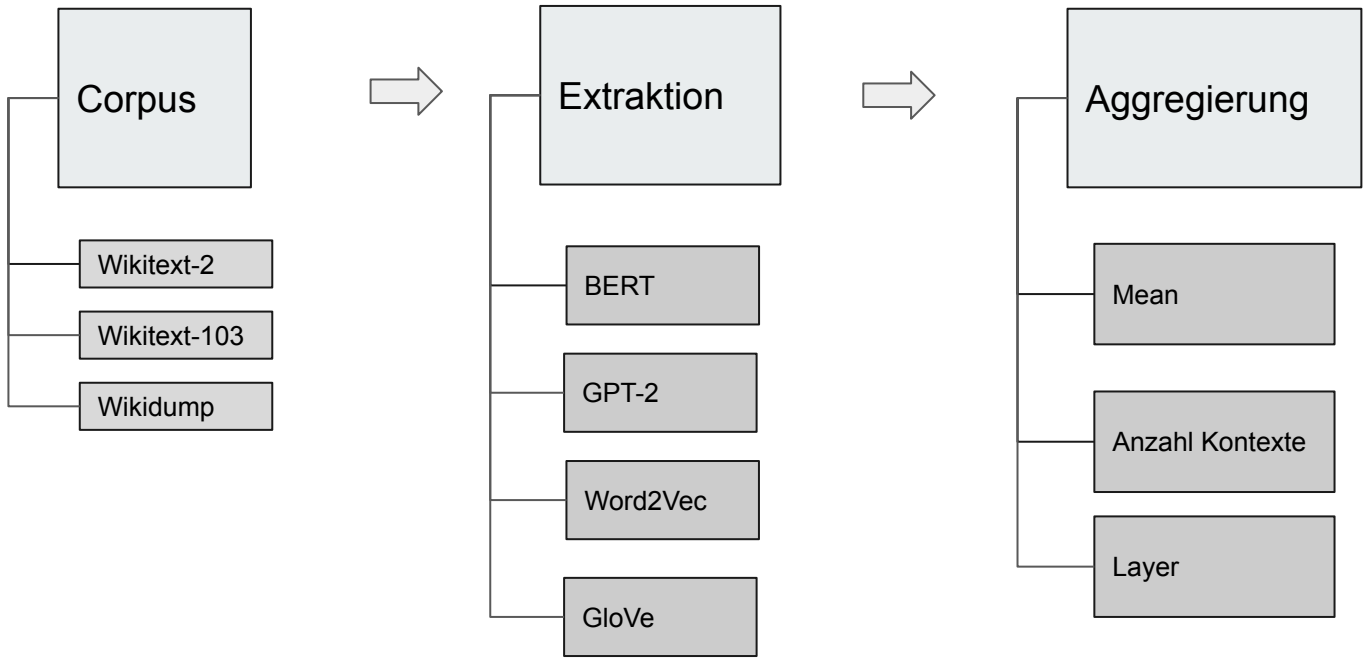
Ähnlichkeit
basierend auf Word Embeddings

THINGS

Forschungsfragen und Ziele

1. Wie gut lassen sich menschliche Ähnlichkeitseinschätzungen anhand von Word Embeddings nachempfinden?
   a. Word Similarity (Wordsim-353, Simlex-999)
   b. Dimensionen-basierte Ähnlichkeit (THINGS)
   c. Wie ist der Einfluss von Aggregation und verschiedenen Hyperparametern?
2. Können dekontextualisierte Word Embeddings mit Hilfe von Transformern erstellt werden?
   a. mit dem Ziel diese Ähnlichkeit nachzubilden
   b. Was ist der Effekt von Aggregation?

Extraktion & Aggregation



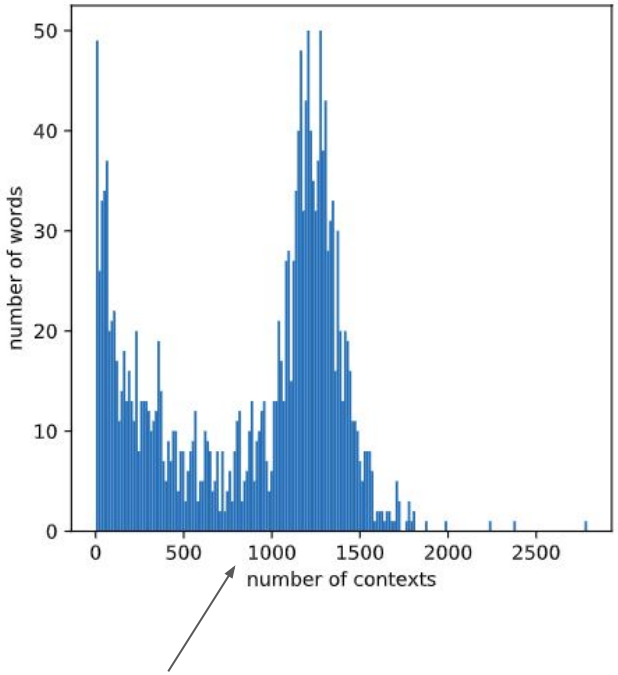
Problem: Corpus - Worthäufigkeiten

Wieviele von n annotierten Wörtern kommen im Corpus vor?	THINGS n = 1.854	Wordsim-353 ∪ Simlex-999 n = 1.341
Wikitext-2	770	1.247
Wikitext-103	1.728	**1.341**
Wikidumps	**1.854**	**1.341**

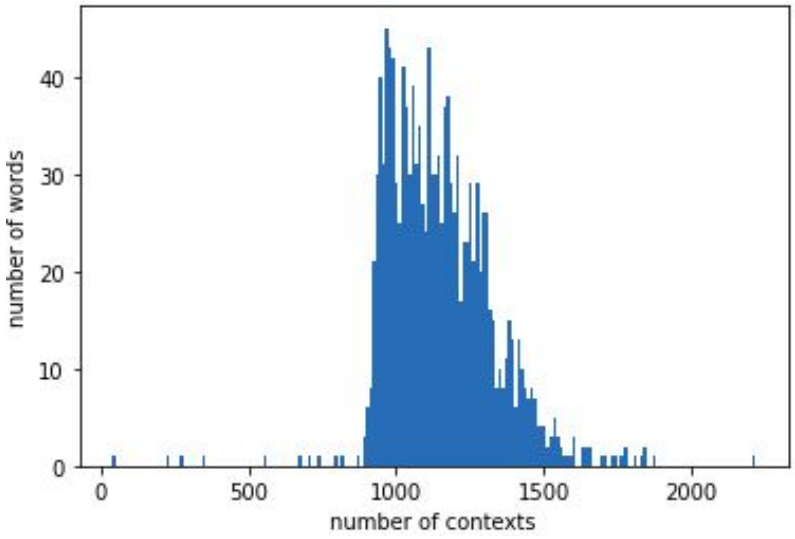
Entscheidung für Wikidumps, da es alle Wörter abgedeckt sind und mehr Kontexte liefert

Wikidumps

THINGS



Wordsim/Simlex

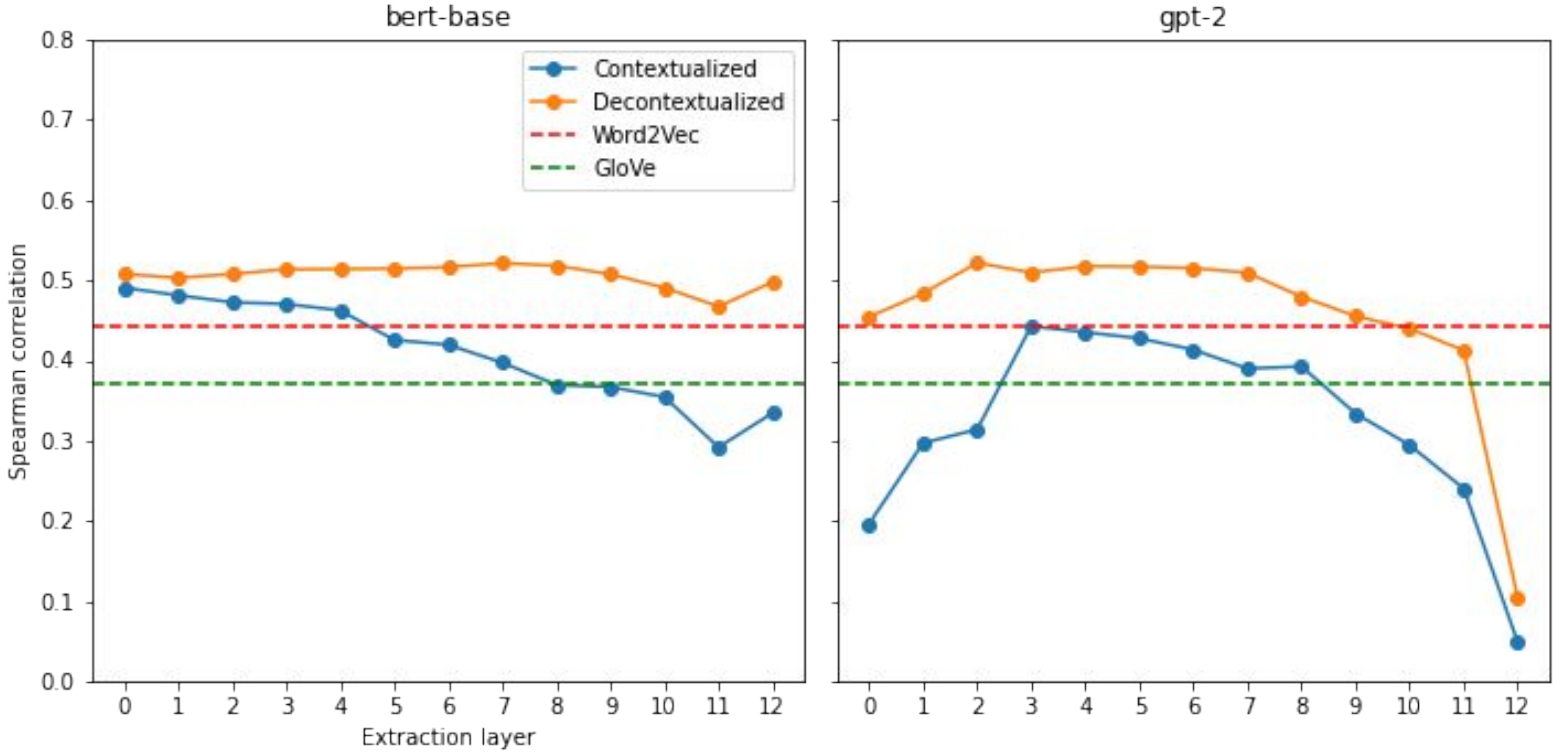


Ziel: 1000 Kontexte pro Wort

Schwer zu erreichen für selten verwendete Wörter

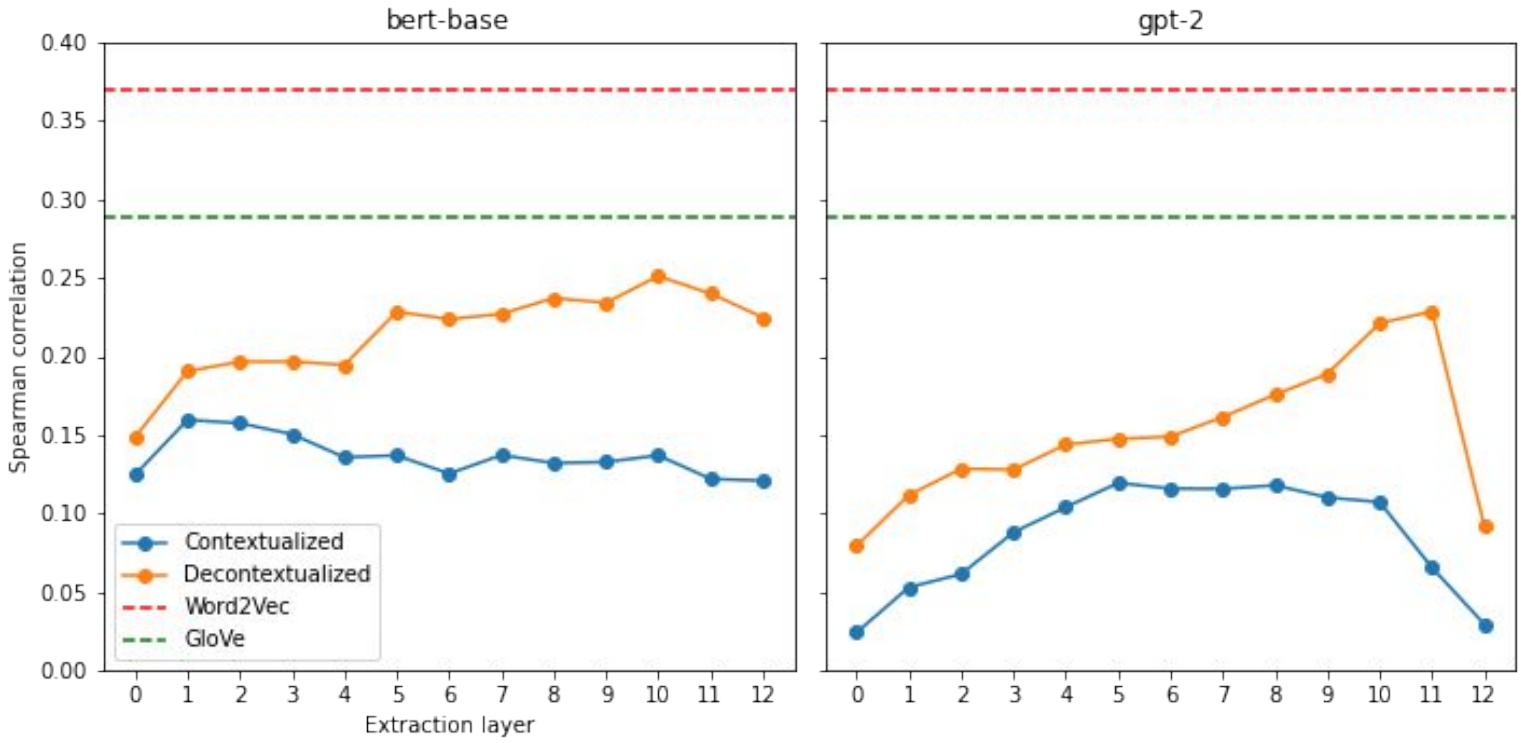
Ergebnisse - Simlex-999

- Ergebnisse:
  - Höhere Korrelation für Transformer-basierte Embeddings
- Diskussion:
  - Transformer-basierte Embeddings, vor allem aggregiert, spiegeln direkt erhobene menschliche Einschätzungen am besten wider (Bommasani et al., 2020)



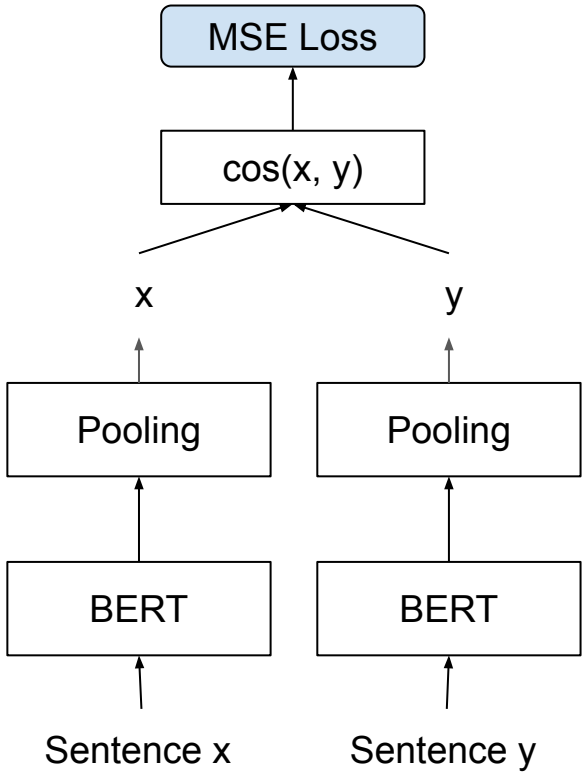
Ergebnisse - THINGS

- Ergebnisse:
  - Transformer-basierte Embeddings erreichen niedrigere Korrelation als Word2Vec-Embeddings
  - Aggregation führt zu höherer Korrelation
- Diskussion:
  - Word2Vec-Embeddings reflektieren die dimensionsbasierte Ähnlichkeit besser



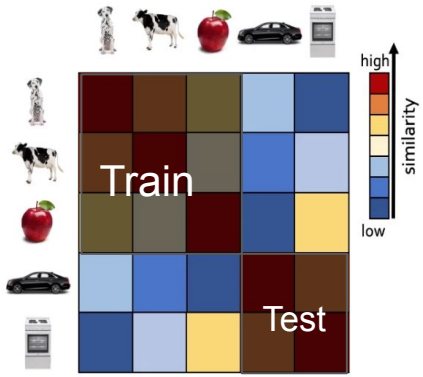
Retraining eines Heads

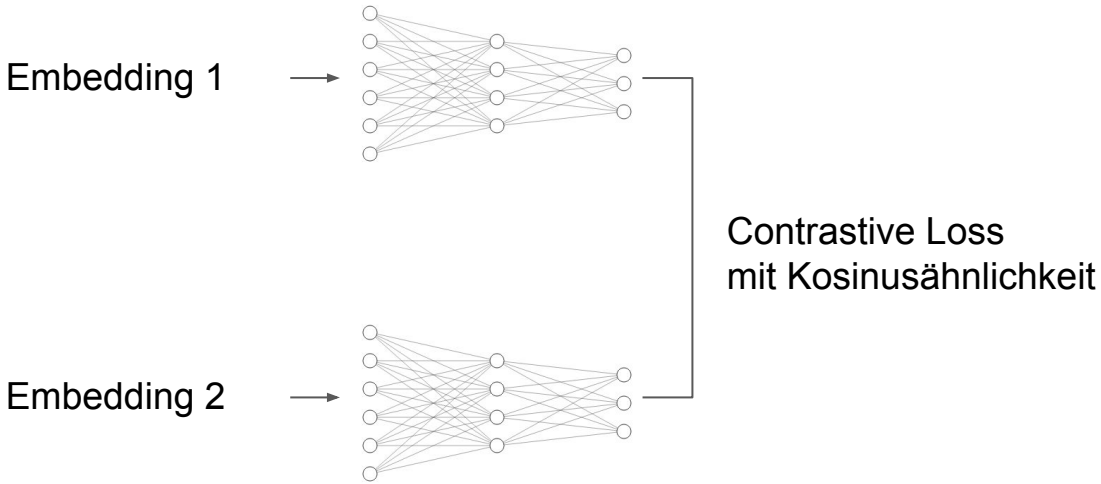
- Motivation:
  - Herausfinden, ob dimensions-basierte Ähnlichkeit wirklich schlechter reflektiert ist oder die Information in pretrained Embeddings nicht ausgenutzt wird
  - Modell zur Vorhersage von Ähnlichkeitswerten erstellen (als Ersatz für Menschen)
- Ansatz:
  - Sentence-BERT: Siamese Netzwerk mit BERT und verschiedene Loss Funktionen (Reimers et al., 2019)
  - kleines Netzwerk als Head auf den Embeddings
  - Groundtruth: menschliche Ähnlichkeitseinschätzung



Retraining eines Heads

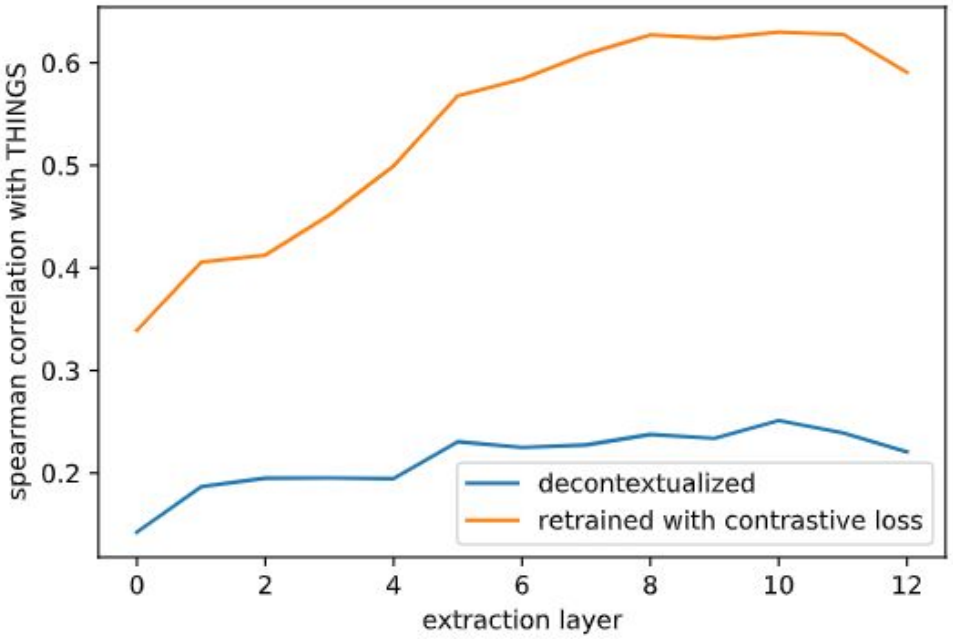
- Contrastive Loss:
  - Distanz ähnlicher Paare verkleinern
  - Distanz unähnlicher Paare vergrößern
- Architektur:
  - 3 Layer (768x600x300)
  - 1 ReLU
- Train-Test Split
- BERT-base





Retraining eines Heads

- Ergebnisse:
  - Korrelation von retrained Embeddings steigt auf 0.6
- Diskussion:
  - Modell zur Vorhersage basierend auf BERT Word Embeddings
  -



Zusammenfassung

- Methode zur Erstellung von dekontextualisierten Word Embeddings
- Intrinsische Evaluierungen von verschiedene Word Embeddings durchgeführt
  - Wortähnlichkeiten (Simlex-999, Wordsim-353)
  - Neue Ähnlichkeitsmetrik angewendet (THINGS)
  - Effekt von Aggregierung untersucht

- Ergebnisse:
  - Kontextualisierte Embeddings kodieren Wortähnlichkeit bereits sehr gut (Wordsim-353/Simlex-999)
  - Aggregierung erhöht Korrelation mit allen Ähnlichkeitsmetriken über alle Modelle und Hyperparameter hinweg

Diskussion

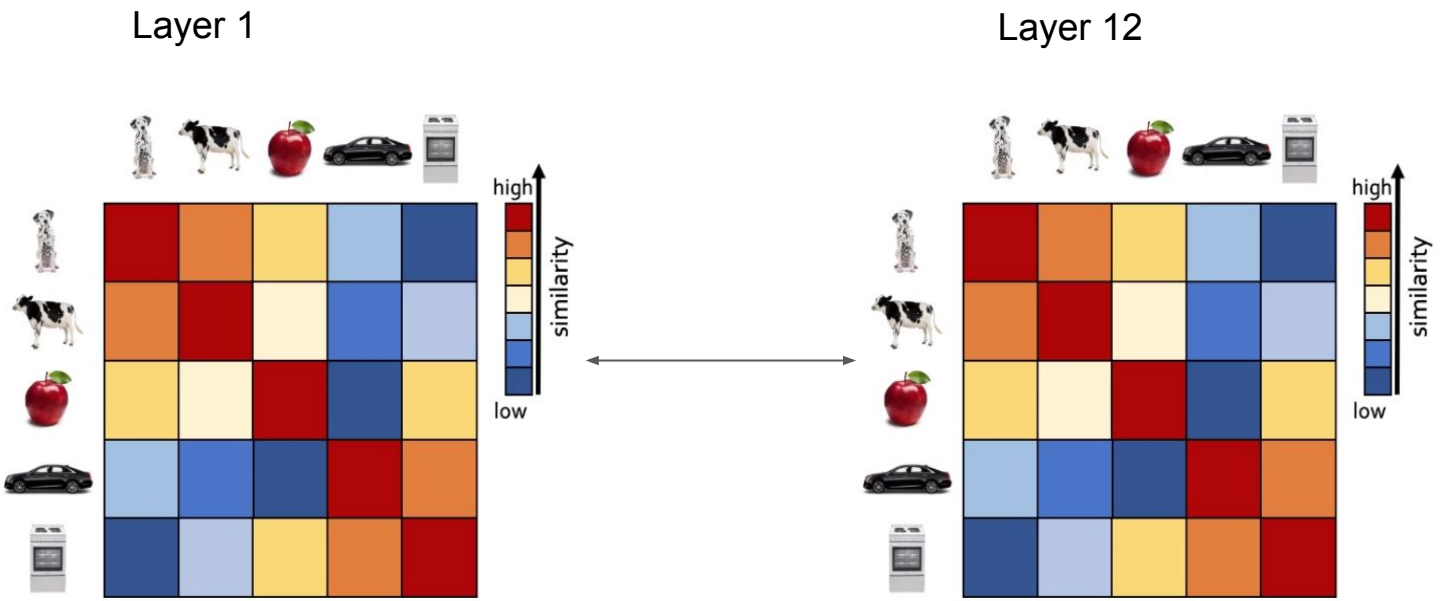
1. Warum sinkt die Korrelation in späteren Layern?
   - wegen (zu) hoher Kontextualisierung? (Ethayarajh et. al, 2019)
   - wegen Lernens von Informationen für Training Objective? (Li et. al, 2020)
2. Warum hilft Aggregierung?
   - verschiedene Kontexte spiegeln verschiedene Bedeutungen wieder?
   - Aggregation wirkt Kontextualisierung entgegen?
3. Warum ist die Korrelation mit THINGS Ähnlichkeit schlechter?
   - THINGS auch selten vorkommende Objekte beinhaltet?
   - Viele visuell geprägte Dimensionen - schlechter in Text kodierbar?
   - nur Substantive

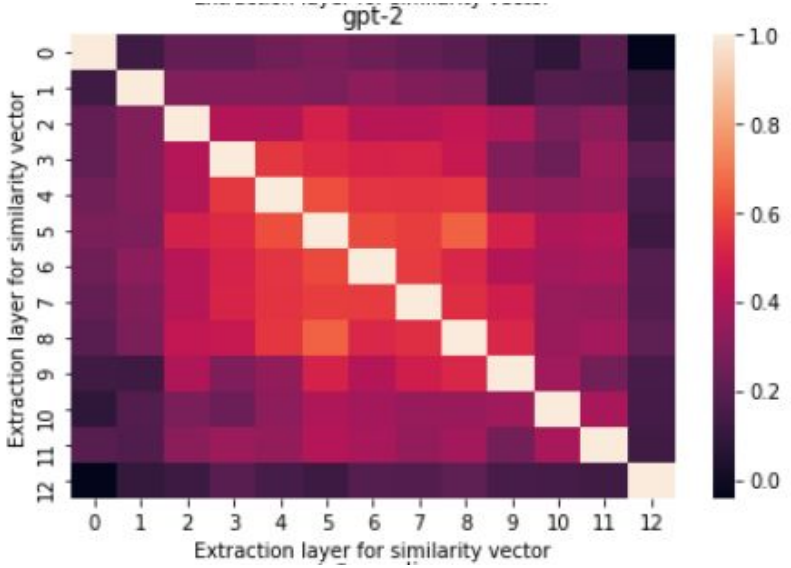
Ausblick

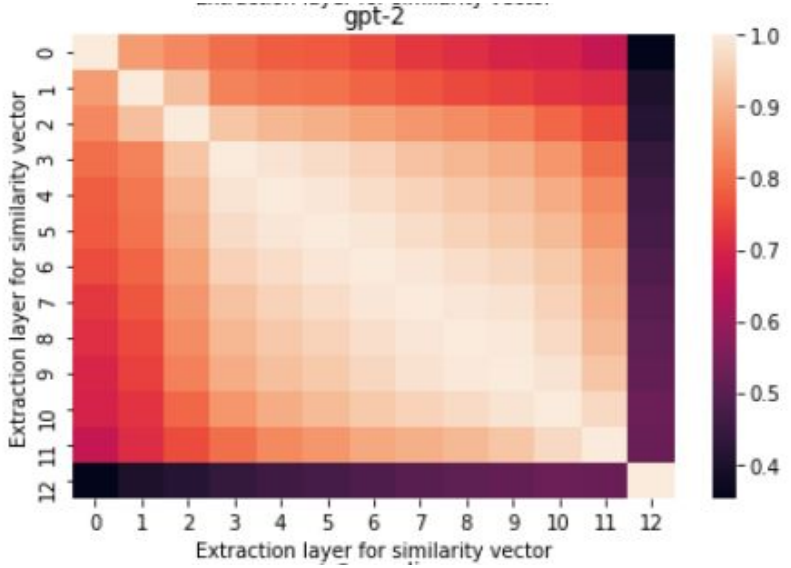
- Vorhersage der Dimensionswerte
- Einfluss von Textcorpora
- End-to-End finetuning
- Einfluss von Kontext Fenster
  - Word2vec 5-10
  - BERT 512 tokens

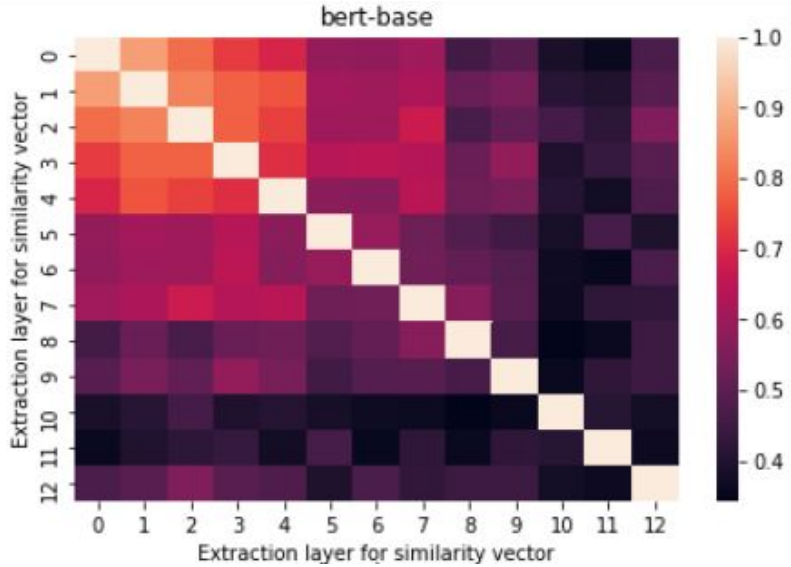
Backup

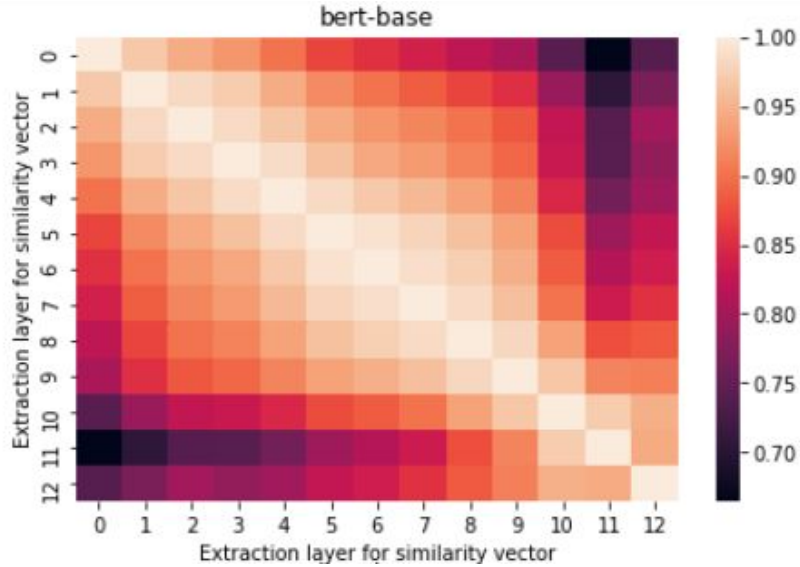
Kodieren Layer Ähnlichkeit anders?




gpt-2
Extraction layer for similarity vector
Extraction layer for similarity vector


gpt-2
Extraction layer for similarity vector
Extraction layer for similarity vector

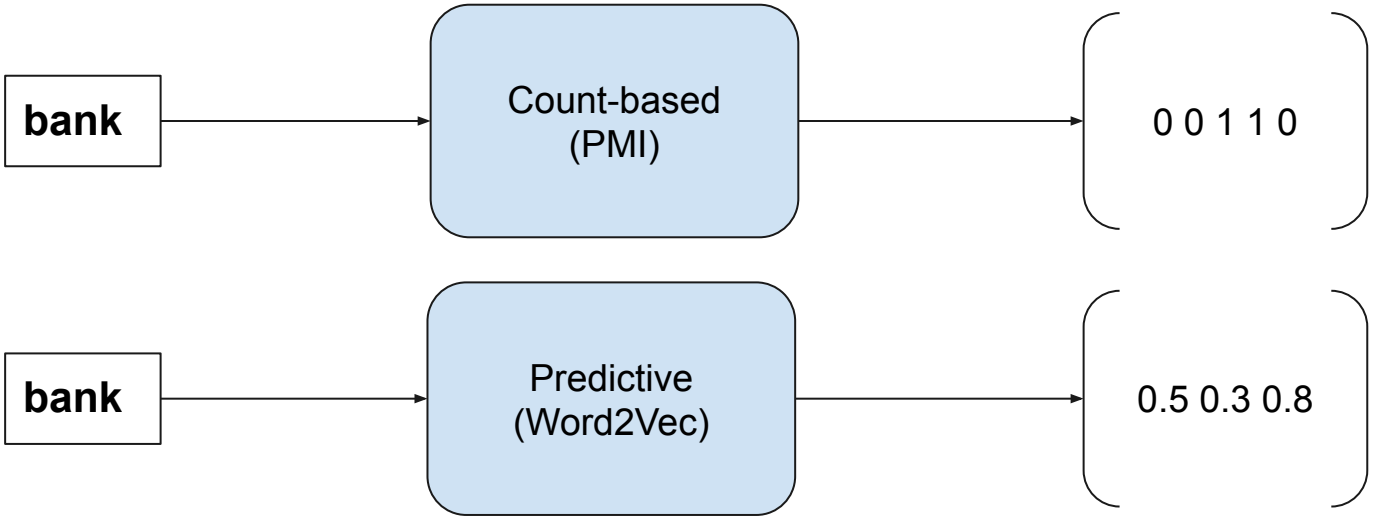

bert-base
Extraction layer for similarity vector
Extraction layer for similarity vector


bert-base
Extraction layer for similarity vector
Extraction layer for similarity vector

Diskussion - Word Similarity

1. einfache contextualized embeddings kodieren Wortähnlichkeit
2. aber schlechter in späteren Layern
   - wegen Anisotropie und/oder wegen (zu) hoher Kontextualisierung? (Ethayarajh, Kawin 2019)
   - wegen Lernens von Informationen für Training Objective?
3. mehr Kontext verbessert allgemein
   - verschiedene Kontexte spiegeln verschiedene Bedeutungen wieder?
   - Aggregation wirkt Anisotropie entgegen?
   - aber limitiert
4. GPT-2 insgesamt schlechter geeignet für Word Embeddings
   - unidirectional
5. Empfehlung:
   - Wenn Word Embeddings, dann entweder
     - Aggregierte Embeddings aus späteren Layern aus Encoder Modellen
     - oder statische Embeddings aus ersten Layer
   - kontextualisierte Word Embeddings nur wenn Satzkontext wichtig ist

Word Embeddings

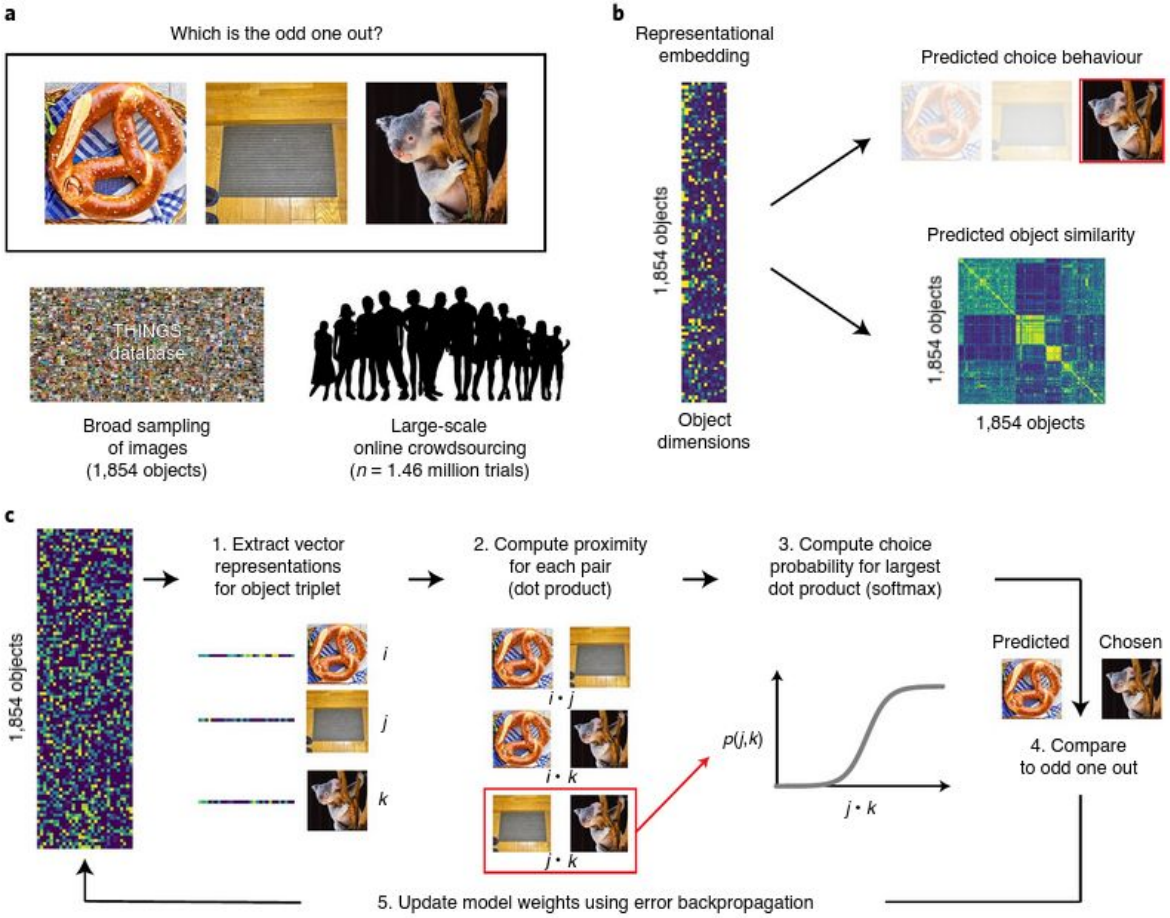


Statisch

Unterschiede Similarity

	Wordsim (0-10)	**Simlex (0-10)**	**THINGS (0-1)**
clothes - closet	8.0	3.2	0.87
woman - man	7.0	3.3	0.88
dog - cat	-	1.8	0.92
money - bank	8.5	-	0.4
train - car	6.3	-	0.96

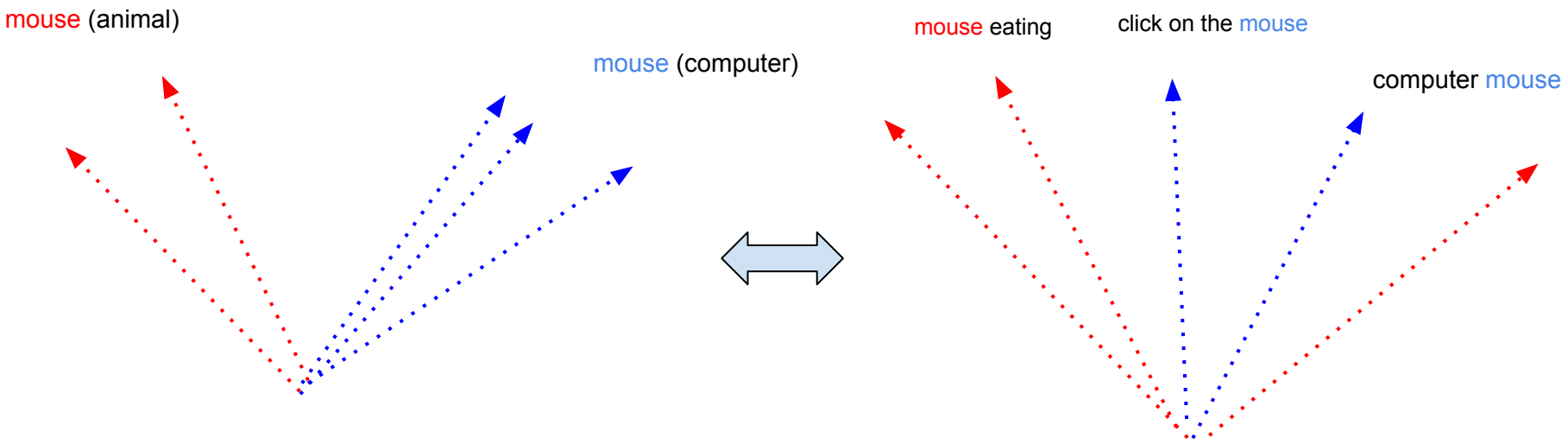
THINGS Dimensionen



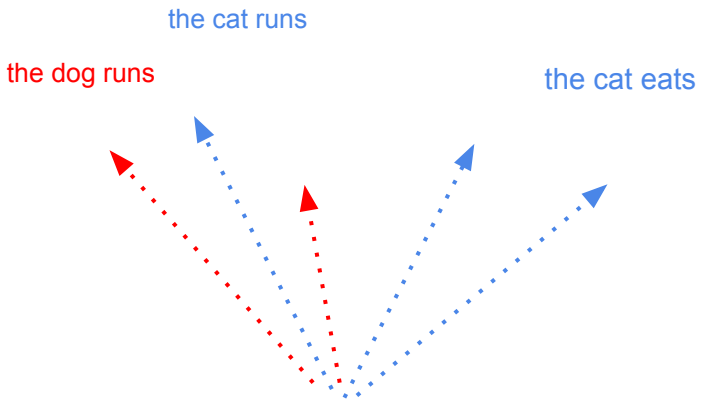
- 1854x90 embedding space
- Cross Entropy Loss + L1 Regularisierung

Quelle: Hebart et al. 2020

Kontextualisierung



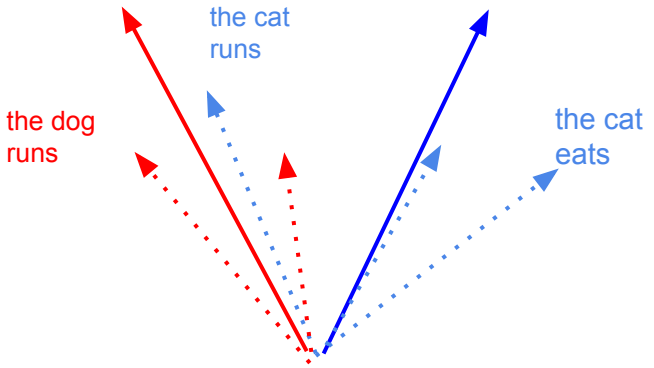
Ethayarajh, Kawin 2019

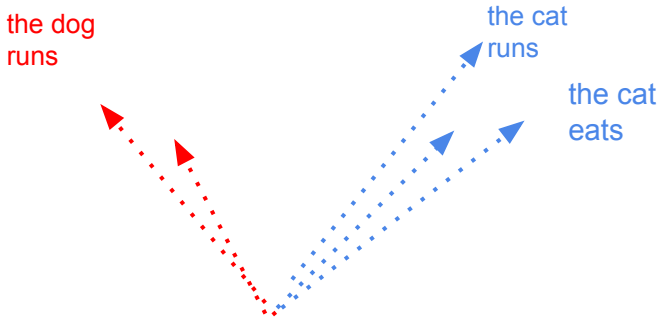


Ethayarajh et al. 2019

Aggregierung

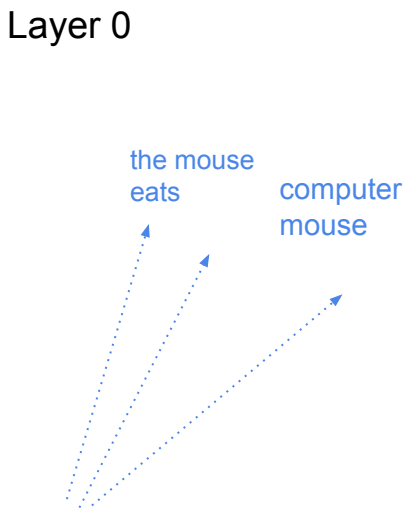
Finetuning

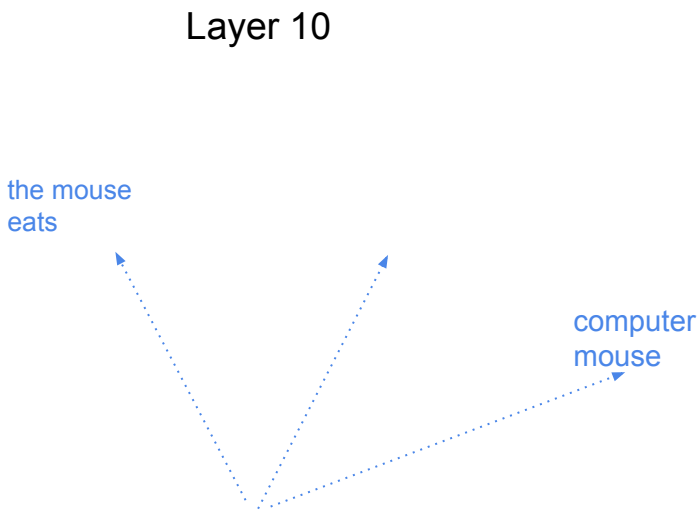




Reimers et al. 2019

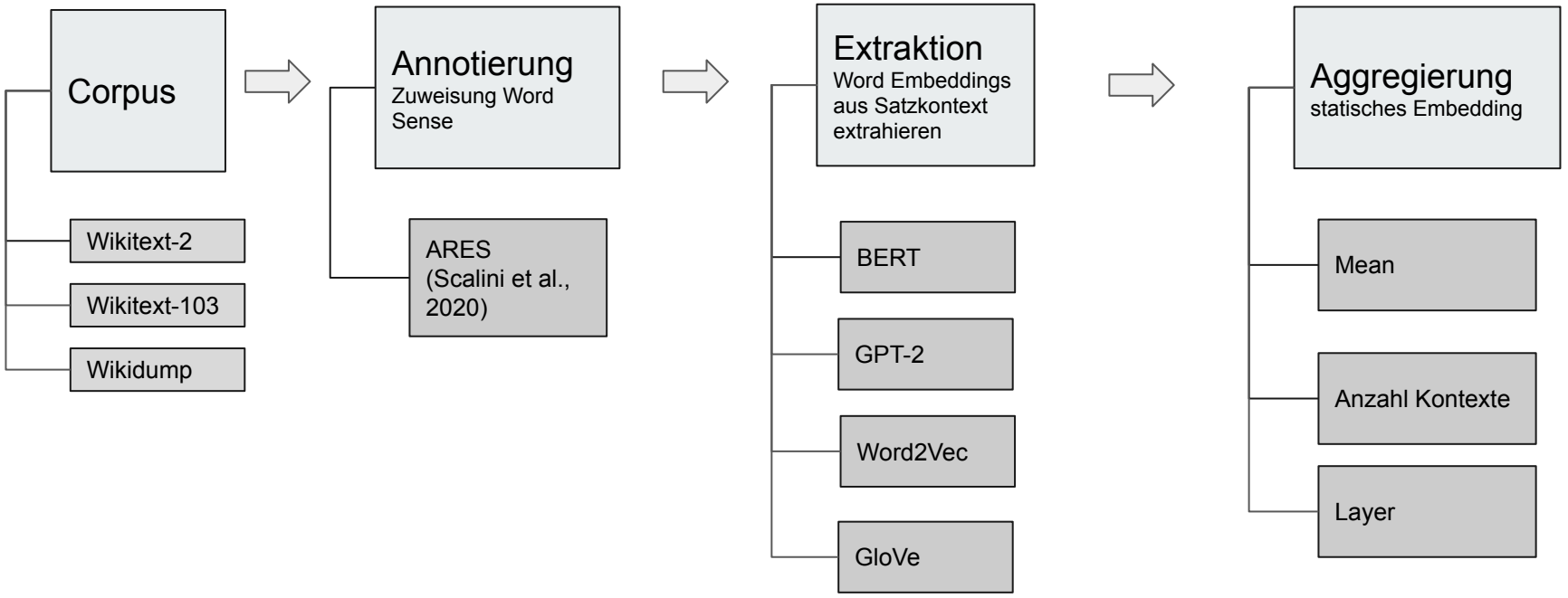
Kontextualisierung





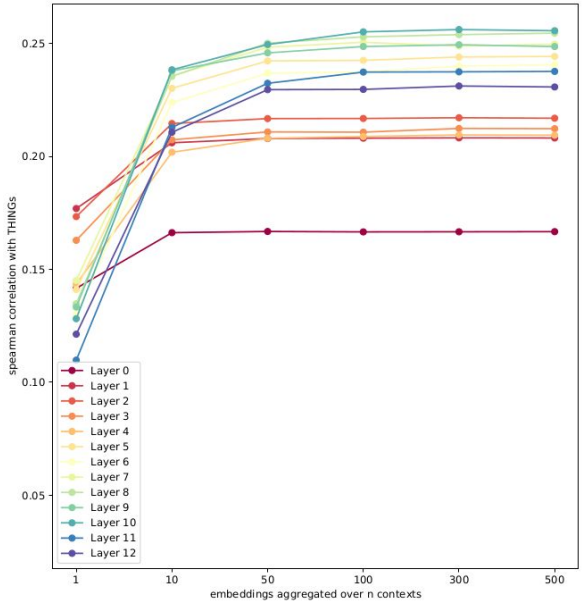
Ethayarajh et al. 2019

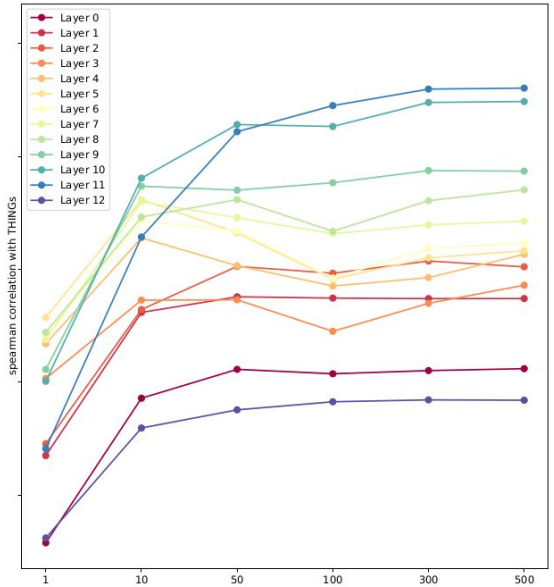
Extraktion & Aggregation



Anzahl Kontexte und Layer

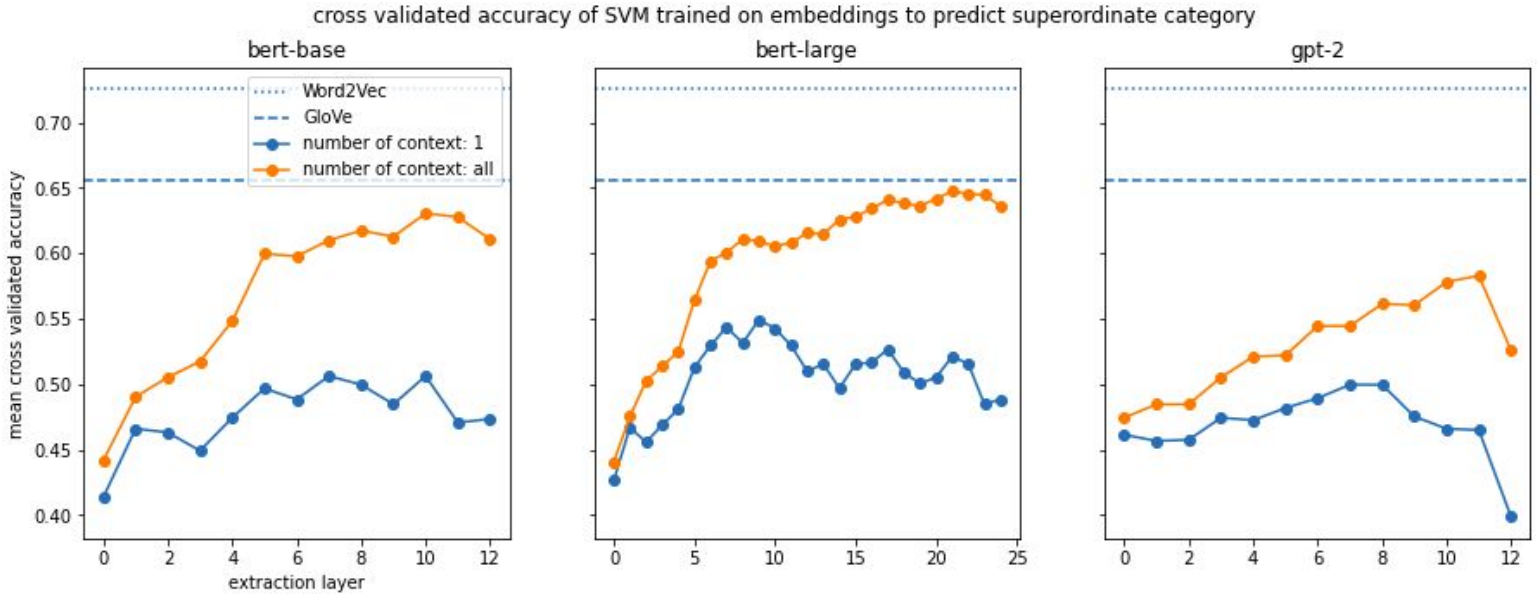
- Motivation:
  - Wieviele Kontexte?
- Ansatz:
  - Wörter mit > 500 Kontexten
  - then use 1-1000 contexts to aggregate
- Ergebnisse
  - mehr Kontexte erhöhen
  - vor allem in späteren Layern
  - aber ab 50-100 keine Veränderung mehr
- Diskussion:
  - Aggregierung verbessert die Embeddings nicht unbegrenzt

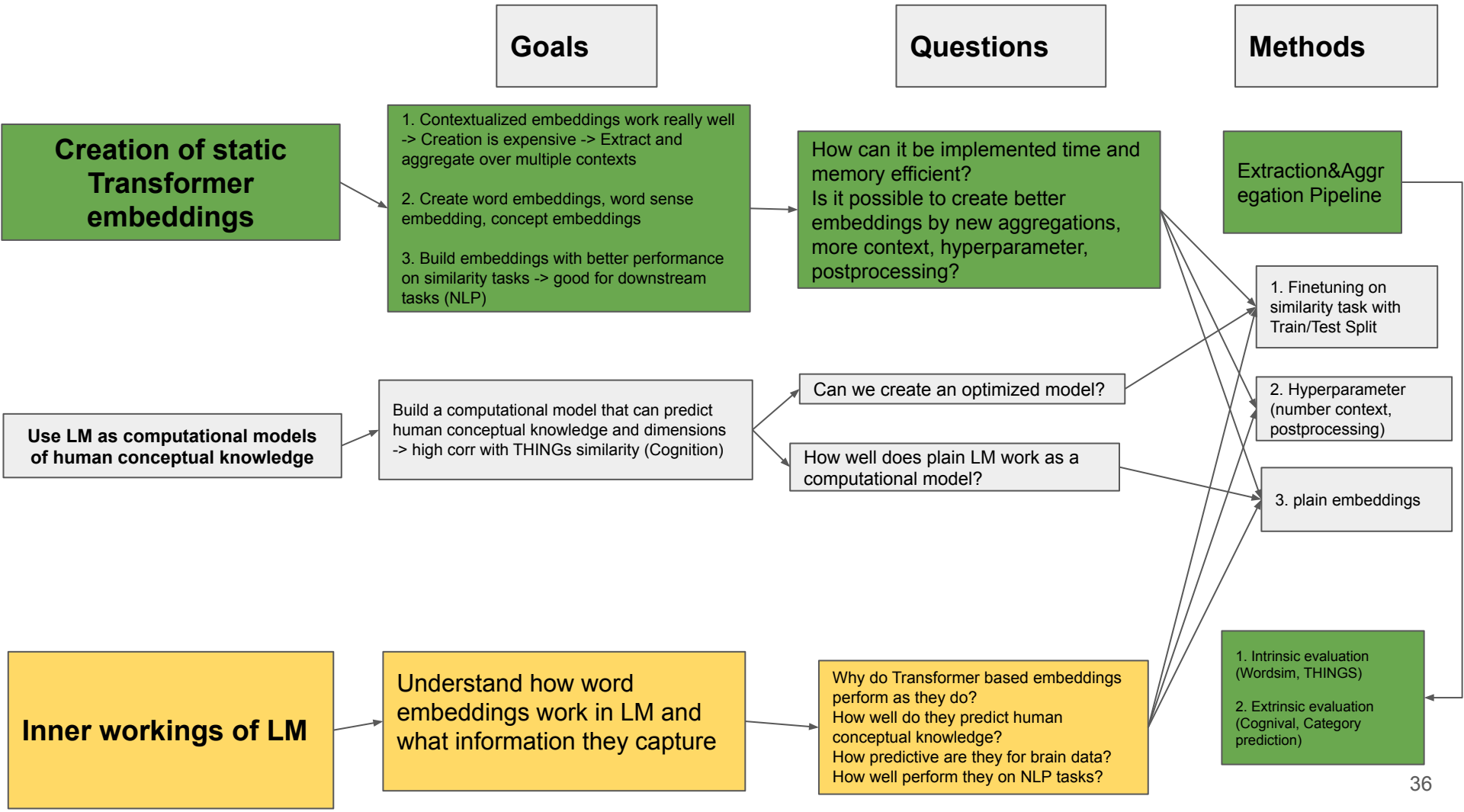




Category prediction

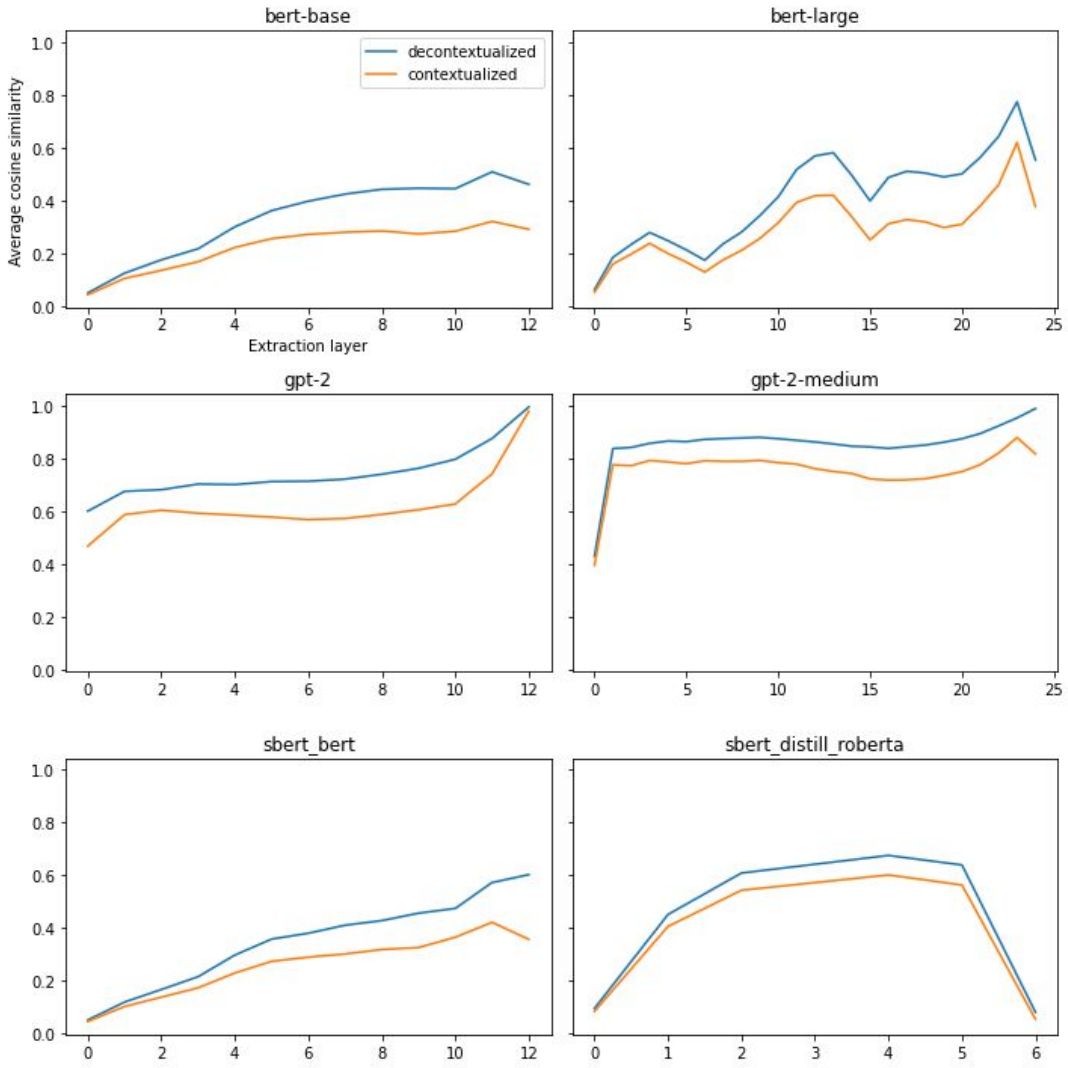
- Results:
  - Transformer based embeddings perform worse than static embeddings
  - decontextualized embedding perform better than single contextualized embeddings
  - later layers are more predictive
  - especially when aggregation over contexts is used
- Discussion:
  - Word2Vec captures global similarity really well
  - later layers seem to encode more information about global conceptual similarity



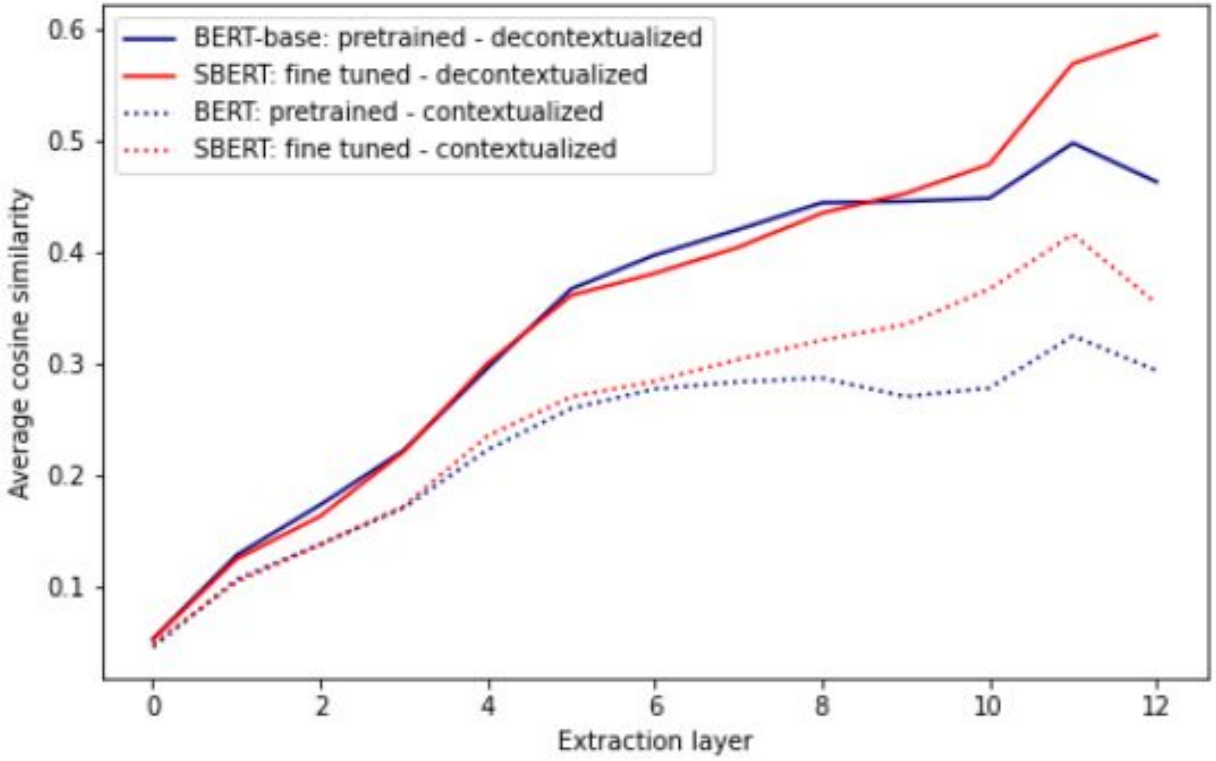

Goals
Questions
Methods
Creation of static Transformer embeddings
1. Contextualized embeddings work really well -> Creation is expensive -> Extract and aggregate over multiple contexts
2. Create word embeddings, word sense embedding, concept embeddings
3. Build embeddings with better performance on similarity tasks -> good for downstream tasks (NLP)
How can it be implemented time and memory efficient?
Is it possible to create better embeddings by new aggregations, more context, hyperparameter, postprocessing?
Extraction&Aggregation Pipeline
1. Finetuning on similarity task with Train/Test Split
2. Hyperparameter (number context, postprocessing)
3. plain embeddings
Use LM as computational models of human conceptual knowledge
Build a computational model that can predict human conceptual knowledge and dimensions -> high corr with THINGs similarity (Cognition)
Can we create an optimized model?
How well does plain LM work as a computational model?
Inner workings of LM
Understand how word embeddings work in LM and what information they capture
Why do Transformer based embeddings perform as they do?
How well do they predict human conceptual knowledge?
How predictive are they for brain data?
How well perform they on NLP tasks?
1. Intrinsic evaluation (Wordsim, THINGS)
2. Extrinsic evaluation (Cognival, Category prediction)

36

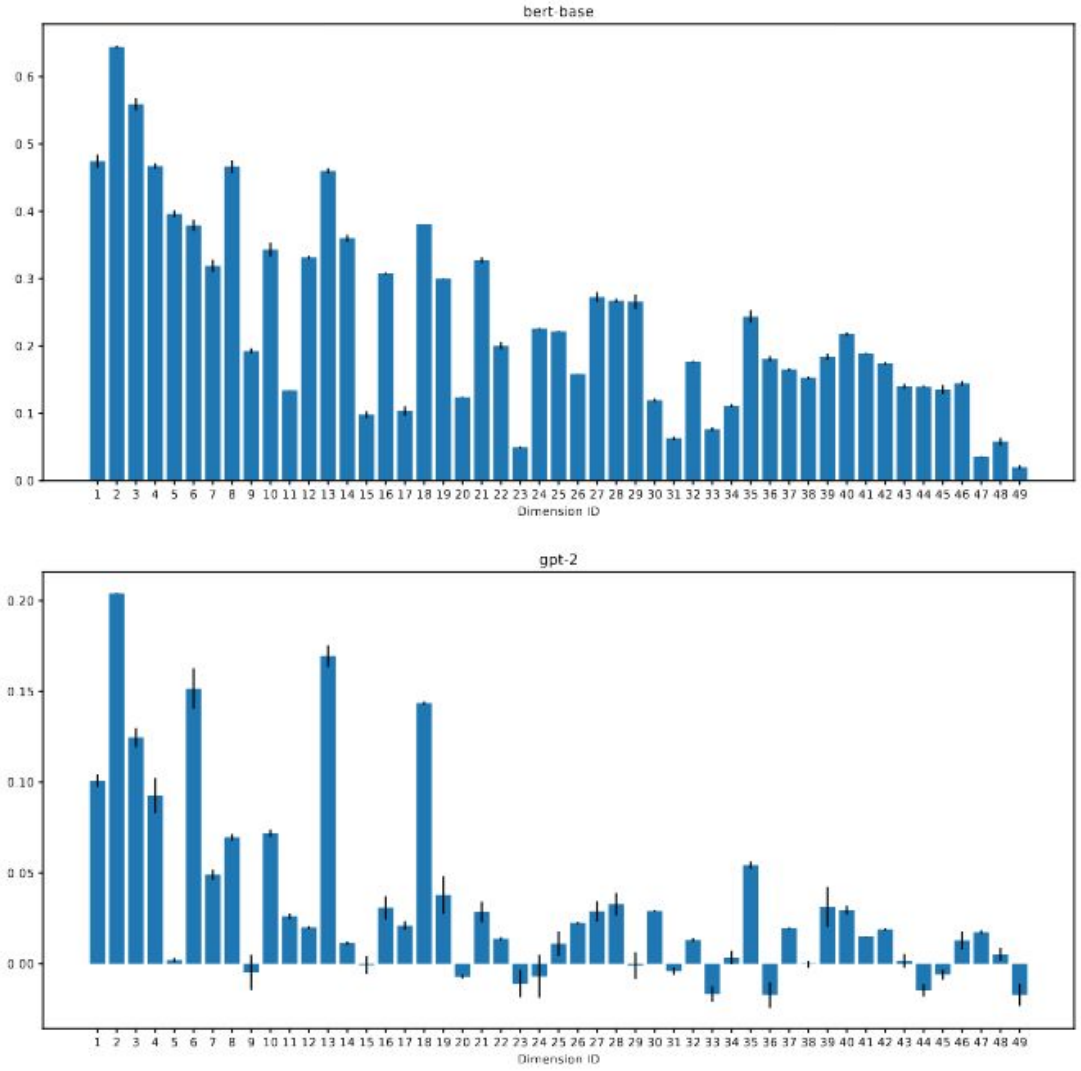
Anisotropy - Modelle



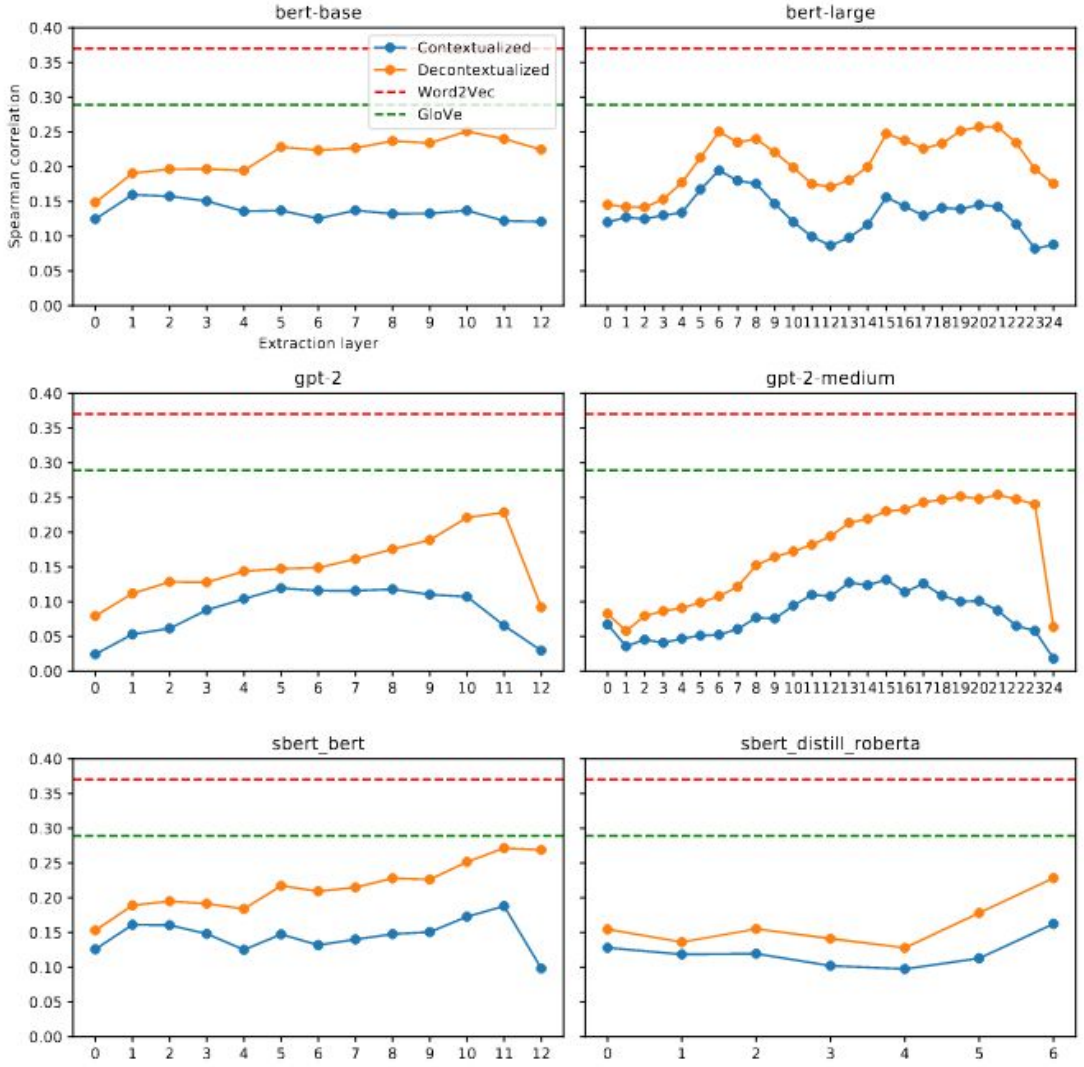
Anisotropy - Finetuning



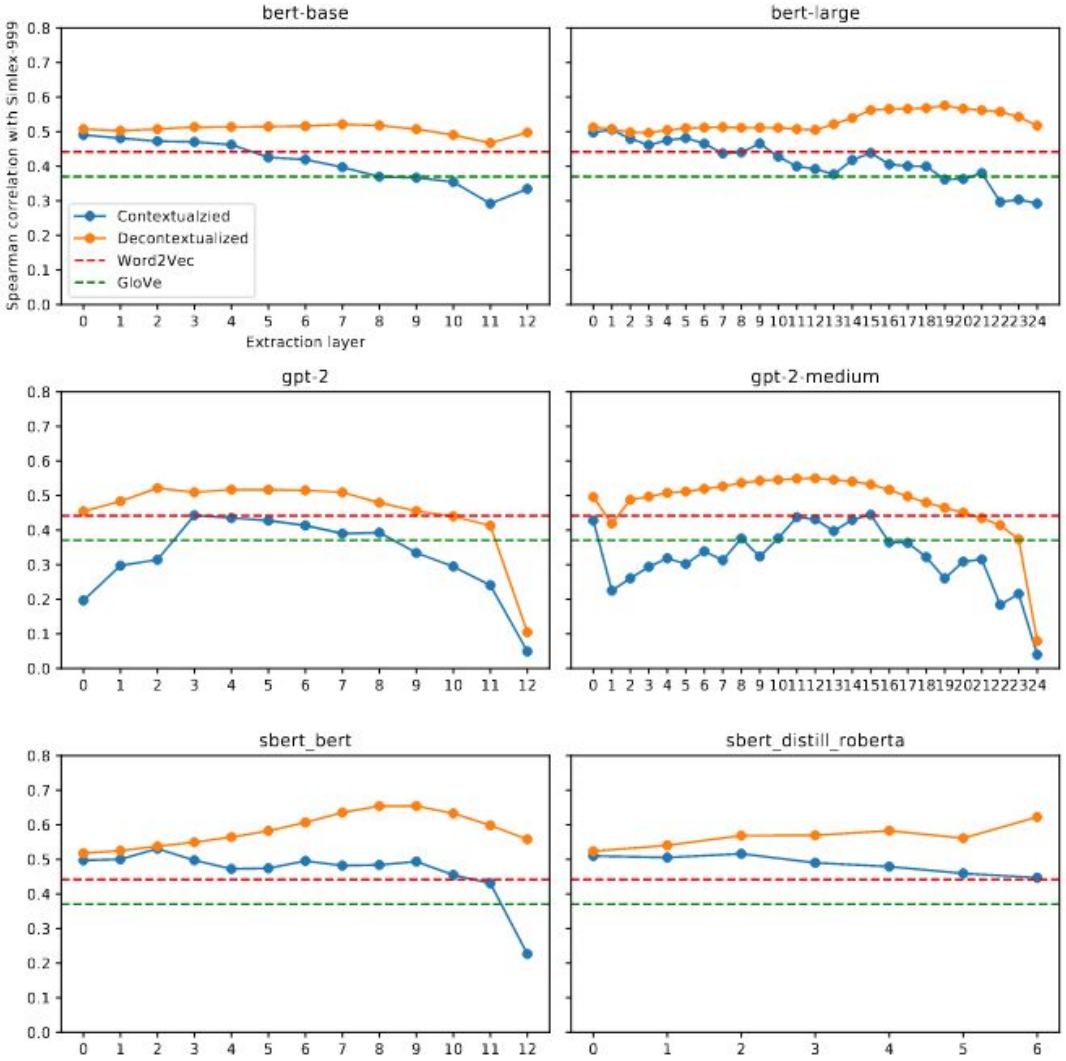
Dimensionsvorhersage



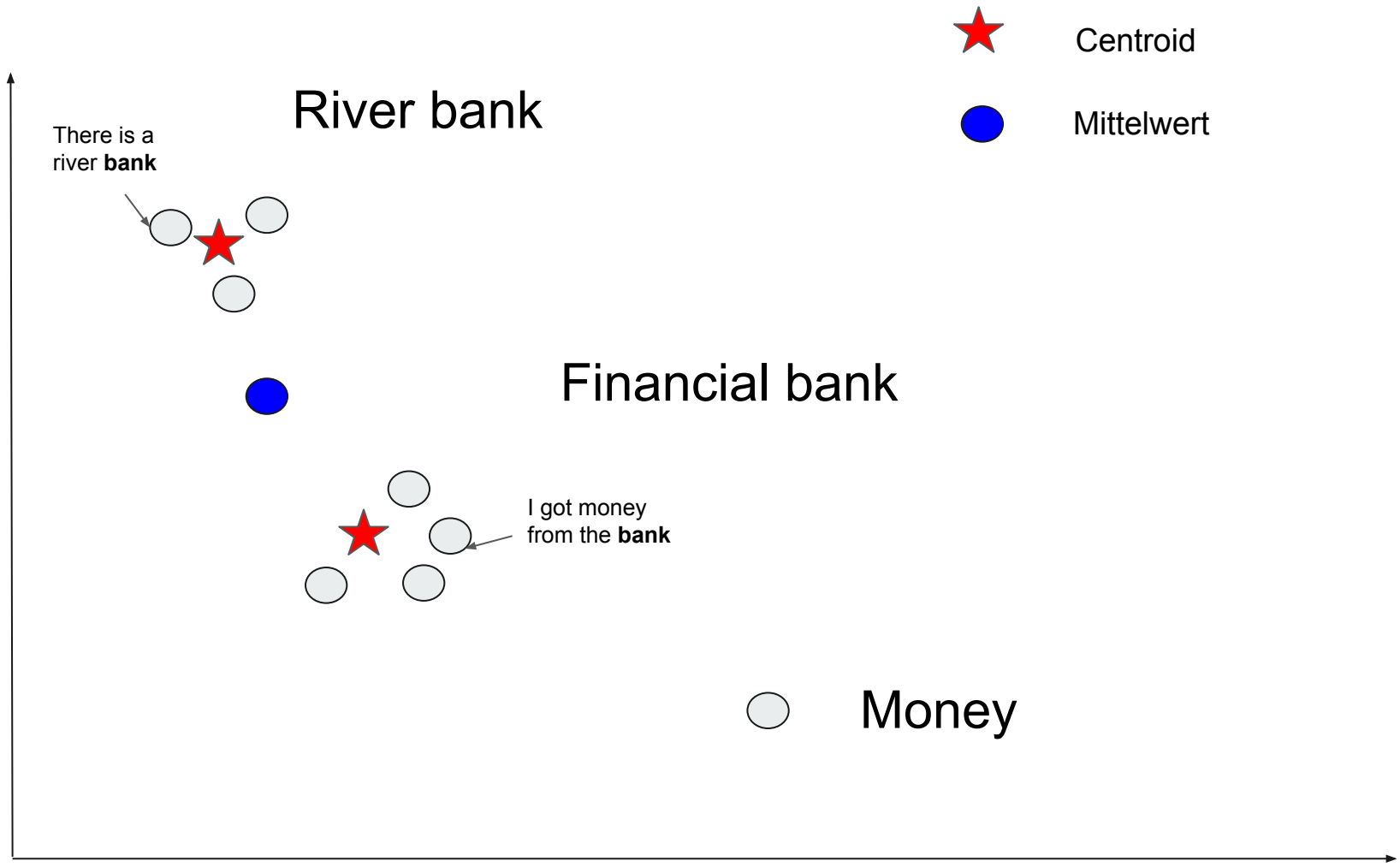
THINGS


bert-base
bert-large
gpt-2
gpt-2-medium
sbert_bert
sbert_distill_roberta
Contextualized
Decontextualized
Word2Vec
GloVe
Spearman correlation
Extraction layer

Simlex


bert-base
bert-large
gpt-2
gpt-2-medium
sbert_bert
sbert_distill_roberta
Spearman correlation with Simlex 999
Extraction layer
Contextualzied
Decontextualized
Word2Vec
GloVe

Homonyme


Centroid
Mittelwert
River bank
There is a river **bank**
Financial bank
I got money from the **bank**
Money

Ausblick

- End to end learning on whole LM instead of finetuning
- RSA with THINGs brain data
- fine tuning on different task not similarity
- DeBERTa statt BERT
- BERT-flow
- Einfluss von Textcorpus -> preprocessed Wikitext vs Wikidump
- Einfluss von Context Window -> LM haben großes Window ~500 Tokens in BERT (Word2Vec ~5-10)
- STS with static word embeddings verglichen mit SBERT
  - performance vs accuracy
- Vorhersage der Dimensionswerte