

Anwendung von BERT im Document Retrieval

Informationsverarbeitung II

Vanessa Reichel und Sophie Vonderbank

Am 16.06.21

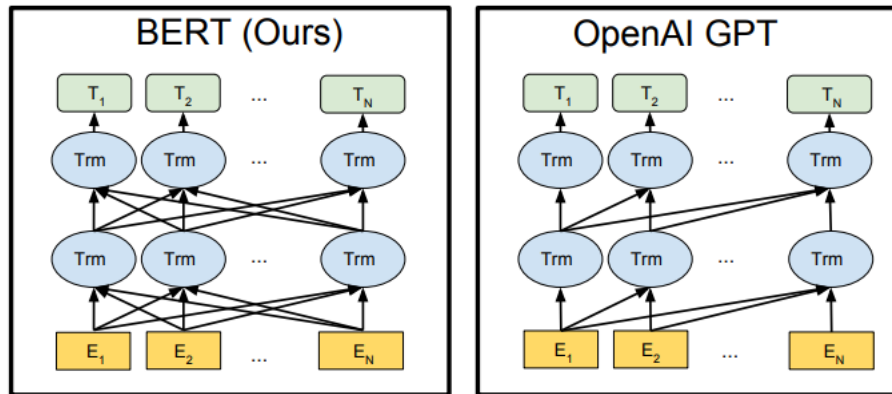
Gliederung

- ▶ 1. Einführung BERT
 - ▶ Was ist BERT?
 - ▶ Pre-Training
 - ▶ Fine-Tuning
 - ▶ Beispiel Finetuning: Q&A
- ▶ 2. Anwendung im Information Retrieval
 - ▶ Bert im ad hoc Document Retrieval (Yang et al. 2019)
 - ▶ Bert für Semantic Search in Background Linking (Deshmukh und Sethi 2020)
- ▶ Zusammenfassung und Fazit

Was ist BERT?

- ▶ BERT = Bidirectional Encoder Representations from Transformers
- ▶ Ist ein Deep Learning Algorithmus
- ▶ Von Google AI Language 2018 entwickelt
- ▶ Wird im Natural Language Processing (NLP) für das Sprachverständnis eingesetzt
- ▶ Wurde auf 2.500 Mio. Wörtern aus Wikipedia und 800 Mio. Wörtern aus Google Books trainiert
- ▶ Zweistufiger Aufbau:
 - ▶ 1. Pre-training: allgemeines Sprachverständnis
 - ▶ 2. Fine-tuning: task-spezifische Anpassung

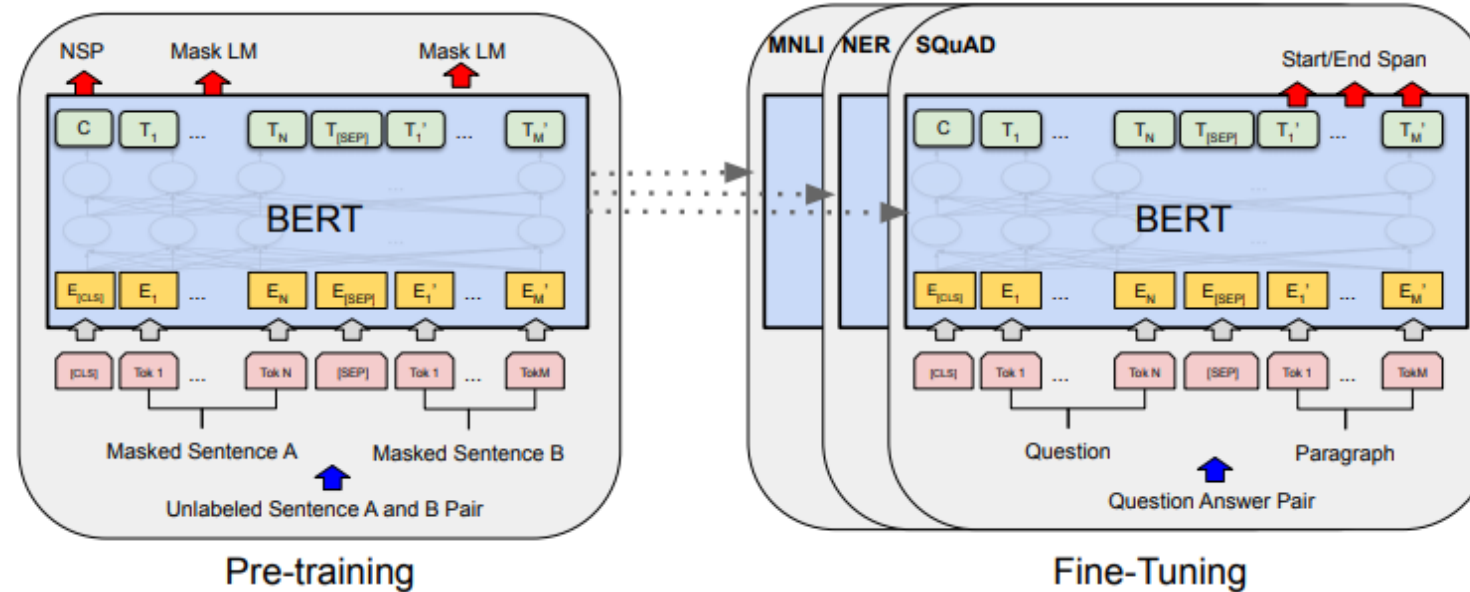
BERT: Allgemeines



B wie bidirectional:

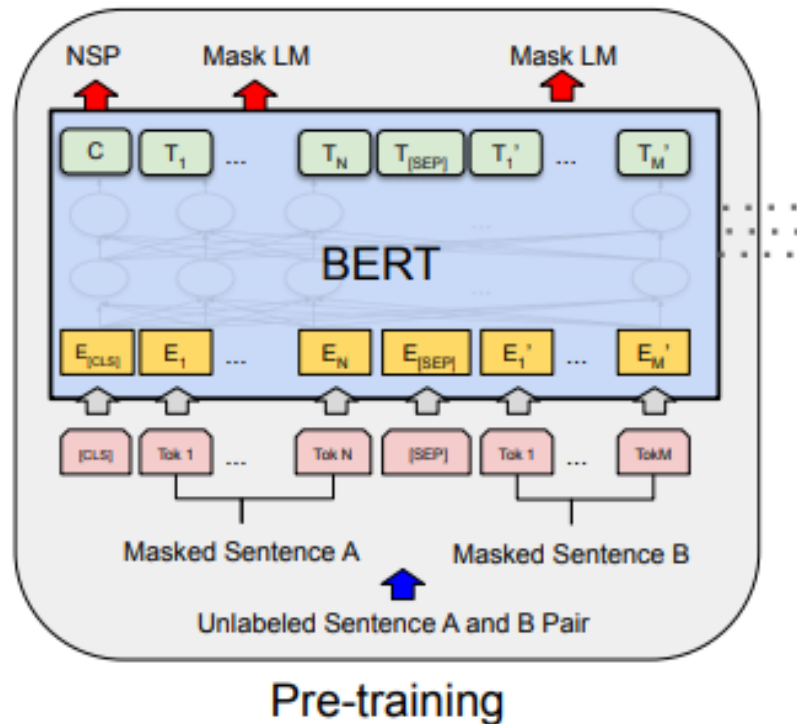
- Rechter UND linker Kontext des Token
 - Unterschied zu GPT: GPT= left to right (unidirectional) vs. BERT = bidirectional
- ERT wie Encoder Representations from Transformers:
- BERT basiert auf der Encoding-Komponente der Transformer Architektur
 - Encoder-Struktur (Multihead Attention + Feedforward neural network)

BERT: Überblick



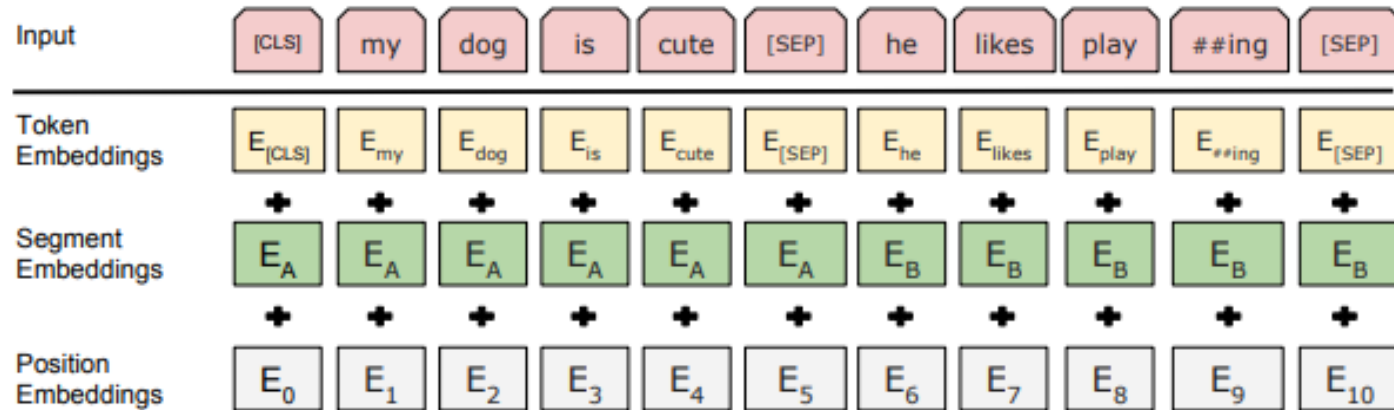
- 2-stufiger Aufbau:
 - 1. Pre-Training: allgemeines Sprachverständnis
 - 2. Fine-Tuning: task-spezifische Anwendung (z.B. Document Retrieval, Question-Answer etc.)
- Modell aus dem Pre-Training wird im Fine-Tuning optimiert

1. Schritt: Pre-Training



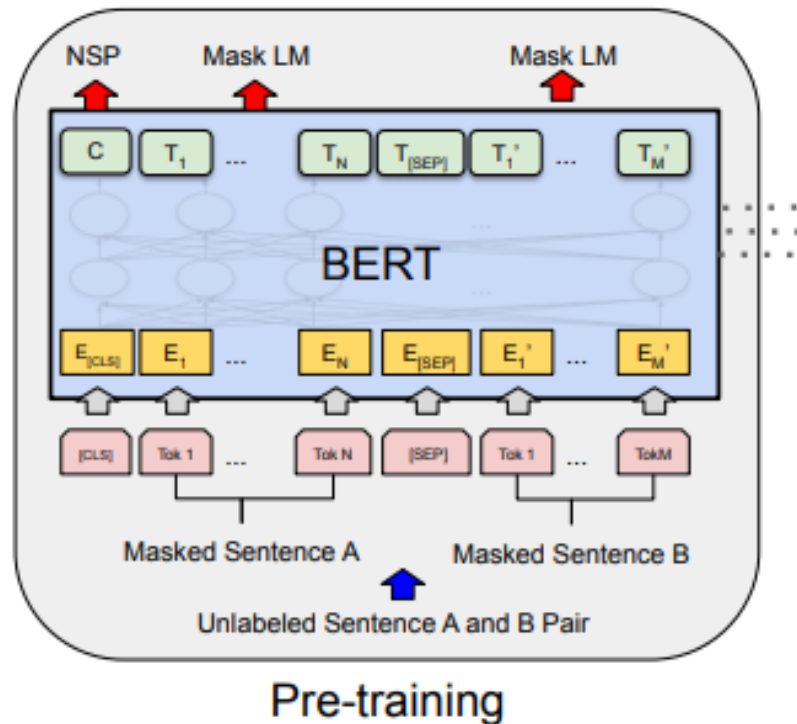
- ▶ Sprachverständnis wird durch 2 Aufgaben gelernt:
- ▶ 1. Next Sentence Prediction (NSP): Folgt Satz B auf Satz A? → Ja/Nein
- ▶ 2. Masked Language Modeling (MLM): Pro Satz wird ein zufälliges Wort maskiert: Welches Wort ist hinter der Maske?
- ▶ Laufen Parallel ab

1. Schritt: Pre-Training



- Input: Kombination aus 3 Embeddings
 - 1. Token Embeddings: Wörter + CLS + SEP
 - 2. Segment Embeddings: Zuordnung zum Satz
 - 3. Position Embeddings: Position eines Wortes im Input
- Diese 3 Schichten repräsentieren den Input

1. Schritt: Pre-Training



- ▶ Output 1: Wort-Vektoren
- ▶ Output 2: C ist für die Klassifikation (NSP) → binäres Ergebnis
- ▶ Mittels Softmax-Layer und Cross-entropy werden die maskierten Token gelernt
- ▶ Darstellung im one-hot-encoding (maskiertes Wort entspricht der 1)
- ▶ Nach abgeschlossenem Training: Sprachverständnis wurde erlernt

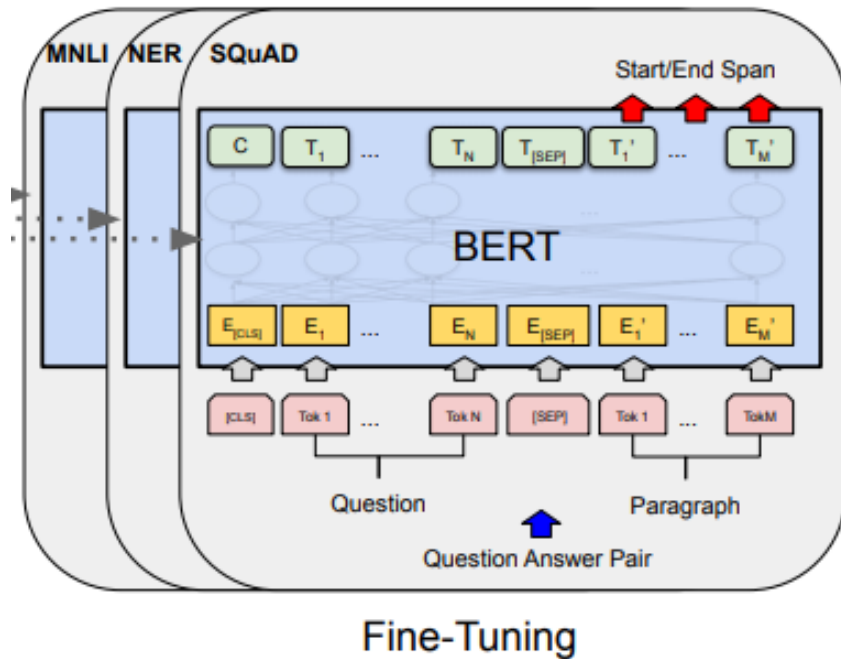
2.Schritt: Fine-Tuning

- ▶ = Anpassung von BERT an spezifische Tasks, z.B. QA, Textzusammenfassungen, Document Retrieval
- ▶ Input und Output des Fine-Tunings ist taskspezifisch
- ▶ Die Parameter des Modells aus dem Pre-Training werden optimiert
- ▶ Vorteil: Kurze Trainingsdauer



<https://towardsml.com/2019/09/17/bert-explained-a-complete-guide-with-theory-and-tutorial/>

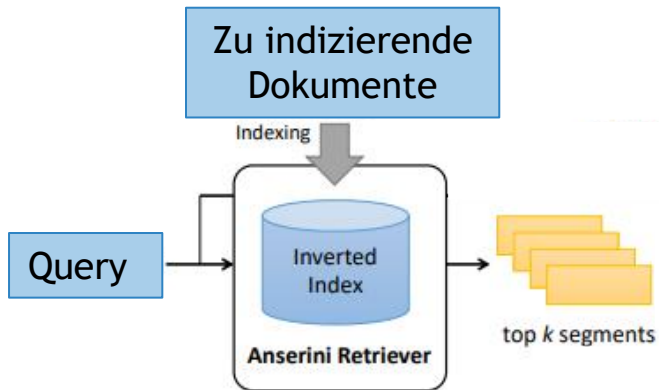
2. Schritt: Fine-tuning



Beispiel: Question-Answer

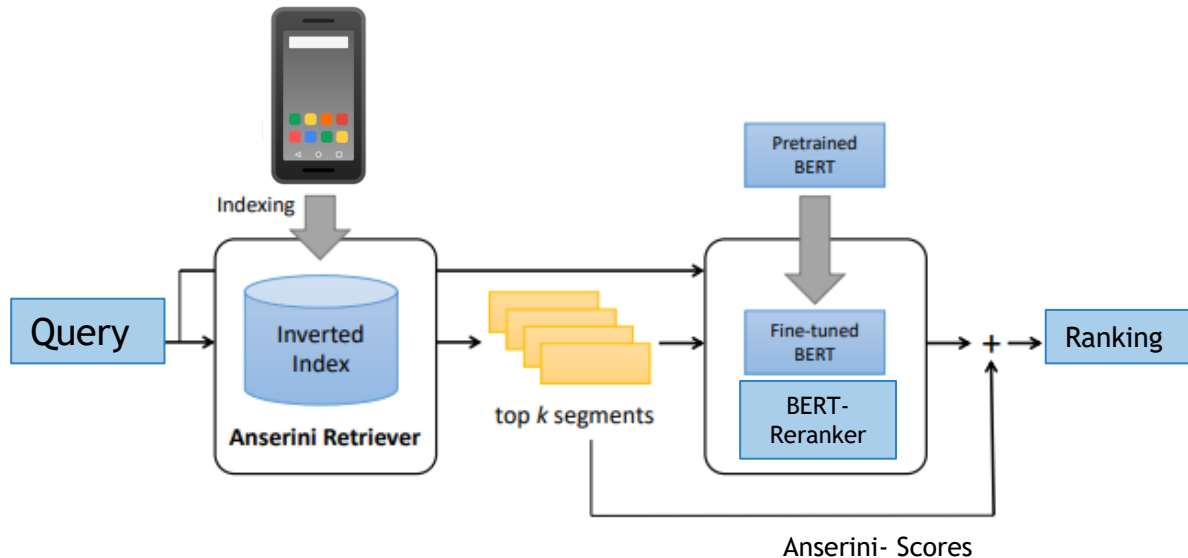
- ▶ Input:
 - ▶ Zu beantwortende Frage
 - ▶ Paragraph, der die Antwort zur Frage enthält
- ▶ Output: Wortvektoren rechts und links von der Antwort

Fine-Tuning: Bert im ad hoc Document Retrieval (Yang et al. 2019)



- ▶ Ad hoc Retrieval: gerankte Ergebnisliste gegeben einer Anfrage
- ▶ QA und Document Retrieval sind recht ähnlich
 - ▶ Unterschied: Inputlänge und Output
- ▶ Anwendung von BERT für Retrieval von (1) Social Media Posts und (2) Zeitungsartikel
- ▶ Ziel: Dokumenten Retrieval und Ranking
- ▶ Architektur: Anserini + BERT
- ▶ Anserini Retriever: Toolkit für initiales Retrieval

Fine-Tuning: Bert für Social Media Posts (Yang et al. 2019)



- ▶ Verwendung von modifiziertem BERTserini
- ▶ Indizieren von Posts
- ▶ Grobes Ranking der Posts durch Anserini
- ▶ Fine-Tuning: top k Sequenzen (Posts) + pre-trained BERT, damit diese final gerankt werden können
- ▶ Kombination aus Ranking von Anserini und BERT-Scores ergeben finales Ranking der Posts

Modifizierte Abbildung; Original aus Yang, W. et al. (2019) End-to-End Open-Domain Question Answering with BERTserini. arXiv:1902.01718v2

Bild: <https://emojiterria.com/de/mobiltelefon/>

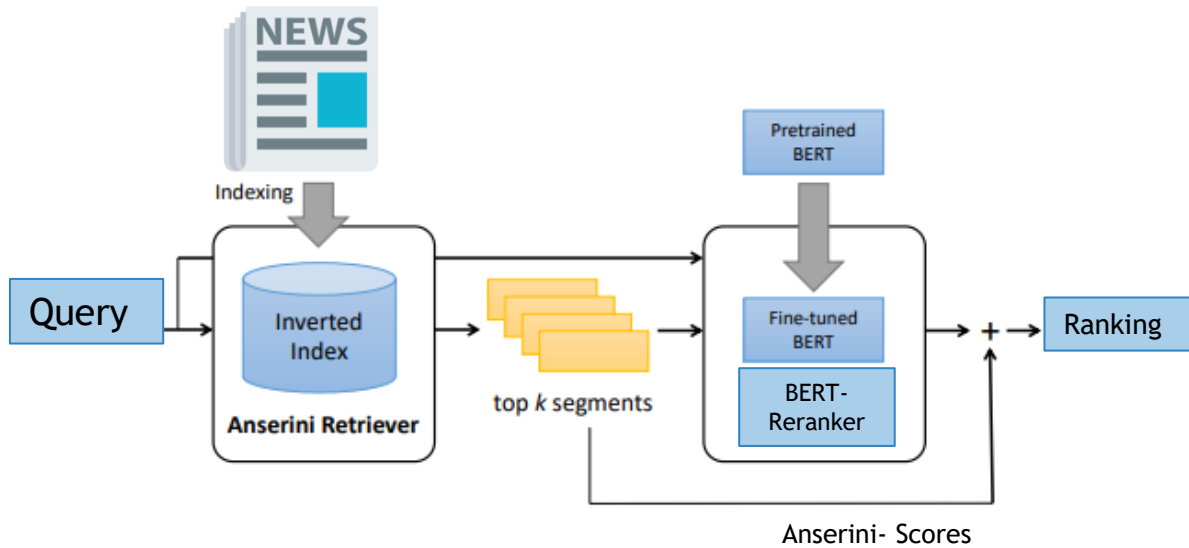
Bert im ad hoc Document Retrieval (Yang et al. 2019): Social Media Posts

- ▶ Bei social Media Posts: Dokumentlänge ist noch nicht problematisch
- ▶ Im Vergleich zu allen anderen Modellen erzielt BERT die besten Ergebnisse

Model	2011		2012		2013		2014	
	AP	P30	AP	P30	AP	P30	AP	P30
QL	0.3576	0.4000	0.2091	0.3311	0.2532	0.4450	0.3924	0.6182
RM3	0.3824	0.4211	0.2342	0.3452	0.2766	0.4733	0.4480	0.6339
DRMM (Guo et al., 2016)	0.3477	0.4034	0.2213	0.3537	0.2639	0.4772	0.4042	0.6139
DUET (Mitra et al., 2017)	0.3576	0.4000	0.2243	0.3644	0.2779	0.4878	0.4219	0.6467
K-NRM (Xiong et al., 2017)	0.3576	0.4000	0.2277	0.3520	0.2721	0.4756	0.4137	0.6358
PACRR (Hui et al., 2017)	0.3810	0.4286	0.2311	0.3576	0.2803	0.4944	0.4140	0.6358
MP-HCNN (Rao et al., 2019)	0.4043	0.4293	0.2460	0.3791	0.2896	0.5294	0.4420	0.6394
BiCNN (Shi et al., 2018)	0.4293	0.4728	0.2621	0.4147	0.2990	0.5367	0.4563	0.6806
BERT	0.4697	0.5040	0.3073	0.4356	0.3357	0.5656	0.5176	0.7006

Table 1: Results on test collections from the TREC Microblog Tracks, comparing BERT with selected neural ranking models. The first two blocks of the table contain results copied from Rao et al. (2019).

Fine-Tuning: Bert im ad hoc Document Retrieval: Zeitungsartikel



- ▶ Grobes Ranking der Artikel durch Anserini
- ▶ Inputlänge ist problematisch: Dokument in „Chunks“ segmentiert
- ▶ Fine-Tuning: Fine-Tuning mit verschiedenen Datensätzen: Social Media Posts, da zuvor erfolgreich + QA-Data (TrecQA + WikiQA), da diese Zeitungsartikeln ähneln
- ▶ Task: Finde den relevantesten Satz eines Dokuments (BERT-Score)
- ▶ Kombination aus Anserini-Scores (Dokument-Scores) und BERT-Scores ergeben finales Ranking der Dokumente

Modifizierte Abbildung; Original aus Yang et al.(2019) End-to-End Open-Domain Question Answering with BERTserini. arXiv:1902.01718v2

Bild: <https://emojipedia.org/joypixels/3.1/newspaper/>

Bert im ad hoc Document Retrieval (Yang et al. 2019): Zeitungsartikel

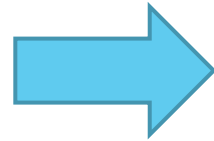
Model	AP	P20
Paper 1 (two fold)	0.2971	0.3948
BM25+RM3	0.2987	0.3871
1S: BERT FT(QA)	0.3014	0.3928
2S: BERT FT(QA)	0.3003	0.3948
3S: BERT FT(QA)	0.3003	0.3948
1S: BERT FT(Microblog)	0.3241	0.4217
2S: BERT FT(Microblog)	0.3240	0.4209
3S: BERT FT(Microblog)	0.3244	0.4219
Paper 2 (five fold)	0.272	0.386
BM25+RM3	0.3033	0.3974
1S: BERT FT(QA)	0.3102	0.4068
2S: BERT FT(QA)	0.3090	0.4064
3S: BERT FT(QA)	0.3090	0.4064
1S: BERT FT(Microblog)	0.3266	0.4245
2S: BERT FT(Microblog)	0.3278	0.4267
3S: BERT FT(Microblog)	0.3278	0.4287

Table 2: Results on Robust04. FT indicates the dataset used for fine tuning; *n*S indicates inference using the top *n* scoring sentences of the document.

- ▶ Mikroblogs sind für das fine-tuning besser geeignet →
 - ▶ Mikroblogs simulieren die Task
 - ▶ QA-Datensätze simulieren den Inhalt (Zeitungsartikel)
 - ▶ → Task-Ähnlichkeit ist wichtiger als inhaltliche Ähnlichkeit
- ▶ BERT erzielt bessere Ergebnisse als BM25+RM3

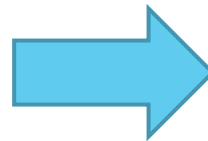
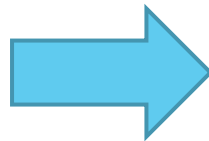
Anwendung BERT: Background Linking

Query
(1 Dokument)



Ranking Liste:
1. Bestes Dokument
2. Zweitbestes Dokument
3. ...

- ▶ Background Linking: zu einem Dokument weitere relevante Artikel finden (Hintergrundinformationen, Hilfe für Verständnis)
- ▶ Vergleich zweier Ansätze:
 - ▶ Keywords extrahieren mit tf-idf → Gewichtung und Verbinden der Keywords → Suchanfrage → Ranking mit BM25 der Artikel aus der query



Rankingliste

- ▶ IR-BERT: Semantische Suche

Background Linking: IR-BERT

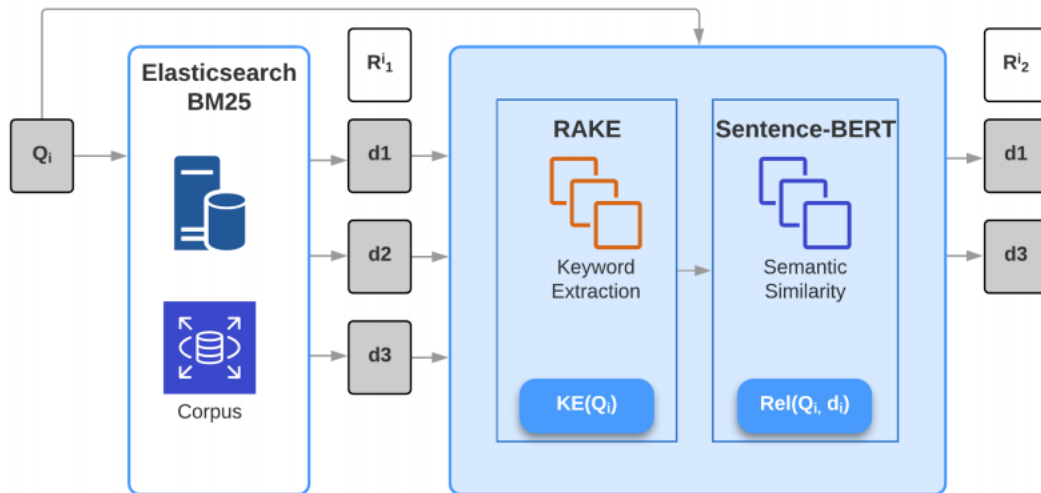


Figure 1: IR-BERT pipeline

- ▶ Semantische Suche
- ▶ Query
- ▶ Elasticsearch: Preprocessing +Indizieren
- ▶ BM25: Ranking der Dokumente und Vorauswahl der Suche
- ▶ RAKE: Keyword-Extrahierung
- ▶ Sentence-BERT: Bestimmung der semantischen Ähnlichkeit
- ▶ Output: inhaltlich passende Artikel zur Query

Background Linking: Ergebnisse IR-BERT

- nDCG = Normalized Discounted Cumulative Gain

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

$\xrightarrow{\hspace{1.5cm}}$ $DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$

$\xrightarrow{\hspace{1.5cm}}$ $IDCG_p = \sum_{i=1}^p rel_i \mid \text{Ideal ordering by relevancy}$

- IR-BERT erzielt besten Ergebnisse
- Wieso? Kombination der retrieval Power von BM25 und Kontextverstehen von BERT

Methods	nDCG@5
TREC 2018 Median	0.3448
BM25	0.3251
anserini_1000w	0.3529
umass_cbrdm	0.4173
(A1.1) wBT+BM25	0.4088
(A1.2) wQ+BM25	0.3942
(A2.1) IR-RoBERTa	0.394
(A2.2) IR-BERT	0.4199

Tabelle aus: Deshmukh, A., Sethi, U. (2020) IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles. arXiv:2007.12603v1

Formeln aus: <https://www.cis.uni-muenchen.de/~stef/seminare/Suchmaschinen20/evaluation.pdf> (letzter Aufruf am 14.06.21)

Fazit: BERT im IR



<https://twitter.com/bertspahn>

- ▶ Taskspezifische Anpassung des pre-trained Modells möglich: schnell und effizient
- ▶ BERT ist auch bei Semantic Search erfolgreich
 - ▶ Anwendung um ad hoc Document Retrieval → erfolgreichster Ansatz
 - ▶ Anwendung im Background Linking → erfolgreichster Ansatz

Vielen Dank
für eure Aufmerksamkeit 😊

Quellen:

▶ Textquellen:

- ▶ Dai, Z., Callan, J. (2019) Deeper Text Understanding for IR with Contextual NeuralLanguage Modeling. arXiv:1905.09217
- ▶ Deshmukh, A., Sethi,U. (2020) IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles. arXiv:2007.12603v1
- ▶ Devlin, J. et al. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
- ▶ Yang, W. et al.(2019) End-to-End Open-Domain Question Answering with BERTserini. arXiv:1902.01718v2
- ▶ Yang, W. et al.(2019) Simple Applications of BERT for Ad Hoc Document Retrieval. [arXiv:1903.10972v1](https://arxiv.org/abs/1903.10972v1)
- ▶ <https://www.cis.uni-muenchen.de/~stef/seminare/Suchmaschinen20/evaluation.pdf> (letzter Aufruf am 14.06.21)

▶ Bildquellen:

- ▶ <https://sdsclub.com/bert-google-nlp-algorithm/> (letzter Aufruf am 30.05.21)
- ▶ <https://emojiterria.com/de/mobiltelefon/> (letzter Aufruf am 06.06.21)
- ▶ <https://emojipedia.org/joypixels/3.1/newspaper/> (letzter Aufruf am 06.06.21)
- ▶ <https://towardsml.com/2019/09/17/bert-explained-a-complete-guide-with-theory-and-tutorial/> (letzter Aufruf am 11.06.21)
- ▶ <https://twitter.com/bertspahn> (letzter Aufruf am 11.06.21)
- ▶ <https://icon-icons.com/de/symbol/text-Dokument-beschrieben-Schnittstelle-symbol-mit-Linien/57757> (letzter Aufruf am 14.06.21)
- ▶ <https://icon-icons.com/de/symbol/Gewicht/135019> (letzter Aufruf am 14.06.21)