NEURONALE SUCHE, TRANSFORMER, LLM

MASTERSEMINAR SUCHMASCHINEN UND RAG

COMPUTERLINGUISTIK

SOMMERSEMESTER 2024

STEFAN LANGER

STEFAN.LANGER@CIS.UNI-MUENCHEN.DE

"Big data"

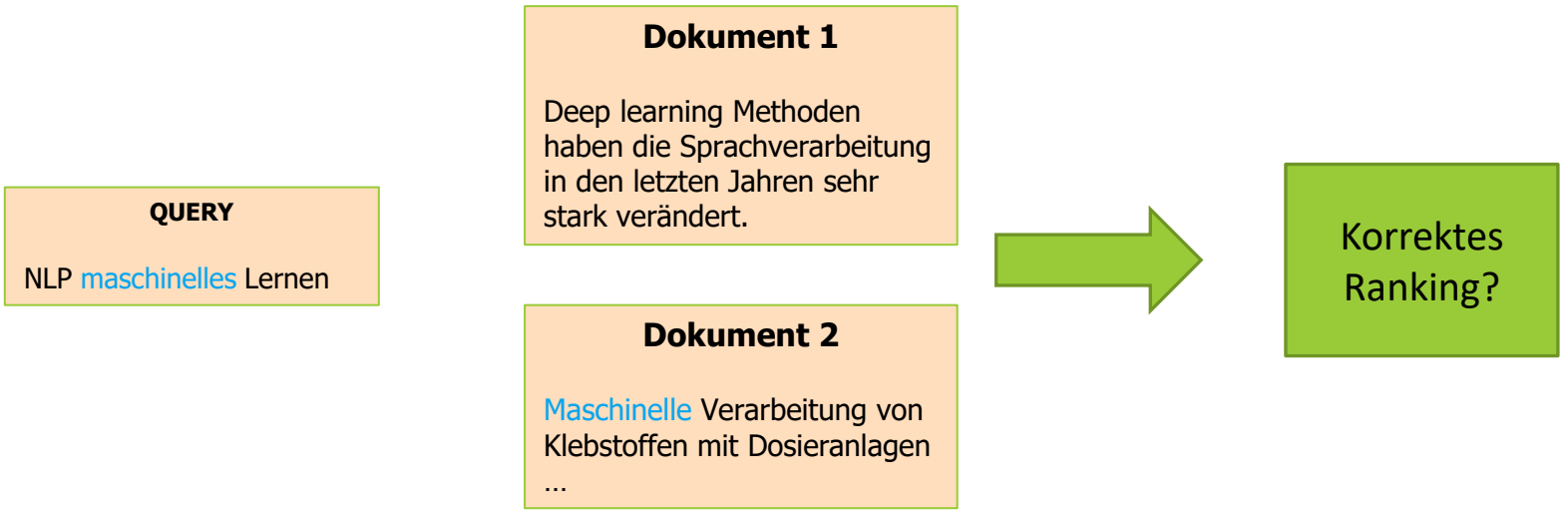
IR ist eigentlich schon lange 'big data':

- Internet-Suchmaschinen indizieren Milliarden von Dokumenten
- Suchmaschinen-Software wie Solr und Elasticsearch kann ohne weiteres mehrere Millionen oder gar Milliarden von Dokumenten indizieren und verwalten, auch repliziert
- Ähnlich neueren NLP-Modellen basiert der Index (i.e. das Datenmodell) auf sehr vielen Daten
- Corporate-Suchmaschinen enthalten in ihren Indexen große Ausschnitte des Wissens eines Unternehmens und sind sehr häufig der umfassendste Informationsspeicher eines Unternehmens; sie machen die Information zugänglich

Stichwortsuche - termbasiert

- Beruht auf Wortformen

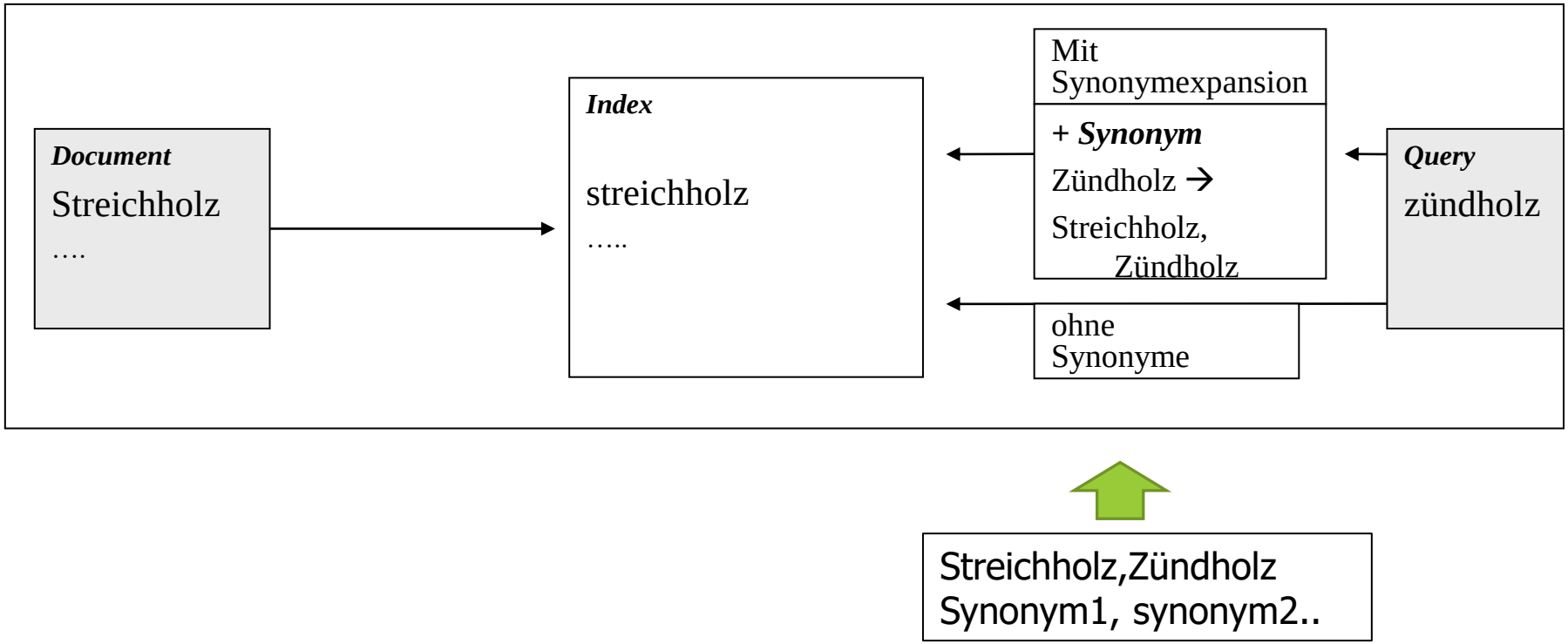
  → Recall-Problem für nur semantische verwandte Dokumente



Term-basierte Suche – Semantik & approximative Suche

- Normalisierung
- Stemming
- Phonetische Suche
- Synonyme

Synonyme: Anfrageexpansion



Was fehlt…

- Kontextsensitive Semantik
- Datengetriebene Semantik (i.e. semantische Modell für Domäne)
- Semantik für längere Sequenzen
  - Phrasen
  - Sätze
  - Abschnitte
  - Dokumente

Neuronale Netze und Word Embeddings

Neuronale Netze haben NLP umgekrempelt (Spracherkennung, maschinelle Übersetzung, Textklassifikation) bzw. sind dabei diese stark zu verändern (Textzusammenfassung, Entitätenextraktion…)

- Embeddings: Repräsentation eines Zeichens/Wortes/Phrase/Satzes/Dokuments als N-dimensionaler Vektor
  - Word-Embedding mit neuronalen Netzen (e.g. word2vec)
    - Skip gram – Training: Kontext vorhersagen
    - CBOW (continuous bag of words) – Training: Zielwort vorhersagen
    - Glove, Fasttext …
  - Satz- und Dokumentenembeddings
- Transformer – Kontextsensitive dichtbesetzte Vektoren
  - Bert, Roberta, … , GPT-3

SIGIR **2018** Workshop
Learning from Limited/Noisy data for IR

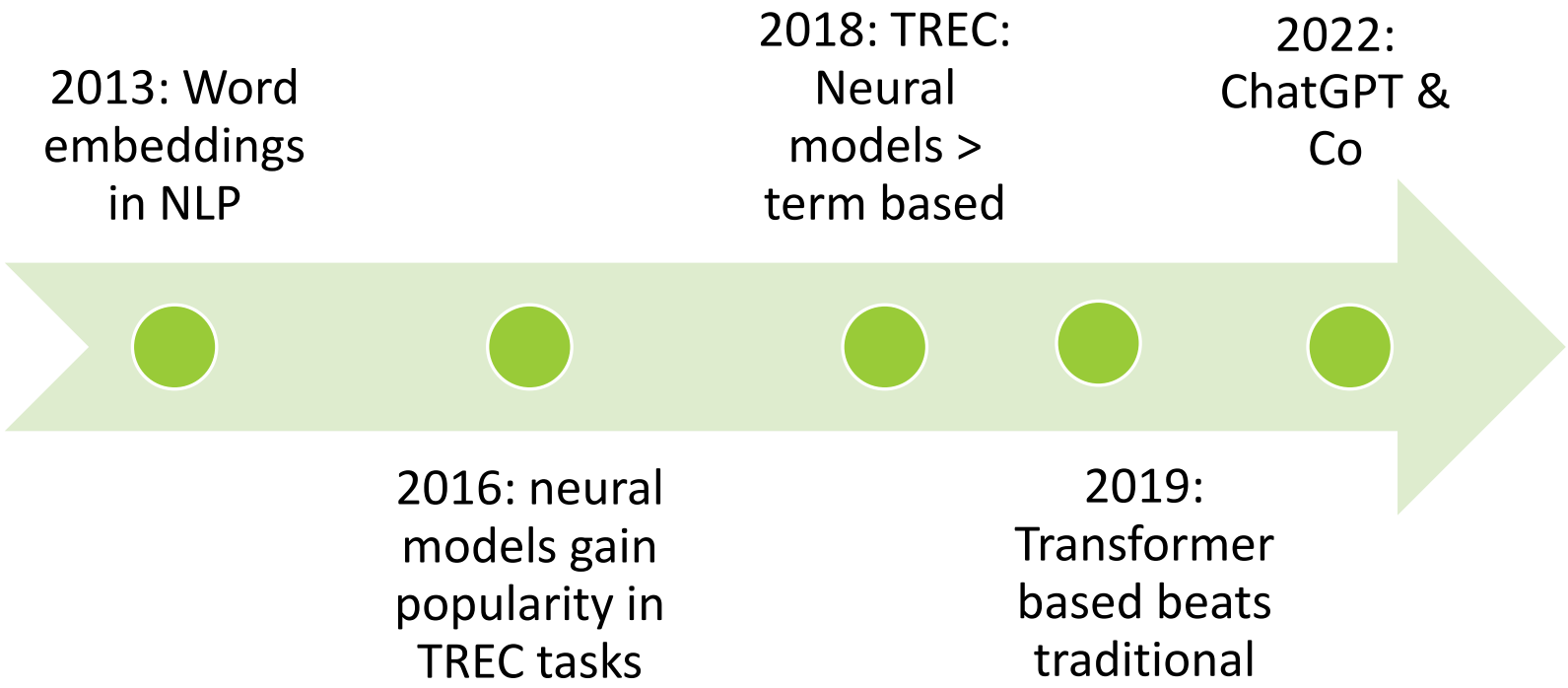
Organizers: Hamed Zamani (UMass Amherst), Mostafa Dehghani (Univ. of Amsterdam), Fernando Diaz (Spotify), Hang Li (Toutiao AI Lab), Nick Craswell (Microsoft)

In recent years, machine learning approaches, and in particular deep neural networks, have yielded significant improvements on several natural language processing and computer vision tasks; **however, such breakthroughs have not yet been observed in the area of information retrieval.**

Besides the complexity of the IR tasks, such as understanding the user's information needs, a main reason is the lack of high-quality and/or large-scale training data for many IR tasks. This necessitates studying how to design and train machine learning algorithms where there is no large-scale or high-quality data in hand. Therefore, considering the quick progress in development of machine learning models, this is an ideal time for a workshop that especially focuses on learning in such an important and challenging setting for IR tasks. The goal of this workshop is to bring together researchers from industry—where data is plentiful but noisy—with researchers from academia—where data is sparse but clean to discuss solutions to these related problems.

(http://sigir.org/sigir2018/program/workshops/ ; + Hervorhebungen)

Time line – Neural Information retrieval



Sparse and dense vector

Dokument 1

Schöne Männer gehören nach Cannes wie die Aschewolke an den isländischen Himmel

Dokument 2

Getrocknetes Wasser, das vom Himmel fällt und Tiere ohne Flügel, die trotzdem fliegen können?

Index

schöne.0,4
männer.0,3
….

0
0
0
0
0
0
0
0
0
0
0.4
0
0
0
0
0
0
0.3
0
0
0
0
0
0
0
0
0
0
0

The representation of a document in an inverted index can be viewed as a sparse vector:

Dimension = size of vocabulary

Matching is done based on the query vector (with normally just very few filled vector positions)

Latent Semantic Indexing

Relativ altes Verfahren zur Erstellung von Embeddings

Grundidee:

- Reduziere die Term-Dokument Matrix auf eine Begriffs-Dokument-Matrix
- Fasse Terme, die häufig zusammen auftreten, zu einem Begriff zusammen
- → Reduziere den Vektorraum auf weniger Dimensionen

Term-Dokument-Matrix

	D1	D2	D3	D4	D5	D6	D7	D8	D9
Tisch	1	1			1			1	1
Stuhl	1	1						1	1
Sofa		1			1				1
Frosch			1	1		1			
Kröte		1		1		1			
Molch		1	1	1		1			
Auto			1				1	1	
Rikscha	1		1				1		
Fahrrad	1						1	1	

LSI: Weitere Grundlagen

Matrizenrechnung

Singulärwertzerlegung SVD (singular value decomposition)

Methode der kleinsten Quadrate (least squares method)

Eigenvektor

Embeddings

Dokument 1

Schöne Männer gehören nach Cannes wie die Aschewolke an den isländischen Himmel

0.3
0.6
0.12
0.3
0.21
....

Dokument 2

Getrocknetes Wasser, das vom Himmel fällt und Tiere ohne Flügel, die trotzdem fliegen können?

0.3
0.6
0.12
0.3
0.21
....

The representation of a word, a phrase, a sentence of a document is a dense vector (typically with dimensions between 100 and 1000)

How to use this in search?

Unterschiede IR / andere NLP

Queries: Sehr kurze Texte mit fehlendem Kontext

Oft extrem variable Dokumentlänge

Dokumente enthalten eine Mischung aus relevanten und irrelevanten Passagen in Bezug auf die Query; relevanter Inhalte kann verteilt sein über das Dokument

Extrem große Datenmengen (Queries/Dokumente) müssen u.U. permanent verarbeitet warden

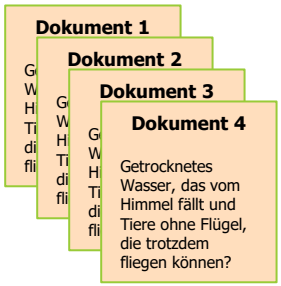
Unterschiedliche Verwendung des Vokabulars in Query/Dokumenten

Query-Kontext of nicht im Query-Text (Ort, Zeit…) → Metadaten

Textlänge

QUERY: xxx

Textlänge



Queries sind kurz, sehr kurz, variabel

(anders in Frage-Antwort-Systemen)

Dokumente können zwischen sehr kurz und extrem lang variieren

Kontextdisambiguierung Queries?

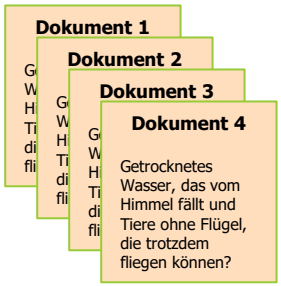
Welche Embedding-Methode passt?

Welches Modell passt?

Training

Ein termbasiertes Retrieval-System funktioniert sofort mit beliebiger Anzahl von Dokumenten. Training=Indizierung

Trainingsdaten

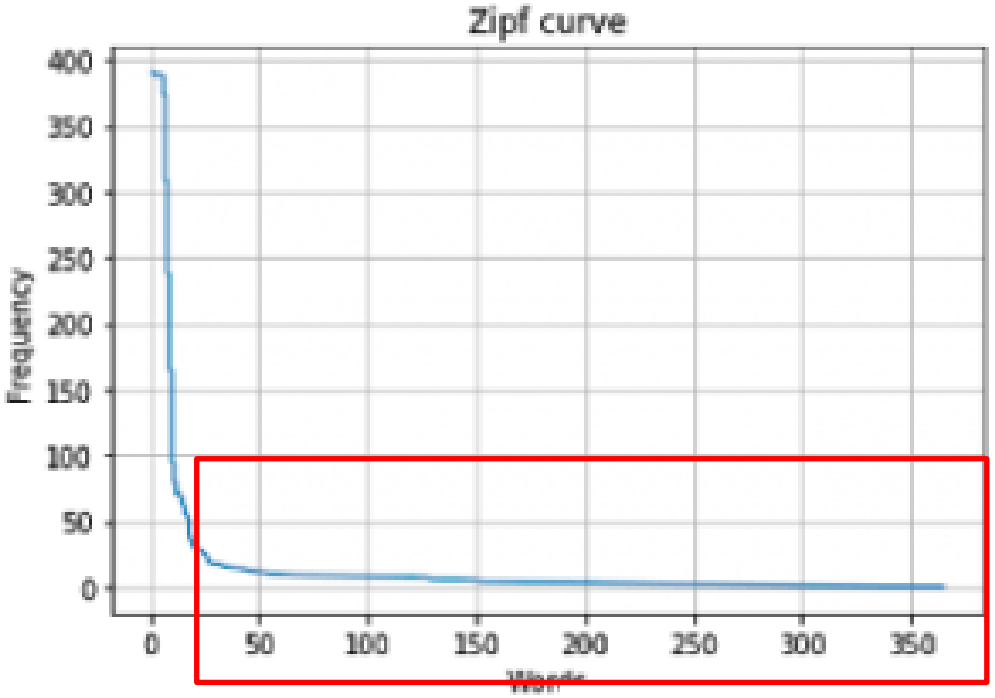


Word Embeddings müssen auf sehr vielen Dokumenten trainiert werden

Transformer brauchen noch mehr Dokumente

Vortrainierte Modelle (passend?)

Unbekannte Terme



N queries

Sehr viele Queries wurden nie zuvor gesehen

Queries mit unbekanntem Vokabular

Modellanpassung

Anpassung des Modells an neue Daten, Dokumente

Indexierung 10 000 Dokumente in Elasticsearch:

< 1 min auf einem einfachen Laptop
weitere Optimierung möglich

Trainingszeit 10 000 Dokumente Embeddings (bsp. Fasttext)

- Mittelhoch, ca. 20 min, abhängig von Parametern
- GPU erforderlich

Trainingszeit Transformer

- Sehr hoch
- GPU erforderlich

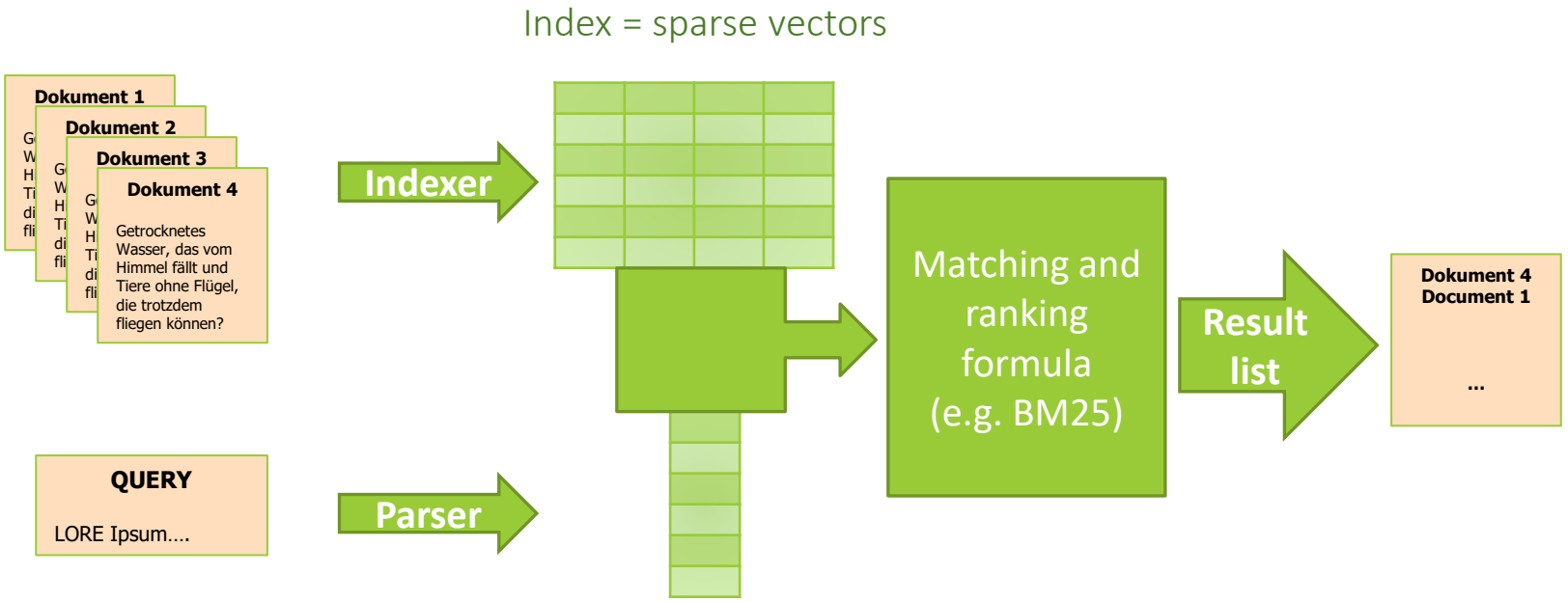
Sparse vector – dense vector

Sparse (Document) – ultrasparse (Query)	Dense
Find all document vectors where at least on dimension of the query vector is similar	Find the document vectors which match the query vector best
Known method: inverted index + boolean operations	See architectures on following slides
Errors can be easily analyzed, potentially fixed (e.g. lemmatization, synonyms, tokenization…)	Experiment until best model setup is found

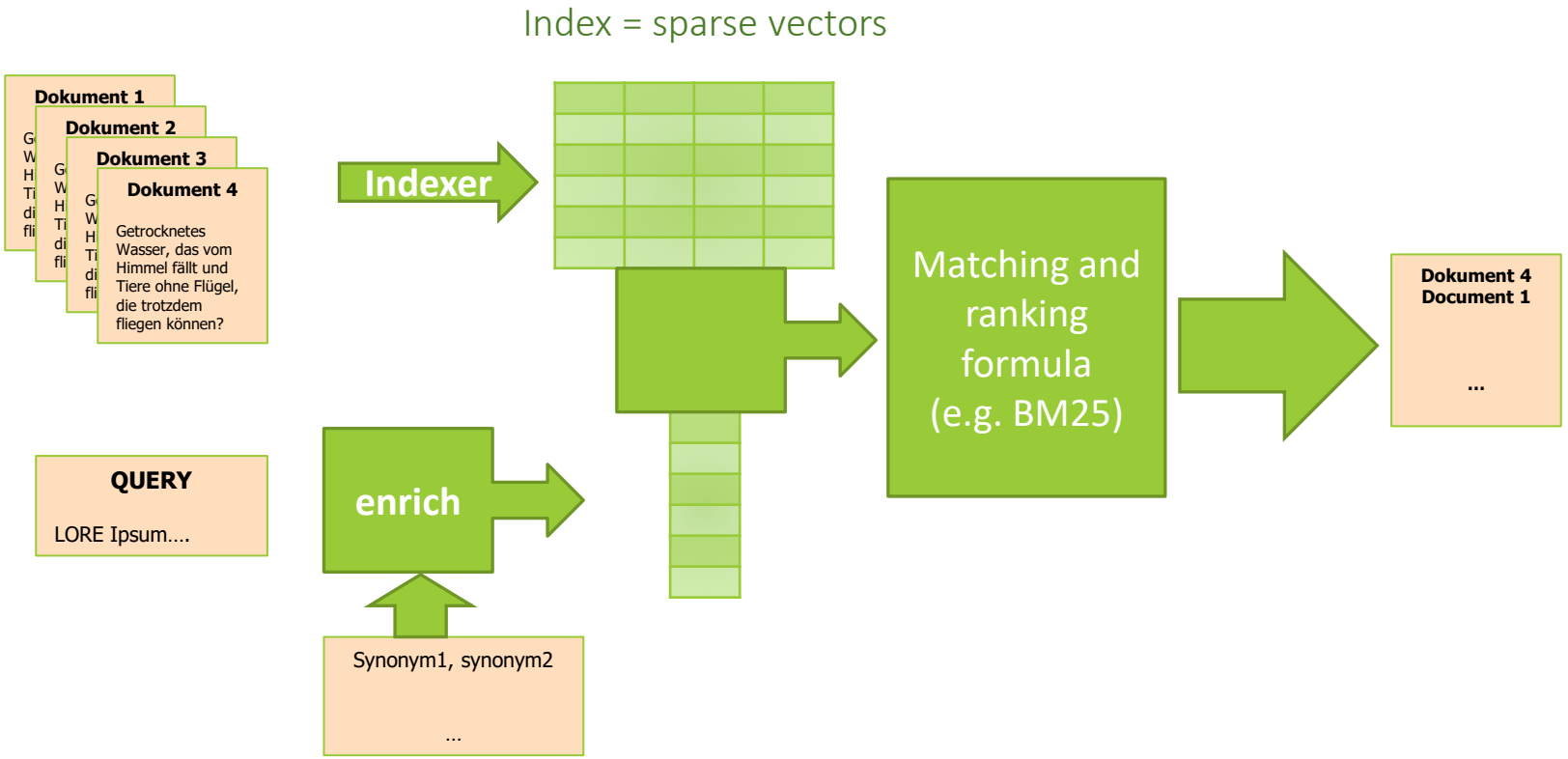
Trainieren von neuronalen Modellen im IR

- unsupervised: Just learn the presentation from texts (documents, query logs)
  - Word2vec
  - Document, sentence embeddings
  - Transformers
- Supervised: Query document pairs with similarity rating
  - Requires a lot of training data – often not available
  - Alternative
    - Use term-based results to train the model
      - → difficult to beat the term based models
    - artificial training data

Term-based retrieval

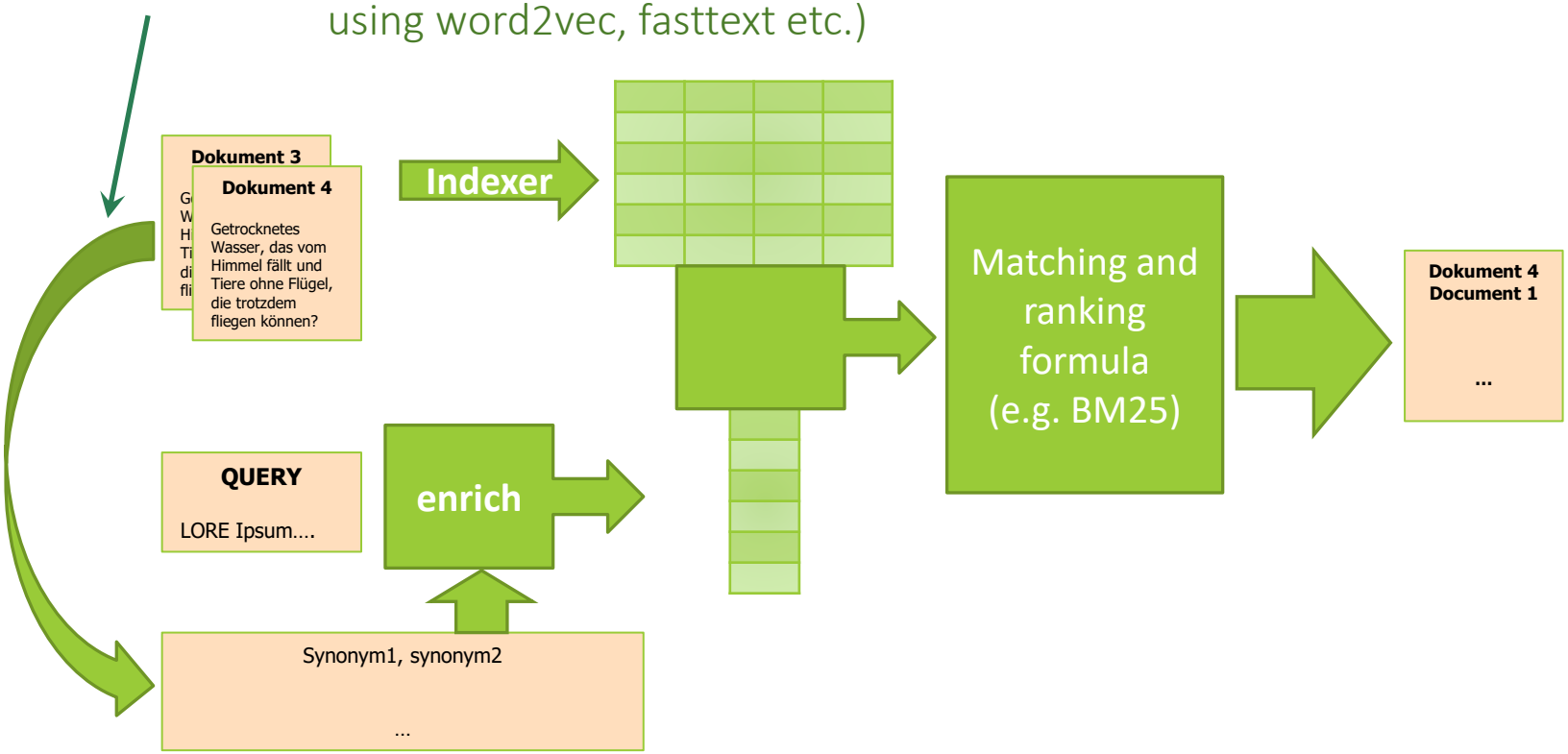


Term-based retrieval + synonyms



Neural search I – synonyms

Learn synonyms from data (word / phrase embeddings using word2vec, fasttext etc.)



Word embedding synonyms

- (re-)Train word embedding model from data (e.g. fasttext)

- Remove stopwords (e.g. using DF)

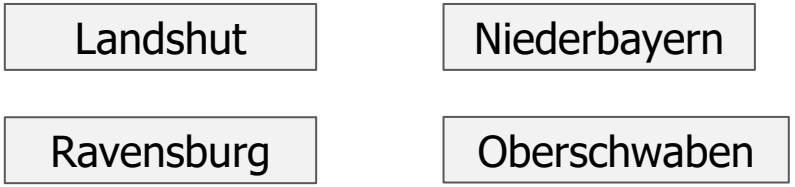
- create synonym list based on document content using a similarity measure

- deploy synonyms to elasticsearch

Was ist semantische Ähnlichkeit?

… im IR-Kontext?

- Selbes Thema
- Echte Synonyme
- Partonyme, Holonyme



Probleme mit Synonymen und einfachen Word-Embeddings

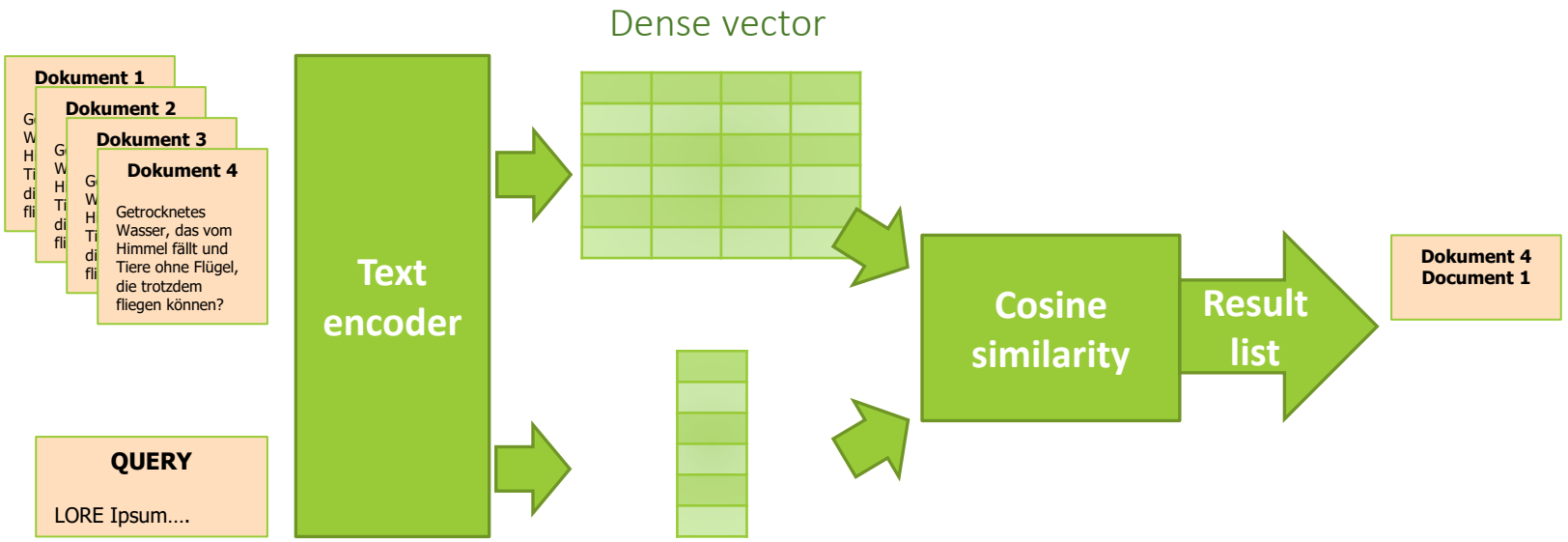
Synonyme sind kontextabhängig

- young dog → puppy
- 'hot dog' ←→ warm puppy

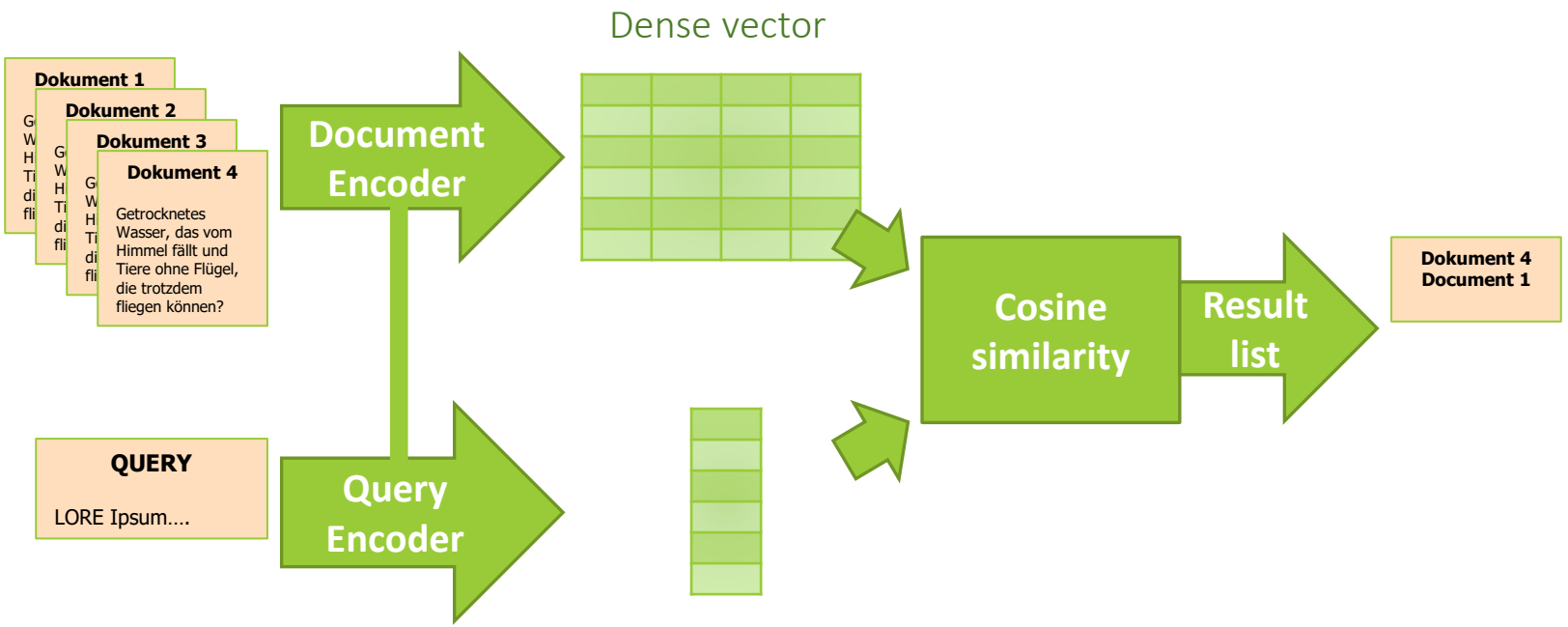
Wortbedeutungen sind ebenfalls kontextabhängig

- Die Maus nagt am Käse
- Die Programmiererin nagt an ihrer Maus

Neural search IIa – general encoder



Neural search IIb – dual encoders



Dokumentenembeddings

Bereiche:

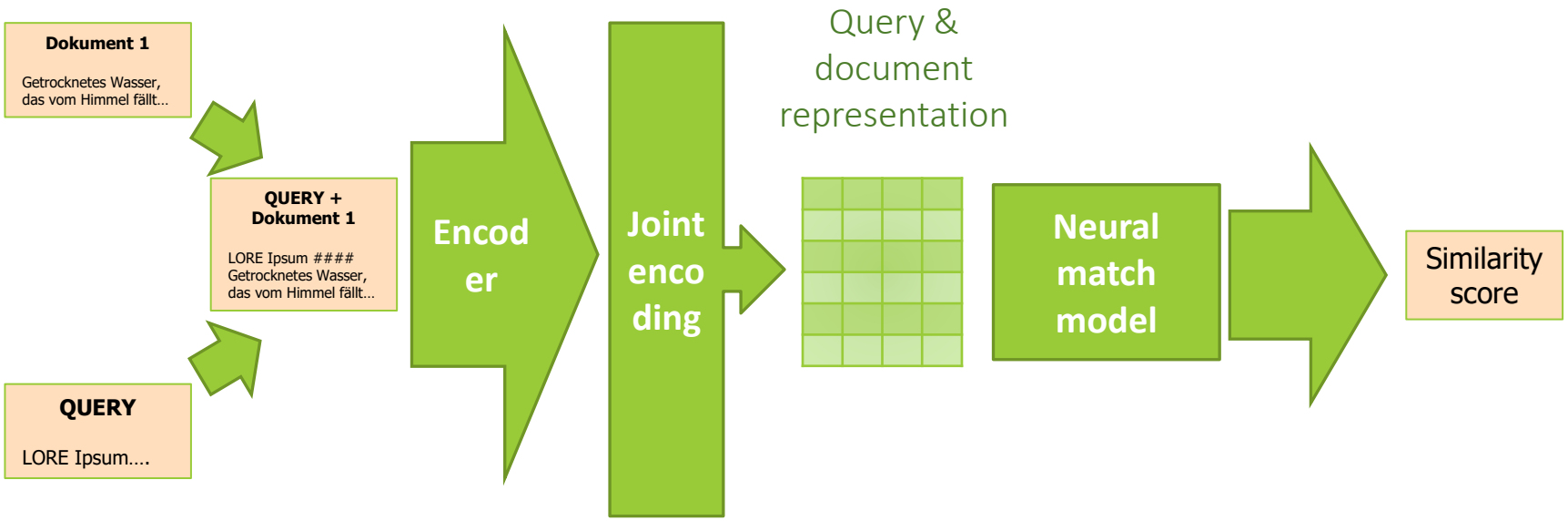
- Dokumente
- Kapitel- / Unterkapitel
- Abschnitte
- Sätze

- Methoden:
  - Durchschnitt für Wortembeddings
  - Encoder / decoder: doc2vec
  - Transformers – e.g. SBERT

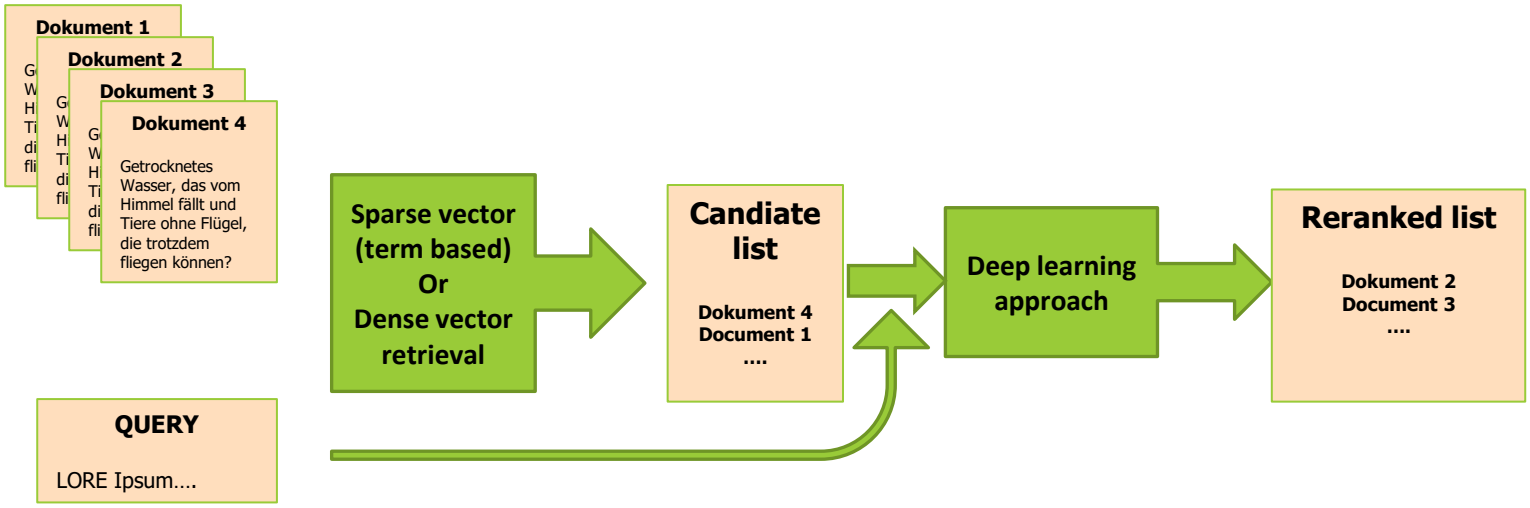
Efficient dual encoder retrieval

- Naïve similarity approach very time consuming
- nearest neighbour vector search methods

Neural search III – cross attention



Neural search IV - re-ranking



Challenges with re-ranking

- Transformers are not good at converting very long texts (i.e. documents with multiple subjects) into vectors

- Speed (see next slide)

- Training data

When to use semantic / neural search?

Pro neural search

Vocabulary of queries and documents is very different; we do not have a good synonym list

Few results for most queries, although semantically related documents are available

General domain (not very specific terminology)
OR
Training texts for domain are abundant

Other solution required

Queries and document use same vocabulary, but best documents are ranked too low
- document sources cleanup
- intent classification

Differences in vocabulary, but synonym lists are available for domain
- enhance search with synonym

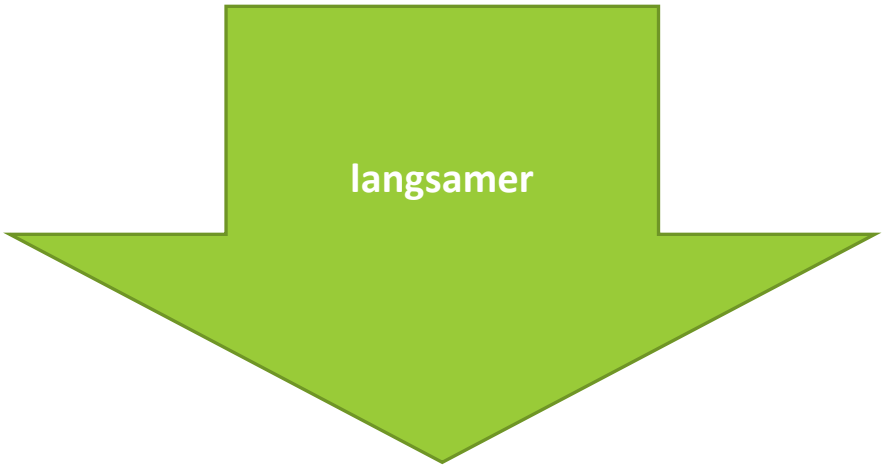
Document contain word variants
- stemming
- fuzzy search approaches
- phonetic search

Too many documents in my results
- clean up documents sources
- assign static rank

Performanz-Anforderungen

Anfragen / Sekunde

Indexierungsgeschwindigkeit



Term-basiert (sparse vector)

Vektorvergleich, Dual Encoder

Ad-hoc-Vergleich – Cross Encoder

Components for neural search

Candidate retrieval engine:
Retrieves the candidates for later reranking. Term based or vector based (nearest neigbour search)

Re-Ranking engine:
Takes query and candidate documents and re-ranks them. based on the relevancy model

Query encoder
Encodes the query as a vector

Document encoder
Encodes the document as a vector

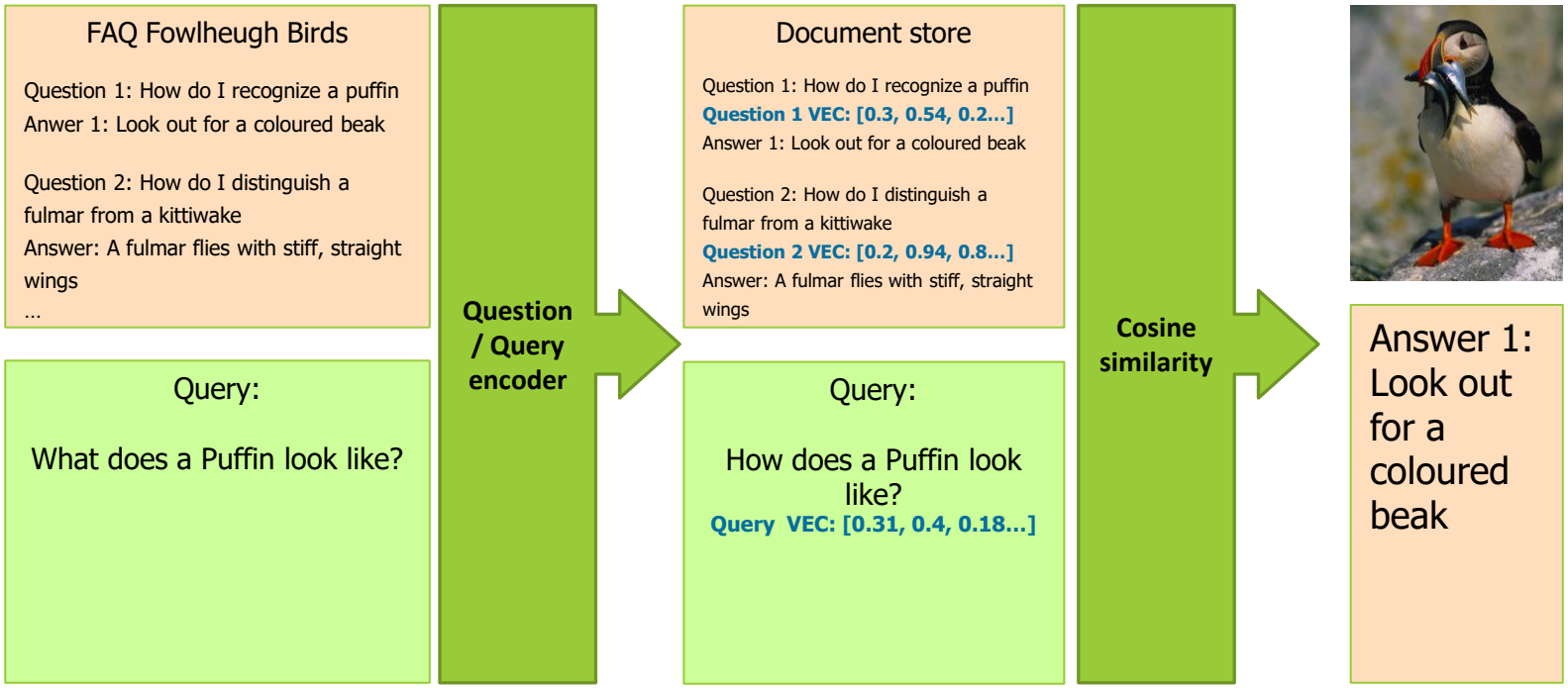
Query-Doc pair encoder
Creates a representation of the query document pair

Encoding model:
Model used for query vector creation

Encoding model:
Model used for doc vector creation

Relevancy model:
Given a query and a document representation, calculates the relevancy

Example – FAQ to QnA



Frameworks for neural search

Frameworks

- Lucene based
  - Solr
  - Elasticsearch:
    - - vector field
    - kNN search(nearest neighbour) – preview in V 8.X

Haystack
(deepset)

Jina AI

Search & vector similarity

- FAISS – Facebook
- Milvus
- Elasticsearch, Opensearch

- Weaviate
- Qdrant

Haystack-Terminologie

https://docs.haystack.deepset.ai/docs/nodes_overview

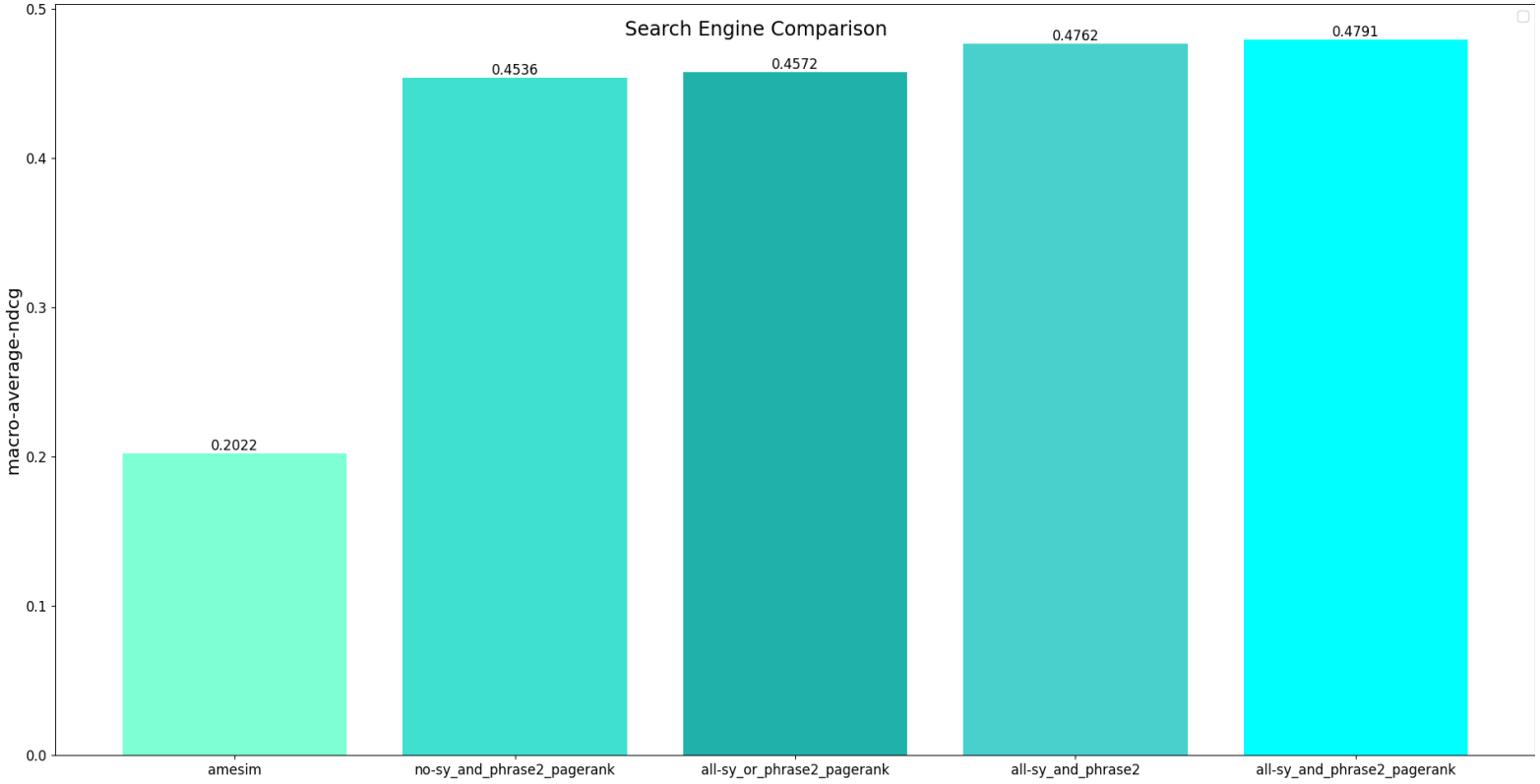
Retriever: Suche auf eigenem Dokumentenset

Ranker: Re-ranking auf gefundenen Dokumenten

SearchEngine: Nur Websuche

Reader: Weiterverarbeitung der Suchergebnisse (z.B. QnA)

Example evaluation results



Literatur

Bhaskar Mitra and Nick Craswell (2018): An Introduction to Neural Information Retrieval. https://www.microsoft.com/en-us/research/uploads/prod/2017/06/fntir2018-neuralir-mitra.pdf

Übersicht über Optionen für Neural Retrieval.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, Sanjiv Kumar (2020): Pre-training Tasks for Embedding-based Large-scale Retrieval. ICLR.

Erklären den Dual Encoder / TwoTower-Ansatz.

Andrew Yates Rodrigo Nogueira and Jimmy Lin (2021): Pretrained Transformers for Text Ranking: BERT and Beyond.

Ganz kurzer Überblick über ein Tutorial, WSDM 21

Aditya Menon, Sadeep Jayasumana, Ankit Singh Rawat, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar (2022): In Defense of Dual-Encoders for Neural Ranking. ICML.

Vergleich Dual Encoder / Reranking

Yi Luan , Jacob Eisenstein , Kristina Toutanova , Michael Collin: Sparse, Dense, and Attentional Representations for Text Retrieval. Transactions of the Association for Computational Linguistics Vol 9, 2021.

Evaluierung verschiedener Konfigurationen für Neural Retrieval

Palangi et al: Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. http://sigir.org/sigir2018/program/workshops/

LSTM-Ansatz