INFORMATION RETRIEVAL

L05. NEURAL IR

1

SUZAN VERBERNE 2022

Universiteit Leiden

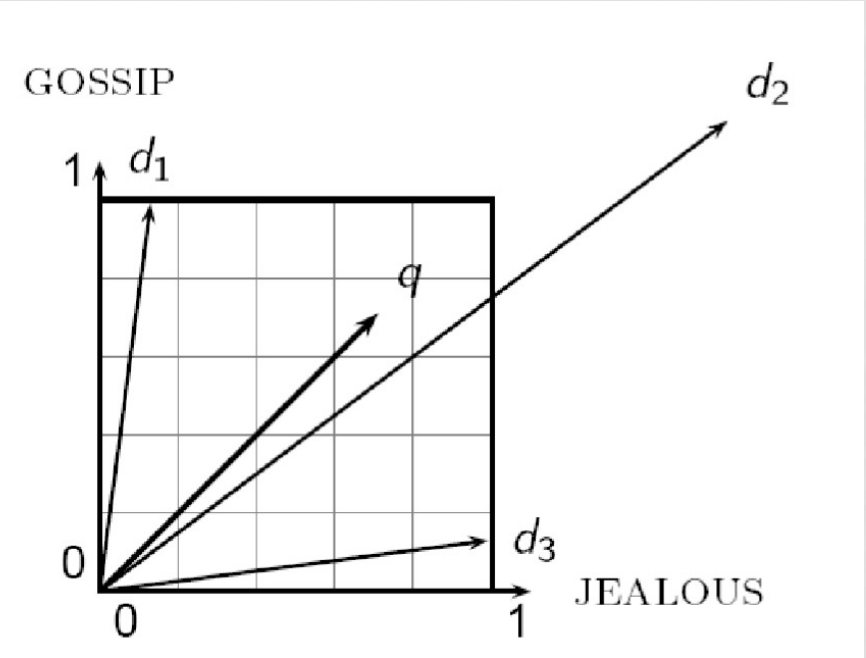
TODAY'S LECTURE

- Introduction to neural IR
- Term representations/embeddings
- Deep neural networks for IR
- BERT-based ranking models

INTRODUCTION TO NEURAL IR

RECAP OF WEEK 4

- Vector space model
  - High-dimentional vector space with one dimension for each term
  - Queries and documents are vectors in that vector space based on the terms they contain -> angle represents similarity
  - Sparse: many dimensions are zero for a given document



FROM VSM TO NEURAL IR

- Neural methods are based on the vector space model
- Neural IR architectures use dense representations instead of sparse representations

BI-ENCODER ARCHITECTURE

Encoding

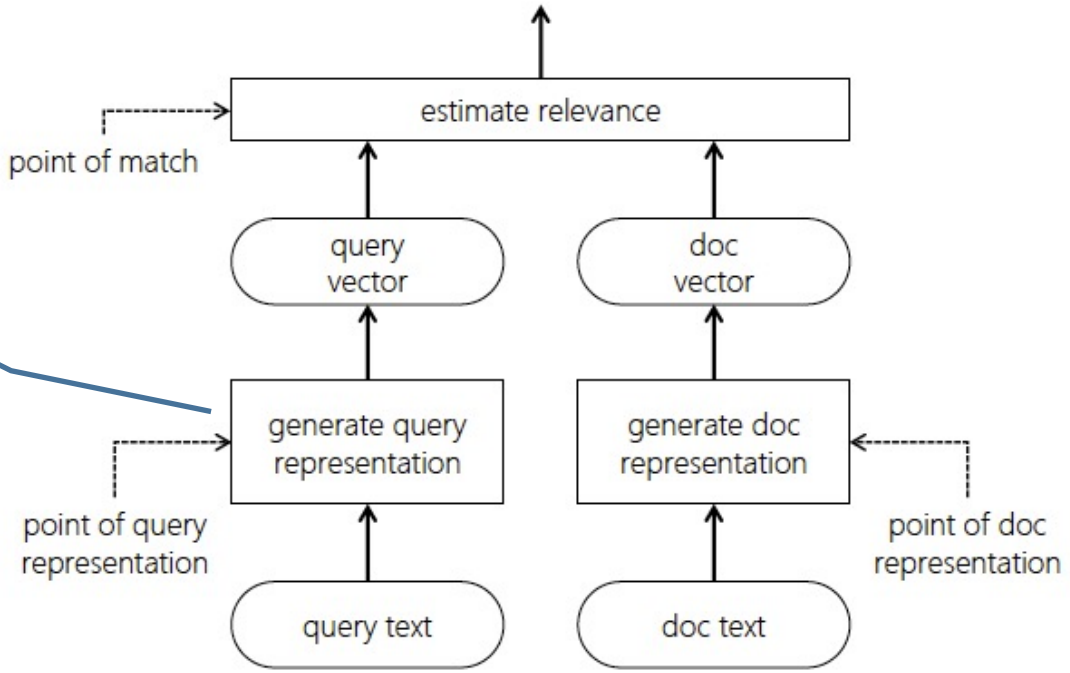


Figure 2.3: Document ranking typically involves a query and a document representation steps, followed by a matching stage. Neural models can be useful either for generating good representations or in estimating relevance, or both.

Mitra & Craswell section 2.6

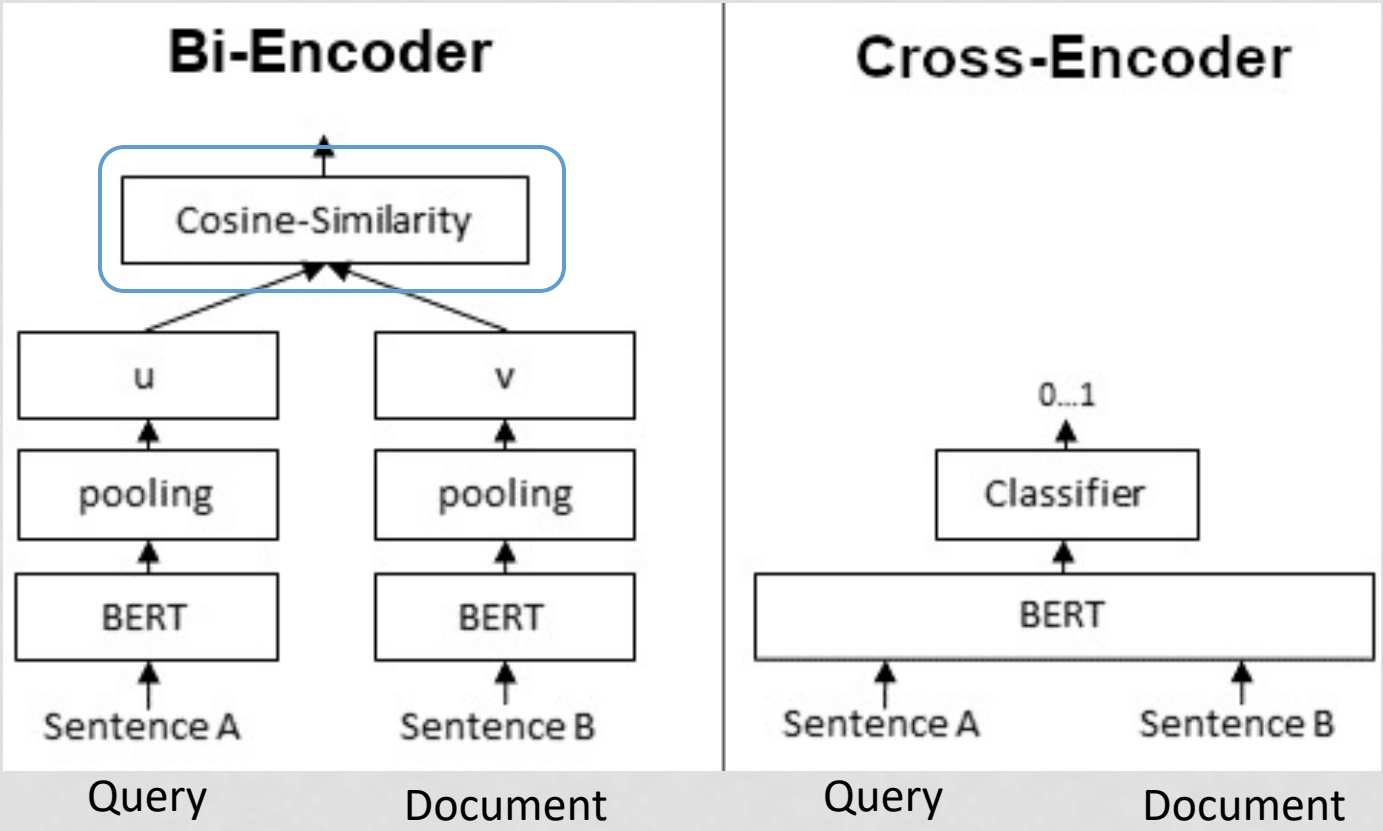
GENERAL ARCHITECTURE

Bi-encoder architecture

1. Generate a representation of the query that specifies the information need
2. Generate a representation of the document that captures the distribution over the information contained
3. Match the query and the document representations to estimate their mutual relevance

What would be a metric for relevance (similarity) between the query and the document vectors?

BI-ENCODER VS CROSS-ENCODER ARCHITECTURES



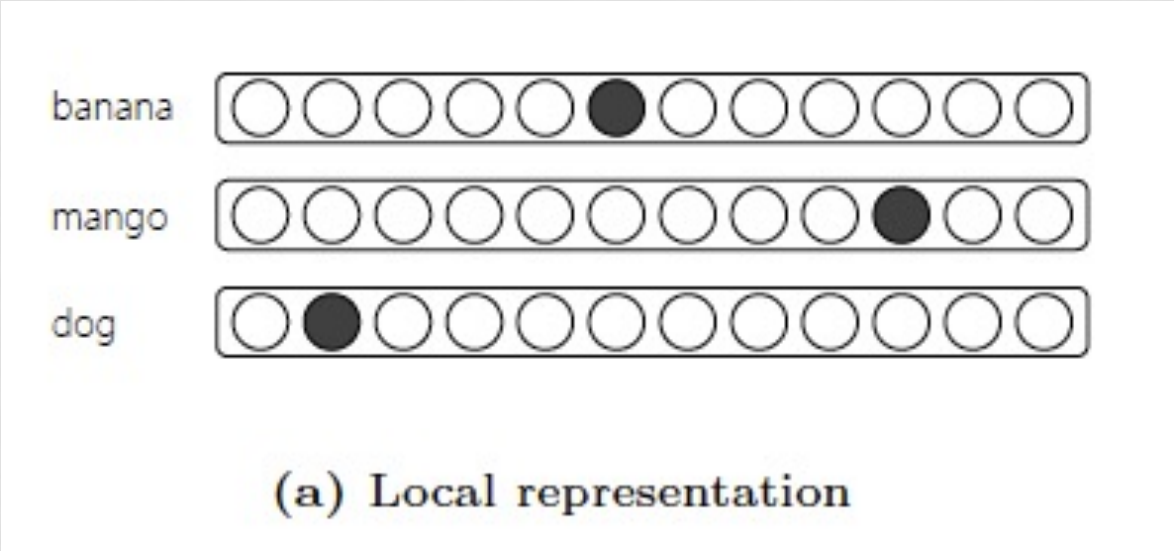
https://www.sbert.net/examples/applications/cross-encoder/README.html

TERM REPRESENTATIONS

MITRA & CRASSWELL CHAPTER 3

TERM REPRESENTATIONS

- Sparse vs dense representations
- Sparse: one-hot encoding / local representation

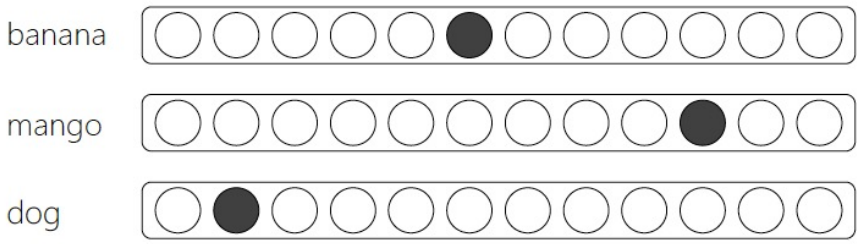


(a) Local representation

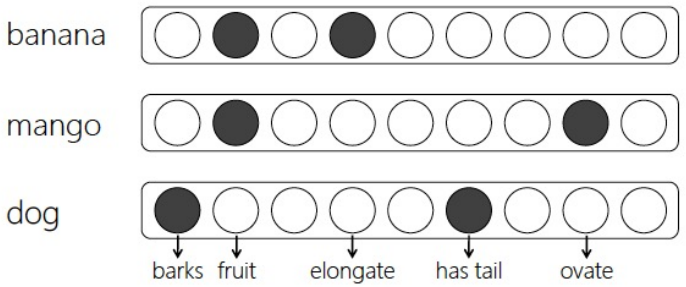
Mitra & Craswell figure 3.1

TERM REPRESENTATIONS

- Traditional IR models use local (one-hot) representations of terms
- Distributed representation:
  - Every item represented by multiple features



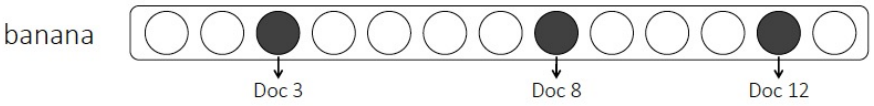
(a) Local representation



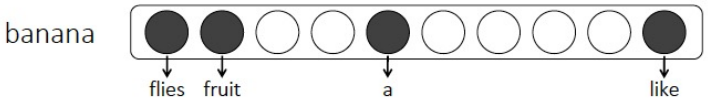
(b) Distributed representation

DISTRIBUTED REPRESENTATIONS

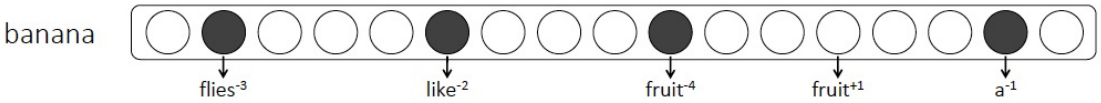
- Examples of distributed representations



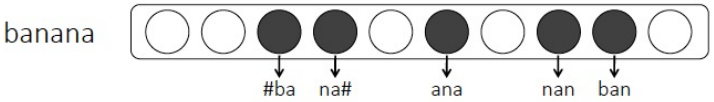
(a) In-document features



(b) Neighbouring-term features



(c) Neighbouring-term w/ distance features



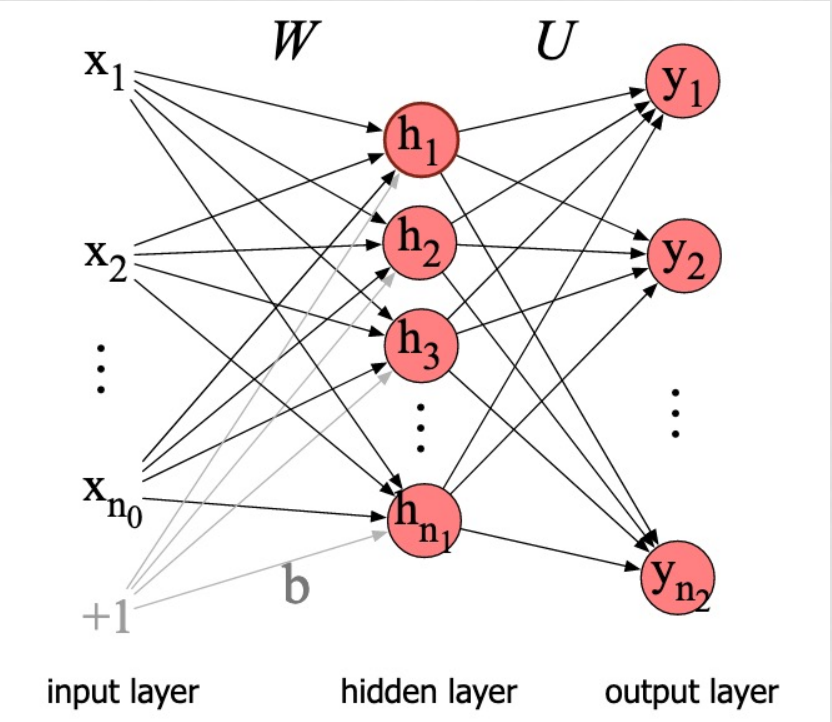
(d) Character-trigraph features

CHOICE OF FEATURES

- In IR, we want an informative term representation
  - representing meaning/semantics
- Principle: representing terms by their distributional properties
- Distributional hypothesis (Harris, 1954): terms that occur in similar context tend to be semantically similar
- **Embeddings**: dense, distributed features, learnt from distributional properties (context)

CHOICE OF FEATURES

- In the context of neural networks:
  - Local representation: each term is represented within a neural network by the activation of a single neuron
  - Distributed representations (embeddings). each term is represented by the combined pattern of activations of several neurons (Hinton, 1984)
  - The number of neurons is the dimensionality



Mitra & Craswell section 3.2
Jurafsky & Martin figure 7.8

VECTOR SPACE MODEL & EMBEDDINGS

- When the vectors are high-dimensional, sparse, and based on observable features (e.g. words) we refer to them as observed vector representations
- When the vectors are dense, lower-dimensional ($k << |T|$), and learnt from data then we instead refer to them as embeddings
  - The dimensions are latent (not labelled with words)

Dimensionality much smaller than the number of terms in the collection (vocabulary size)

TERM REPRESENTATIONS

One-hot / Term vector spaces

- Local
- Sparse
- High-dimensional
- Observable

Embeddings

- Distributed
- Dense
- Lower-dimensional (still 100s)
- Latent

DISTRIBUTED REPRESENTATIONS

- What would be an advantage of using term embeddings instead of one-hot term encodings in IR?

- Allows for in-exact matching of query to document:
  a document can be relevant to a query without the query terms occurring in the document
  - In the traditional vector space model, a document with the word *bicycle* will be far from the query *bike*.
  - The embeddings representations of bicycle and bike are highly similar

TERM EMBEDDINGS FOR IR

- More details about query-document matching with embeddings (2013-2017) in Mitra & Craswell section 4.1

- For who needs more background about neural models for text data:
  - Chapters 6 and 7 from Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed), December 2021
    - Chapter 6, Vector Semantics and Embeddings,
    - Chapter 7, Neural Networks and Neural Language Models

COMBINING LEXICAL AND EMBEDDING MODELS

- The errors made by embedding based models and exact matching (lexical) models may be different

The President of the United States of America (POTUS) is the elected head of state and head of government of the United States. The president leads the executive branch of the federal government and is the commander in chief of the United States Armed Forces. Barack Hussein Obama II (born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States.

(a) Lexical model

The President of the United States of America (POTUS) is the elected head of state and head of government of the United States. The president leads the executive branch of the federal government and is the commander in chief of the United States Armed Forces. Barack Hussein Obama II (born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States.

(b) Semantic model

Figure 7.2: Analysis of term importance for estimating the relevance of a passage to the query "United States President" by a lexical and a semantic deep neural network model. The lexical model only considers the matches of the query terms in the document but gives more emphasis to earlier occurrences. The semantic model is able to extract evidence of relevance from related terms such as "Obama" and "federal".

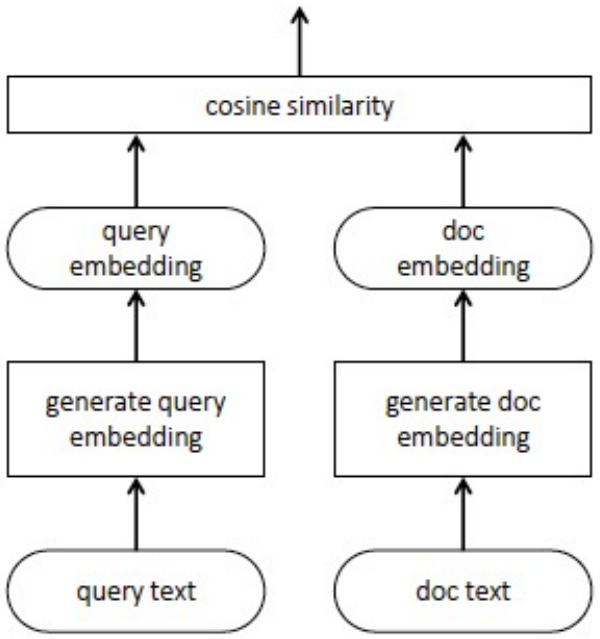
COMBINING LEXICAL AND EMBEDDING MODELS

- The errors made by embedding based models and exact matching (lexical) models may be different—and the combination of the two is often preferred
- Another technique is to use the embedding based model to re-rank the top documents retrieved by a lexical IR model
- Most common set-up nowadays:
  - Step 1: probabilistic retrieval and ranking with BM25 (lecture 6)
  - Step 2: re-ranking with a supervised neural model (lecture 9)

DEEP NEURAL NETWORKS FOR IR

MITRA AND CRASWELL CHAPTER 7

BI-ENCODER ARCHITECTURE



(c) Learning query and document representations for matching (*e.g.*, (Huang *et al.*, 2013; Mitra *et al.*, 2016a))

- Can be used in a supervised as well as an unsupervised setting
  - Supervised: learn embeddings from labelled data (relevant and non-relevant query-document pairs)
  - Unsupervised: learn embeddings for documents from unlabelled collection. Project query in same space and rank by cosine similarity

SUPERVISED LEARNING

- Training data for information retrieval consists of:
  - a collection of search queries
  - a corpus of candidate documents
  - labels—in the form of either explicit human relevance judgments or implicit labels (e.g., from clicks)—for query-document pairs

Relevance assessments

SUPERVISED LEARNING

- Large test collections for common retrieval tasks
  - MS Marco
  - Robust 04
  - TREC collections, e.g. the TREC Deep Learning track
    - see all past tracks on https://en.wikipedia.org/wiki/Text_Retrieval_Conference
- For tasks in other domains:
  - use unsupervised representation learning
  - + a smaller collection for model fine-tuning

CHALLENGES OF LONG DOCUMENTS

- Retrieval tasks for long documents
  - archival documents
  - scientific papers
  - patents
  - long web pages

- Challenges:
  - mixture of many topics, query matches may be spread
  - neural model must aggregate the relevant matches from different parts

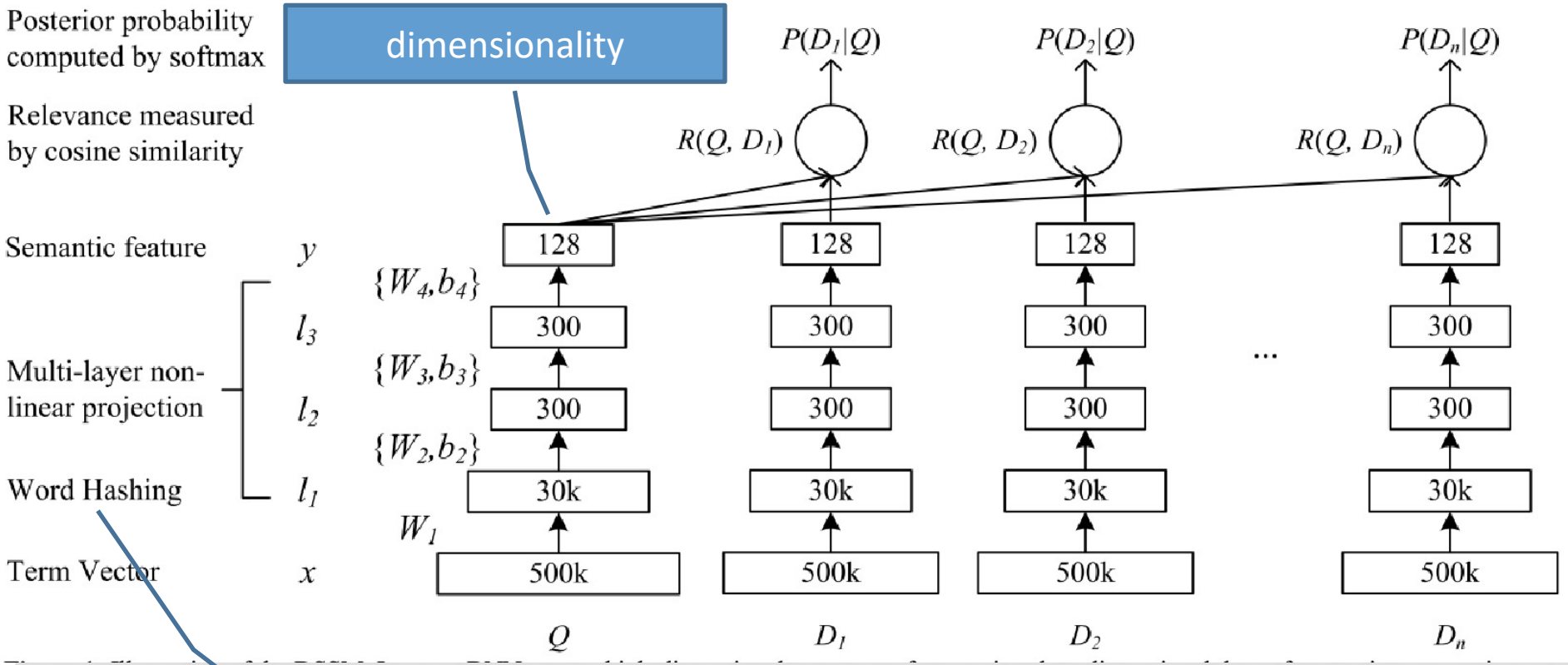
CHALLENGES OF SHORT DOCUMENTS

- Retrieval tasks for short documents
  - social media data
  - question answering
  - sentence retrieval/claim retrieval for tasks such as stance classification

- Challenges:
  - fewer query matches
  - but neural model is more robust towards the vocabulary mismatch problem compared to lexical term-based matching models

NEURAL IR ARCHITECTURES

DEEP SEMANTIC SIMILARITY MODEL



Huang et al. (2013). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management* (pp. 2333-2338).

DEEP SEMANTIC SIMILARITY MODEL

- Both the pieces of texts are represented as bags-of-character-trigrams (‘word hashing’)
- The DSSM architecture consists of two deep models—for the query and the document
- with all fully-connected layers
- and cosine distance as similarity function between them

called a ‘Siamese network’ by Mitra and Crasswell

Mitra & Craswell section 7.2

DEEP SEMANTIC SIMILARITY MODEL

Train on clickthrough data:

- Each training sample consists of
  - a query q
  - a positive document d+: a document that was clicked by a user on the search engine result page (SERP) for that query
  - a set of negative documents D− randomly sampled from the full collection.
- trained by minimizing the cross-entropy loss

$$\mathcal{L}_{dssm}(q, d^+, D^-) = -log\left(\frac{e^{\gamma \cdot cos\left(\vec{q},\vec{d^+}\right)}}{\sum_{d \in D} e^{\gamma \cdot cos\left(\vec{q},\vec{d}\right)}}\right)$$

$$\text{where,} \quad D = \{d^+\} \cup D^-$$

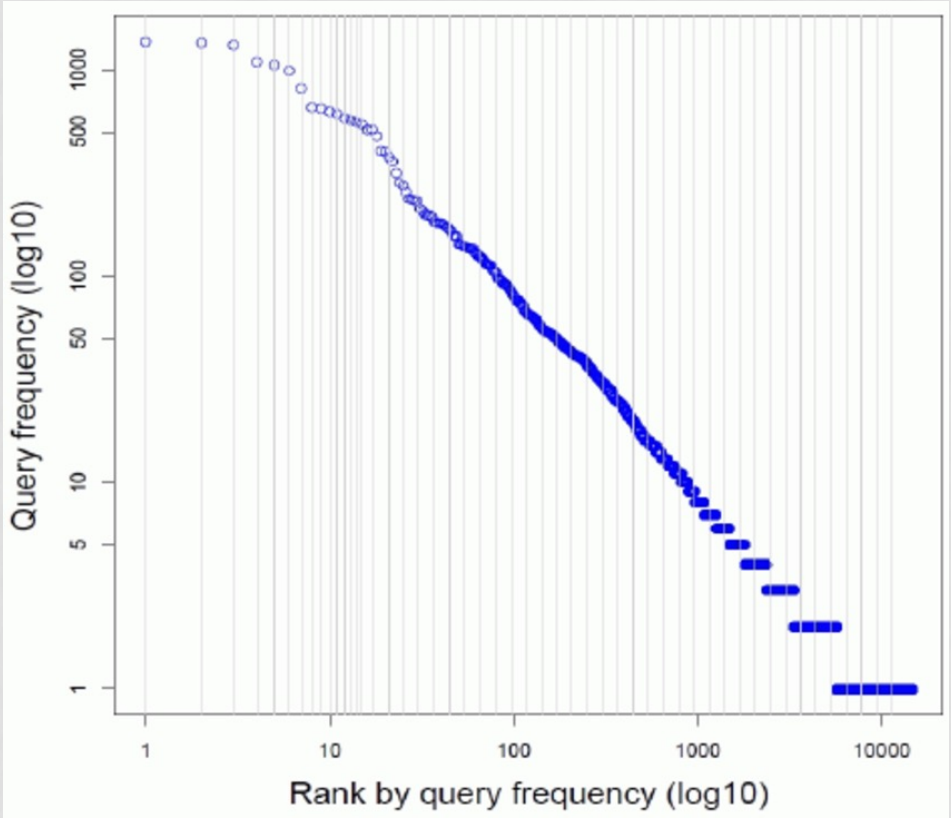
THE LONG TAIL PROBLEM

- Focus of neural IR research: learning good representations of text
- Challenge: rare terms, documents, and search intents
- Query frequencies in most IR tasks follow a Zipfian distribution. E.g. In the AOL query logs
  - > 70% of the distinct queries are seen only once in the period of three months from which the queries are sampled
  - > 50% of the distinct documents are clicked only once

Mitra & Craswell section 7.4

THE LONG TAIL PROBLEM

- A good IR method must
  - be able to retrieve infrequently searched-for documents
  - perform reasonably well on queries with rare terms



https://dl.acm.org/doi/pdf/10.1145/1555400.1555445

THE LONG TAIL PROBLEM

- Example query: Yavoriv base rocket attack
  - Yavoriv is rare -> recent documents with Yavoriv are likely to be relevant (exact matching)
  - but because it is rare, latent pre-trained models don't have a good representation of it

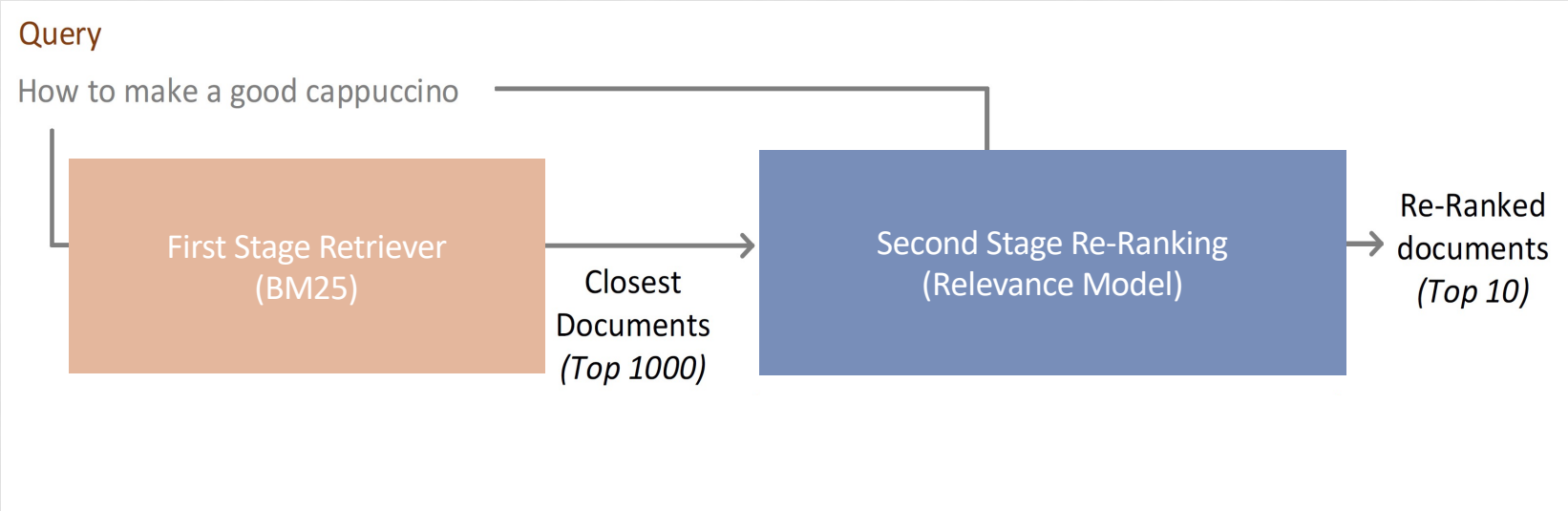
- A good neural IR model should incorporate both lexical and semantic matching signals

BERT-BASED RANKING MODELS

NOT IN MITRA AND CRASWELL

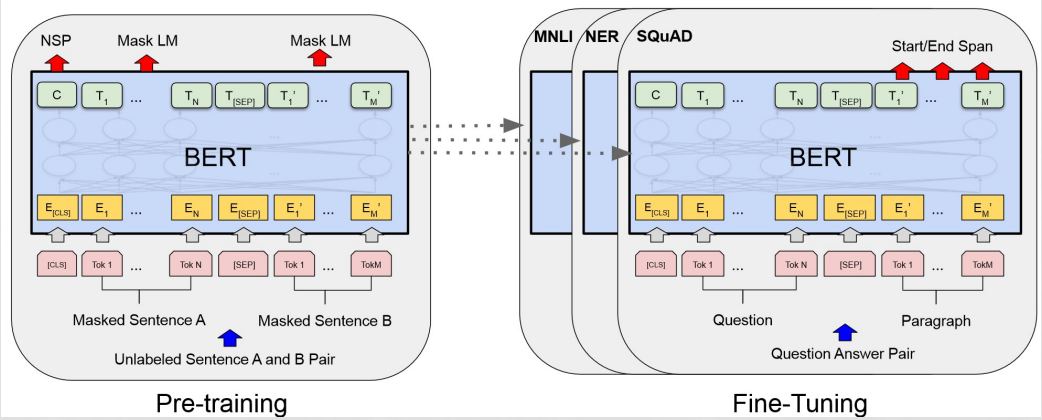
TWO-STAGE RETRIEVAL

- Given a query
  - First stage retrieval from the whole corpus with lexical matching (highly effective probabilistic model: BM25, next week)
  - Re-ranking of top-n retrieved documents with a neural model



NEURAL RE-RANKING

- State-of-the-art neural re-ranking models are based on pre-trained BERT embeddings



Devlin et al. (2019) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" https://aclanthology.org/N19-1423.pdf

- For who is new to Transformer architectures and BERT:
  Jurafsky & Martin, https://web.stanford.edu/~jurafsky/slp3/ , chapter 9, section 9.7, "Self-Attention Networks: Transformers"

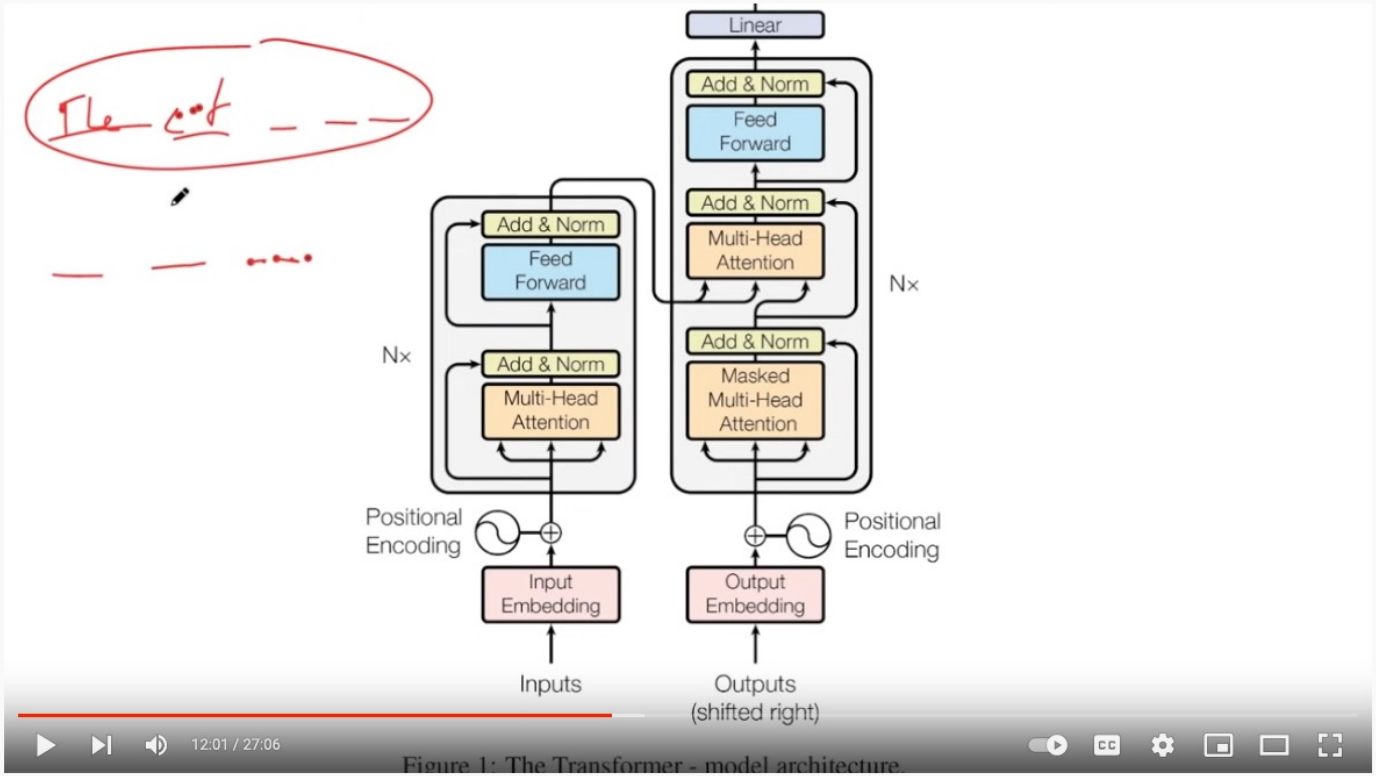
TRANSFORMER ARCHITECTURE

- The Transformer architecture is a sequence-to-sequence architecture
  - Uses self-attention: computes strength of relation (dot product) between pairs of input words/embeddings
  - Efficient because input is processed in parallel
  - Can model long-distance term dependencies because the complete input is processed at once
    - (but if too long, it is too heavy because of quadratic complexity)

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

TRANSFORMER ARCHITECTURE

- If you are interested:



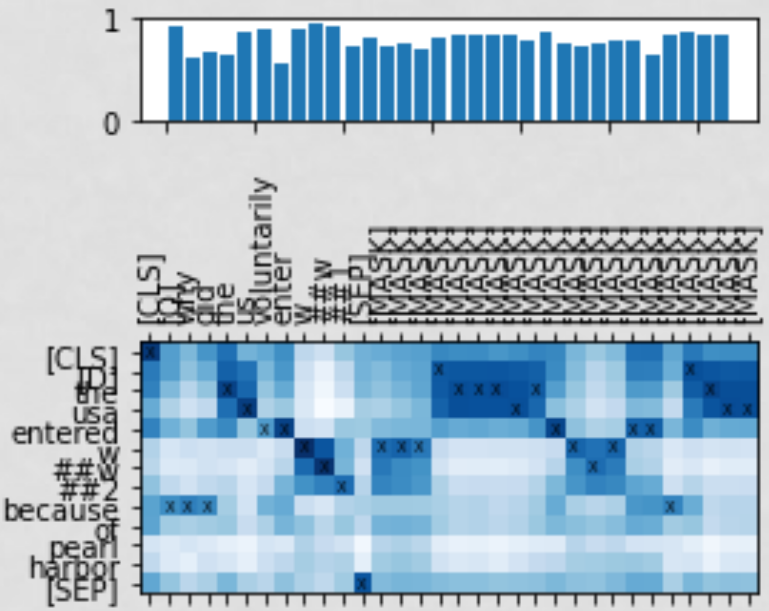
https://www.youtube.com/watch?v=iDulhoQ2pro

Universiteit Leiden

VIEWING ATTENTION

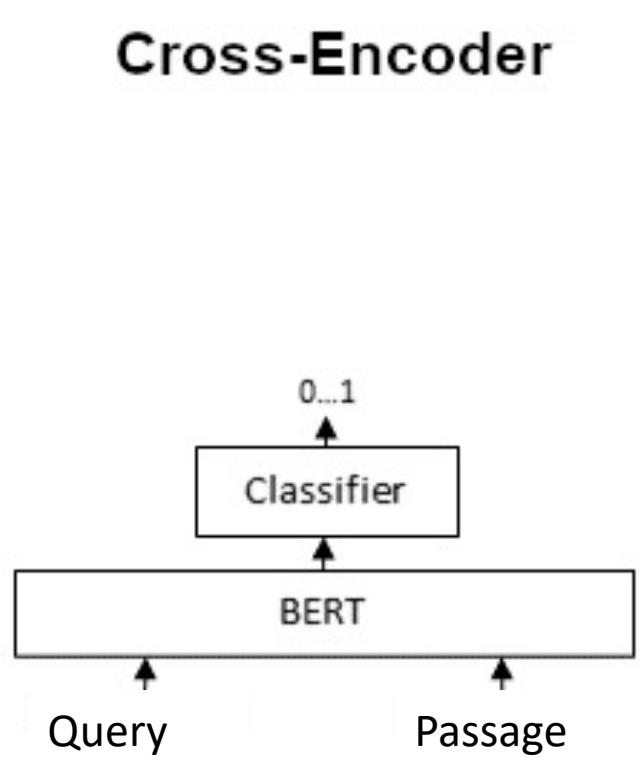
- One of the properties of the self-attention mechanism is that we can output the amount of attention from one token to another
- We can visualise that e.g. in a heatmap
- In this visualisation, the x indicates for each query term embedding the most similar document embedding

```
factory.explain_text("why
did the us voluntarily
enter ww1", "the USA
entered ww2 because of
pearl harbor")
```

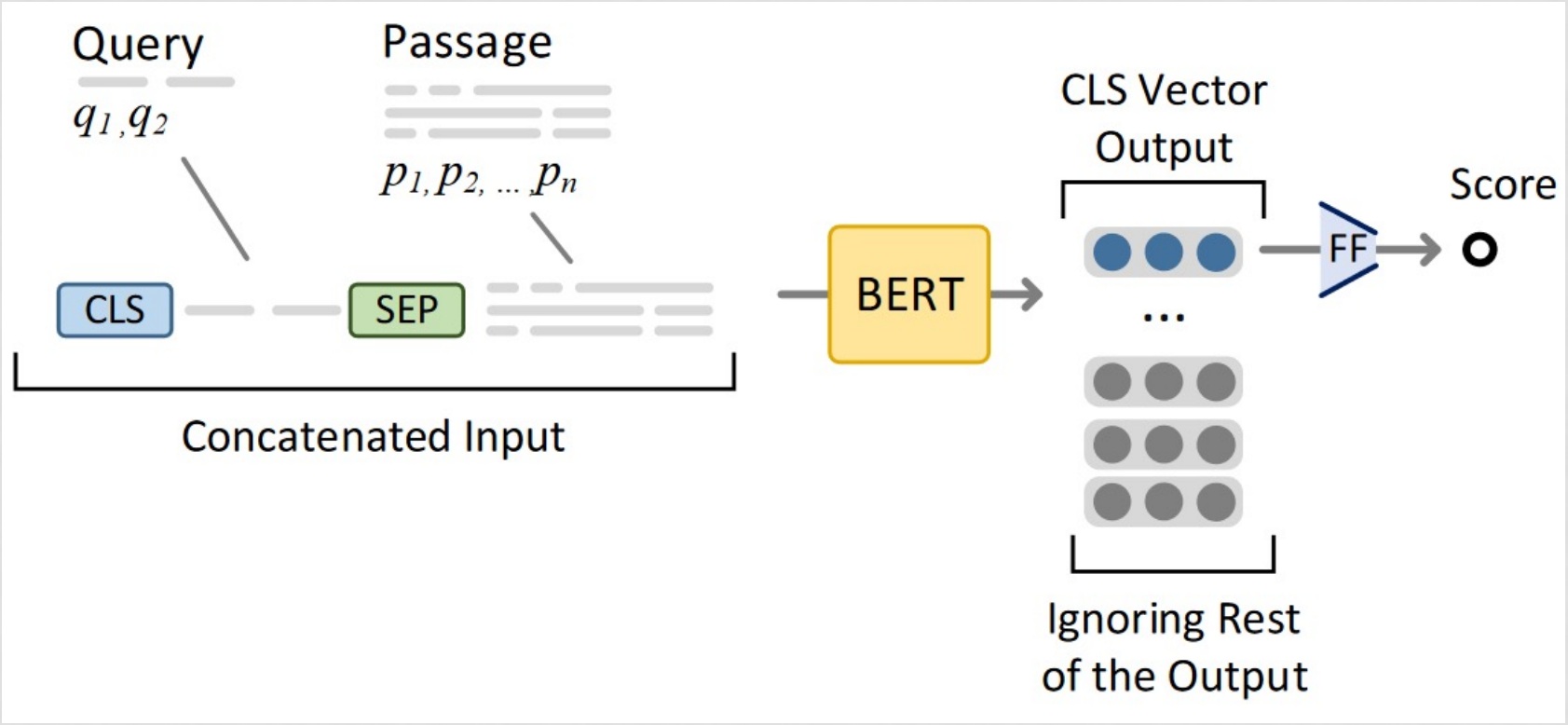


$BERT_{CAT}$ MODEL

- Fine-tune BERT on relevance classification of query-passage pair
- Concatenate query and passage
  - input: "[CLS] <query> [SEP] <passage>"
  - Predict the score with a single linear layer
- Needs to be repeated for every passage



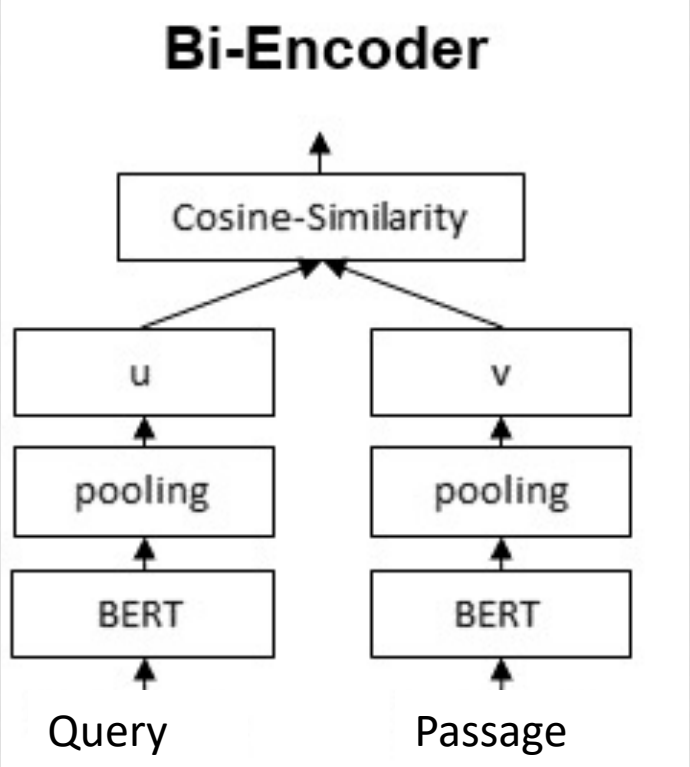
$BERT_{CAT}$ MODEL



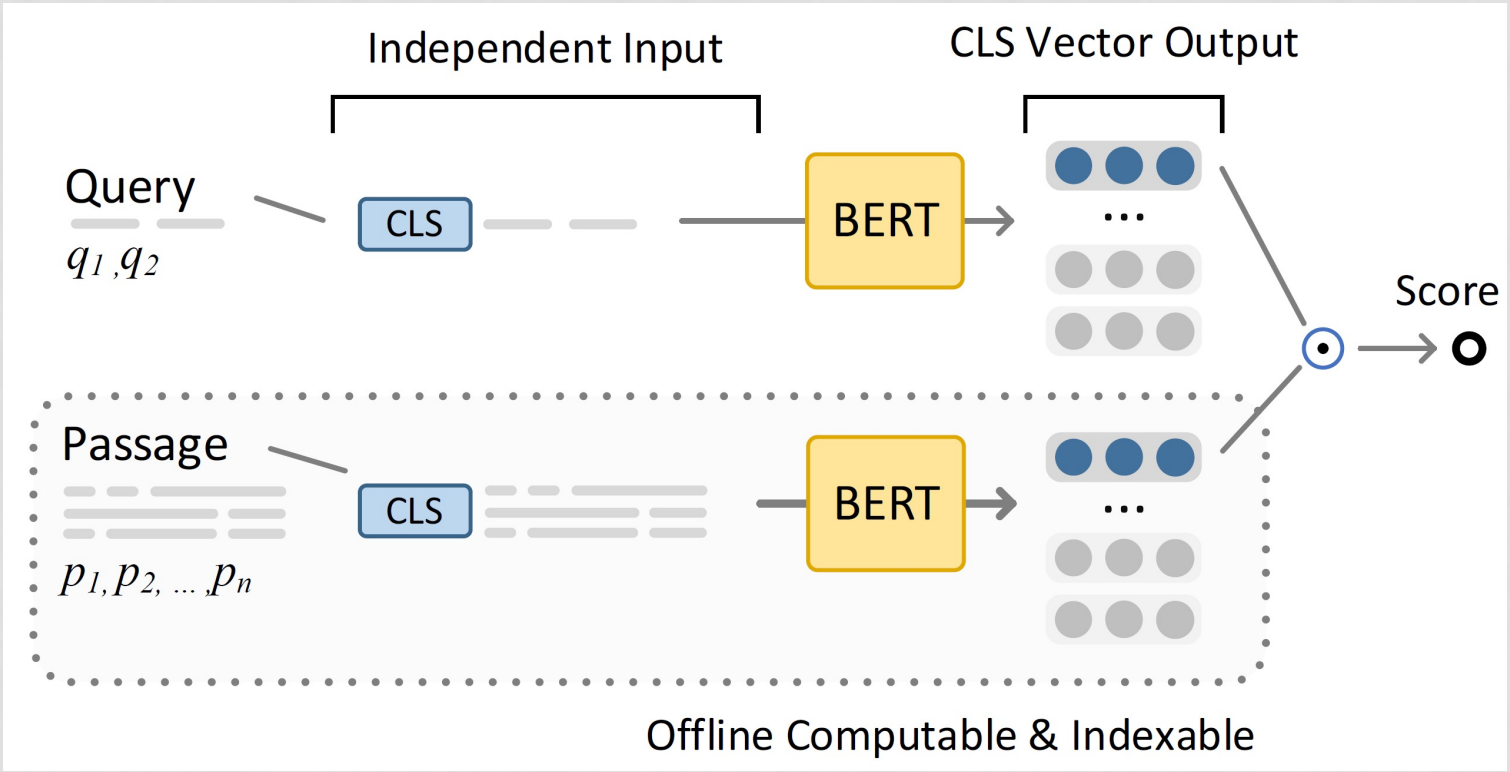
Nogueira and Cho (2019), "Passage Re-ranking with BERT"

$BERT_{DOT}$ MODEL

- Passages and queries are both compressed into a single vector
  - Passages are completely independent -> moves most computation into the indexing phase
  - Only need query encoding at runtime
- Relevance is scored with a dot-product
  - Cosine-similarity variants also exist
  - This allows easy use of an (approximate) nearest neighbor index



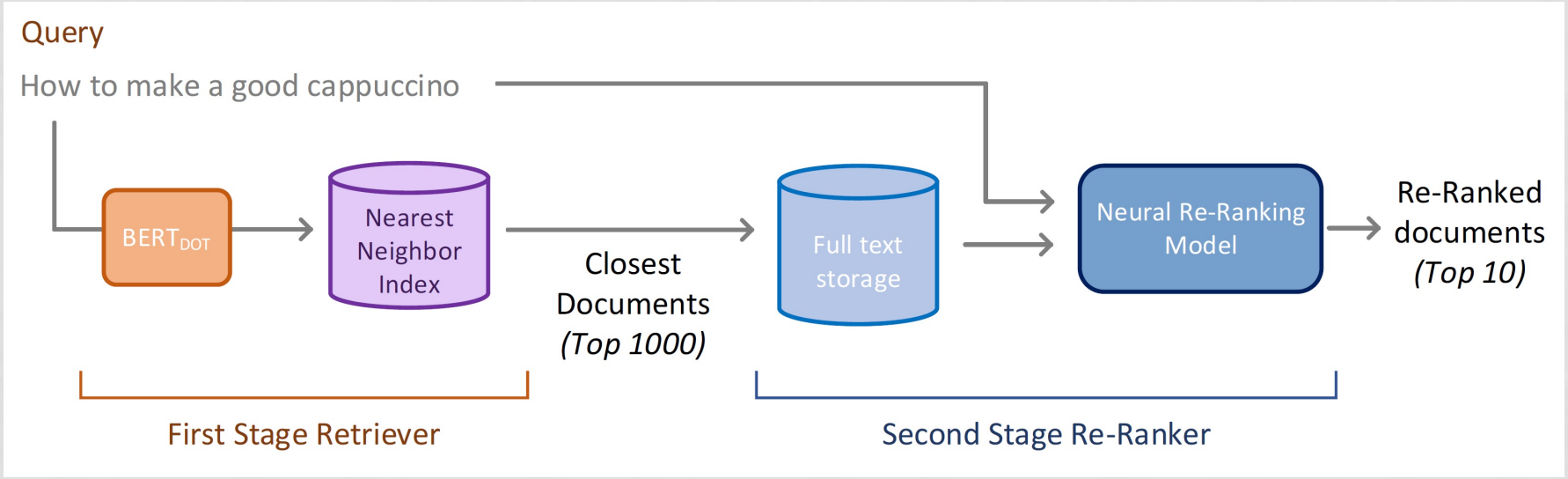
$BERT_{DOT}$ MODEL



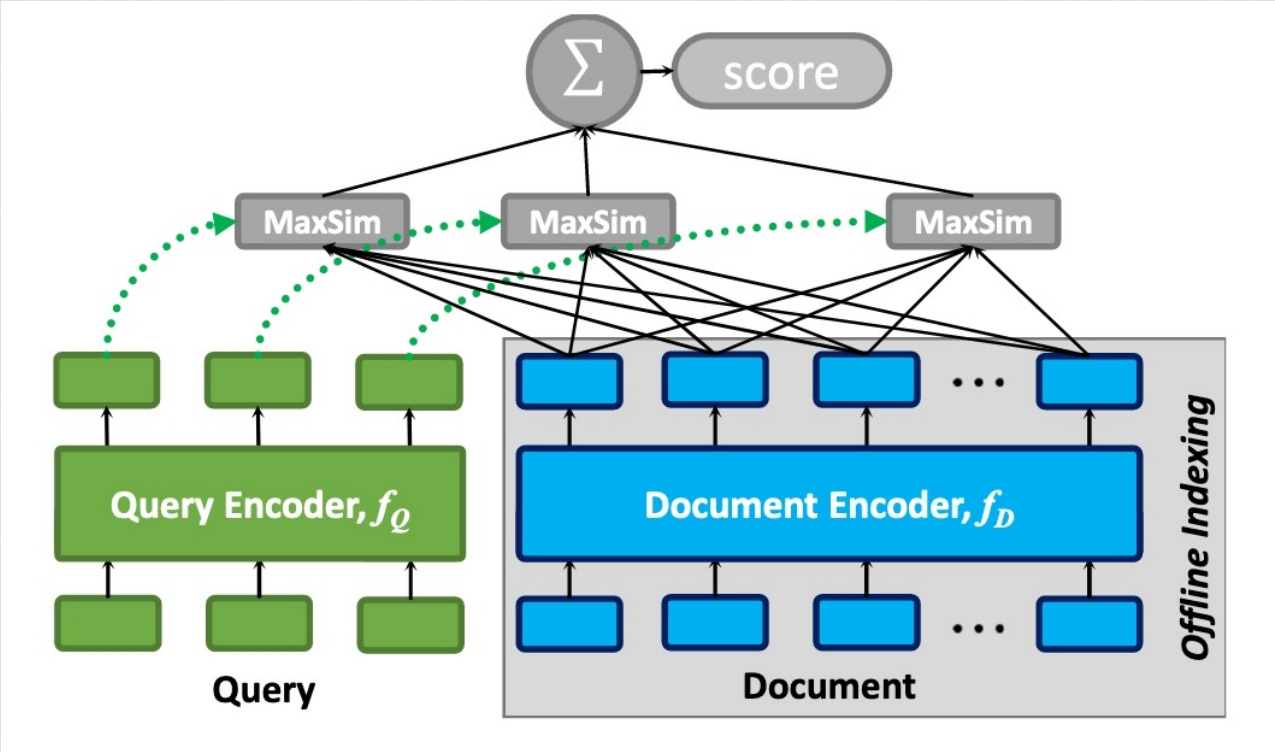
Karpukhin et al. (2020), "Dense Passage Retrieval for Open-Domain Question Answering"

DENSE RETRIEVAL

- Dense retrieval is neural first-stage retrieval (replaces lexical stage)
  - Using a neural query encoder & nearest neighbor vector index
  - Can be used as part of a larger pipeline

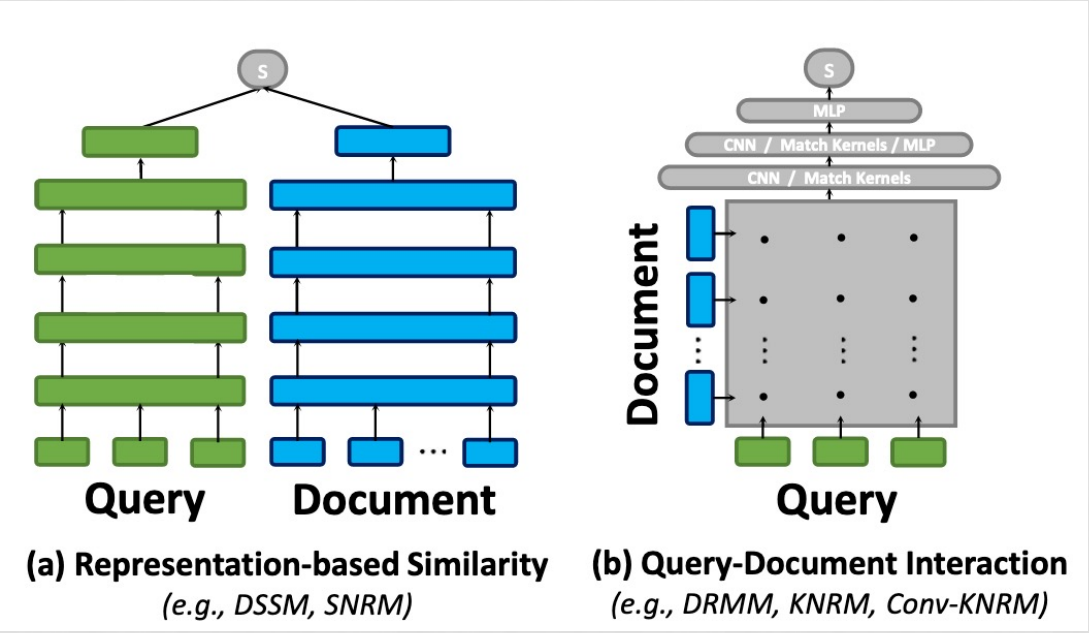


COLBERT



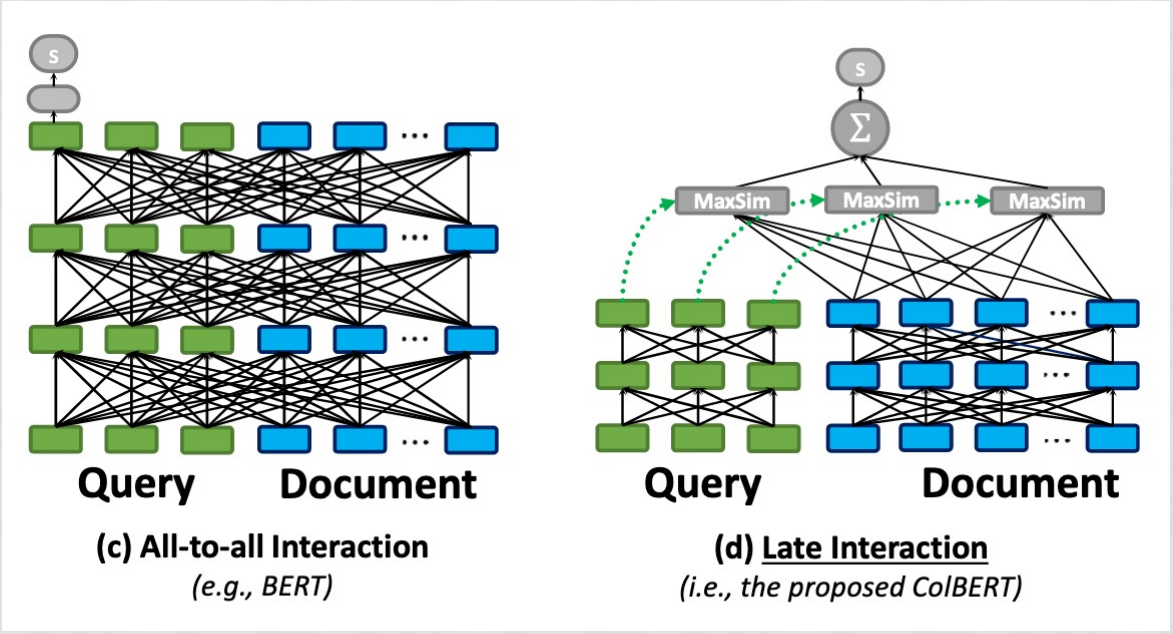
Khattab & Zaharia (2020), "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT", https://dl.acm.org/doi/abs/10.1145/3397271.3401075

Deep Semantic Similarity Model (2013)



Interaction-based methods (2013-2018)

Cross-encoder $BERT_{cat}$ (2019)



ColBERT (2020)

CONCLUSIONS

HOMEWORK

- Read Mitra and Craswell, Introduction to Neural Information Retrieval, chapter 4 and 7 (Brightspace), skip 4.2

- Homework exercises on Brightspace (Assignments -> week 5)
  - This time a practical exercise with colBERT in PyTerrier
  - Submit your answers through Brightspace before or on Sunday, March 20, 23.59
  - Submit 1 PDF file (any formatting is good)
  - If you need help, you can contact the TAs at ircourse@liacs.leidenuniv.nl

AFTER THIS LECTURE…

- You can list the differences between one-hot encodings and embeddings as term representations
- You can explain why embeddings based on distributional semantics can be beneficial for IR
- You can explain why combining lexical matching and semantic matching leads to the best models
- You can explain how to use a neural IR model in an unsupervised and in a supervised manner
- You can list the particular challenges of long and short documents
- You can define and describe the long-tail problem in IR
- You can explain the following architectures: deep semantic similarity model (DSSM), $BERT_{cat}$, $BERT_{dot}$, and colBERT