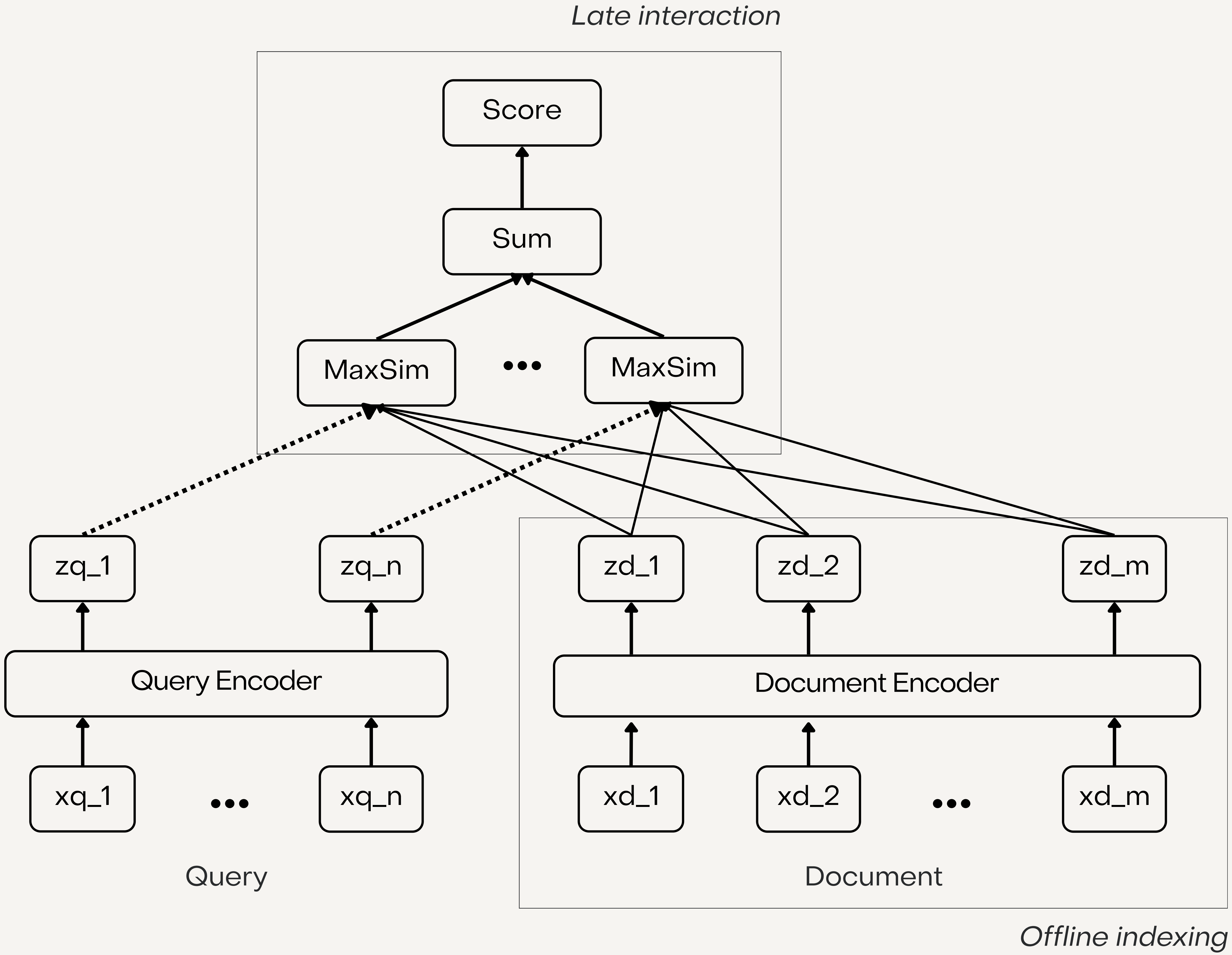
The ColBERT Dense Retrieval Model

Weronika Łajewska

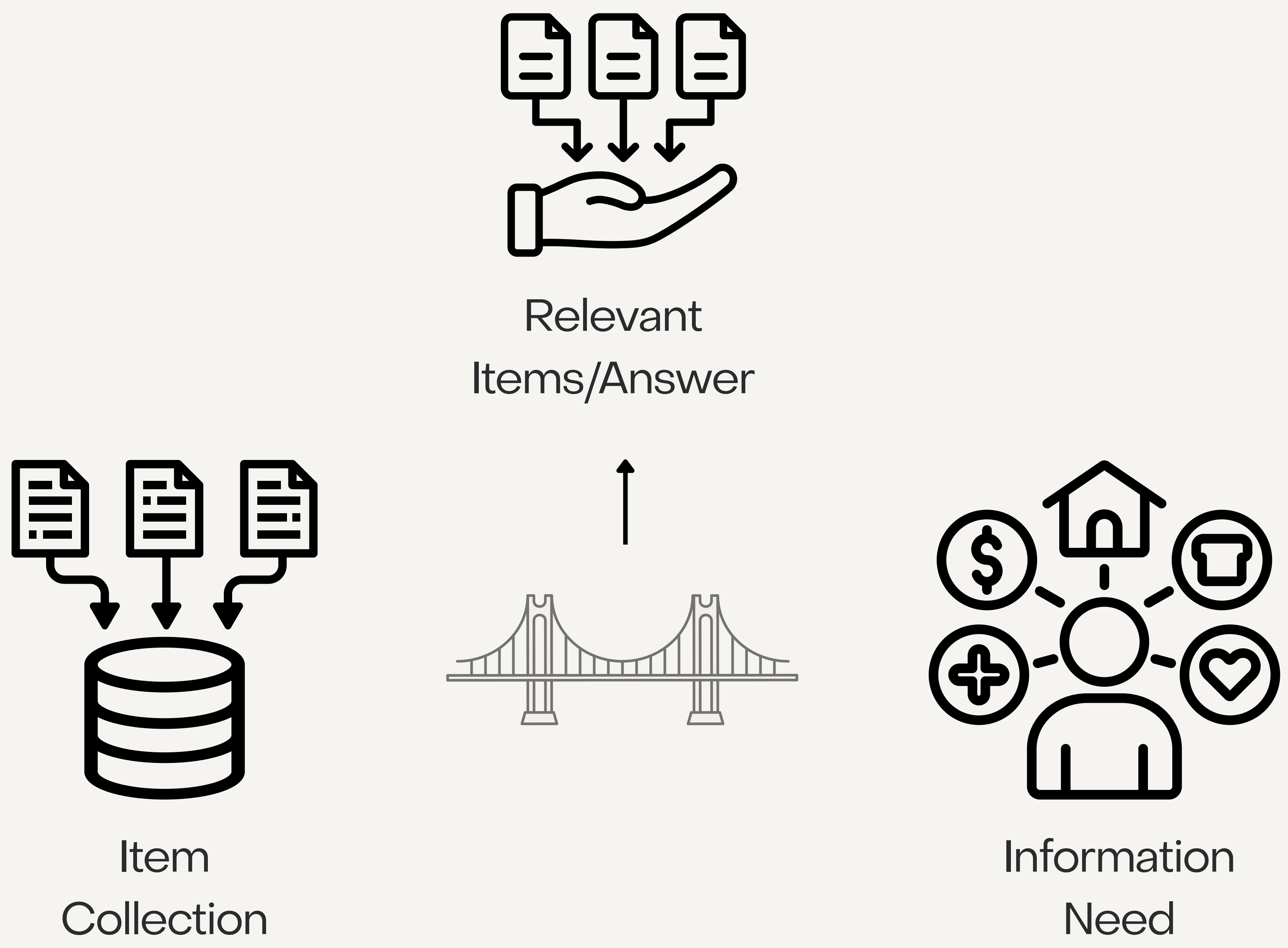
Information Access and Artificial Intelligence Research Group
University of Stavanger, Norway

ColBERT



Omar Khattab and Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.* In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)

Information Access

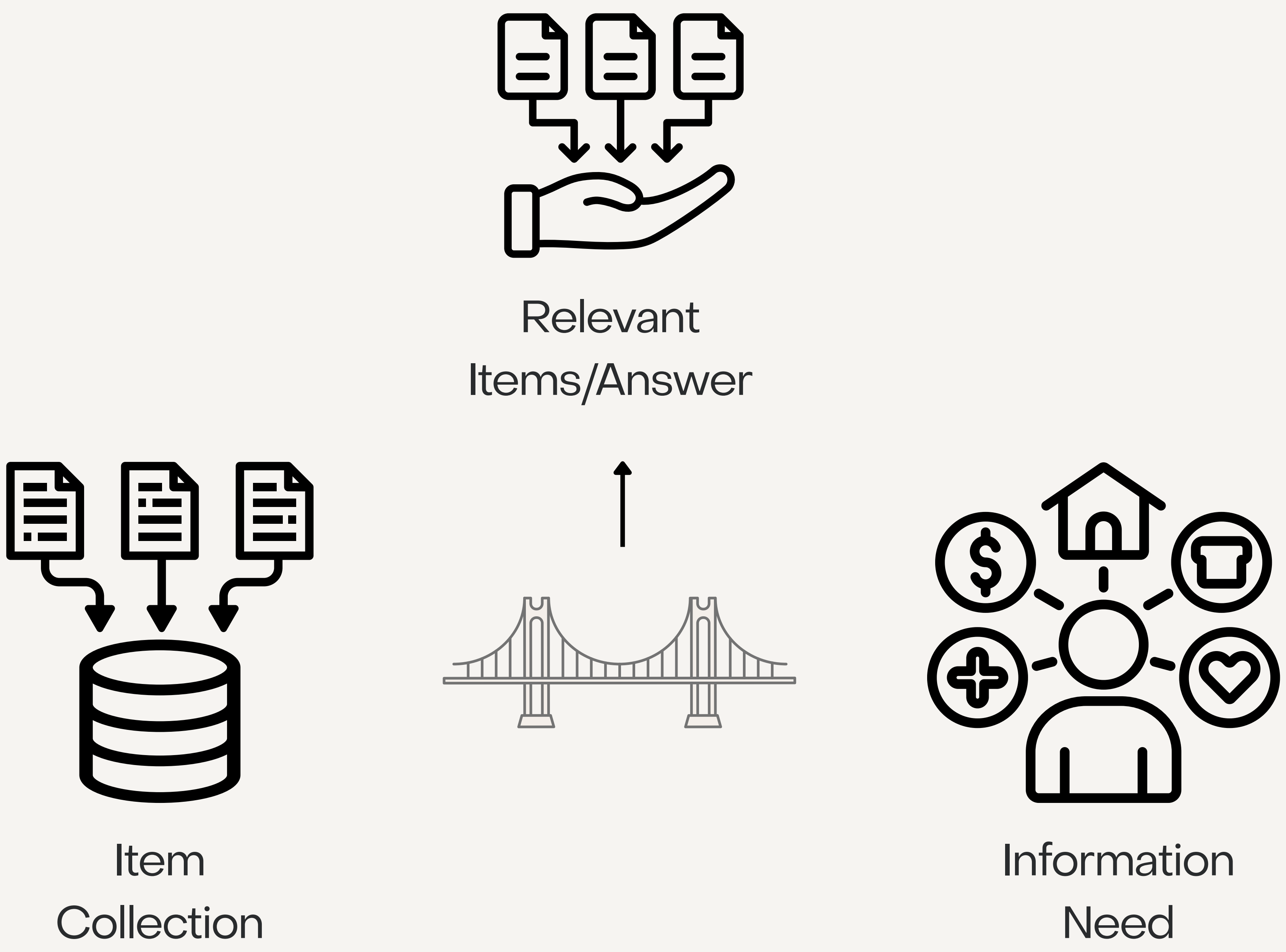


Goal: help people find, retrieve, and use relevant information from large collections

Given an item collection and a user's information need, how can the system retrieve the most relevant items to meet that need?

Information access systems bridge the gap between a collection of items (traditionally, documents) and a user's information need

Information Access



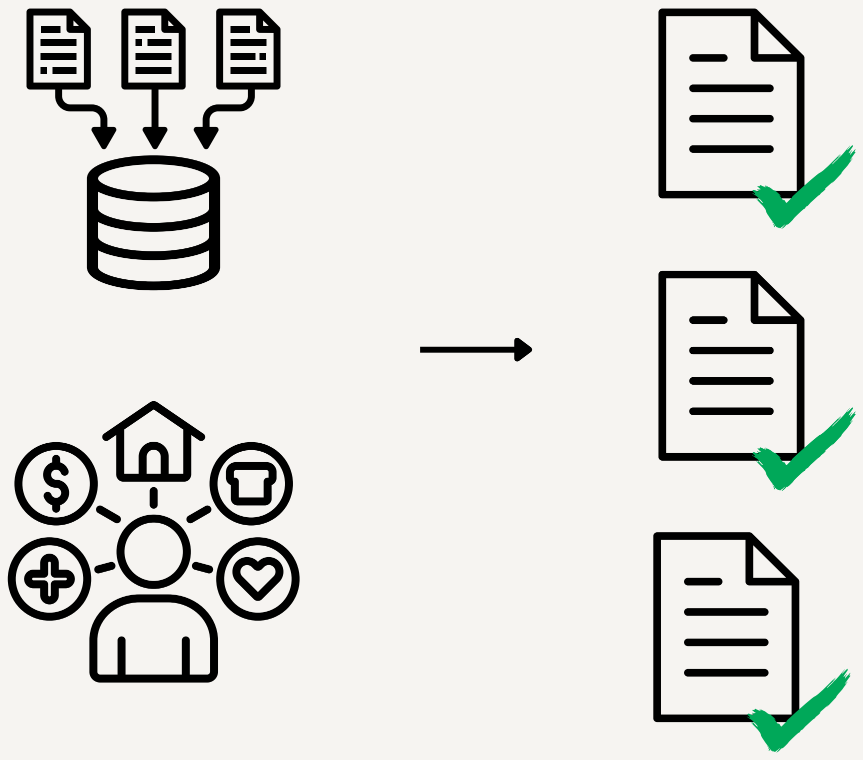
Goal: help people find, **retrieve**, and use relevant information from large collections

Given an item collection and a user's information need, how can the system retrieve the most **relevant items** to meet that need?

Information access systems bridge the gap between a **collection of items** (traditionally, documents) and a **user's information need**

Information Retrieval (IR)

How to match information needs and information objects?



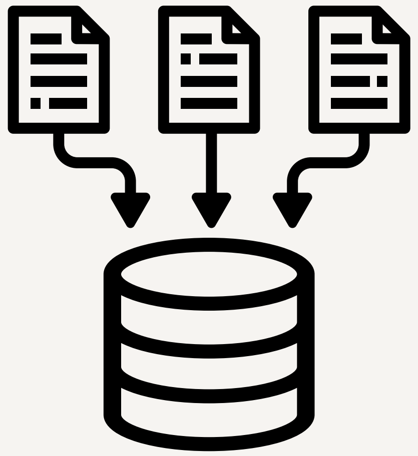
IR is the process of obtaining some information resources relevant to an information need from within large collections

Traditional search systems provide a ranked list of documents in response to a query

Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information [1]

[1] Salton, G. (1968). *Automatic Information Organization And Retrieval.* McGraw Hill Text.

Document Corpus



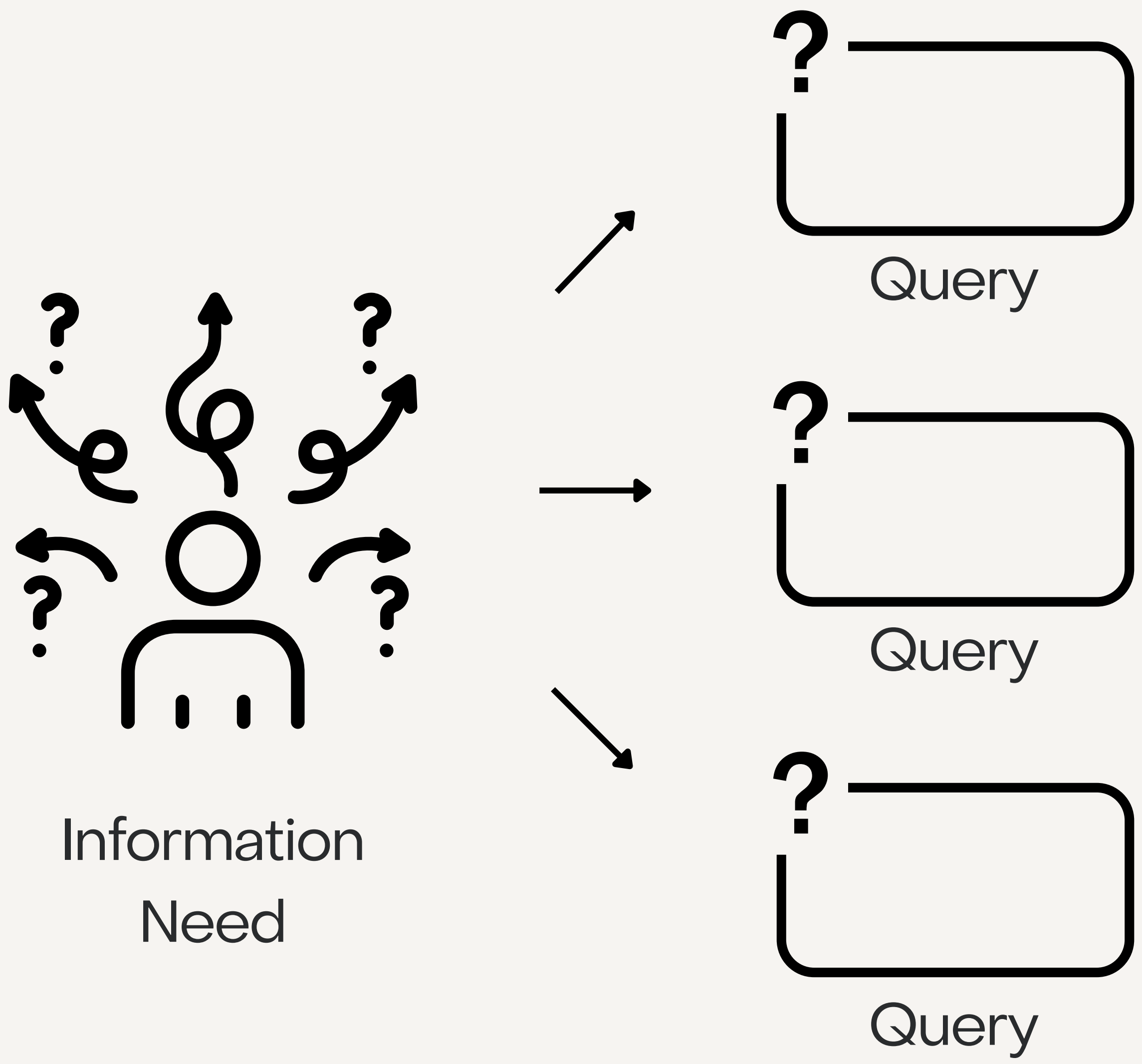
Document corpus in a large set of documents used to evaluate and develop search systems

What Do They Contain?

- Web pages (e.g., ClueWeb, GOV2)
- News articles (e.g., Washington Post Corpus)
- Scientific papers (e.g., CORD-19)
- Social media posts, forums, product reviews, etc.

- Size: Thousands to billions of documents
- Domain: General, biomedical, legal, etc.
- Structure: Plain text, structured fields, multimodal

User's Information Need



Information need is the underlying motivation or goal that drives a person to seek out information

Understanding information needs is central to designing effective information access systems because users often express them in incomplete or imprecise ways

Query is not synonymous with information need

The same information need might give rise to different manifestations with different systems

User’s Information Need – Example

Information Need

User wants to track their diet more carefully to lose weight and is trying to estimate their daily calorie intake. They just ate a banana and want to know how many calories it added to their total.

banana calories

Web Search Engine (typed)

Concise, keyword-based query to quickly look up nutritional info

Hey Ciri, how many calories are in a banana?"

Voice Assistant (spoken)

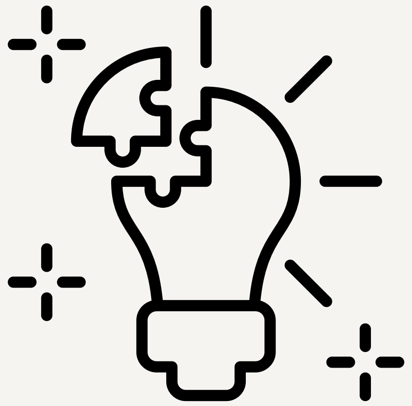
Conversational and direct, expecting a spoken response

Navigates to Add Food → Fruit → Banana, expects calorie count in

Nutrition Tracking App

Structured input, integrated with goal tracking in dedicated application

Relevance



Relevance refers to how well a retrieved document meets the user's information need

Relevance is not an absolute property of a document but rather a subjective and dynamic judgment that can vary across users and contexts

Topical relevance—whether a document pertains to the subject at hand—differs from user relevance, which is subjective and shaped by the individual's abstract information need

Relevance – Example

But I still haven't found what I'm looking for … U2

Initial Information Need

A user is planning their first trip to Nepal and types "best places to visit in Nepal" into a search engine

Relevance is Subjective

User A is an adventure seeker looking for trekking experiences. They find an article about the Annapurna Circuit and Everest Base Camp highly relevant.

User B is more interested in culture and spirituality, and prefers content about Lumbini and Pashupatinath Temple.

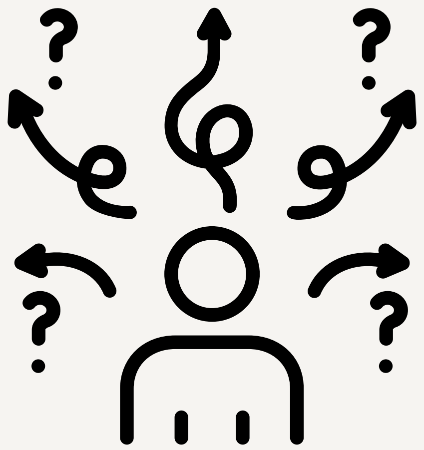
Same query, different users, different perceptions of relevance

Relevance is Dynamic

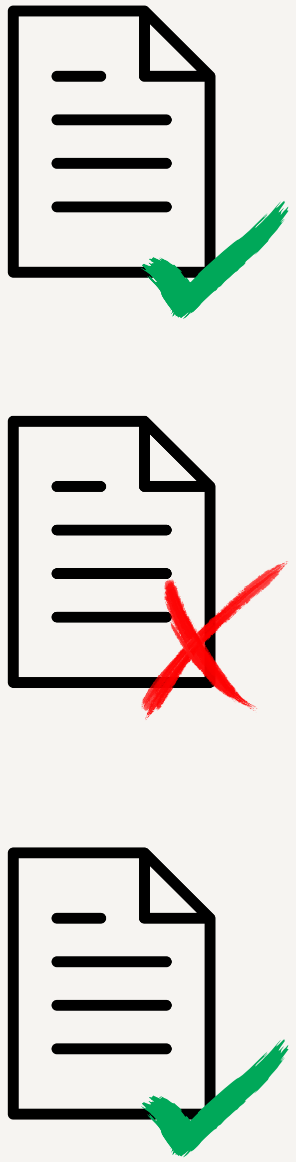
1. Searching for general sightseeing → *"Top 10 places to visit in Nepal"* blog post highly relevant
2. Choosing Pokhara as a destination, new query: *''things to do in Pokhara.")* → the general article less helpful
3. New query *"best gear shops in Pokhara for trekking"* → only content with very specific recommendations relevant

Same user, evolving needs, shifting perception of what's relevant

Relevance Judgments



Information
Need



Relevance judgments are human-provided relevance annotations on query-document pairs

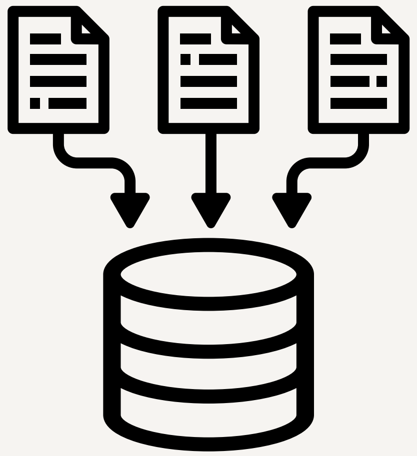
Relevance judgments are used both to train ranking models in supervised settings and to evaluate the effectiveness of those models

Relevance judgments reflect a particular individual’s opinion, making them inherently subjective

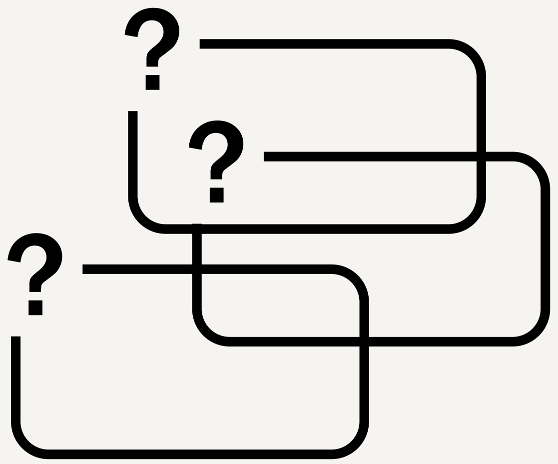
Assessor agreement on relevance judgments is typically low, with a commonly cited overlap of just 60% [2]

[2] Voorhees, E. M. (1998). *Variations in relevance judgments and the measurement of retrieval effectiveness.* In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’98, pages 315–323

Retrieval Evaluation



Document
Corpus



Queries

Relevance
Judgments

The Text Retrieval Conference (TREC) series organized by the National Institute of Standards and Technology (NIST), facilitates large-scale, community-wide evaluations of IR methods, enabling collaboration between academia, industry, and government

System evaluation at TREC follows the Cranfield paradigm:

- system-oriented methodology
- search as an optimization problem
- quantitative ranking metrics based on relevance judgments

MS MARCO

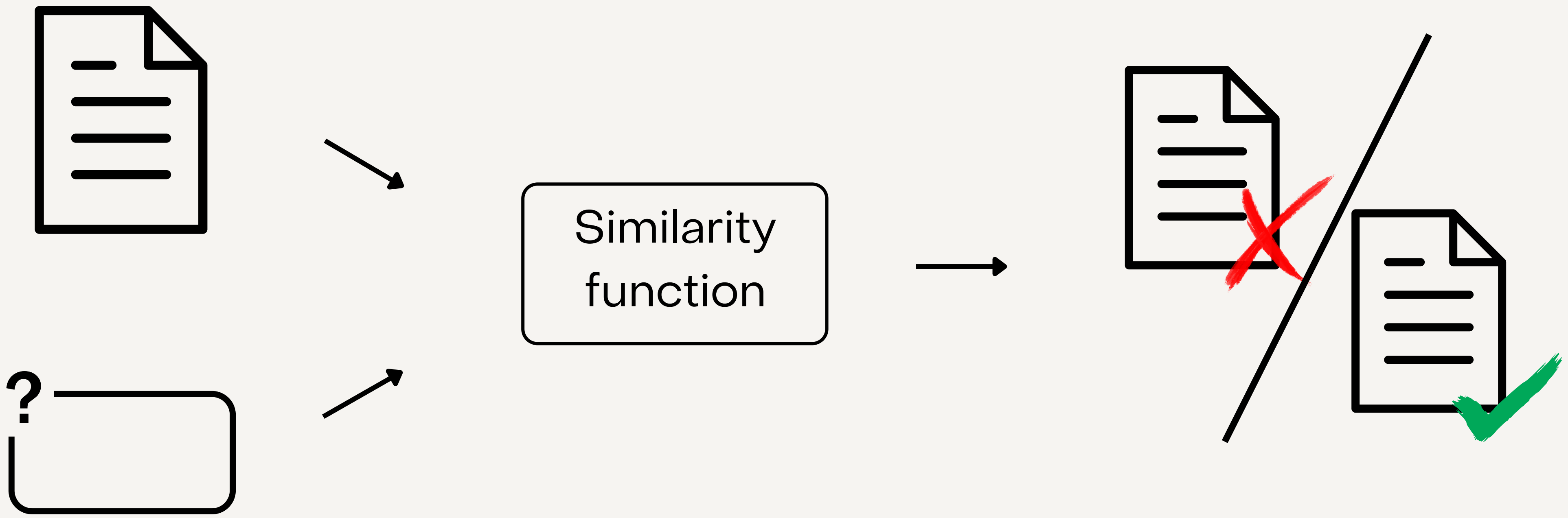
MAchine Reading COmprehension dataset comprising questions sampled from Bing's search query logs, each with multiple candidate passages retrieved from web documents

MS MARCO contains: 1M queries, 3,6M documents, 8,8M passages

Each question in MS MARCO is accompanied by an average of 10 passages, extracted from relevant web documents using a state-of-the-art Bing retrieval system

Query	*Query type*
cortical functions of the brain	description
what animal is a possum	entity
what is the discriminant of an equation	description
what temp does meat need to be cooked to	numeric
who sings on the song moves like jagger original	person
where mangroves are generally found	location

Retrieval Model



- The retrieval model defines how a relevance score between a document and a query is computed using their respective representations
- It estimates the utility of the document to an information need using a scoring function
- Based on the relevance scores, the system retrieves items that match the need

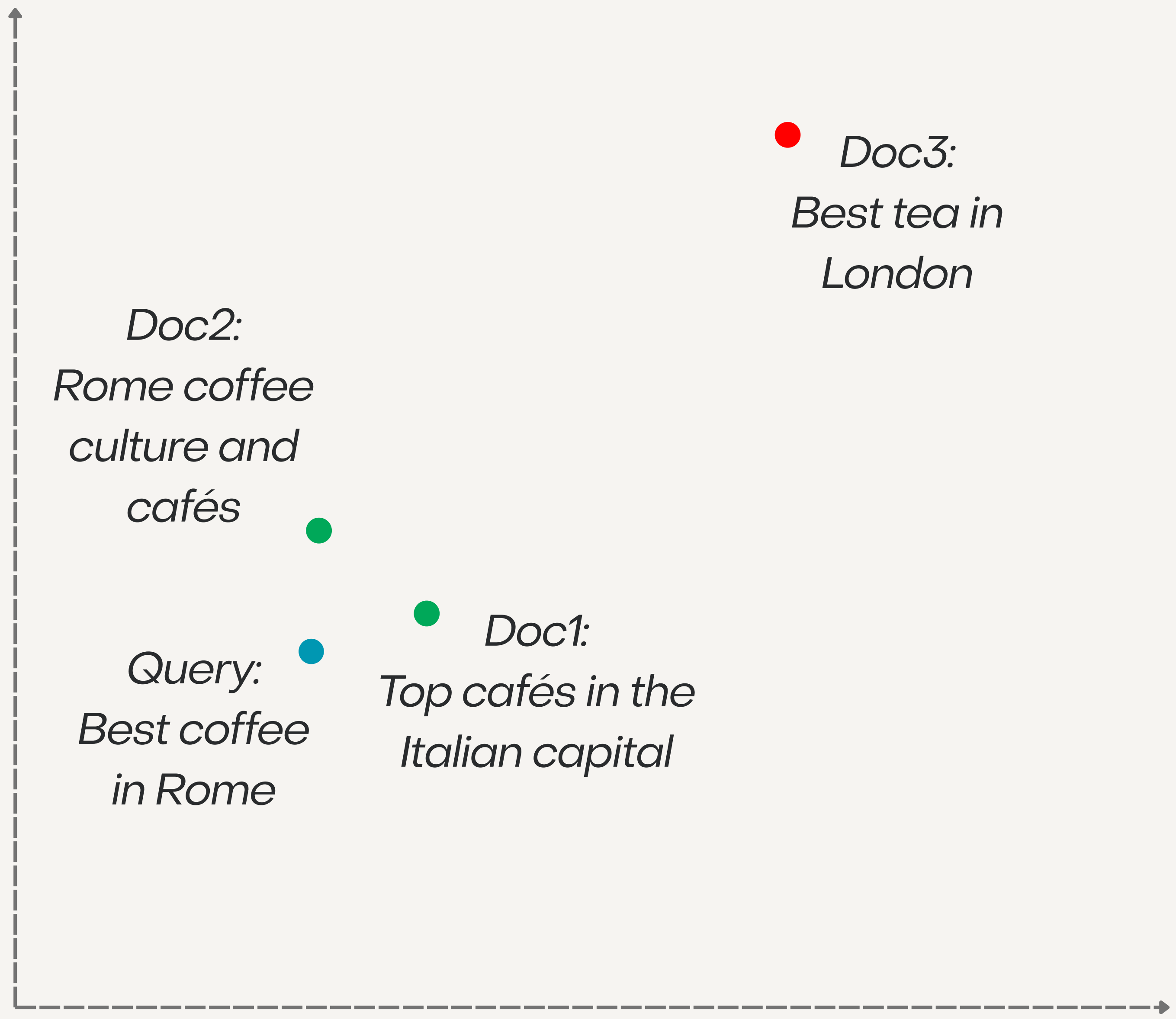
Dense Retrieval

Dense Retrieval

Dense retrieval aims to match texts in a continuous representation space learned via deep neural networks

Dense retrieval calculates the relevance score using similarities in a learned embedding space (based on cosine similarity or dot product between vectors)

Differently from sparse retrieval, it retrieves relevant results even when there is little or no lexical overlap between the query and the document



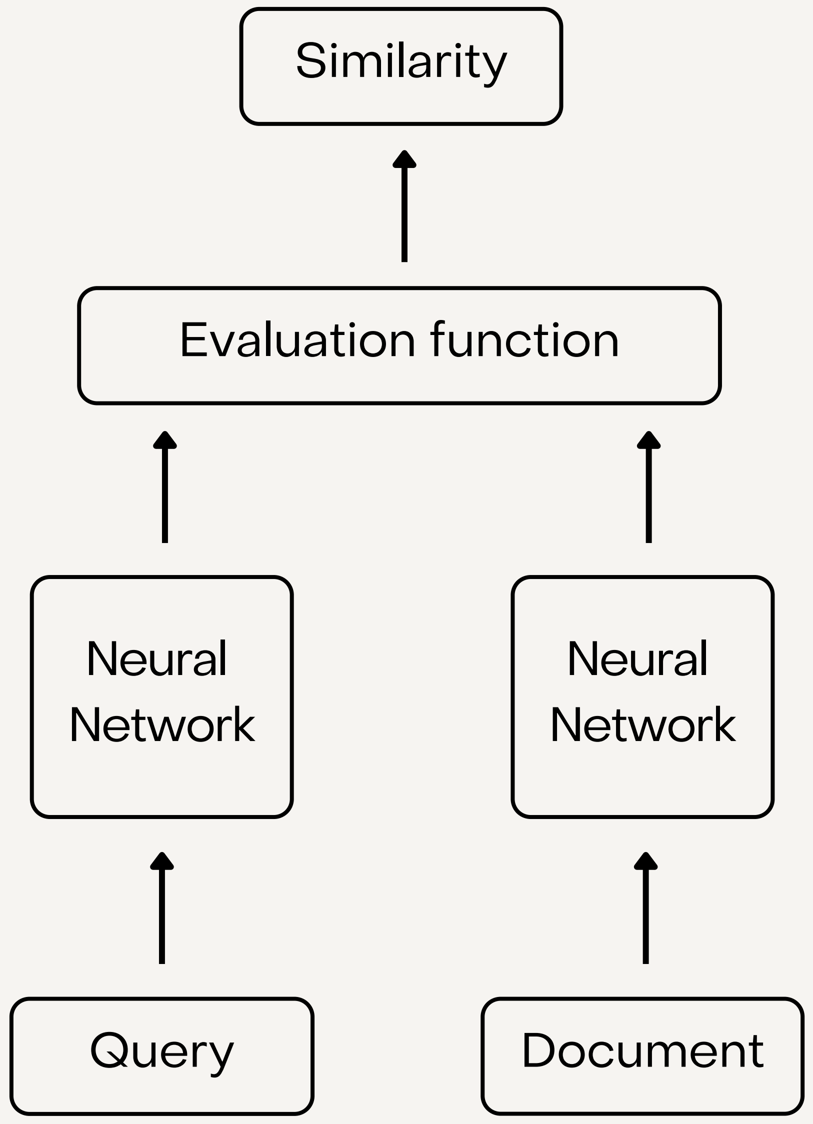
Representation-Focused Approaches

Main assumption: relevance depends on compositional meaning of the input texts

Both the query and the document are represented using single embedding vectors computed independently of each other

Relevance is estimated as a single similarity score between two high-level representations of input texts

Document representations can be precomputed offline making this method fast and scalable



[4] Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W.B., & Cheng, X. (2019). *A Deep Look into Neural Ranking Models for Information Retrieval.* Inf. Process. Manag., 57, 102067.

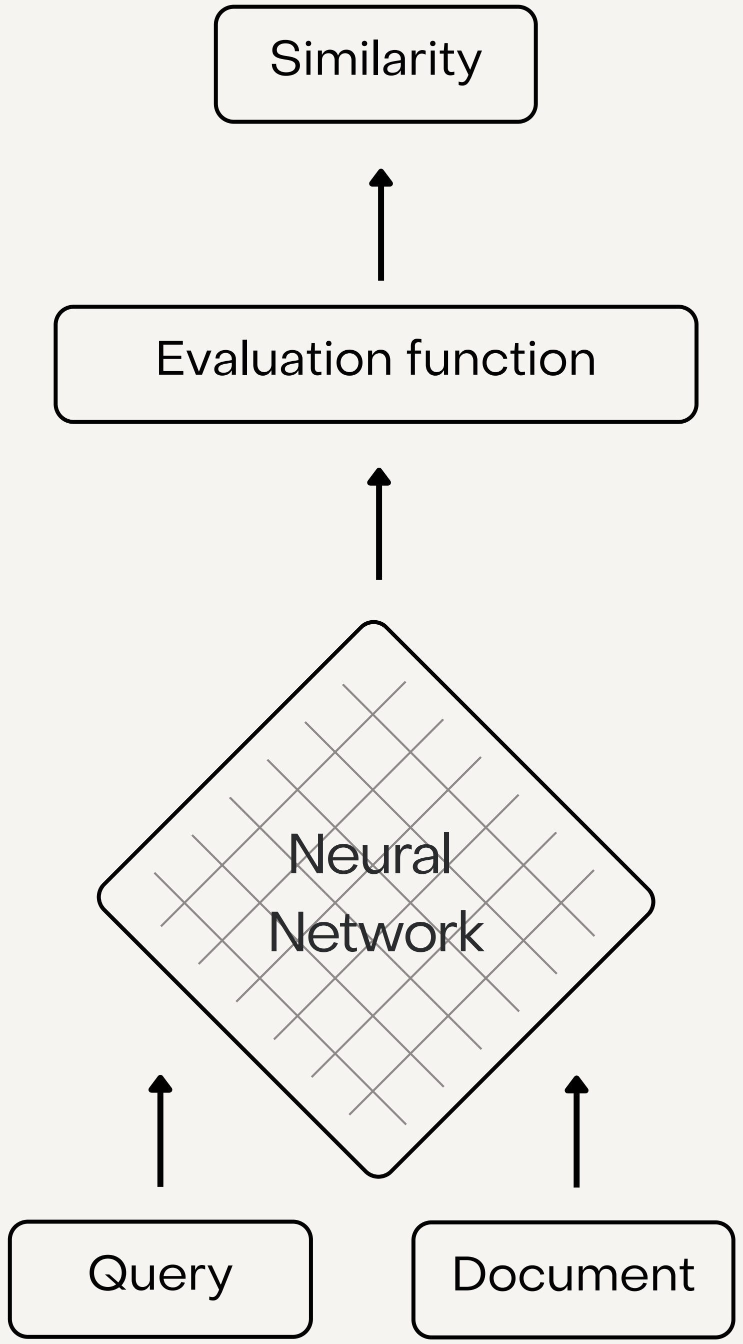
Interaction-Focused Approaches

Main assumption: relevance is in essense about the relation between the input texts

Word and n-gram relationships across a query and a document are modelled using deep neural networks

Relevance is modelled based on the query-document representation with a complex evaluation function (e.g., deep neural network)

Interaction function cannot be precomputed until the query-document pair is available making this method slower and more compute-heavy



[4] Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W.B., & Cheng, X. (2019). *A Deep Look into Neural Ranking Models for Information Retrieval.* Inf. Process. Manag., 57, 102067.

Dense Text Representation

Distributional hypothesis

Words that are occur in similar context tend to be semantically similar

	Dense Representations
Vocabulary	Implicit — learned embeddings encode semantic meaning
Text representation	Dense, continuous, low-dimensional vectors
Similarity	Vector similarity (e.g., dot product or cosine) in embedding space

Embeddings

Static (Global) Embeddings

She sat by the river bank.
- *"bank" → vector A*

He works at a bank downtown.
- *"bank" → vector A*

Words with multiple senses are mapped into an average or most common-sense representation based on the training data used to compute the vector

Contextualized (Local) Embeddings

She sat by the river bank.
- *"bank" → vector A*

He works at a bank downtown.
- *"bank" → vector B*

The representation of each token is conditioned on the context in which it is considered

Embeddings

Static (Global) Embeddings

She sat by the river bank.

- *"bank" → vector A*

He works at a bank downtown.

- *"bank" → vector A*

Words with multiple senses are mapped into an average or most common-sense representation based on the training data used to compute the vector

Contextualized (Local) Embeddings

She sat by the river bank.

- *"bank" → vector A*

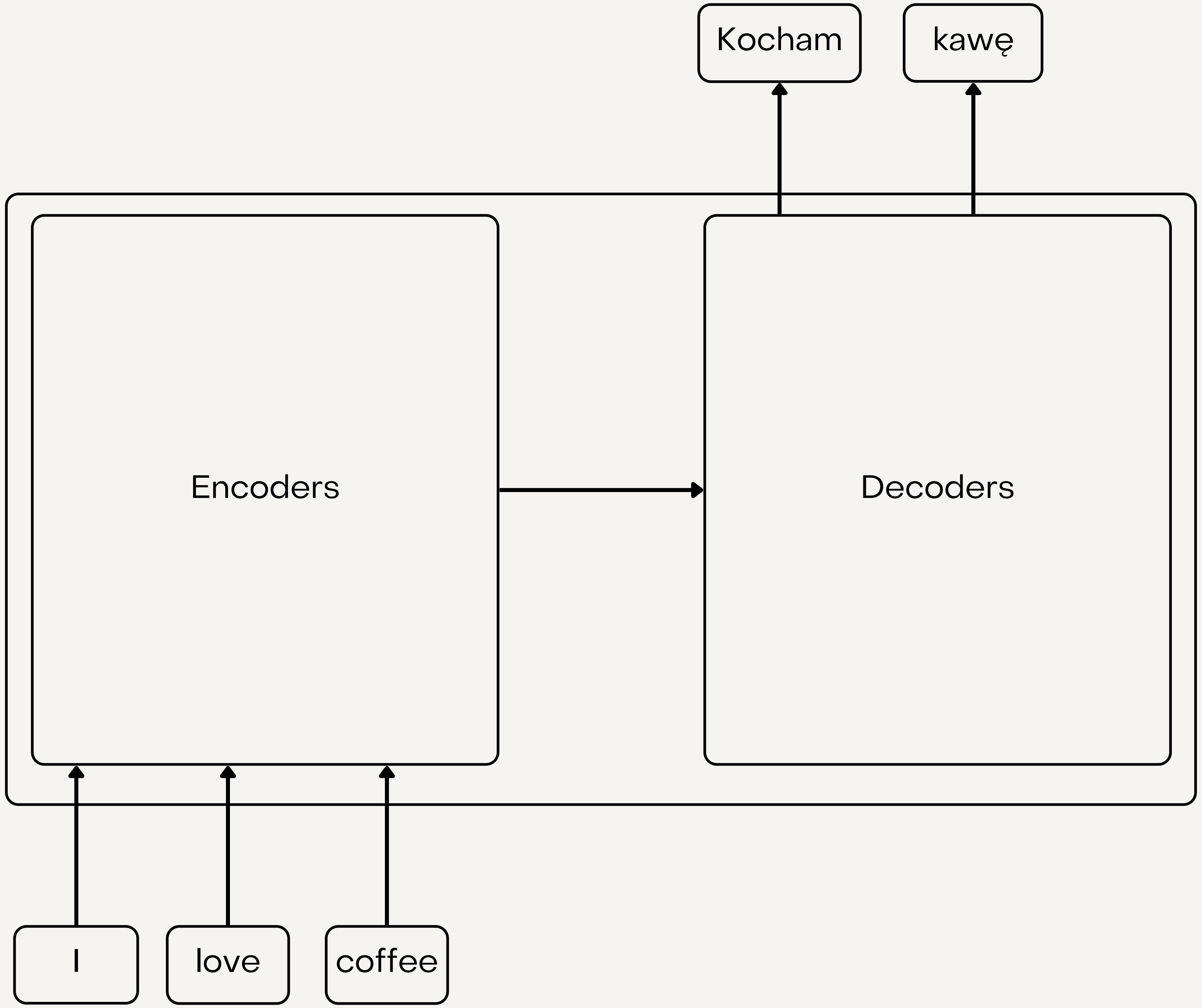
He works at a bank downtown.

- *"bank" → vector B*

The representation of each token is conditioned on the context in which it is considered

BERT Model

Transformer

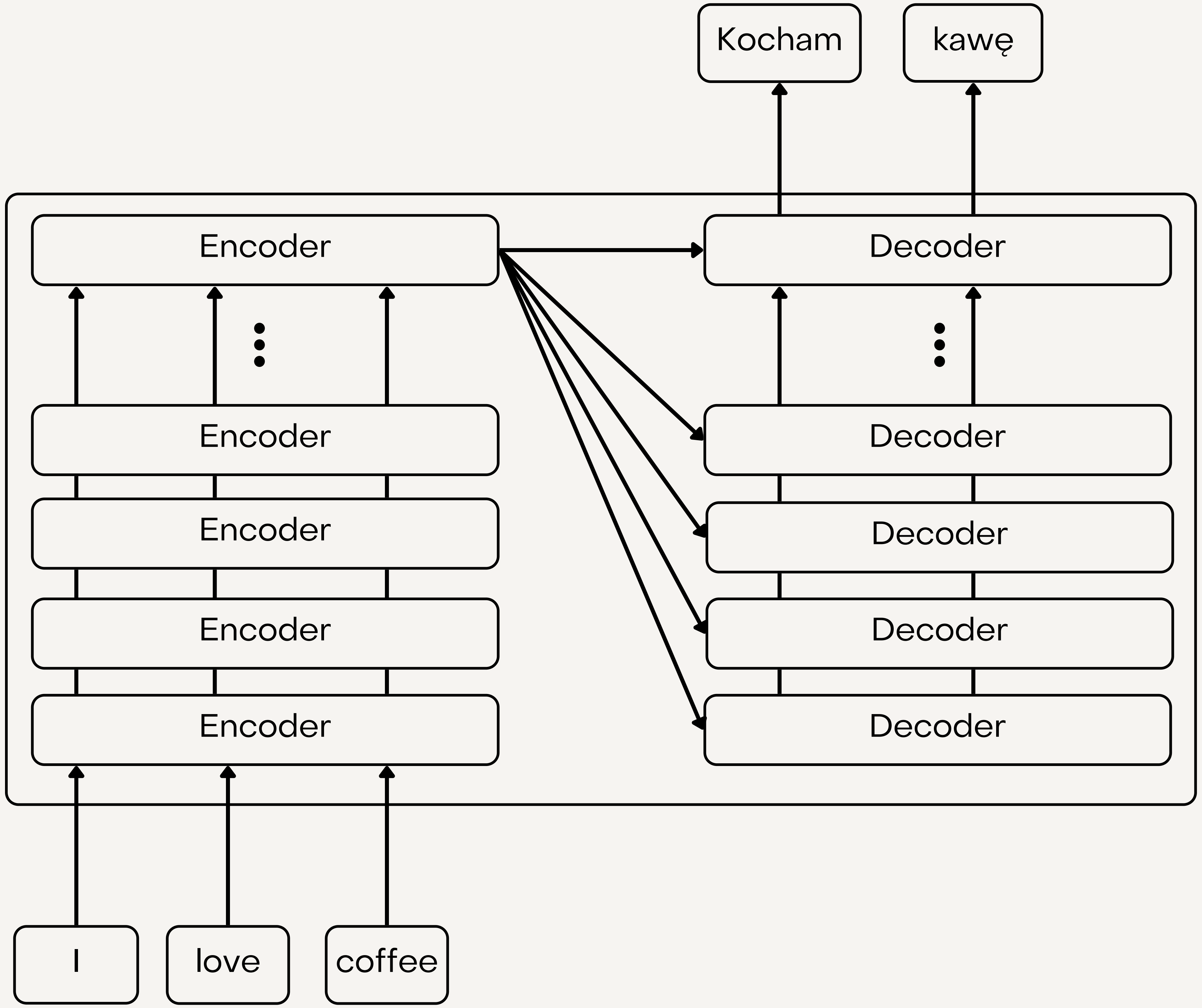


Neural network designed to explicitly take into account the context of arbitrary long sequences of text

An example of sequence-to-sequence models that transforms an input vectors to some output vectors of the same length

[5] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). *Attention is All you Need.* Neural Information Processing Systems.

Transformer

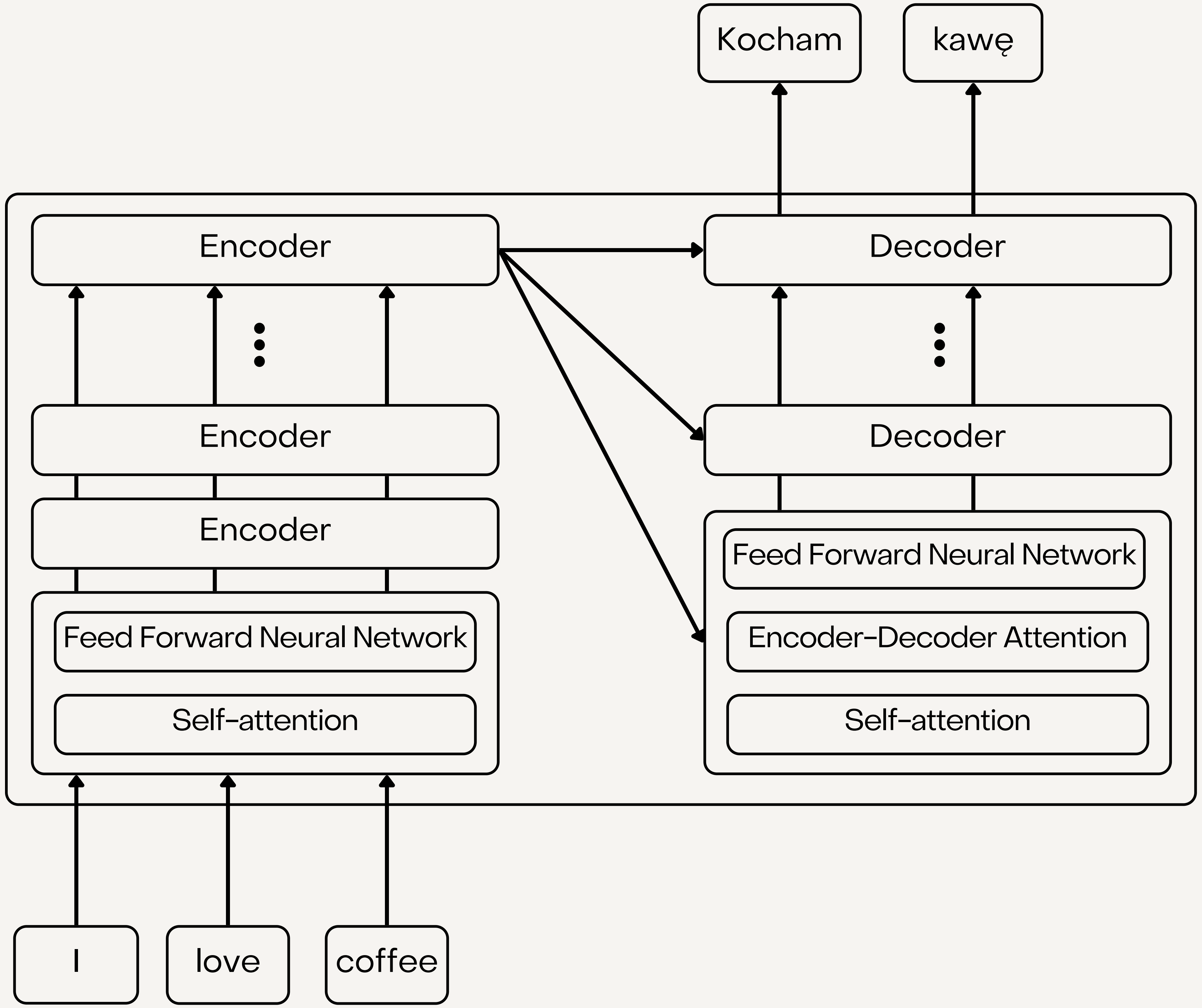


Neural network designed to explicitly take into account the context of arbitrary long sequences of text

An example of sequence-to-sequence models that transforms an input vectors to some output vectors of the same length

[5] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). *Attention is All you Need.* Neural Information Processing Systems.

Transformer



Neural network designed to explicitly take into account the context of arbitrary long sequences of text

An example of sequence-to-sequence models that transforms an input vectors to some output vectors of the same length

Attention allows to directly extract and use information from arbitrarily long contexts

[5] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). *Attention is All you Need.* Neural Information Processing Systems.

Transformer – Self-attention

Example

Morning coffee is the best part of the day. I always have it in my favourite mug.

Attention Links:

- *"it" → "coffee": high attention weight*
- *"it" → "morning": lower weight*
- *"it" → "mug" or "day": low or negligible weight*

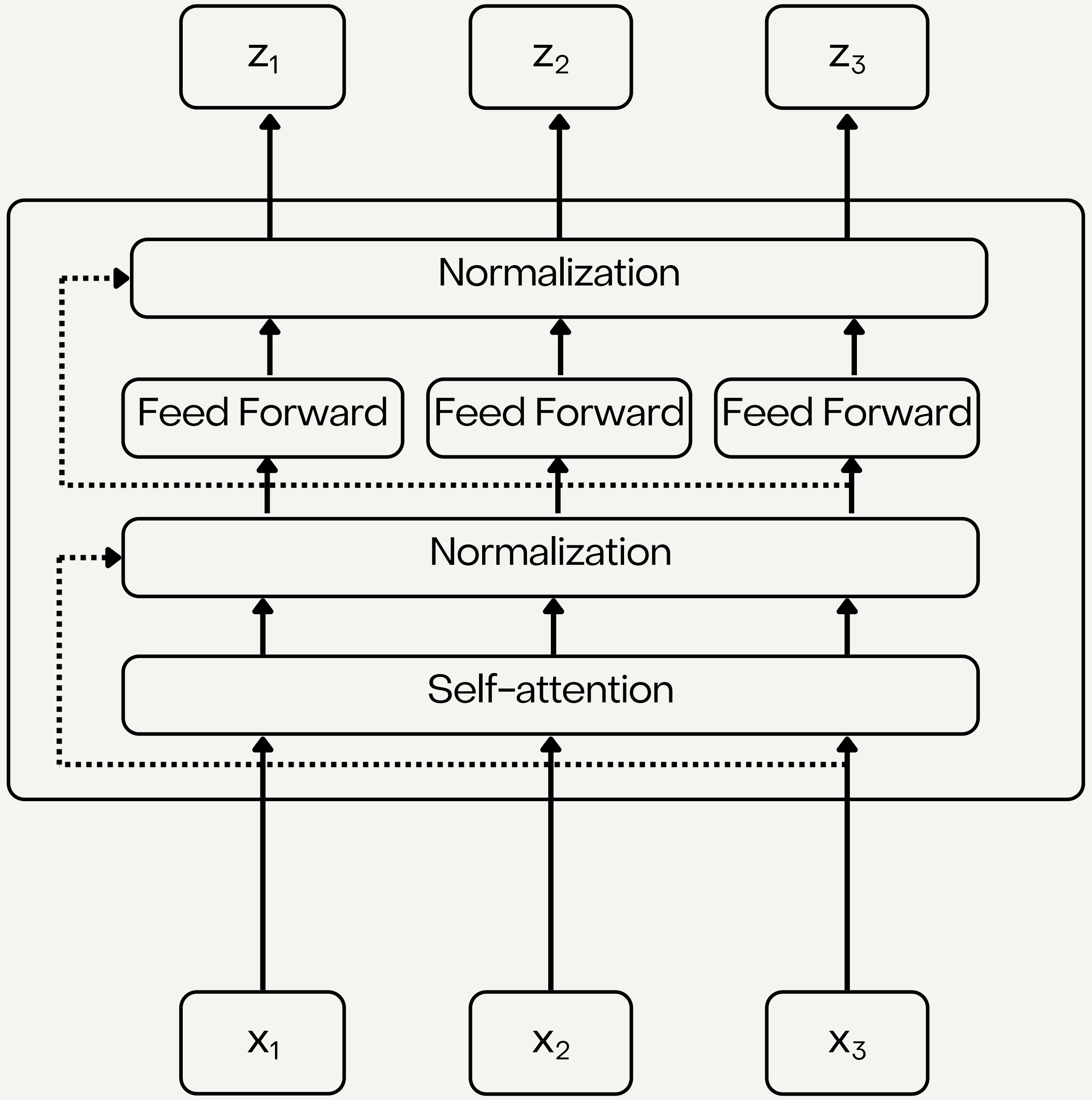
A self-attention layer allows the network to take into account the relationships among different elements in the same input

As the model processes each word, self attention allows it to look at other positions in the input sequence for clues that can help lead to a better encoding for this word

Self-attention differs in the encoder and decoder blocks

[5] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). *Attention is All you Need.* Neural Information Processing Systems.

Transformer – Transformer Block



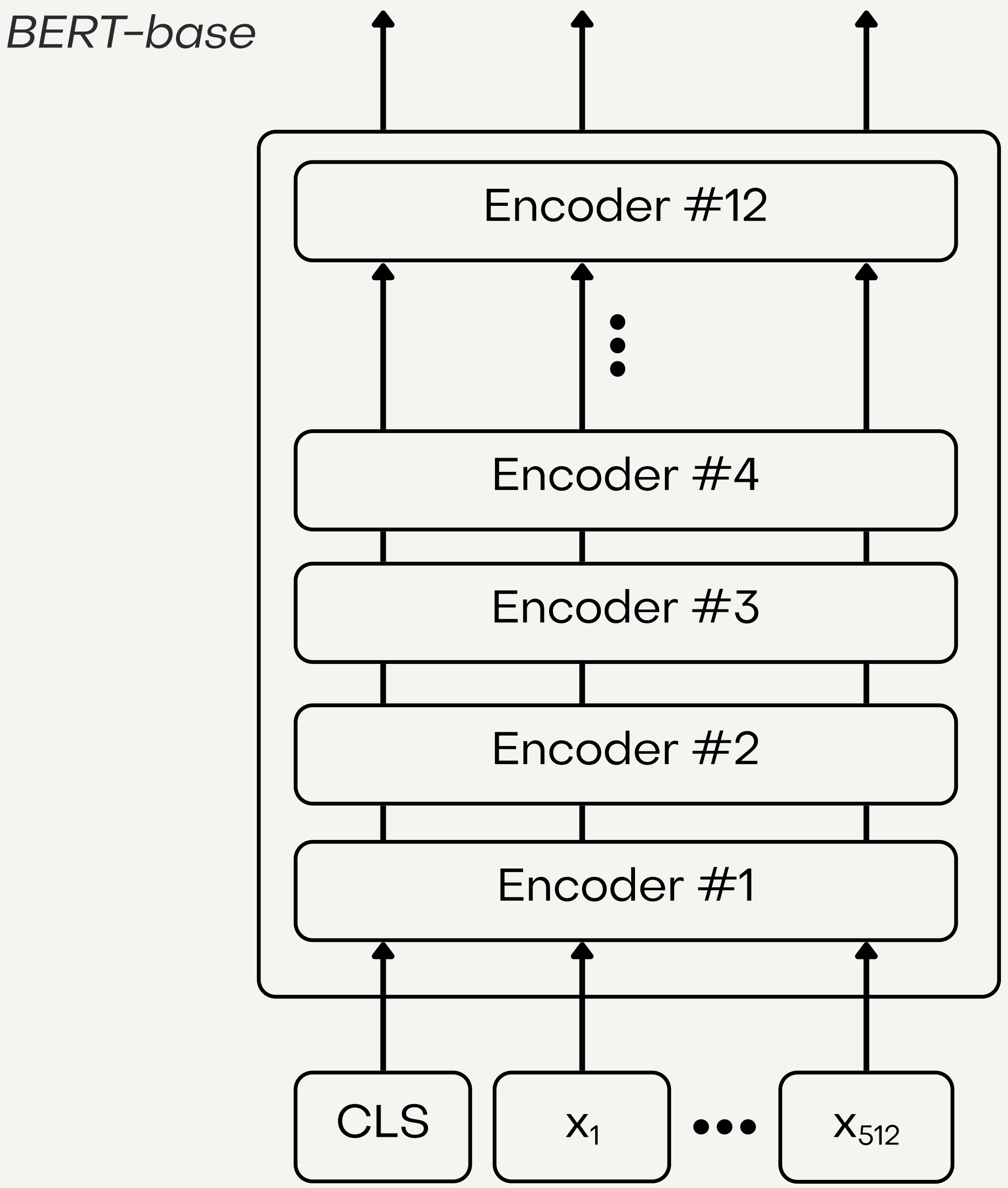
In deep networks, residual connections are connections that pass information from a lower layer to a higher layer without going through the intermediate layer

Residual connections help avoid vanishing gradients, making it easier for the model to learn identity mappings and propagate information through deep networks

Layer normalization is used to improve training performance in deep neural networks by keeping the values of a hidden layer in a range that facilitates gradient-based training

[5] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). *Attention is All you Need.* Neural Information Processing Systems.

Bidirectional Encoder Representations from Transformers



BERT uses a deep stack of Transformer encoders to read text in both directions, allowing it to understand context more effectively than previous models

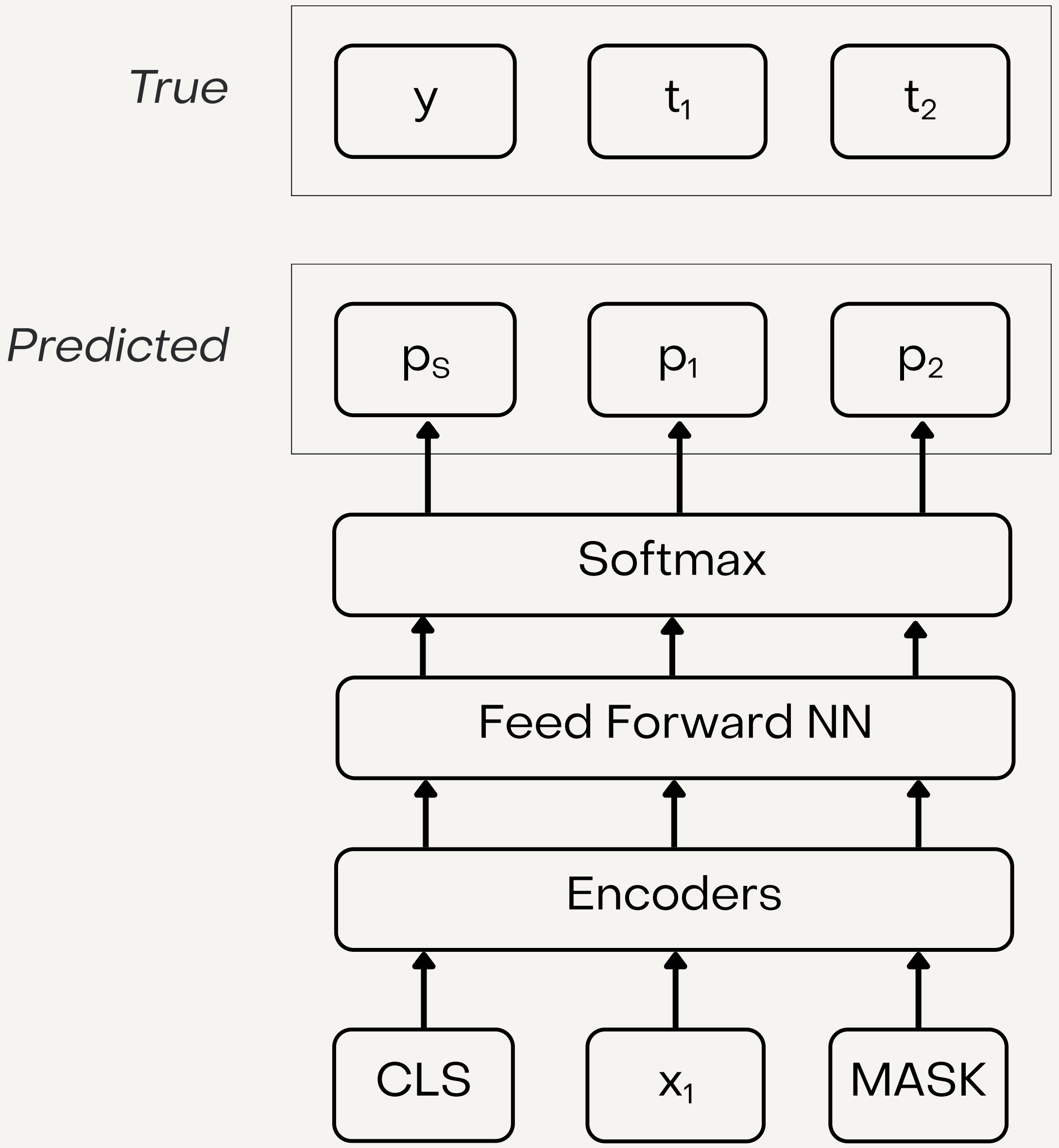
BERT input text is tokenised using the WordPiece tokeniser, where the uncommon/rare words, e.g.,*goldfish*, is divided in sub-words, e.g., *gold## and ##fish*

BERT accepts as input at most 512 tokens, and produces an output embedding for each input token

The most commonly adopted BERT version is $\text{BERT}_{\text{Base}}$, which stacks 12 transformer layers, and whose output representation space has 768 dimensions ($\text{BERT}_{\text{Large}}$ has 24 layers and 1024 hidden units)

[6] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. North American Chapter of the Association for Computational Linguistics.

BERT - Pre-training



Masked Language Modelling

Input: "I love drinking [MASK] in the morning."

The model learns to predict "coffee" by considering the surrounding words.

BERT randomly masks some tokens in the input and trains the model to predict those masked tokens based on their context

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} t_i \log p_i$$

Next Sentence Prediction

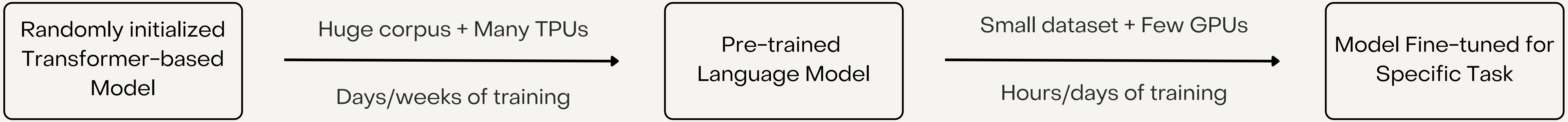
Sentence A: "I want to grab a coffee."

Sentence B: "I'll order a cappuccino."
→ Label: IsNext
Sentence B: "The sky was full of stars."
→ Label: NotNext

BERT is also trained to predict whether one sentence logically follows another

$$\mathcal{L}_{\text{NSP}} = -\left[y \log(p_S) + (1-y)\log(1-p_S)\right]$$

Transfer Learning



The training of the model is a two-step process called transfer learning

The model is pretrained on a certain task that allows it to grasp patterns in a language using unlabelled data

Then, then model is trained in a supervised manner on a labelled data to perform a specific task

ColBERT

Omar Khattab and Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.* In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)

Motivation

Approach	*Effectiveness*	*Efficiency*	*Overall*
Representation-Based Dense Retrieval	✓ Captures semantic meaning ✗ May miss fine-grained interaction signals	✓ Offline documents encoding ✓ Efficient retrieval via ANN search	Balances semantic matching and scalability
Full Interaction Dense Retrieval	✓ Very effective ✓ Rich query-document interactions	✗ Slower at inference ✗ More memory-intensive	Highest relevance but at a higher computational cost

ColBERT improves:

- Efficiency over traditional interaction-based retrieval by avoiding running cross-encoder over every document
- Effectiveness over representation-based retrieval by retaining token-level granularity in matching

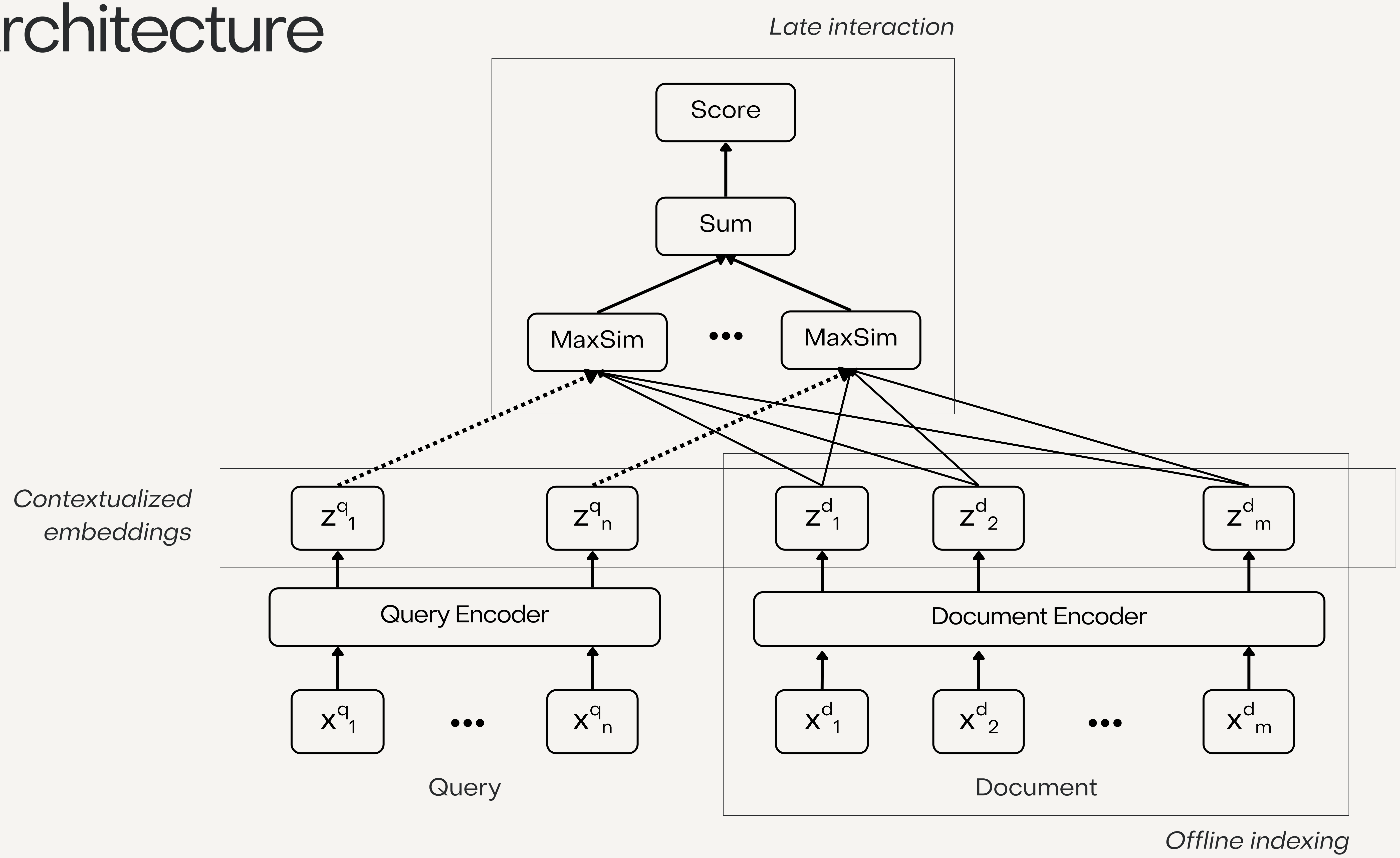
Contextualized Late Interaction over BERT (ColBERT)

Late interaction mechanism – independently encoding the query and the document and modelling their fine-grained similarity

Offline document encoding – precomputing document representations offline to speed up query processing by delaying and yet retaining fine-grained interaction

Pruning-friendly interaction mechanism – leveraging embedding index for end-to-end retrieval directly from a large document collection

Architecture



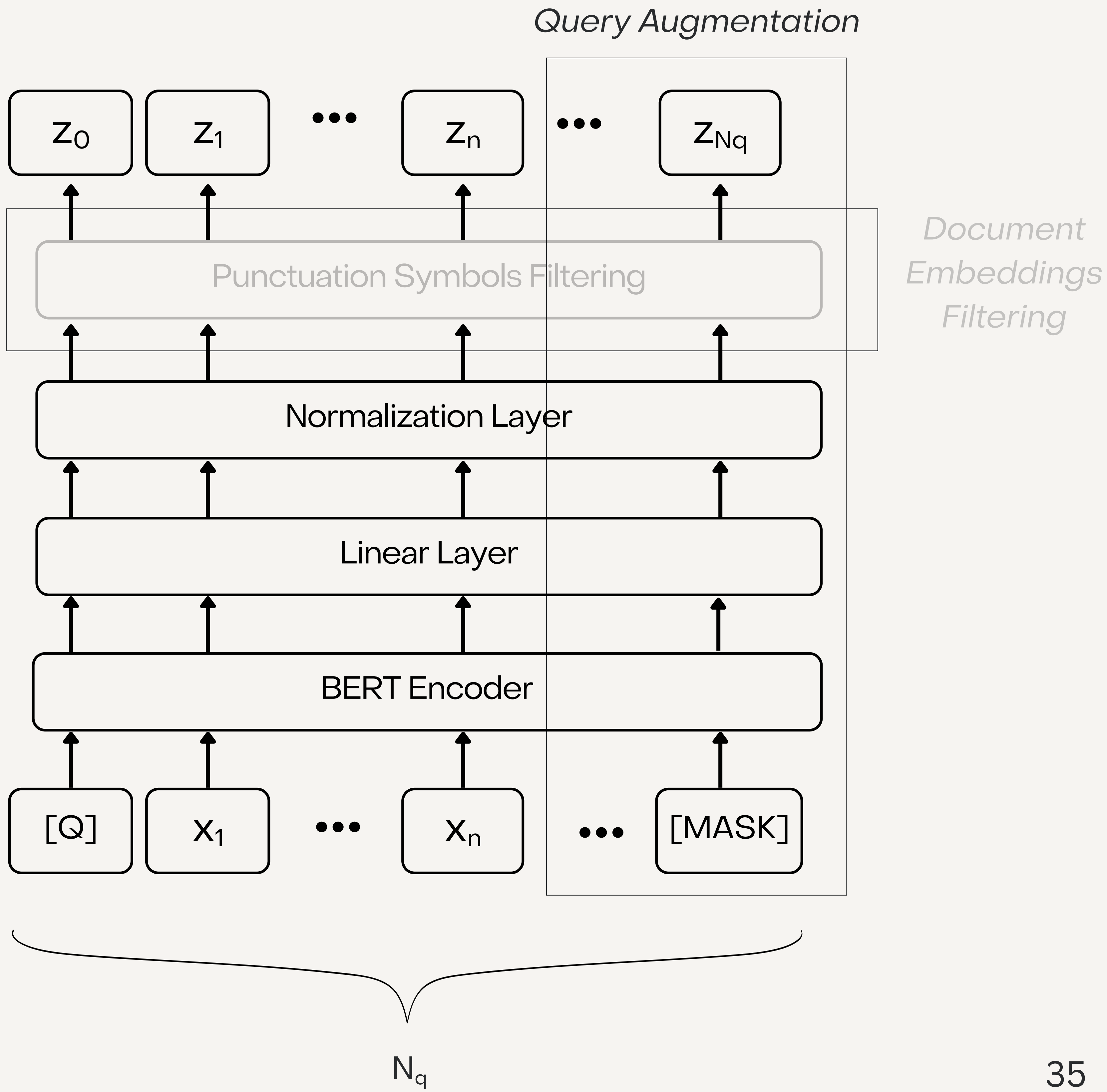
Query & Document Encoders

A single BERT model encodes both queries and documents, with inputs distinguished by special tokens: [Q] for queries and [D] for documents

Query augmentation enables BERT to generate query-based embeddings at mask positions → query expansion + terms re-weighting

The high-dimensional embeddings from BERT are projected to a lower-dimensional space using a linear layer to reduce their size

Each embedding is normalized to ensure efficient similarity computations using dot products equivalent to cosine similarity



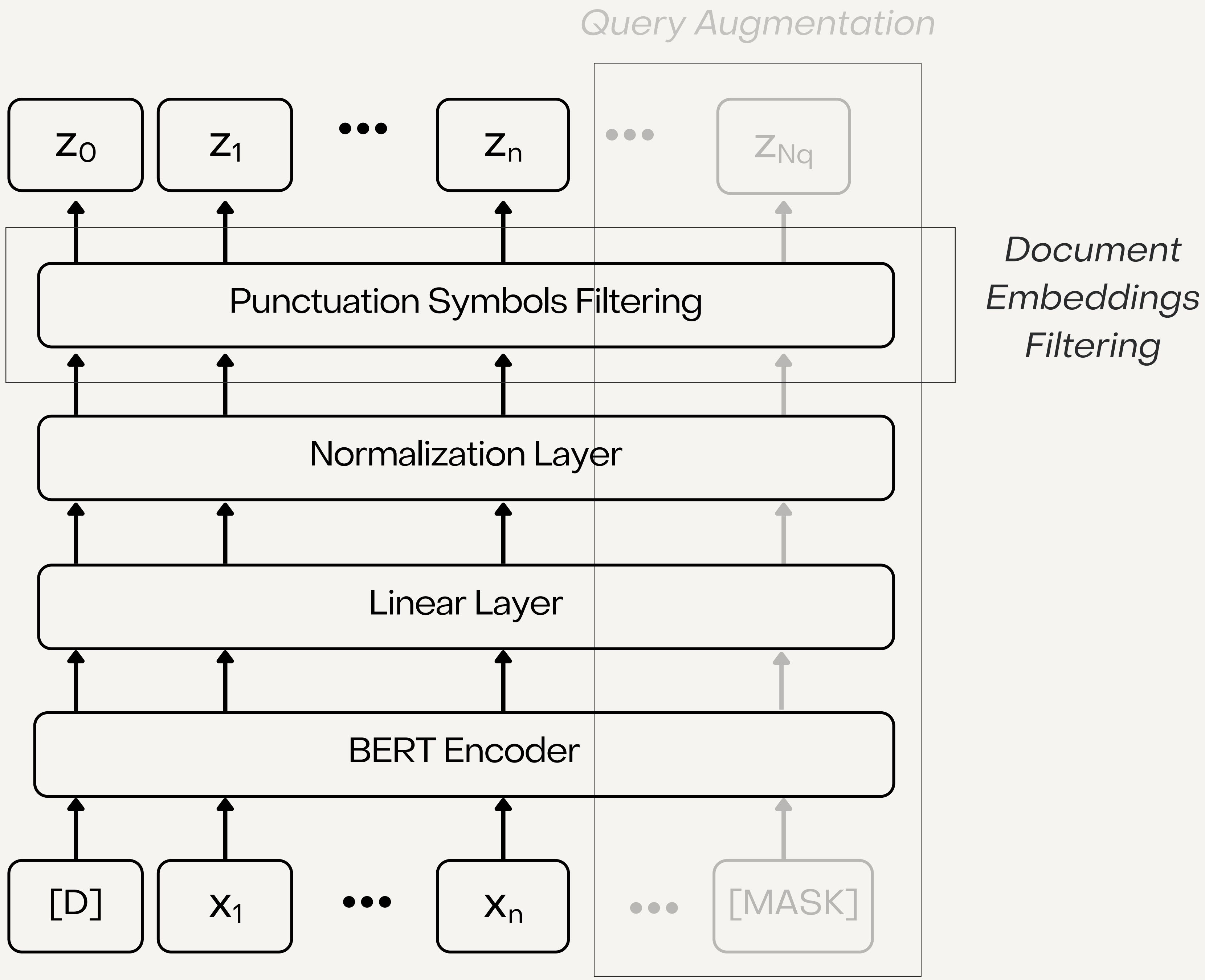
Query & Document Encoders

A single BERT model encodes both queries and documents, with inputs distinguished by special tokens: [Q] for queries and [D] for documents

Query augmentation enables BERT to generate query-based embeddings at mask positions → query expansion + terms re-weighting

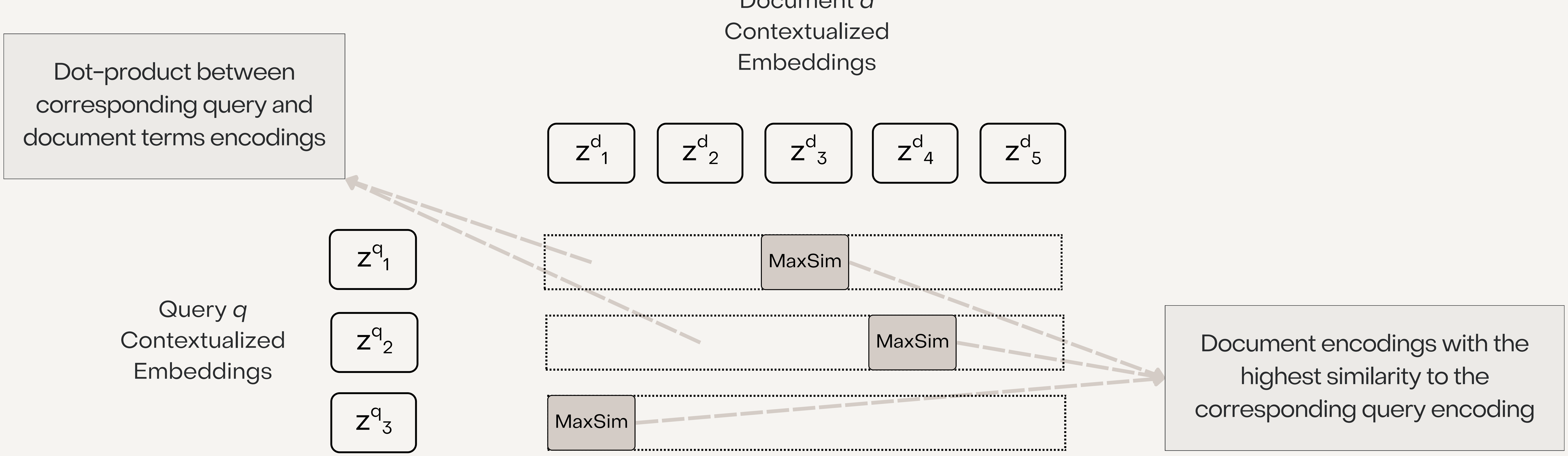
The high-dimensional embeddings from BERT are projected to a lower-dimensional space using a linear layer to reduce their size

Each embedding is normalized to ensure efficient similarity computations using dot products equivalent to cosine similarity



Late Interaction

Intuition: *We align each query token with the most contextually relevant passage token, quantify these matches, and combine the partial scores across the query.*



Relevance score of document to query: SUM (MaxSim , MaxSim , MaxSim)

More formally: $rel(q,d) = \sum_{i=1}^{|q|} \max_{j=1,\ldots,|d|} {z_i^q}^T z_j^d$

Training

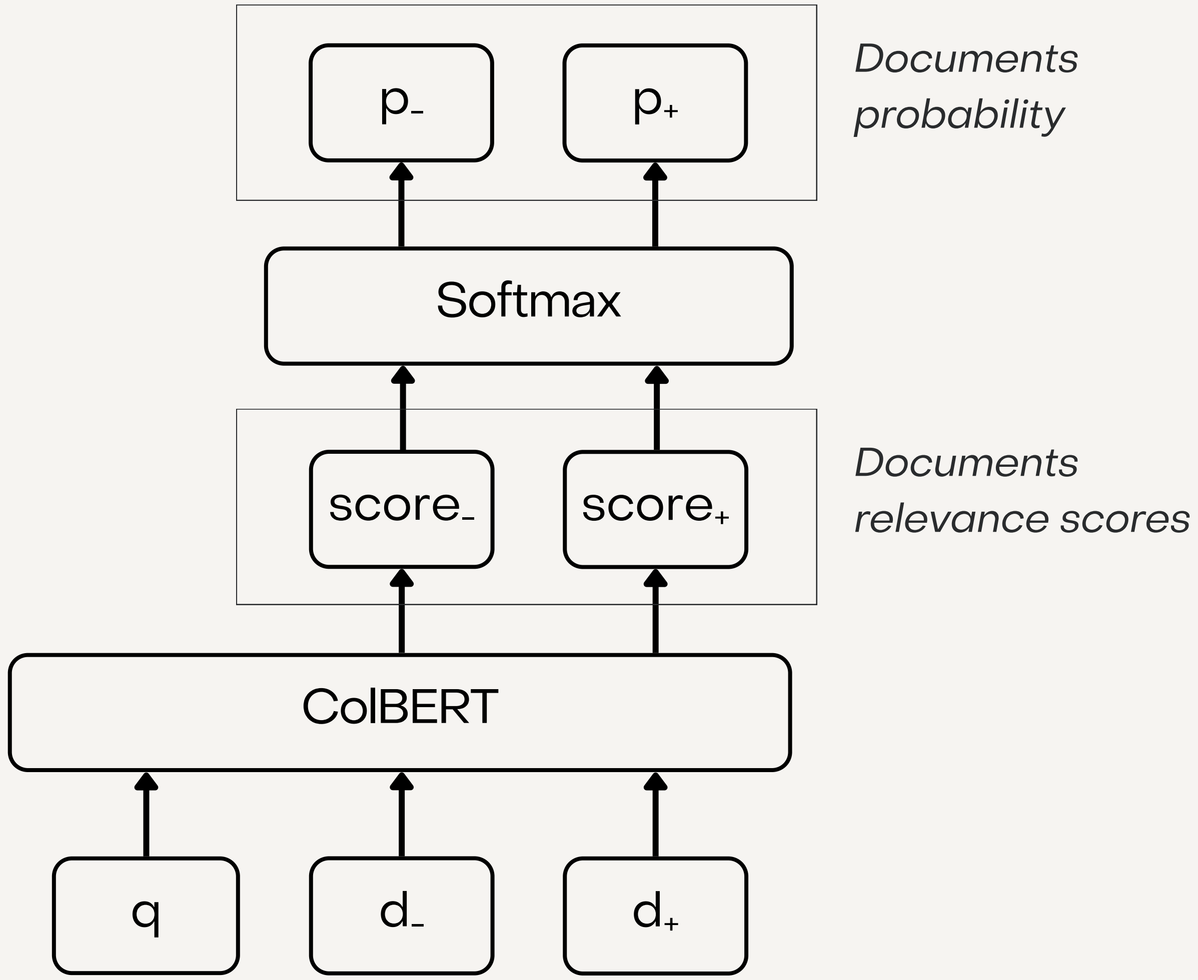
ColBERT is trained end-to-end using pairwise softmax cross-entropy loss

$$\mathcal{L}(q, d_-, d_+) = -\log p_+$$

BERT encoders are fine-tuned and the additional parameters are trained from scratch using the Adam optimizer

Note: the interaction mechanism has no trainable parameters

Given a triple <*query, positive_doc, negative_doc*>, ColBERT is optimized to assign higher scores to positive documents compared to negative documents



Offline Documents Encoding

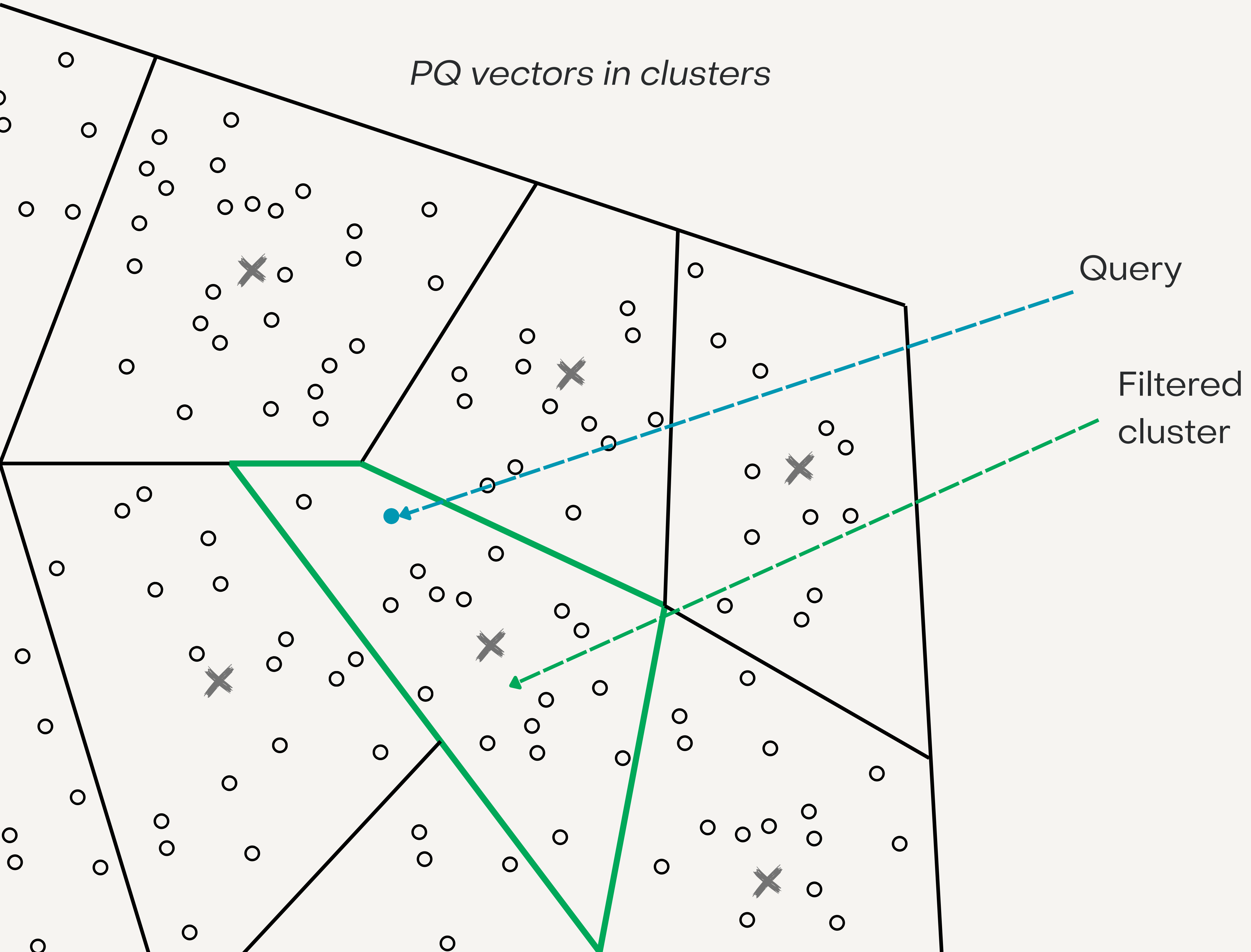
Multiple GPUs, are used for faster encoding of batches of documents in parallel

Indexing in batches with documents of comparable length

Padding documents to the maximum length of a document within the batch

Tokenization with is parallelized across the available CPU cores

Search in IVFPQ Index



Inverted File (IVF) index partitions the vector space into clusters, reducing the search scope

Product Quantization (PQ) is used to compress and quantize vector representations to speed up search

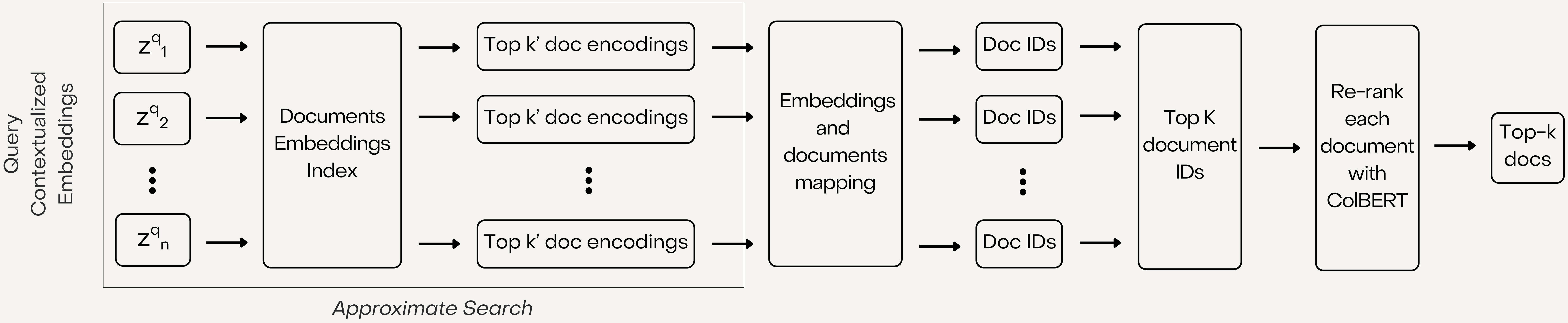
Approximate nearest neighbour search focuses only on the clusters closest to the query vector instead of evaluating the entire dataset

End-to-End Retrieval with ColBERT

Two-stage procedure to retrieve the top-k documents from the entire collection relying on ColBERT's scoring:

1) filtering potentially relevant documents from the entire collection using approximate search
2) re-ranking candidate documents by exhaustively scoring each document against the query using the late interaction mechanism

$$k' < k \leq K$$
$$K \leq n \times k'$$



ColBERT Results

FLOPs measures the total number of floating-point operations performed by a machine learning model.

MRR@k measures the average of the reciprocal ranks of the first relevant document, considering only the top k retrieved results for each query.

Recall@k measures the proportion of relevant documents retrieved in the top k results out of all relevant documents for each query.

Approach	*MRR@10*	*Latency (ms)*	*Recall@50*	*Recall@200*	*Recall@1000*
BM25	18.7	62	59.2	73.8	85.7
docTTTTTquery	27.7	87	75.6	86.9	94.7
ColBERT	36.0	458	82.9	92.3	96.8

End-to-end retrieval results on MS MARCO. Each model retrieves the top-1000 documents per query directly from the document collection.

ColBERT outperforms standard bag-of-words approaches in both MRR and Recall

Approach	*MRR@10*	*Latency (ms)*	*FLOPs/query*
$\text{BERT}_{\text{Base}}$	34.7	10,700	97T (13,900x)
$\text{BERT}_{\text{Large}}$	36.5	32,900	340T (48,600x)
ColBERT	34.9	61	7B (1x)

Re-ranking results on MS MARCO. Each neural model re-ranks the top-1000 results produced by BM25. Latency is reported for re-ranking only.

While highly competitive in effectiveness, ColBERT is orders of magnitude cheaper than standard all-to-all interaction reranking with BERT models

Advantages & Limitations of ColBERT

Advantages

Retrieval performance

Matching explainability

Limitations

Storage requirements

Improving storage efficiency by introducing residual compression of document embeddings (ColBERTv2) [7]

Improving space efficiency with static pruning of embeddings associated with the terms with the lowest IDF [8]

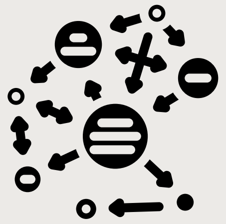
[7] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., and Zaharia, M. 2022. *ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction*. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3715–3734, NAACL '22

[8] Acquavia, A., Tonellotto, N. and Macdonald, C. (2023) *Static Pruning for Multi-Representation Dense Retrieval*. In: 23rd ACM Symposium on Document Engineering DocEng'23

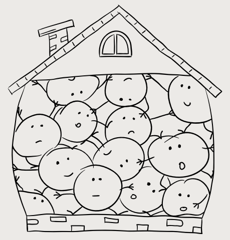
Conclusions

Takeaway Messages

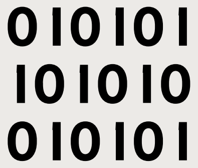
Dense retrieval leverages neural embeddings to capture semantics, enabling better recall—especially for complex or nuanced queries



Transformers power dense retrieval by capturing contextual meaning through self-attention and pretraining tasks



Representation-focused models encode queries and documents independently, while interaction-focused models enable fine-grained matching at the token level



ColBERT combines efficiency of late interaction with effectiveness of token-level semantic alignment