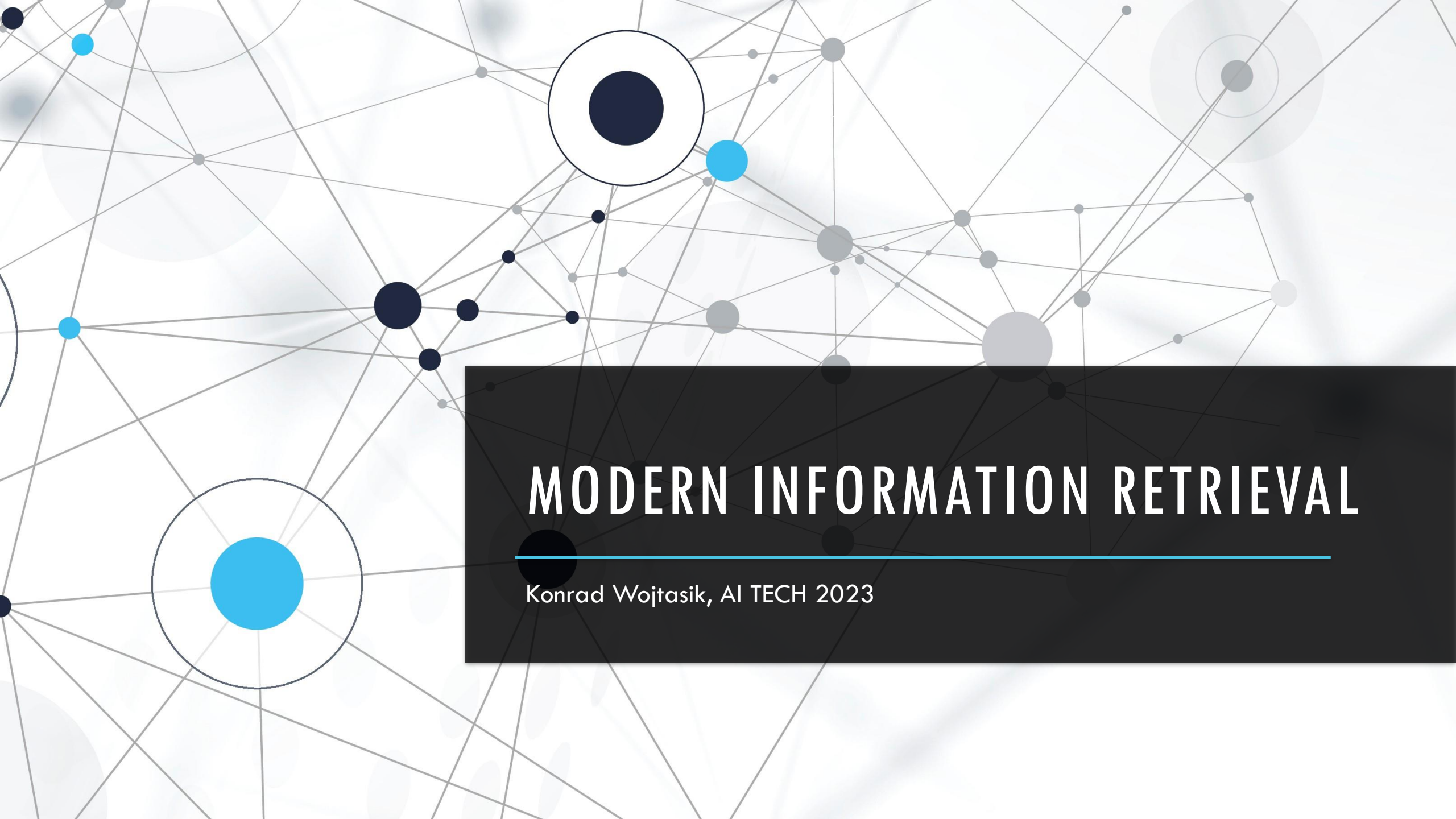


# MODERN INFORMATION RETRIEVAL

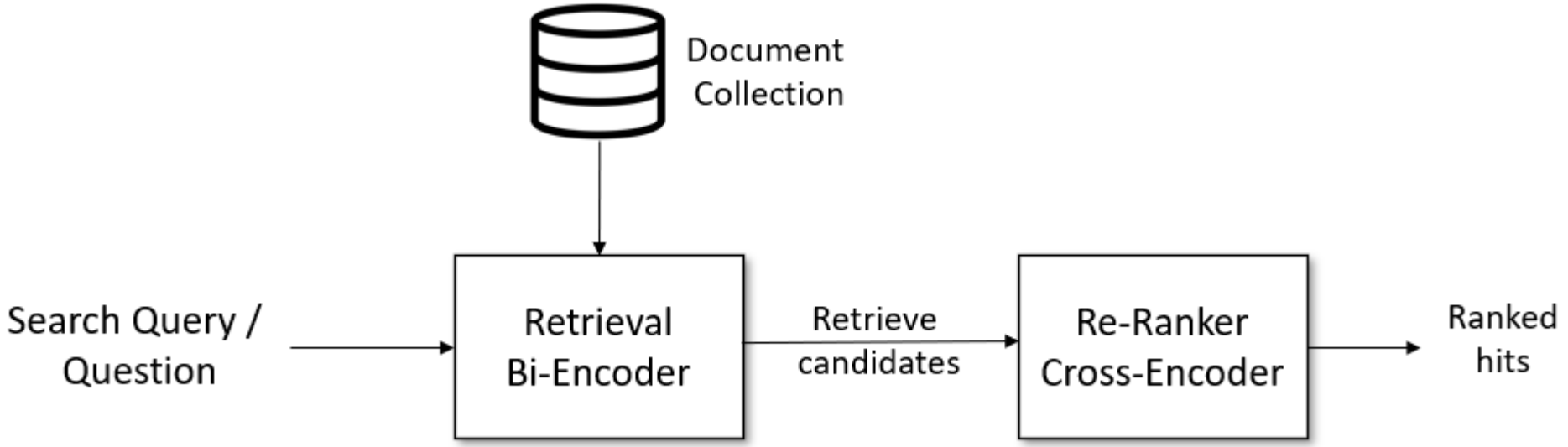
Konrad Wojtasik, AI TECH 2023

# GITHUB REPOSITORY

Github: https://github.com/kwojtasi/modern-ir-aitech

# WHAT IS INFORMATION RETRIEVAL

Information Retrieval (IR) is a process of obtaining relevant information form a collection of documents or data sources. User input is a query and the output of the system is the relevant documents, passages.




https://www.sbert.net/examples/applications/information-retrieval/README.html

# WHY INFORMATION RETRIEVAL

- It is crucial component of Question Answering systems, given user question, system searches for relevant information in data collection.
- Prevents QA model hallucinations providing relevant context.
- Keeps information up-to-date. Large Language Models may not have been trained with the most recent data and to avoid this expensive process IR comes in place.
- Product search, tourist offers search etc.

# INFORMATION RETRIEVAL

Steps in Information Retrieval:

1. Document Collection
2. Indexing
3. Query Processing
4. Retrieval
5. Reranking (optional)
6. Returning relevant documents

# AVAILABLE DATASETS

- MSMARCO – large dataset created by Microsoft from Bing search questions,
- BEIR benchmark – contains diverse range of IR datasets
- Any dataset that contains pairs Query-Passage, ex. SQuAD dataset, NLI datasets

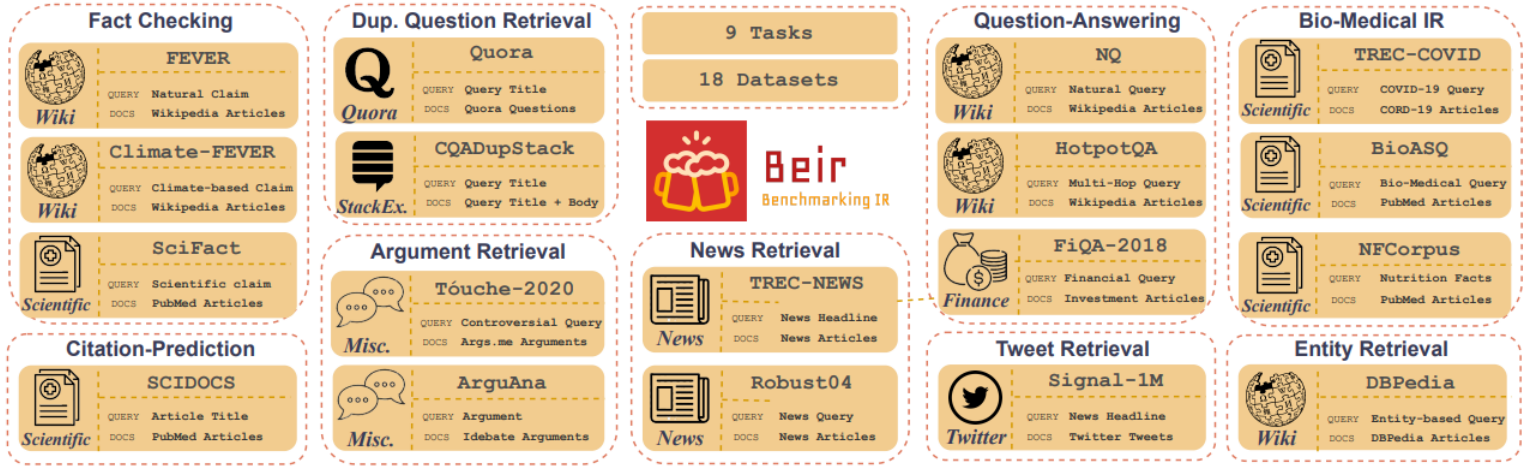



**Figure 1:** An overview of the diverse tasks and datasets in BEIR benchmark.

BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models

# AVAILABLE DATASETS

Number of queries and size of corpuses in BEIR and BEIR-PL benchmarks.

| Dataset | #Test queries | Corpus size | Avg. Q Len | Avg. D Len |
|---|---|---|---|---|
| MSMARCO | 43 | 8.8M | 5.33 | 49.63 |
| TREC-COVID | 50 | 171K | 9.44 | 137.05 |
| NFCorpus | 323 | 3.6K | 3.37 | 205.96 |
| NQ | 3 452 | 2.68M | 7.33 | 66.89 |
| HotpotQA | 7 405 | 5.2M | 15.64 | 38.67 |
| FiQA | 648 | 57K | 9.76 | 113.96 |
| ArguAna | 1 406 | 9K | 168.01 | 142.48 |
| Touche-2020 | 49 | 382K | 7.12 | 125.48 |
| CQADupstack | 13 145 | 547K | 7.86 | 110.76 |
| Quora | 10 000 | 523K | 8.13 | 9.85 |
| DBPedia | 400 | 4.63M | 4.82 | 41.61 |
| SciDocs | 1 000 | 25K | 9.70 | 150.15 |
| SciFact | 300 | 5K | 11.74 | 187.66 |

BEIR-PL: Zero Shot Information Retrieval Benchmark for the Polish Language

# AVAILABLE DATASETS

There are also multilingual resources available.

- mMARCO - Automatic translation of original MS MARCO dataset to 13 languages
- Mr. TyDi – Dataset constructed from TyDI QA dataset

| | | Train | | Dev | | Test | | Corpus Size |
|---|---|---|---|---|---|---|---|---|
| | | # Q | # J | # Q | # J | # Q | # J | |
| Arabic | (Ar) | 12,377 | 12,377 | 3,115 | 3,115 | 1,081 | 1,257 | 2,106,586 |
| Bengali | (Bn) | 1,713 | 1,719 | 440 | 443 | 111 | 130 | 304,059 |
| English | (En) | 3,547 | 3,547 | 878 | 878 | 744 | 935 | 32,907,100 |
| Finnish | (Fi) | 6,561 | 6,561 | 1,738 | 1,738 | 1,254 | 1,451 | 1,908,757 |
| Indonesian | (Id) | 4,902 | 4,902 | 1,224 | 1,224 | 829 | 961 | 1,469,399 |
| Japanese | (Ja) | 3,697 | 3,697 | 928 | 928 | 720 | 923 | 7,000,027 |
| Korean | (Ko) | 1,295 | 1,317 | 303 | 307 | 421 | 492 | 1,496,126 |
| Russian | (Ru) | 5,366 | 5,366 | 1,375 | 1,375 | 995 | 1,168 | 9,597,504 |
| Swahili | (Sw) | 2,072 | 2,401 | 526 | 623 | 670 | 743 | 136,689 |
| Telugu | (Te) | 3,880 | 3,880 | 983 | 983 | 646 | 664 | 548,224 |
| Thai | (Th) | 3,319 | 3,360 | 807 | 817 | 1,190 | 1,368 | 568,855 |
| Total | | 48,729 | 49,127 | 12,317 | 12,431 | 8,661 | 10,092 | 58,043,326 |

Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval

| | | R@1k | | MRR@10 | | | |
|---|---|---|---|---|---|---|---|
| | Language | BM25 | mColB. | BM25 | mT5 | mMiniLM | mColB. |
| (1) | English (Orig.) | 0.857 | 0.953 | 0.184 | 0.366 | 0.366 | 0.352 |
| (2) | Spanish | 0.770 | 0.897 | 0.158 | 0.314 | 0.309 | 0.301 |
| (3) | French | 0.769 | 0.891 | 0.155 | 0.302 | 0.296 | 0.289 |
| (4) | Italian | 0.753 | 0.888 | 0.153 | 0.303 | 0.291 | 0.292 |
| (5) | Portuguese | 0.744 | 0.887 | 0.152 | 0.302 | 0.289 | 0.292 |
| (6) | Indonesian | 0.767 | 0.854 | 0.149 | 0.298 | 0.293 | 0.275 |
| (7) | German | 0.674 | 0.867 | 0.136 | 0.289 | 0.278 | 0.281 |
| (8) | Russian | 0.685 | 0.836 | 0.124 | 0.263 | 0.251 | 0.250 |
| (9) | Chinese | 0.678 | 0.837 | 0.116 | 0.249 | 0.249 | 0.246 |
| *Zero-shot (models were fine-tuned on the 9 languages above)* | | | | | | | |
| (10) | Japanese | 0.714 | 0.806 | 0.141 | 0.267 | 0.263 | 0.236 |
| (11) | Dutch | 0.694 | 0.862 | 0.140 | 0.292 | 0.276 | 0.273 |
| (12) | Vietnamese | 0.714 | 0.719 | 0.136 | 0.256 | 0.247 | 0.180 |
| (13) | Hindi | 0.711 | 0.785 | 0.134 | 0.266 | 0.262 | 0.232 |
| (14) | Arabic | 0.638 | 0.749 | 0.111 | 0.235 | 0.219 | 0.209 |

mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset

# METHODS

Main methods categories in information retrieval:

- Lexical retrieval – BM25
- Dense retrieval – transformer encodings ex. SBERT, DPR
- Sparse retrieval – sparse encodings with transformer models ex. SPLADE, SPARTA

- Reranking – reranking model with transformer model used as cross-encoder

# BM25

BM25(Best Match 25) is a ranking function used in information retrieval systems to estimate the relevance of a document to a given query. It uses the relevance score of a document based on the query terms and the document's content.

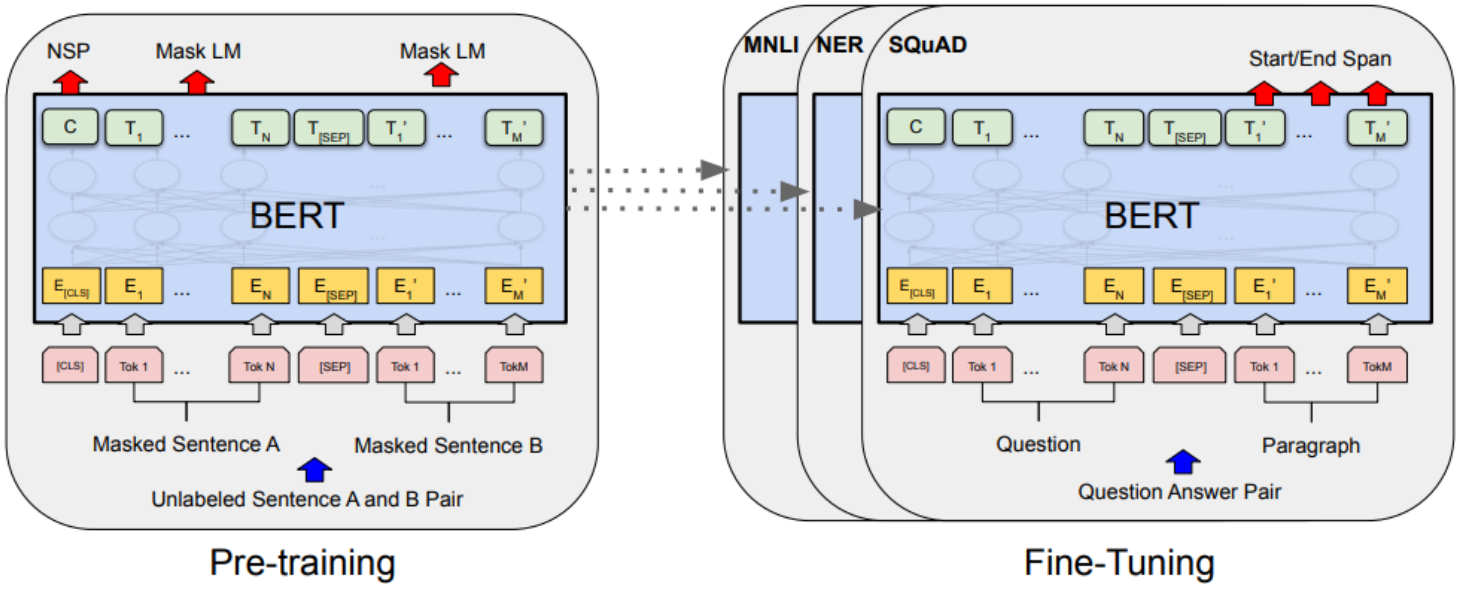
$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

F(qi, D) is a number of times that qi occured in document D
Avgdl is the average document length in the collection
IDF is a inverse document frequency

elasticsearch

# TRANSFORMER MODEL

In order to get whole sentence representation, transformer model like BERT can be used.
The model is pretrained with Masked Language Modeling Task and afterwards fine-tuned to target task.



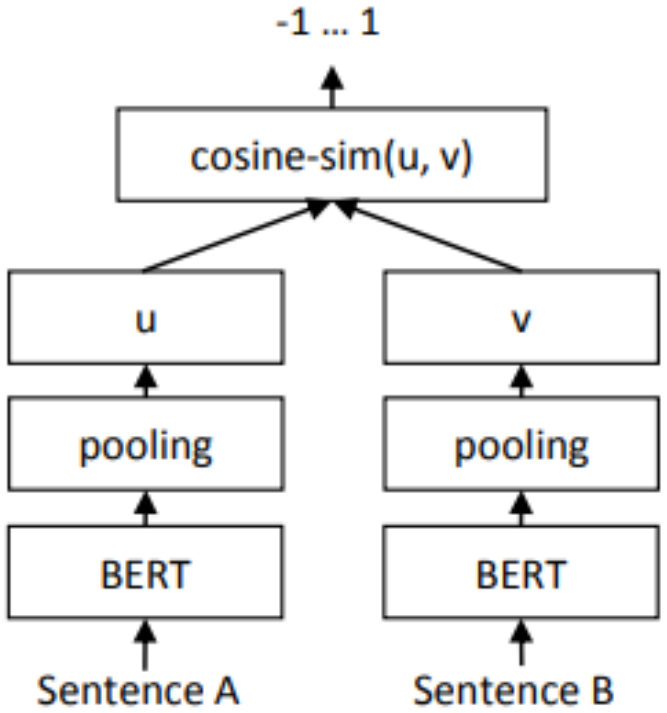

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

# BIENCODER ARCHITECTURE

Get representation of the sentences from the model and calculate similarity with dot product or cosine similarity.

We have to get the representation of the whole sentence and we want to store it in one vector, that is why pooling is needed.

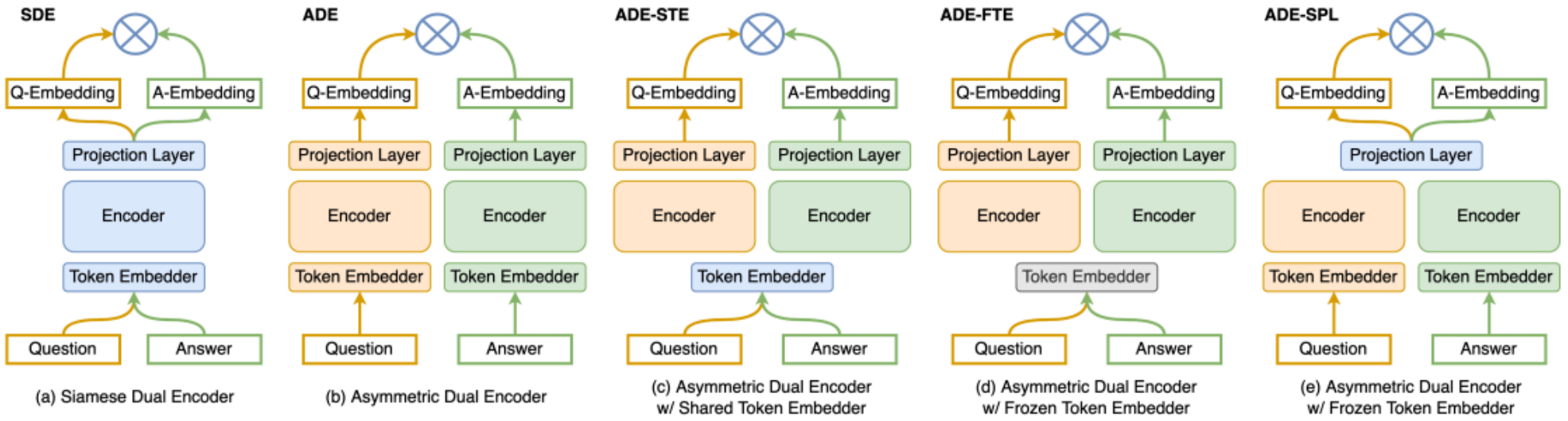
Pooling: CLS or MEAN or MAX(not used right now)




Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

# BIENCODER ARCHITECTURE

There are different approaches to biencoders, they can have shared weights or be completely separate encoders for query and passage – then they are usually called dual encoders. Other variations are also possible.




Exploring Dual Encoder Architectures for Question Answering

# BIENCODER ARCHITECTURE – VECTOR DATABASES

When there is a large number of document embeddings, which is usually the case when we are working with large datasets we can use vector databases as they can perform approximate neighbor search.

Examples:

- FAISS
- Weaviate
- Milvus
- Pincone



# DENSE PASSAGE RETRIEVER

Two encoders: separate encoder for passages and queries.

In-batch negatives – we use all batch examples except positive pair as negatives. It helps to get better generalization.

Batch size: 128, one additional BM25 negative

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, p_{i,j}^-)}}.$$

Dense Passage Retrieval for Open-Domain Question Answering

# E5 RETRIEVER MODELS

Batch size: 32K pre-training, 256 fine-tuning

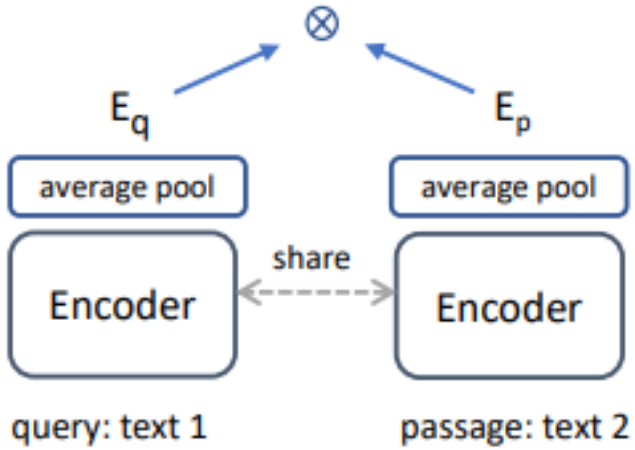
Task prefix: “query: “ and “passage: “

Losses:

InfoNCE contrastive loss and KL Divergence for distillation

$$\min \quad L_{\text{cont}} = -\frac{1}{n}\sum_i \log \frac{e^{s_\theta(q_i,p_i)}}{e^{s_\theta(q_i,p_i)} + \sum_j e^{s_\theta(q_i,p_{ij}^-)}}$$

$$\min \quad D_{\text{KL}}(p_{\text{ce}}, p_{\text{stu}}) + \alpha L_{\text{cont}}$$




Text Embeddings by Weakly-Supervised Contrastive Pre-training

# UNSUPERVISED PRETRAINING

Information Retrieval requires large amount of annotated data, which is not always available. For that reason, some unsupervised methods were proposed.

**Inverse Cloze Task (ICT)** – pseudo query is generated from a document paragraph, where one sentence is extracted and treated as query and the rest of the document is treated as the corresponding passage.
**Independent cropping** – similar to ICT, but the query and passage are cropped independently from a document.
**Other augmentation** – deletion, replacement or masking.

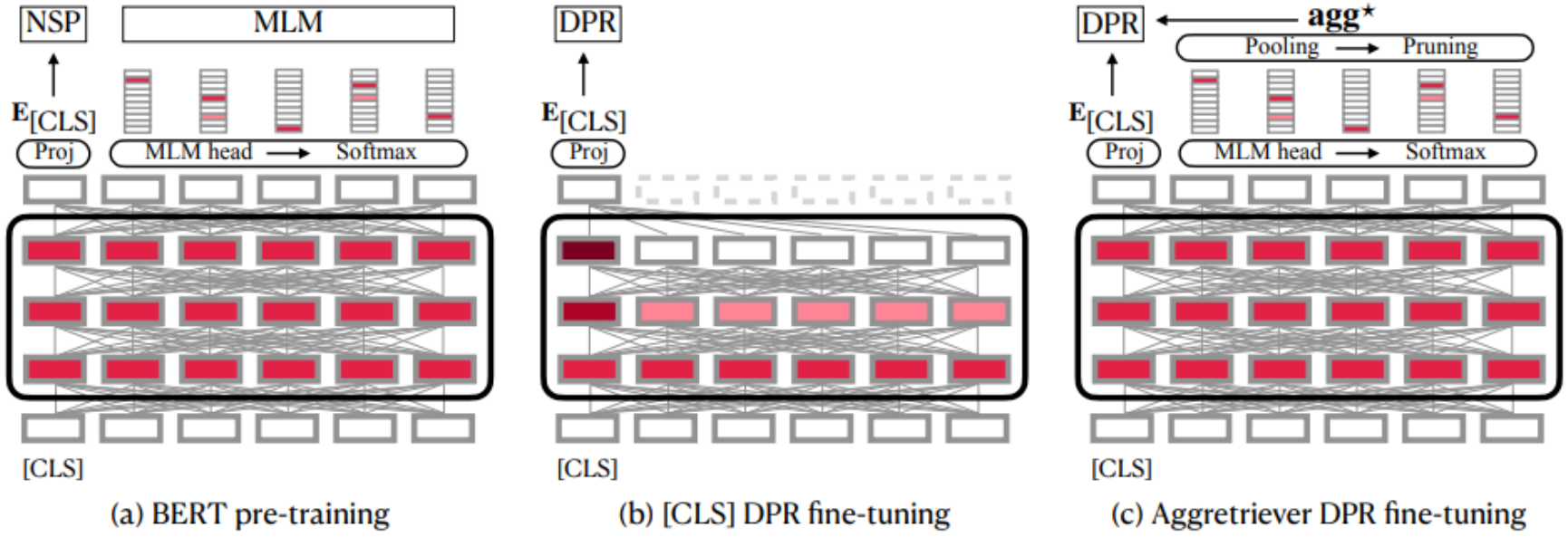
In paper:
Pretraining with In-batch-negatives, 2048 batch size.
Fine-tunning on MS MARCO, 1024 batch size.

Unsupervised Dense Information Retrieval with Contrastive Learning

# SPARSE RETRIEVAL

Reuse the already trained MLM weight matrix in order to obtain the distribution over the model vocabulary and use it as representation. The representation is a 30k length vector, so pooling is required to make it computationally feasable for larger datasets, because indexes may get heavy.

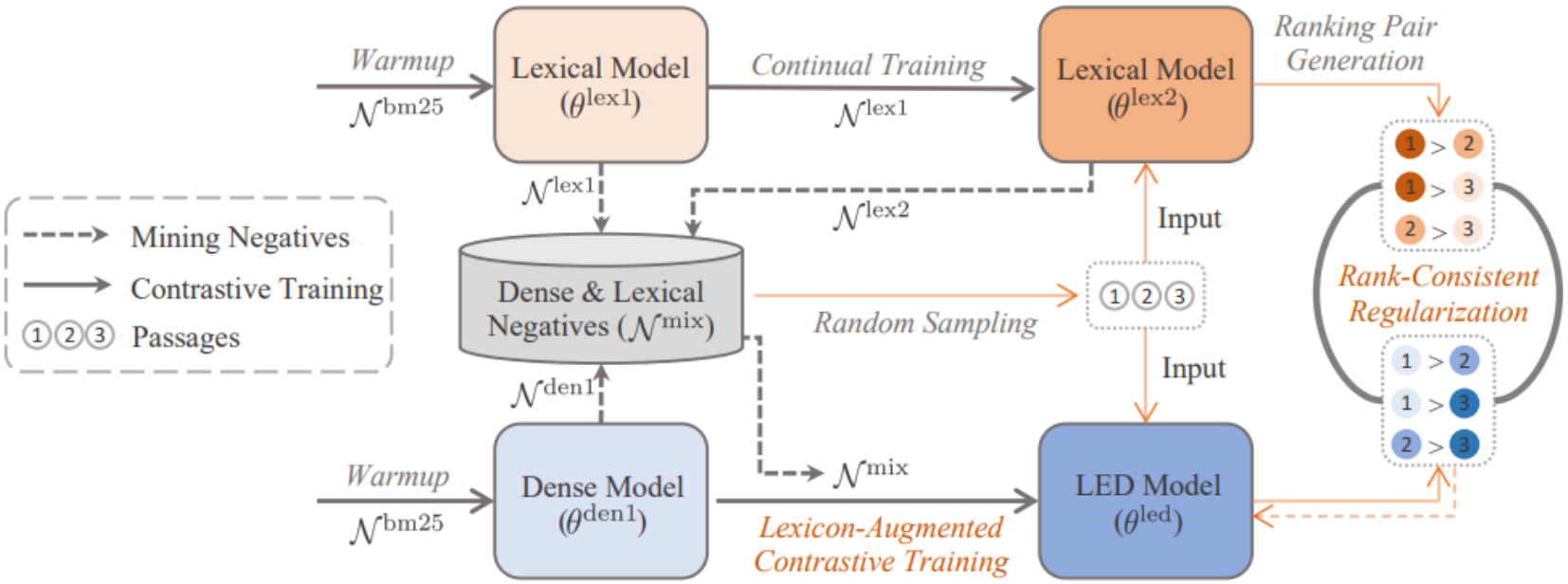
Example models are: SPLADE or Aggretriever.




Aggretriever: A Simple Approach to Aggregate Textual Representations for Robust Dense Passage Retrieval

# IMPORTANCE OF HARD NEGATIVES

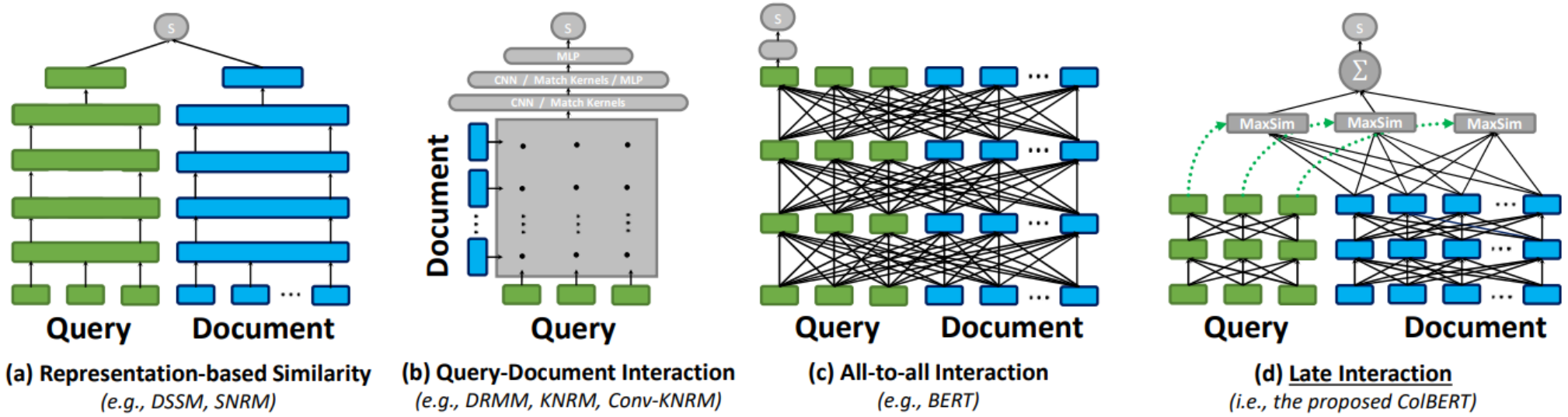
It is beneficial to mine hard negative examples for each training query from different models and train the model on the pool of various hard negative examples.




LED: Lexicon-Enlightened Dense Retriever for Large-Scale Retrieval

# RERANKING

In case of biencoders the query and document are encoded separately. The similarity is measured on already precomputed vector representations. In case of reranking, we are using cross-encoders, where the input is query and document at the same time. The model is able to use attention mechanism to decide if document is relevant for provided query.



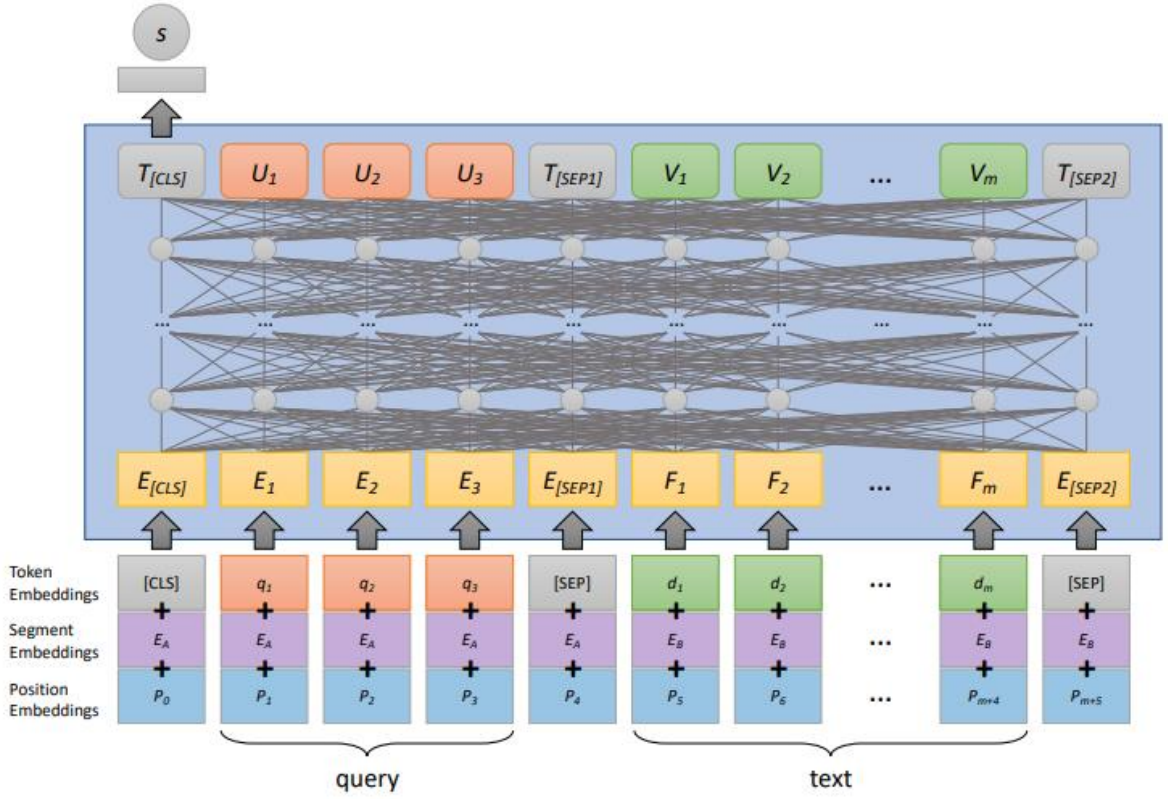

ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

# RERANKING – BERT MODEL

The **monoBERT** ranking model adapts BERT for sequence classification task.

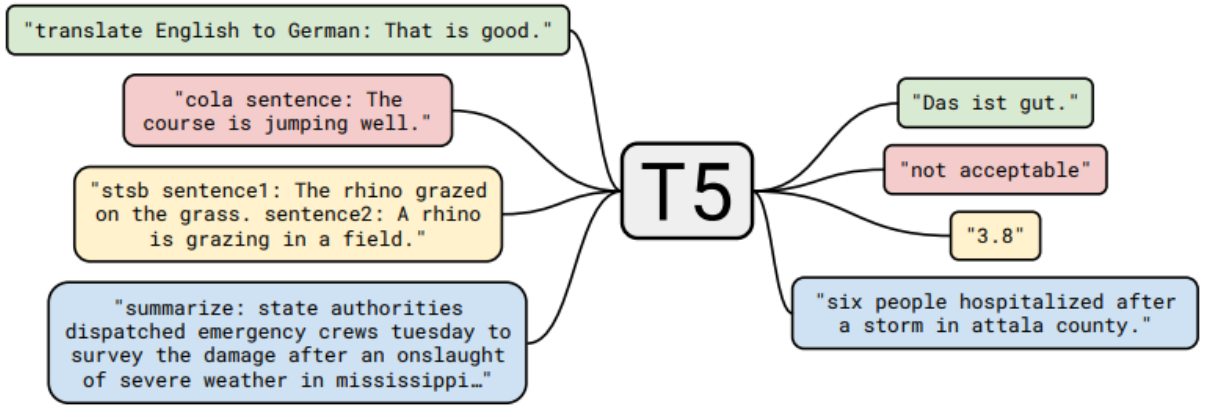
The input is a query and a candidate text separated with [SEP] token.

The final representation of the [CLS] token is fed to a fully-connected layer that produces the relevance score of the text with respect to the query.




Pretrained Transformers for Text Ranking: BERT and Beyond

# RERANKING – T5 MODEL




Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

We can also use sequence-to-sequence models for reranking.

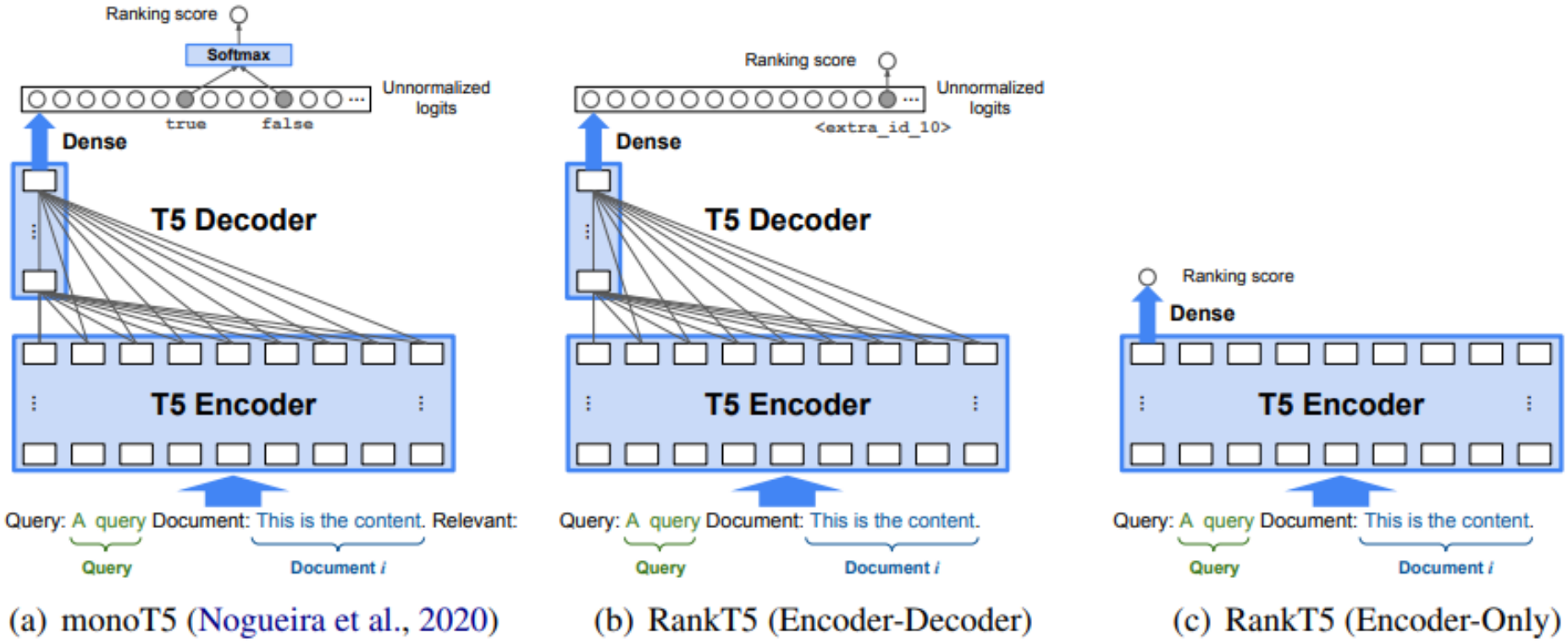
monoT5 – extra tokens are added. One represents **true,** when Query and Document are relevant, or **false** they are not. Model is trained to predict the token based on input.

Example input: Query: [q] Document: [d] Relevant:

Example of new tokens: '_true' , '_false'

Document Ranking with a Pretrained Sequence-to-Sequence Model

# RERANKING – T5 MODEL




RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses

# METRICS

Recall (Recall@k) cut off at the k ranking position. The recall@k informs how many relevant documents from the collection were classified to @k ranking.

$$recall = \frac{|relevant| \cap |retrieved|}{|relevant|}$$

# METRICS

Mean Reciprocal Rate (MRR@k) – the official MS Marco metric. MRR@k measures the quality of rank regarding the first relevant passage in ranking.

$$MRR@k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

# METRICS

Normalised Cumulative Discount Gain(nDCG@k) – reported in the original BEIR benchmark. NCDG@k measures the quality of ranking considering all relevant passages and its position in @k retrieved documents,

$$NDCG@k = \frac{\sum_{i=1}^{k(rank\_order)} \frac{Gain}{log_2(i+1)}}{\sum_{i=1}^{k(real\_order)} \frac{Gain}{log_2(i+1)}},$$

# SUMMARY

Key takeaways:

- BM25 is very strong baseline for open-domain Information Retrieval, it is fast and do not require training. Stemming and lemmatization improve the results significantly.
- Training retrievers require a large amount of data, big batch sizes and hard negative samples. They perform better in-domain that BM25, but just recently we were able to train better dense retrievers for open domain than BM25.
- Rerankers are necessary if we want to increase accuracy of our system, but there is an additional computational cost, as you have to compute to score for each query, document pair.

# THANK YOU FOR YOUR ATTENTION

Contact: konrad.wojtasik@pwr.edu.pl

Github: https://github.com/kwojtasi/modern-ir-aitech

# REFERENCES


1. Document Ranking with a Pretrained Sequence-to-Sequence Model
2. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses
3. Pretrained Transformers for Text Ranking: BERT and Beyond
4. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT
5. LED: Lexicon-Enlightened Dense Retriever for Large-Scale Retrieval
6. Aggretriever: A Simple Approach to Aggregate Textual Representations for Robust Dense Passage Retrieval
7. Unsupervised Dense Information Retrieval with Contrastive Learning
8. Text Embeddings by Weakly-Supervised Contrastive Pre-training
9. Dense Passage Retrieval for Open-Domain Question Answering
10. Exploring Dual Encoder Architectures for Question Answering
11. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
12. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
13. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking

# REFERENCES


14. Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval
15. mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset
16. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models
17. BEIR-PL: Zero Shot Information Retrieval Benchmark for the Polish Language
18. https://www.sbert.net/ - Accessed 13.06.2023