Information Retrieval

Probabilistic Information Retrieval

Last lecture

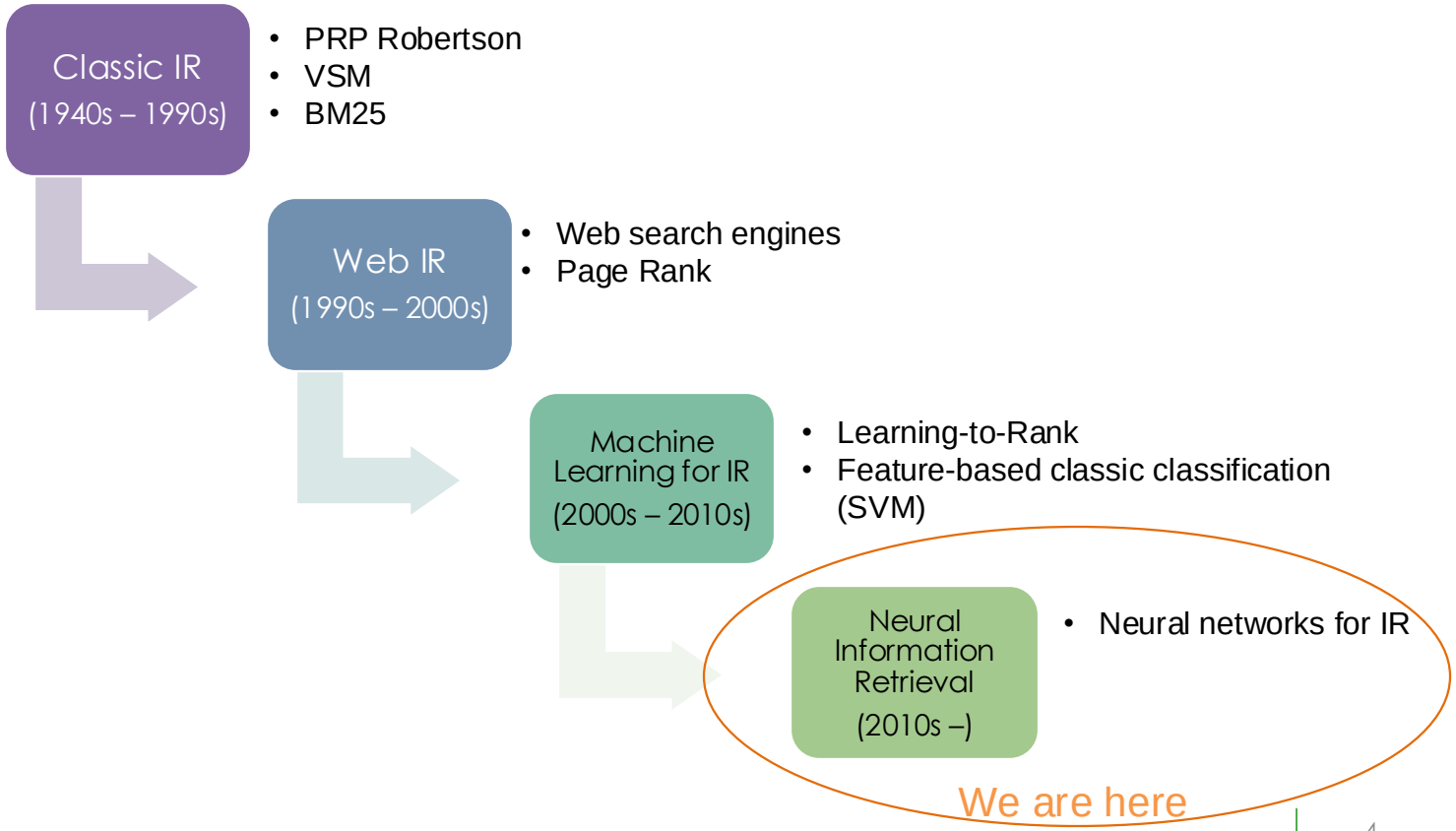
- You have learned the fundamentals of Deep Learning, with special interest in text processing, where we address the critical problem of text representation.
  - DL -> Neural networks -> Mathematical model -> Learn through optimization
  - Distributed representations
    - Static
    - Contextualized

- Two questions from the previous class:
  - When to use local representation vs distributed?
    - Local are only used to obtain distributed representations.
  - How contextualized representation are used/obtained?
    - Check the notebook at the end of this presentation.

This lecture

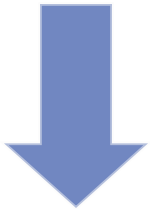
- Introduction to Neural Information Retrieval
  - Contextual history
- Fundamentals in NIR
  - The two Neural Architectures
  - Retrievers vs Rerankers
  - How to train
- Literature examples
  - Timeline
  - Shallow models
  - Deep models

Introduction



Neural Information Retrieval

- Use labelled data to teach an AI model how to search for information!
  - The aim is to let the AI model learn an internal notion of relevance without human intervention



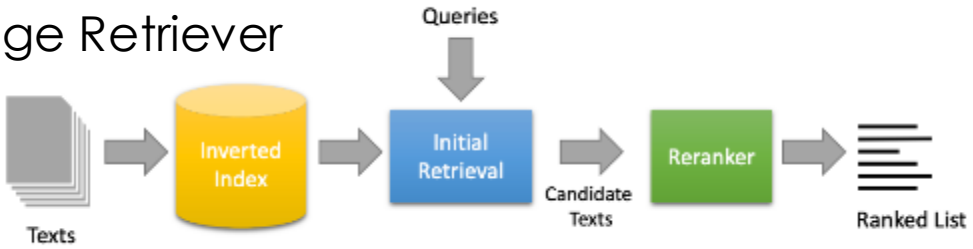
Neural Information Retrieval

Up until now we only used Retrievers

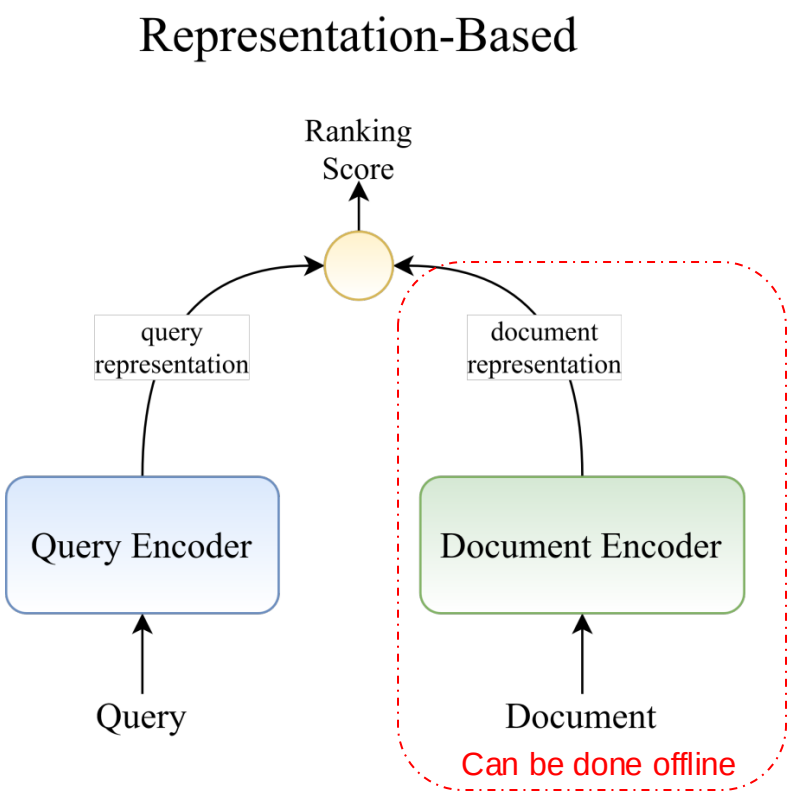
- Retrievers:
  - Focus on efficiently finding relevant documents from a large corpus
  - Optimize for speed and recall

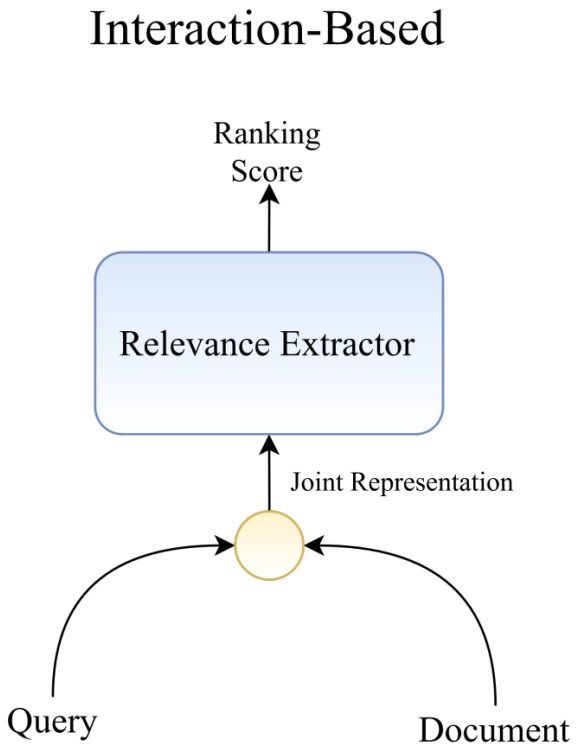
- Rerankers:
  - Focus on precisely ordering already retrieved documents
  - Optimize for precision and ranking quality
  - Can analyse deeper semantic relationships between query and documents
  - Much more expensive to run when compared to Retrievers

- Hybrid approach: Two-stage Retriever

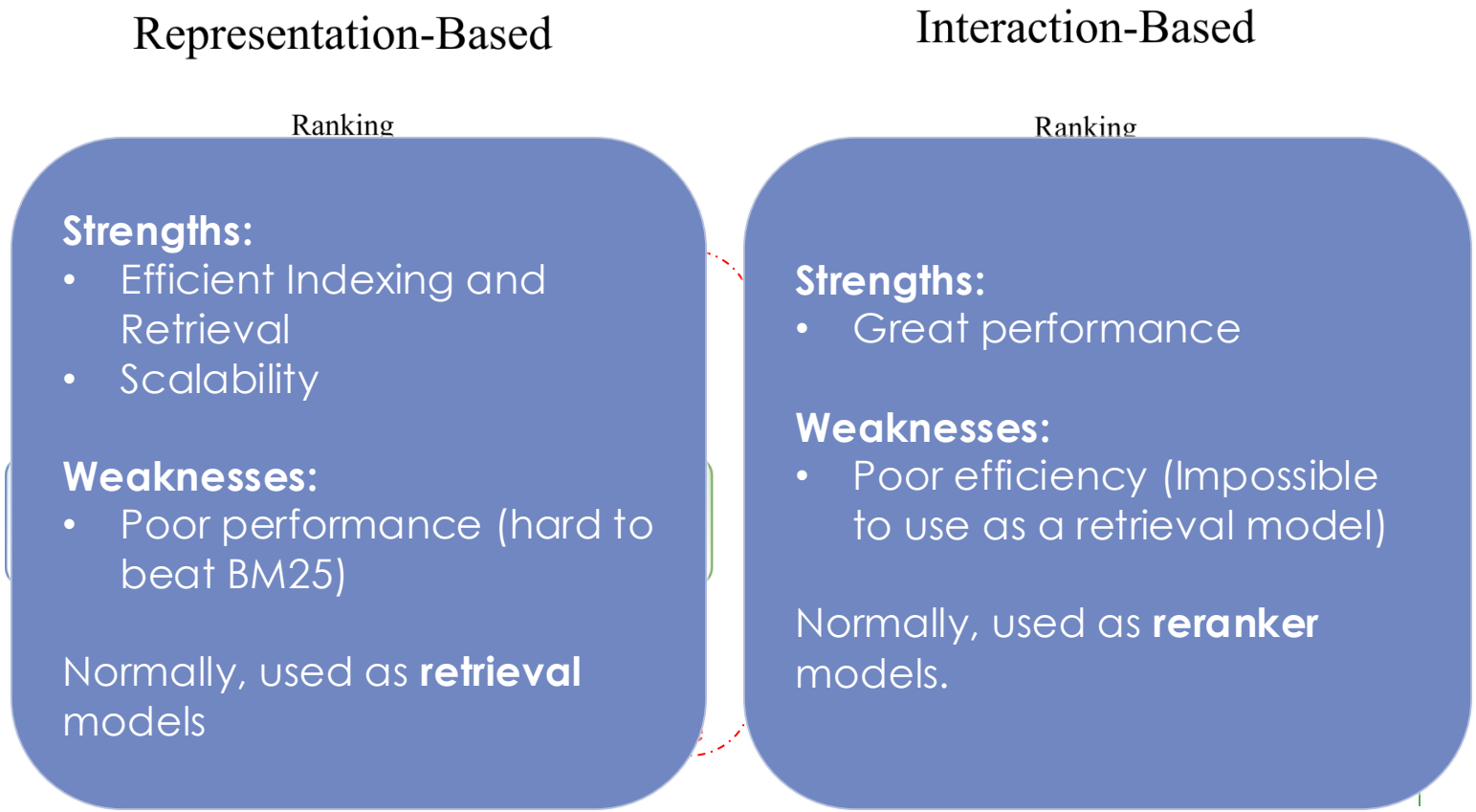


The two architectures for neural IR





The two architectures for neural IR



How to train a neural IR model?

- Training Dataset
  - Collection of queries with a list of **relevant** and **unrelevant** documents.

Positive example

Negative example

In practice, we only have an incomplete list of query-positive pairs


```
{"question": "Is Hirschsprung disease a mendelian or a multifactorial disorder?", "goldstandard_documents": ["PMID:15858239", "PMID:15829955", "PMID:6650562", "PMID:
{"question": "List signaling molecules (ligands) that interact with the receptor EGFR?", "goldstandard_documents": ["PMID:21514161", "PMID:23212918", "PMID:23888072"
{"question": "Is the protein Papilin secreted?", "goldstandard_documents": ["PMID:3320045", "PMID:7515725", "PMID:20805556", "PMID:19297413", "PMID:19724244", "PMID:
{"question": "Are long non coding RNAs spliced?", "goldstandard_documents": ["PMID:22955988", "PMID:22955974", "PMID:24285305", "PMID:22707570", "PMID:21622663", "PM
{"question": "Is RANKL secreted from the cells?", "goldstandard_documents": ["PMID:21618594", "PMID:23698708", "PMID:23827649", "PMID:22901753", "PMID:22948539", "PM
{"question": "Which miRNAs could be used as potential biomarkers for epithelial ovarian cancer?", "goldstandard_documents": ["PMID:21345725", "PMID:22246341", "PMID:
{"question": "Which acetylcholinesterase inhibitors are used for treatment of myasthenia gravis?", "goldstandard_documents": ["PMID:21815707", "PMID:21845054", "PMID
{"question": "Has Denosumab (Prolia) been approved by FDA?", "goldstandard_documents": ["PMID:22826702", "PMID:21208140", "PMID:21113693", "PMID:21129866", "PMID:239
{"question": "List the human genes encoding for the dishevelled proteins?", "goldstandard_documents": ["PMID:12883684", "PMID:24040443", "PMID:19618470", "PMID:88173
{"question": "Name synonym of Acrokeratosis paraneoplastica.", "goldstandard_documents": ["PMID:3819049", "PMID:18775590", "PMID:6225397", "PMID:22146896", "PMID:779
```

How to train a neural IR model?

- Training Dataset
  - Collection of queries with a list of **relevant** and **unrelevant** documents.

Positive example

Negative example

In practice, we only have an incomplete list of query-positive pairs

How can we get the negative examples?


```
{"question": "Is Hirschsprung disease a mendelian or a multifactorial disorder?", "goldstandard_documents": ["PMID:15858239", "PMID:15829955", "PMID:6650562", "PMID:
{"question": "List signaling molecules (ligands) that interact with the receptor EGFR?", "goldstandard_documents": ["PMID:21514161", "PMID:23212918", "PMID:23888072"
{"question": "Is the protein Papilin secreted?", "goldstandard_documents": ["PMID:3320045", "PMID:7515725", "PMID:20805556", "PMID:19297413", "PMID:19724244", "PMID:
{"question": "Are long non coding RNAs spliced?", "goldstandard_documents": ["PMID:22955988", "PMID:22955974", "PMID:24285305", "PMID:22707570", "PMID:21622663", "PM
{"question": "Is RANKL secreted from the cells?", "goldstandard_documents": ["PMID:21618594", "PMID:23698708", "PMID:23827649", "PMID:22901753", "PMID:22948539", "PM
{"question": "Which miRNAs could be used as potential biomarkers for epithelial ovarian cancer?", "goldstandard_documents": ["PMID:21345725", "PMID:22246341", "PMID:
{"question": "Which acetylcholinesterase inhibitors are used for treatment of myasthenia gravis?", "goldstandard_documents": ["PMID:21815707", "PMID:21845054", "PMID
{"question": "Has Denosumab (Prolia) been approved by FDA?", "goldstandard_documents": ["PMID:22826702", "PMID:21208140", "PMID:21113693", "PMID:21129866", "PMID:239
{"question": "List the human genes encoding for the dishevelled proteins?", "goldstandard_documents": ["PMID:12883684", "PMID:24040443", "PMID:19618470", "PMID:88173
{"question": "Name synonym of Acrokeratosis paraneoplastica.", "goldstandard_documents": ["PMID:3819049", "PMID:18775590", "PMID:6225397", "PMID:22146896", "PMID:779
```

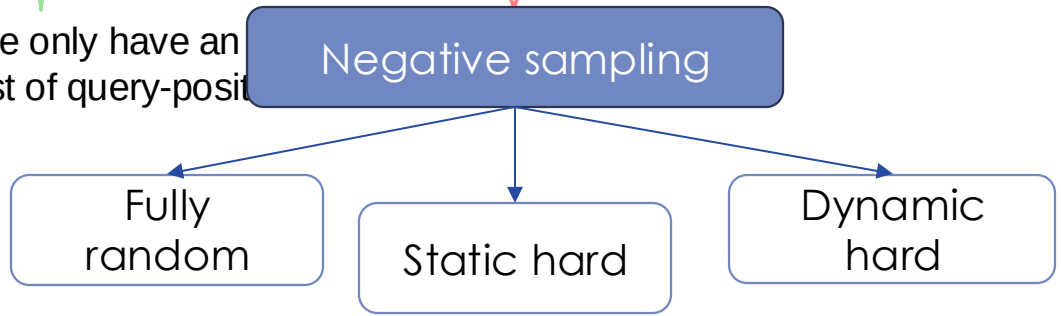
How to train a neural IR model?

- Training Dataset
  - Collection of queries with a list of **relevant** and **unrelevant** documents.

Positive example

Negative example

In practice, we only have an incomplete list of query-posit pairs




```
{"question": "Is Hirschsprung disease a mendelian or a multifactorial disorder?", "goldstandard_documents": ["PMID:15858239", "PMID:15829955", "PMID:6650562", "PMID:
{"question": "List signaling molecules (ligands) that interact with the receptor EGFR?", "goldstandard_documents": ["PMID:21514161", "PMID:23212918", "PMID:23888072"
{"question": "Is the protein Papilin secreted?", "goldstandard_documents": ["PMID:3320045", "PMID:7515725", "PMID:20805556", "PMID:19297413", "PMID:19724244", "PMID:
{"question": "Are long non coding RNAs spliced?", "goldstandard_documents": ["PMID:22955988", "PMID:22955974", "PMID:24285305", "PMID:22707570", "PMID:21622663", "PM
{"question": "Is RANKL secreted from the cells?", "goldstandard_documents": ["PMID:21618594", "PMID:23698708", "PMID:23827649", "PMID:22901753", "PMID:22948539", "PM
{"question": "Which miRNAs could be used as potential biomarkers for epithelial ovarian cancer?", "goldstandard_documents": ["PMID:21345725", "PMID:22246341", "PMID:
{"question": "Which acetylcholinesterase inhibitors are used for treatment of myasthenia gravis?", "goldstandard_documents": ["PMID:21815707", "PMID:21845054", "PMID
{"question": "Has Denosumab (Prolia) been approved by FDA?", "goldstandard_documents": ["PMID:22826702", "PMID:21208140", "PMID:21113693", "PMID:21129866", "PMID:239
{"question": "List the human genes encoding for the dishevelled proteins?", "goldstandard_documents": ["PMID:12883684", "PMID:24040443", "PMID:19618470", "PMID:88173
{"question": "Name synonym of Acrokeratosis paraneoplastica.", "goldstandard_documents": ["PMID:3819049", "PMID:18775590", "PMID:6225397", "PMID:22146896", "PMID:779
```

NIR Training – Negative Sampling

- Finding non-matching document query pairs to contrast with the relevant ones.
  - **Fully random (Random Sampling)**: Randomly samples documents from the collection that are not relevant to the question.
  - **Static Hard (Hard Negatives Sampling)**: Find documents that are similar but irrelevant to the query. For instance, consider the documents retrieved from BM25 as negatives.
    - Positive in BM25 that are not in the gold standard
  - **Dynamic Hard (Curriculum Negative Sampling)**: Start with easy negatives (random) during early training and then start to introduce harder negatives as training progress es.

NIR Training – Optimization

- **Pointwise:** Simple classification problem, we try to predict if a query-document pair is relevant or not according to the GS. Ex: Binary Cross Entropy Loss
- **Pairwise:** Ordering problem, we try to predict where the query-document pair is more relevant than the negative. Ex: hinge loss, RankNet loss
- **Listwise:** Optimizes the entire list, given the full query-document pairs we try to optimize the entire list such that the positive document have the highest scores. Ex: ListNet

NIR Training – Put everything together

- Data Loading and prepare the data loaders
  - Positive and negative examples
- Neural model instantiation
- Defining training objective
- Run training loop
  - Save the model checkpoints during training and register metrics

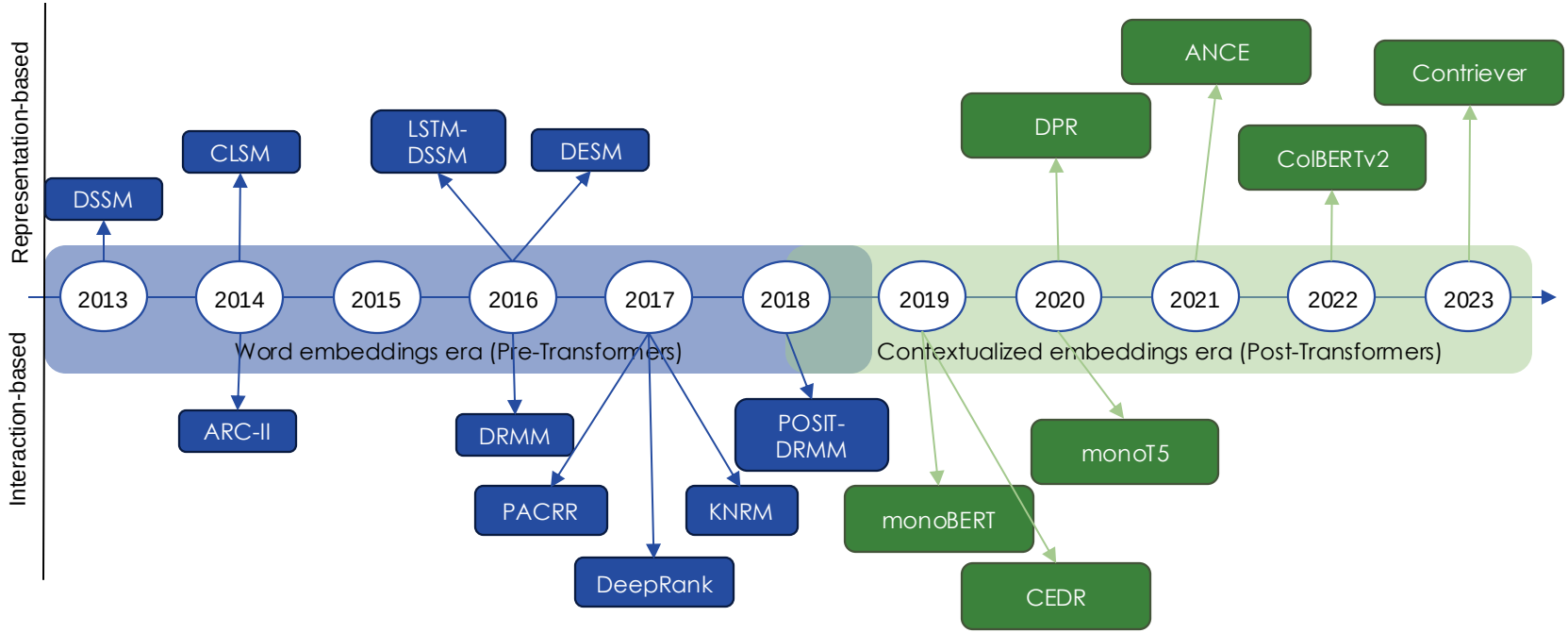
```
import torch
from torch import nn
# 1. Data Loading (text ids for query and document)
train_loader = DataLoader(
    dataset,  # [(query_ids, doc_ids, labels), ...]
    batch_size=32,
    shuffle=True
)

# 2. IR Neural Model
class IRModel(nn.Module):
    ...

# 3. Training objective
loss_function = nn.BCELoss()
model = IRModel()
optimizer = torch.optim.Adam(model.parameters())


# 4. Training Loop
optimizer = torch.optim.Adam(model.parameters())
for epoch in range(epochs):
    for query_ids, doc_ids, labels in train_loader:
        pred_scores = model(query_ids, doc_ids)
        loss = loss_function(pred_scores, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

NIR Training – Put everything together

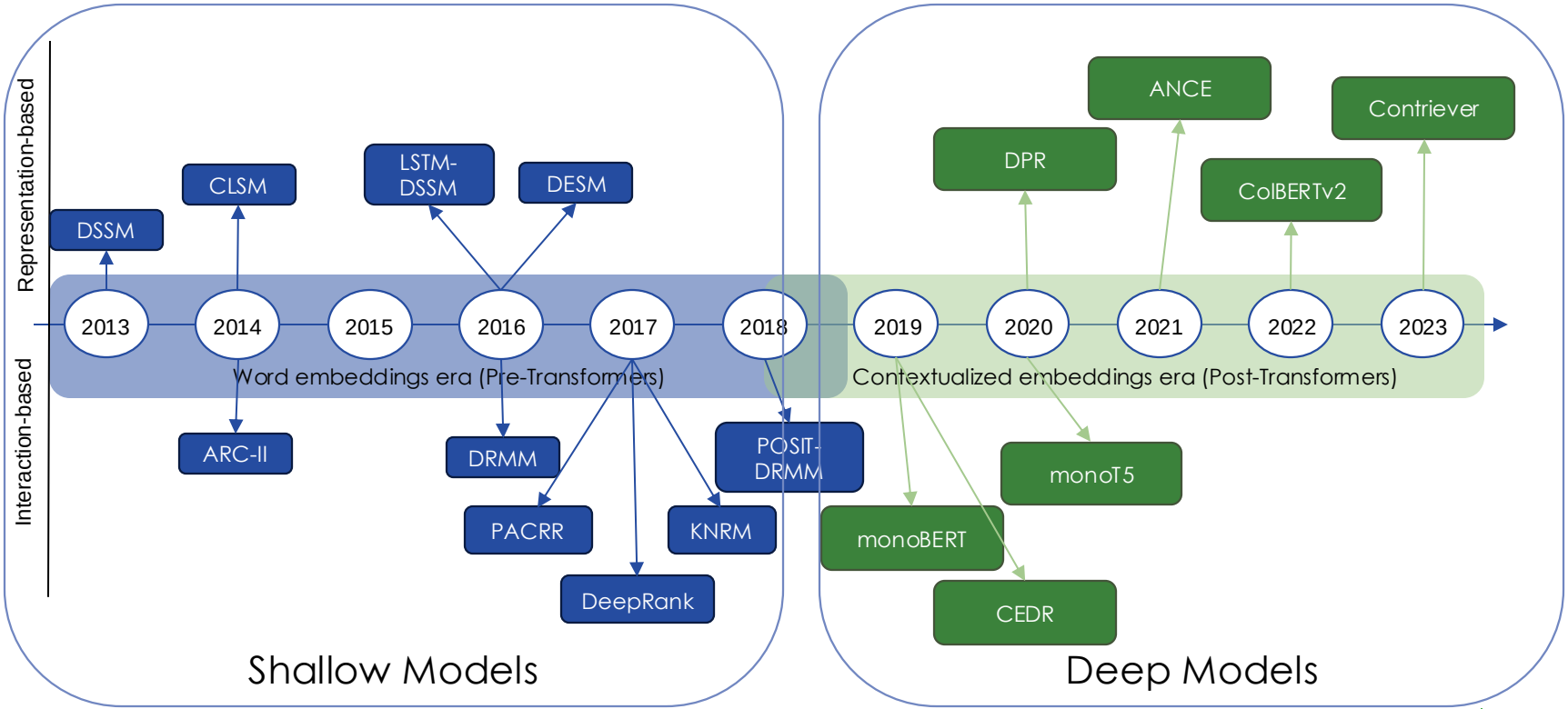
- Data Loading and prepare the data loaders
  - Positive and negative examples
- Neural model instantiation
- Defining training objective
- Run training loop
  - Save the model checkpoints during training and register metrics

```
import torch
from torch import nn

# 1. Data Loading (text ids for query and document)
train_loader = DataLoader(
    dataset,  # [(query_ids, doc_ids, labels), ...]
    batch_size=32,
    shuffle=True
)

# 2. IR Neural Model
class IRModel(nn.Module):
    ...

# 3. Training objective
loss_function = nn.BCELoss()
model = IRModel()
optimizer = torch.optim.Adam(model.parameters())


# 4. Training Loop
trainer = Trainer(train_loader, model, loss_function)
trainer.train()
```

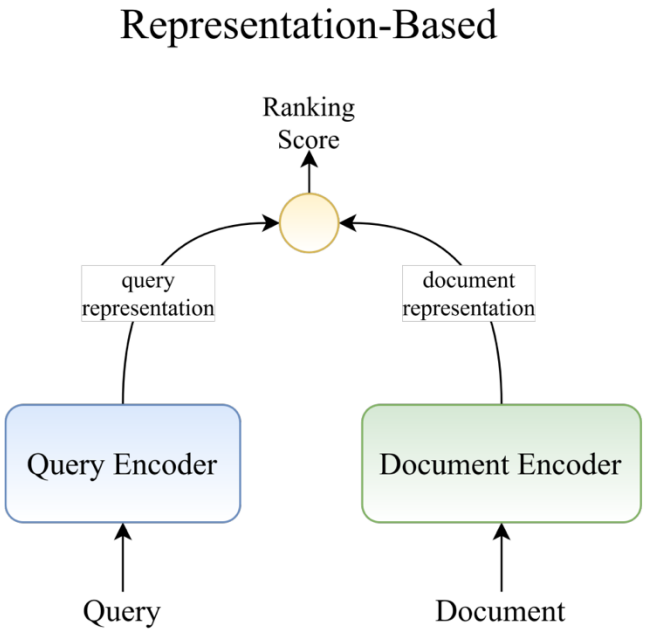
Literature Examples



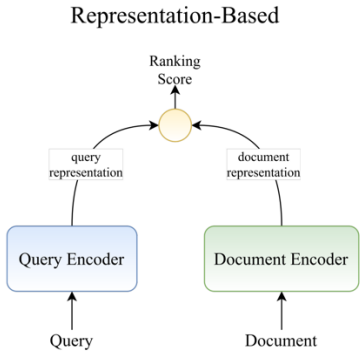
Literature Examples



Shallow representation-based



Average Word Embedding



- Simplest NIR model that creates dense representations by averaging the query and document terms embeddings.

$$\vec{q}^{e} = \frac{1}{|\vec{q}|} \sum_{i=1}^{|\vec{q}|} \frac{\vec{u}_i}{\| \vec{u}_i \|_2},$$

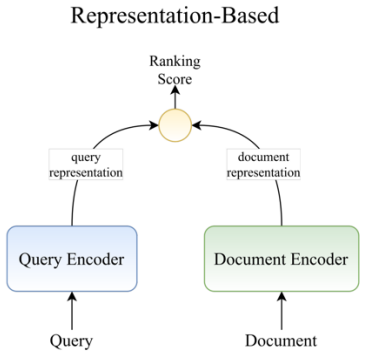
$$\vec{d}^{e} = \frac{1}{|\vec{d}|} \sum_{i=1}^{|\vec{d}|} \frac{\vec{v}_i}{\| \vec{v}_i \|_2}.$$

- Adding a tf-idf weighting scheme

$$\vec{q}^{e} = \frac{\sum_{i=1}^{|\mathbb{V}|} \frac{\vec{u}_i}{\|\vec{u}_i\|} \times tf(u_i, \vec{q}) \times idf(u_i)}{\sum_{i=1}^{|V|} tf(u_i, \vec{q}) \times idf(u_i)},$$

$$\vec{d}^{e} = \frac{\sum_{i=1}^{|\mathbb{V}|} \frac{\vec{v}_i}{\|\vec{v}_i\|} \times tf(v_i, \vec{d}) \times idf(v_i)}{\sum_{i=1}^{|\mathbb{V}|} tf(v_i, \vec{d}) \times idf(v_i)}.$$

Average Word Embedding



- Simplest NIR model that creates dense representations by averaging the query and document terms embeddings.

$$\vec{q}^{e} = \frac{1}{|\vec{q}|} \sum_{i=1}^{|\vec{q}|} \frac{\vec{u}_i}{\| \vec{u}_i \|_2},$$

$$\vec{d}^{e} = \frac{1}{|\vec{d}|} \sum_{i=1}^{|\vec{d}|} \frac{\vec{v}_i}{\| \vec{v}_i \|_2}.$$

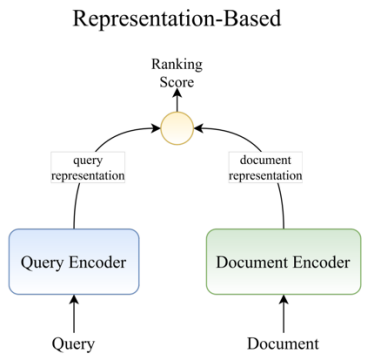
- Adding a tf-idf weighting scheme

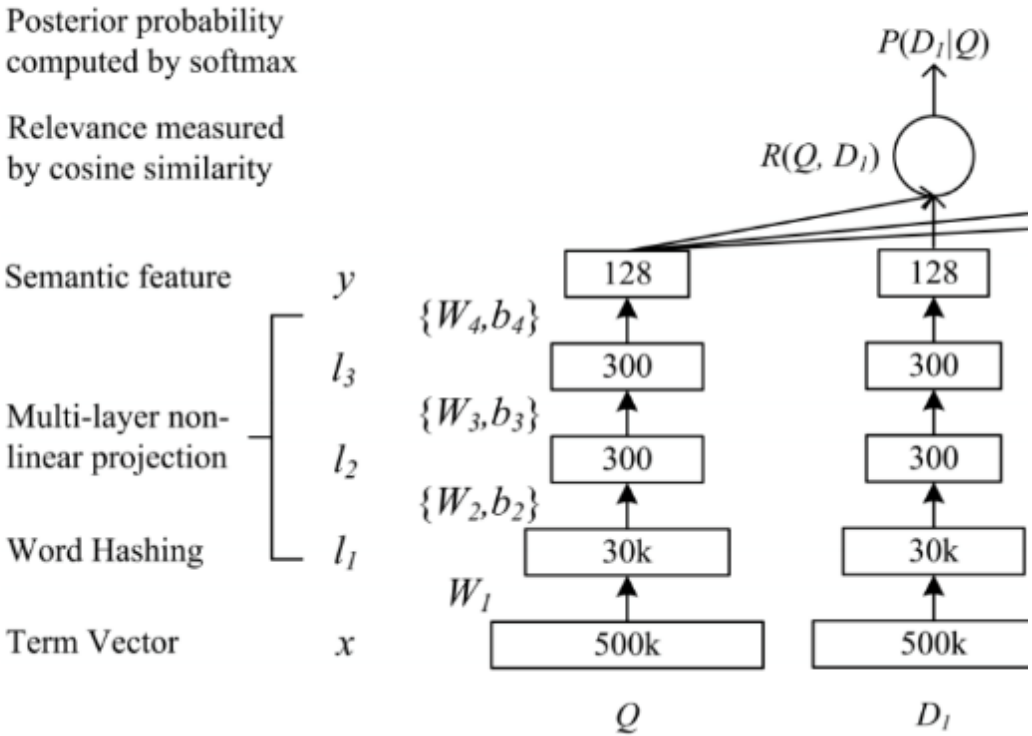
$$\vec{q}^{e} = \frac{\sum_{i=1}^{|\mathbb{V}|} \frac{\vec{u}_i}{\|\vec{u}_i\|} \times tf(u_i, \vec{q}) \times idf(u_i)}{\sum_{i=1}^{|V|} tf(u_i, \vec{q}) \times idf(u_i)},$$

$$\vec{d}^{e} = \frac{\sum_{i=1}^{|\mathbb{V}|} \frac{\vec{v}_i}{\|\vec{v}_i\|} \times tf(v_i, \vec{d}) \times idf(v_i)}{\sum_{i=1}^{|\mathbb{V}|} tf(v_i, \vec{d}) \times idf(v_i)}.$$

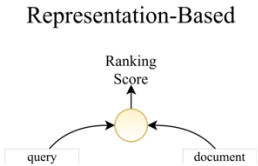
It utilizes pretrained word embeddings. Optionally you can finetune these embeddings

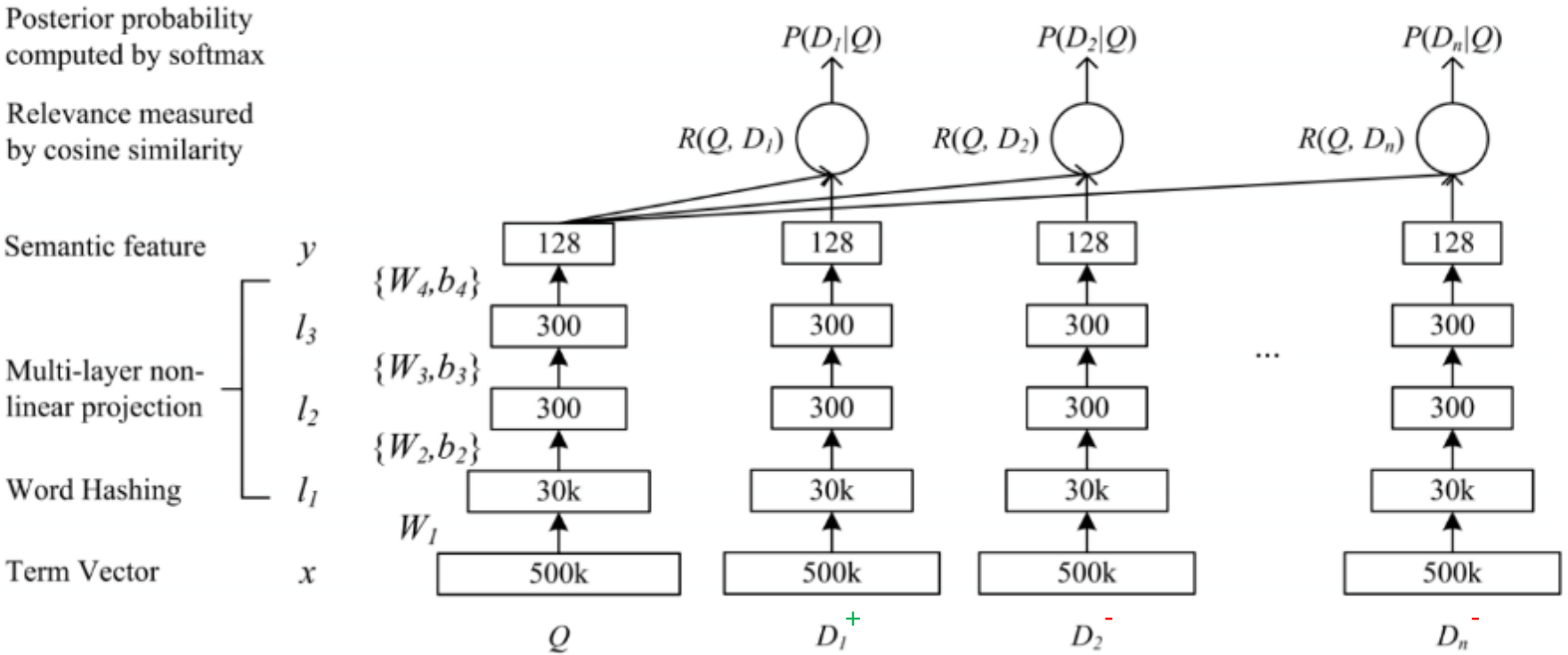
DSSM (Deep Semantic Similarity Model)



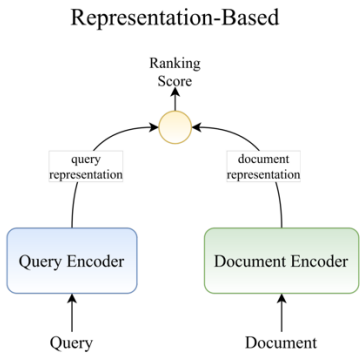


DSSM (Deep Semantic Similarity Model)



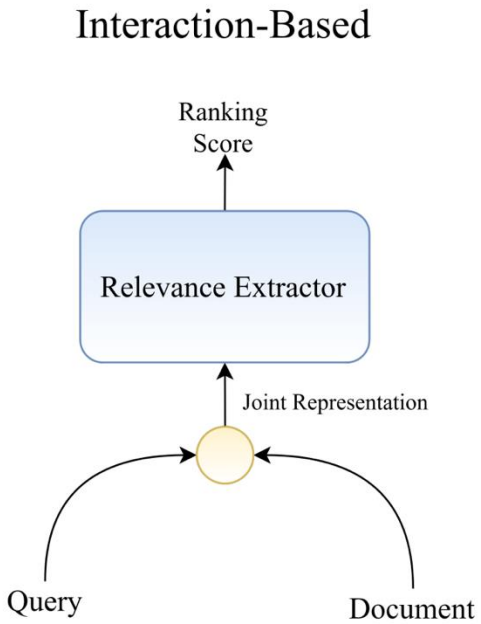


DSSM (Deep Semantic Similarity Model)

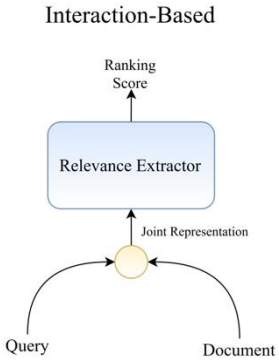


- Other variants that explored embeddings with:
  - Convolutional layers and LSTM layers
- Overall poor performance (vs BM25)

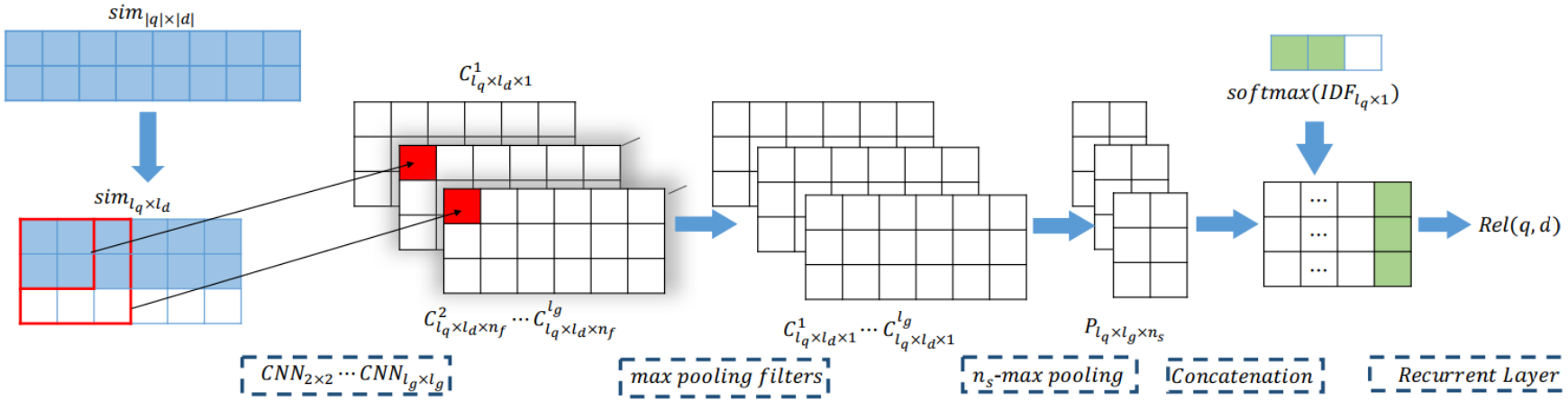
Shallow interaction-based



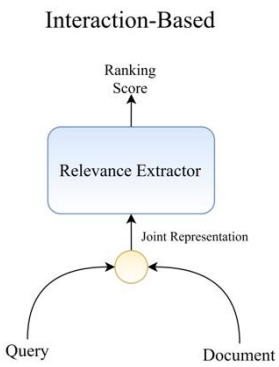
PACRR (2017)



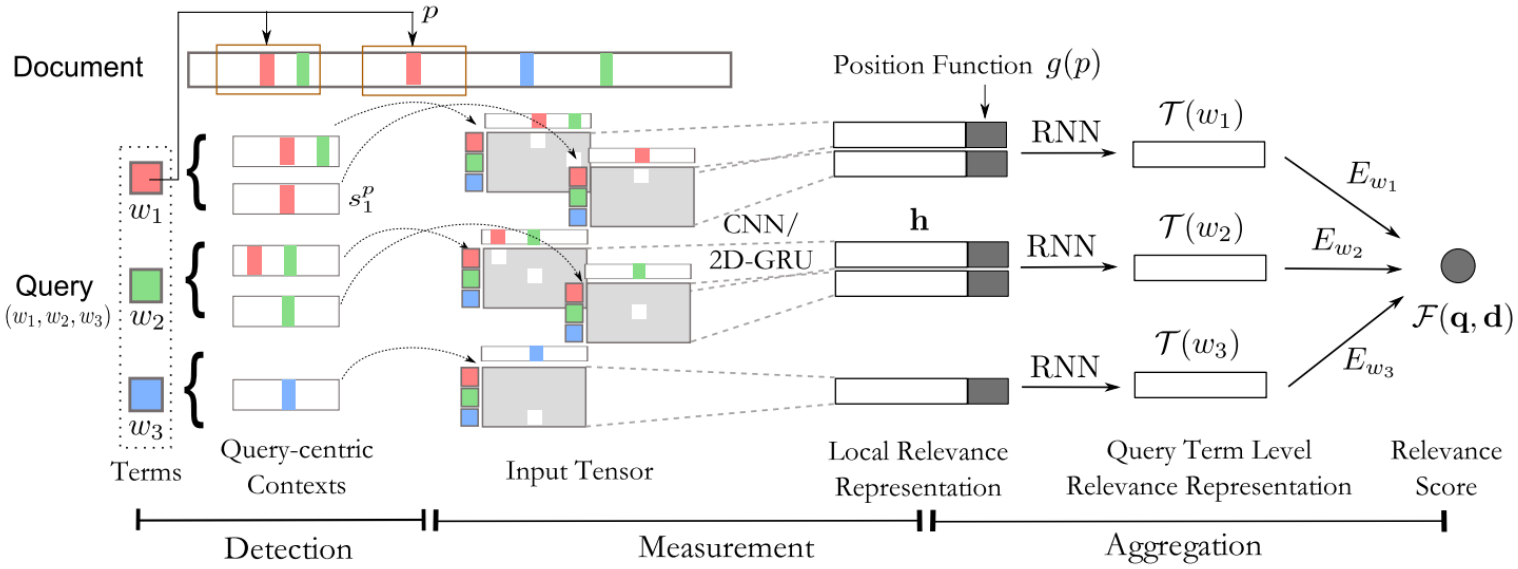
- Mainly uses convolution kernels to extract matching patters over a similarity matrix
- Adds a late IDF feature before the final ranking score computation



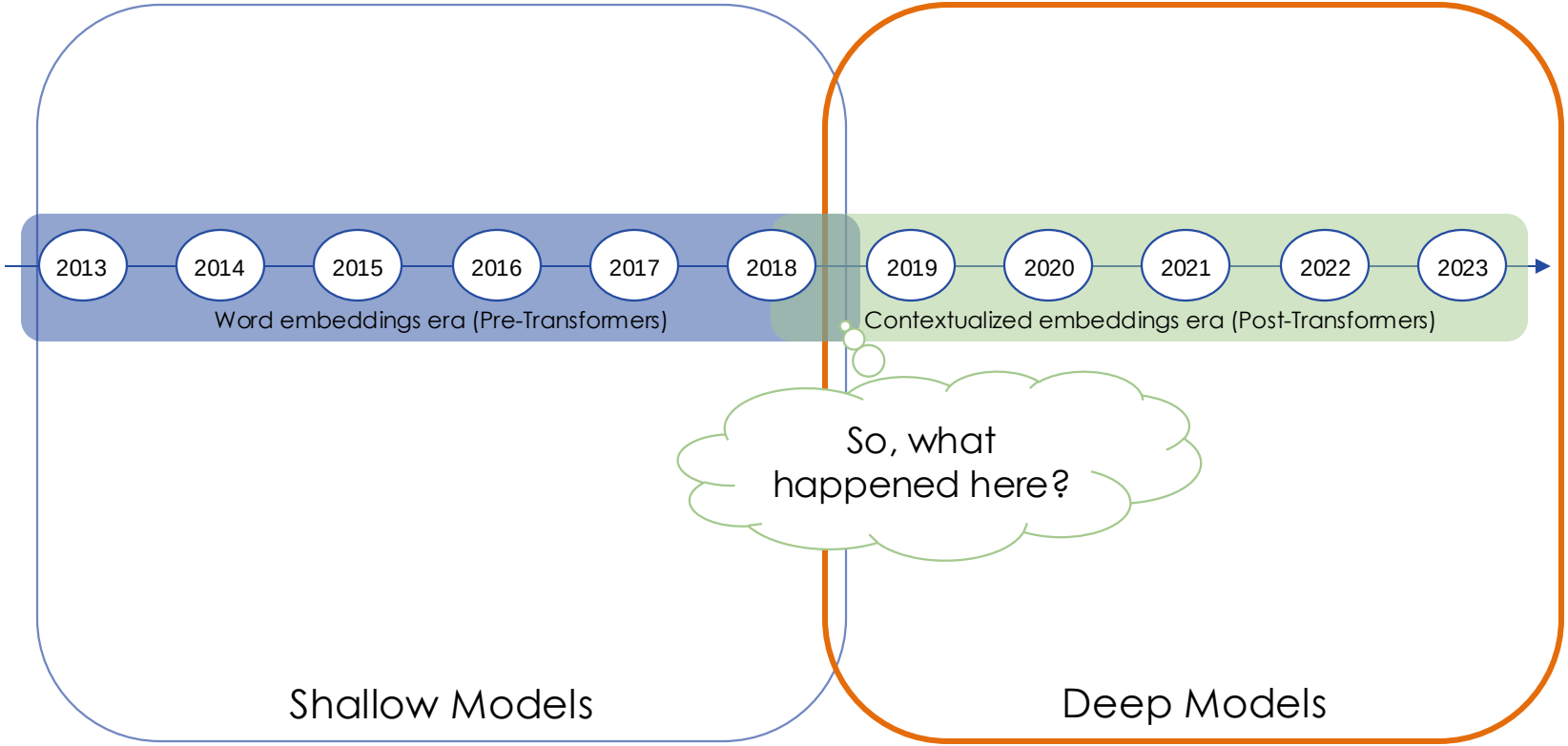
DeepRank (2017)



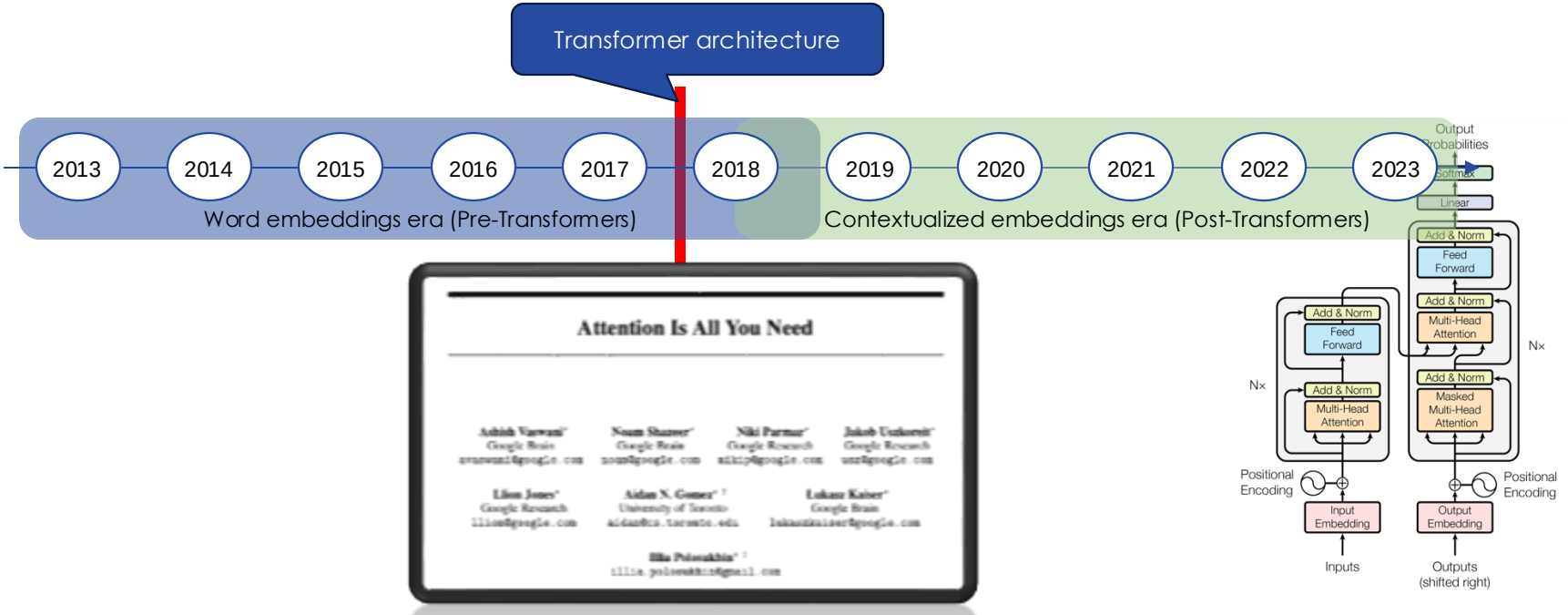
- Tries to mimic the way humans search for information.
- Weights the importance of each query term, since different query terms carry different importance for the information need.



Moving to the Deep Models



Moving to the Deep Models

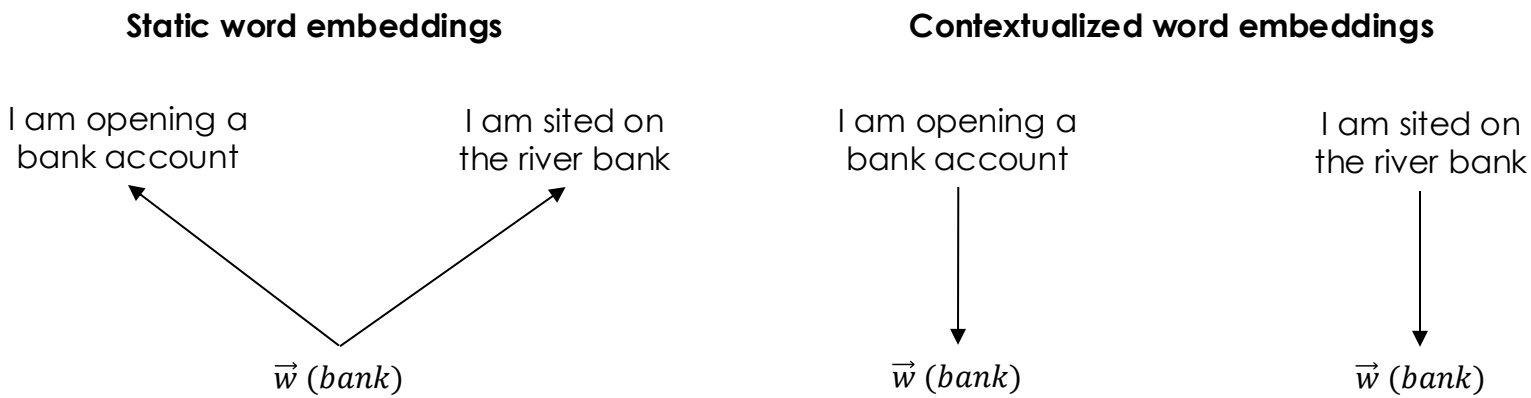


Backbone (magic) behind LLMs (chatGPT)

Contextualize Transformer Models

- They revolutionize language modelling by efficiently processing and learning from billions of documents and web pages.
- They enable the creation of highly **contextualized** embeddings, capturing nuanced meanings and relationships in large-scale textual data.

Contextualize Transformer Models



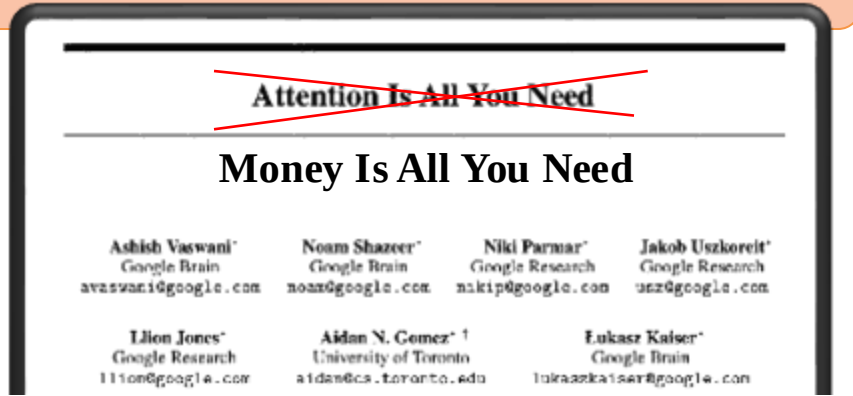
From previous lecture, do you remember this?

- Turns out creating contextualized word embeddings is hard, since it requires processing thousands of billions of tokens! Something that previous RNN based solutions were not mature (efficient) enough.

Contextualize Transformer Models

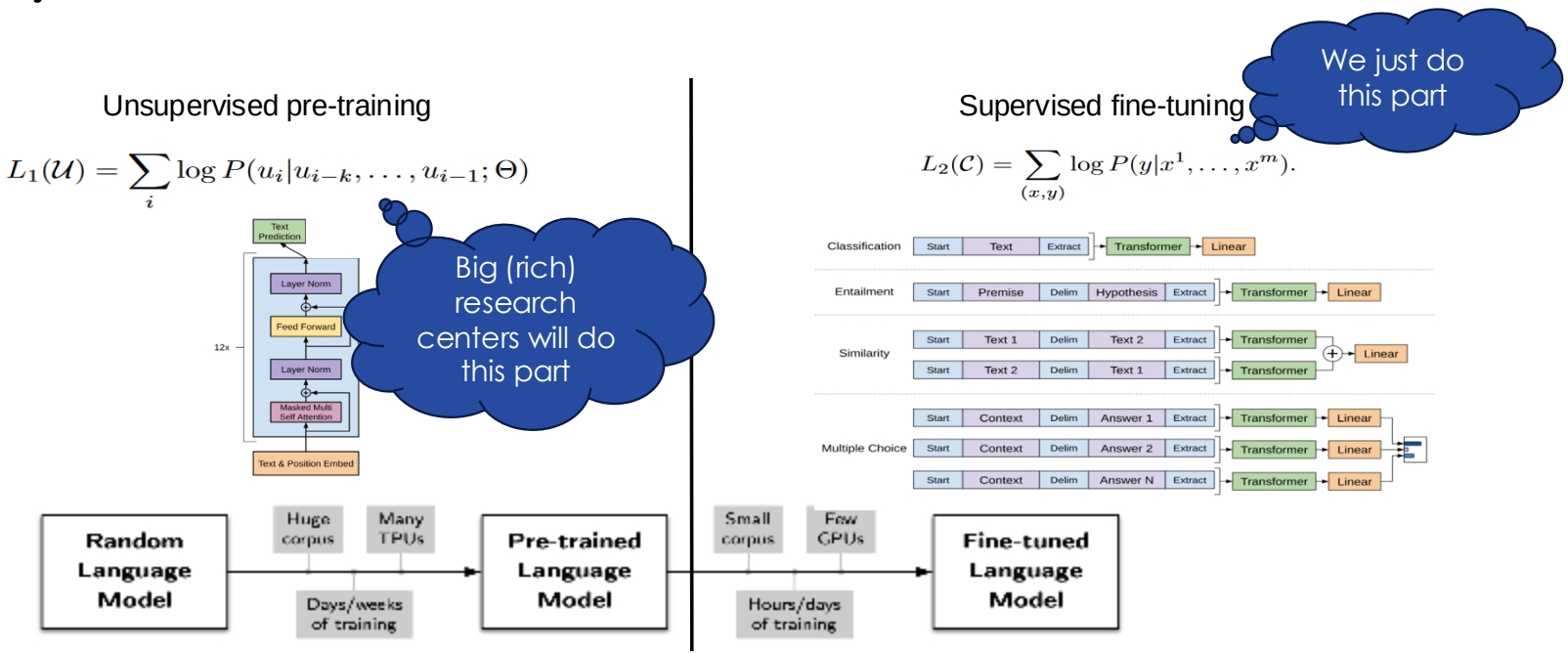
- Transformer models are larger models that highly scalable and can easily being training in parallel taking full advantage of accelerators like GPUs.
- The "downside" is that they are larger models that requires a lot of computational resources:
  - Some research groups use, for example, 2048, 1024 GPUs, 512 TPUs to produce their experiments.

FYI: For today standard of LLMs this is nothing



Contextualize Transformer Models

- Thankfully, we can reuse these large models and just make small adjustments for our tasks.

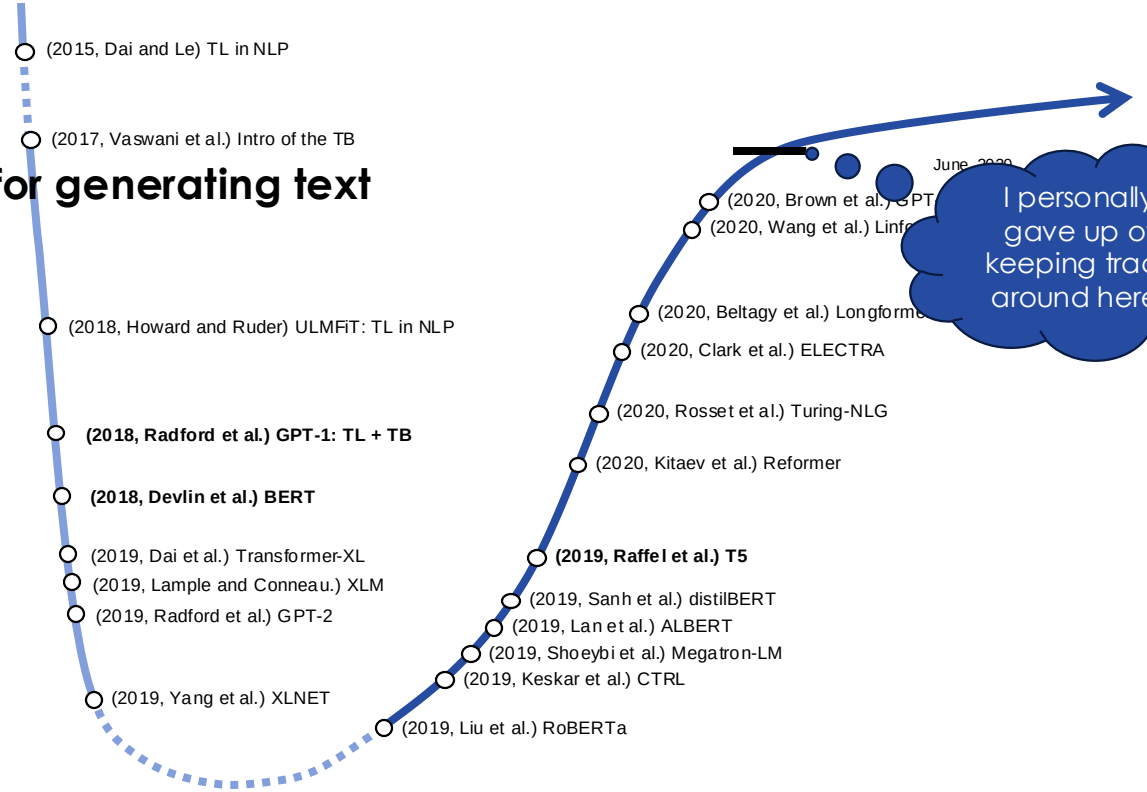


Transformers Architectures

- 3 main architectures:
  - Decoder only (LM)
    - GPT
    - Llama
  - Encoder-Decoder
    - T5
  - Encoder only (MLM)
    - BERT

Good for generating text

Incapable of generating text, but good for creating contextualized representation and for classification tasks.

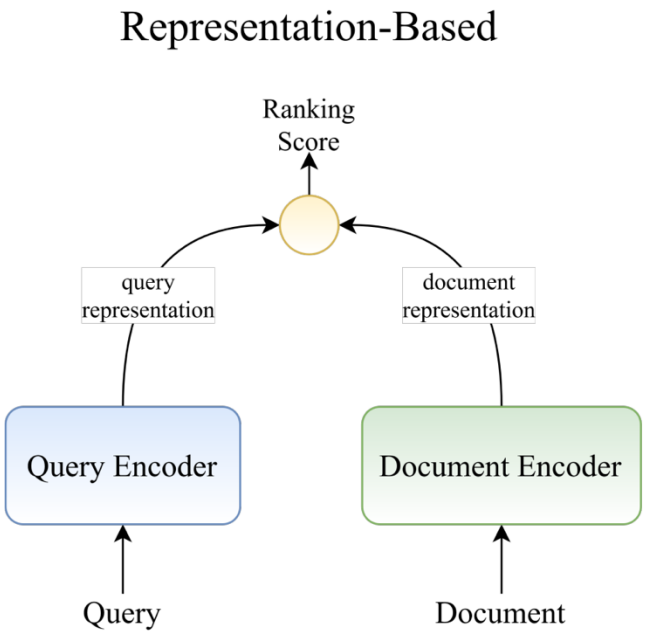


Transformer in IR

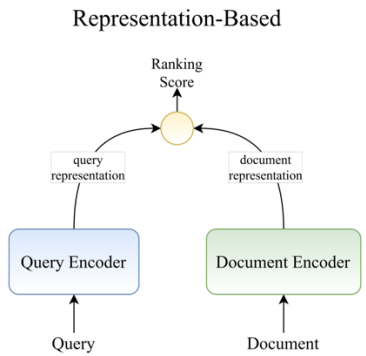
- In IR the encoder type is the mostly widely adopted transformer model to build neural information retrieval models.

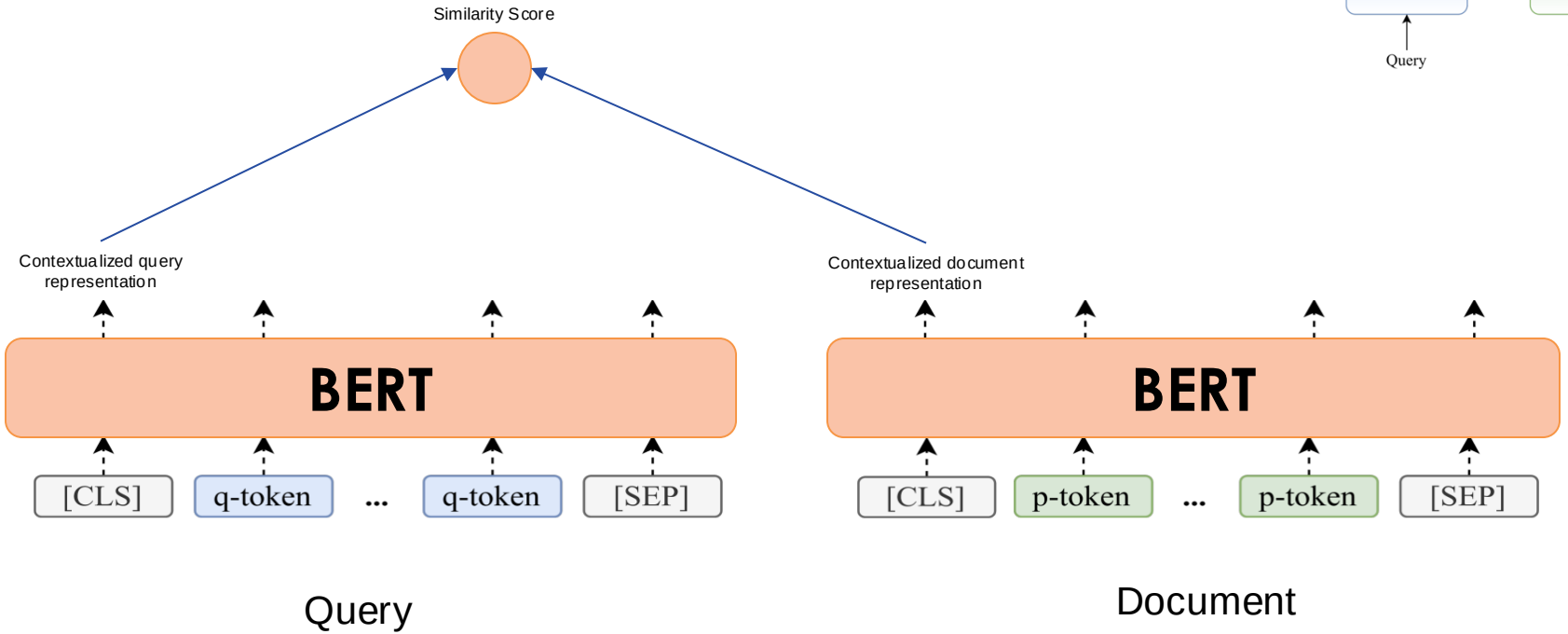
- Main idea:
  - For representation-based, lets leverage the more powerful contextualized embeddings to build query and document representation
  - For interaction-based, lets frame the ranking problem as a classification problem, e.g., is my query relevant to my document (yes or no)

Deep representation-based

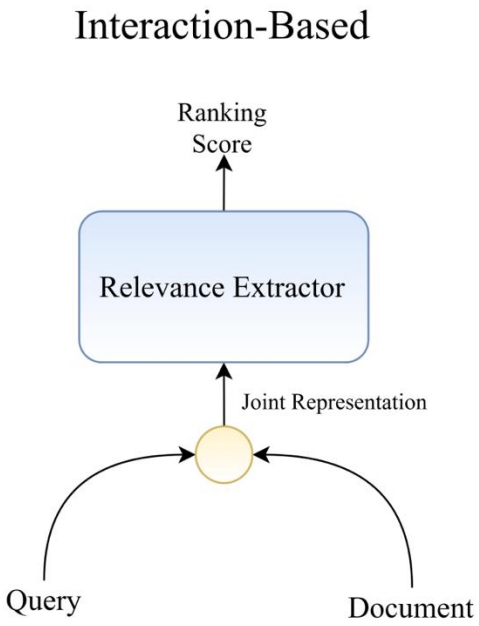


DPR (Dense Passage Retrieval)

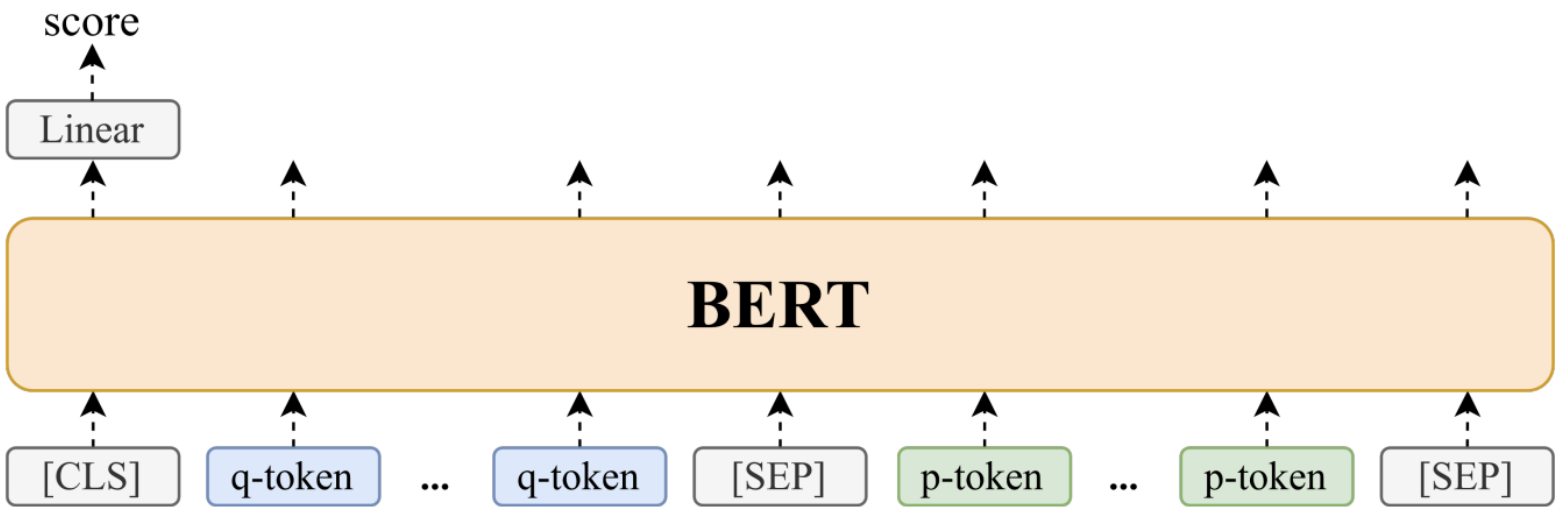




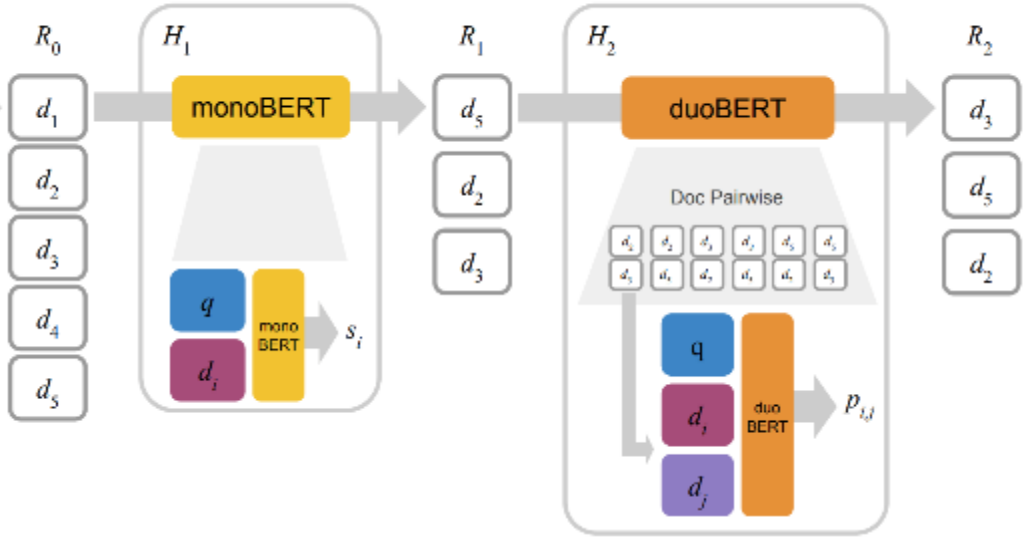
Deep interaction-based



Interaction-based models using Transformers



Mono and duo BERT



CEDR (Contextualized Embeddings for Document Ranking)

- Uses BERT to obtain the contextualized embeddings but then applies the previous neural IR models:
  - PACRR
  - KNRM
  - DRMM
  - DeepRank

Resources

- For transformer-based models having GPU necessary, for that:
  - Use colab from google the free tier offers GPU
  - Use Kaggle, the free tier offers 30h per week (more than enough)

- Pytorch: https://colab.research.google.com/drive/1KX9HUd--QupV_GdROvm3XA28ZyFCuscF?usp=sharing

- Embeddings: https://colab.research.google.com/drive/1OtLgsQD6wPNUcrRixOKAo_eXJL7rD2Px?usp=sharing