DynamicRetriever: A Pre-training Model-based IR System with Neither Sparse nor Dense Index

PRESENTED BY YUJIA ZHOU

RENMIN UNIVERSITY OF CHINA

Outline

Background

Motivation

Model architecture
- Query understanding module
- Document prediction module

Model training
- Vanilla model
- Dense-enhanced model

Experimental results

Discussion

Future work

Background

Sparse retrieval

- building an inverted index based on all candidate documents, where the key is different terms and the value is documents with this term.
- Retrieving relevant documents based on the matching between query terms and document terms

Dense retrieval

- Applying a neural network to encode each of the candidate documents into a dense vector and building a vectorized index.
- Embedding the issued query into the same latent space
- Computing the similarity between the query representation and document vectors to efficiently retrieve relevant document

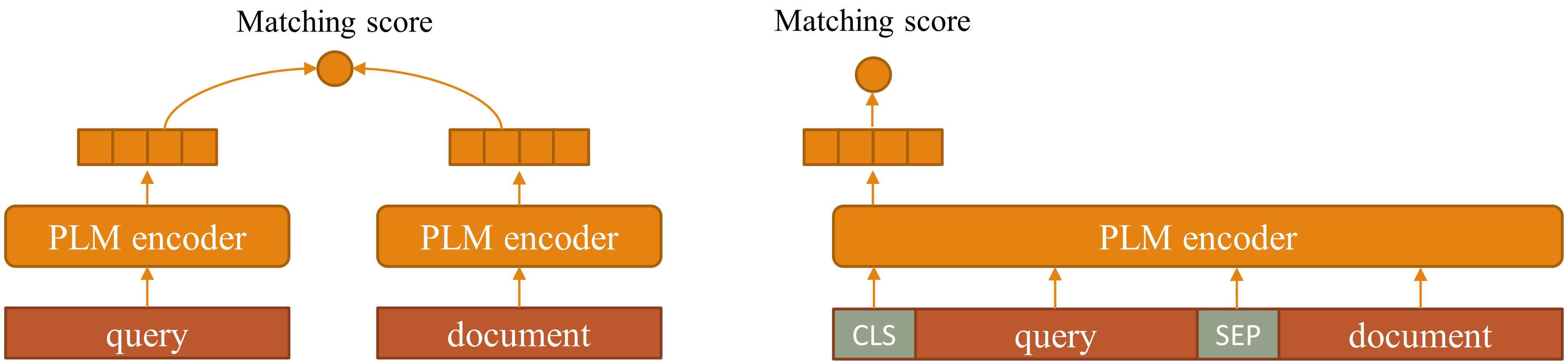
Background

Pre-trained language model (PLM)

- Considering contextual information to understand sentences

PLM for IR

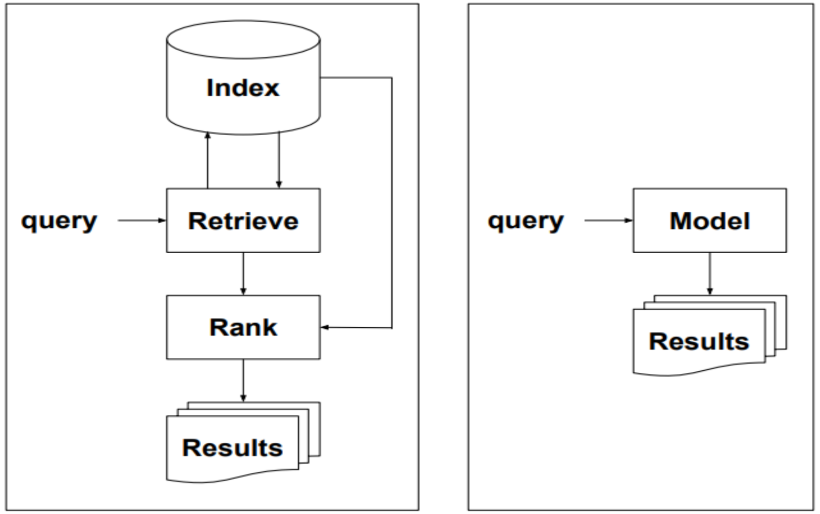
- Retrieval (representation-based matching model)
  - ANCE, HDCT, Col-BERT, STAR, …
- Re-ranking (Interaction-based matching model)
  - ICT, PROP, B-PROP, HARP, …



Representation-based matching

Interaction-based matching

Motivation



(a) Retrieve-then-rank (b) Unified retrieve-and-rank

Rethinking search: making domain experts out of dilettantes. SIGIR Forum 55(1): 13:1-13:27 (2021)

Index-based IR ----- Model-based IR

- From term-level features to document-level features (other meta-information, not only word, such as provenance, authorship, authoritativeness, polarity)

Thinking

Index-based IR systems: train the model with matching tasks

- Input: query, document
- Output: matching score

Model-based IR systems: train the model with generating tasks

- Input: query
- Output: document identifier (docid)

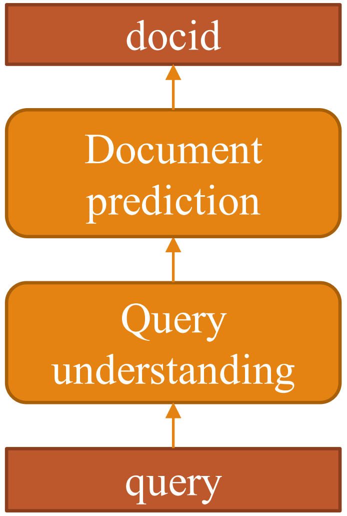
Thinking

Q:What's the workflow of model-based IR systems for ranking?

- Query understanding module
- Document prediction module

Q: What advantages the model-based IR system has for ranking?

- Static index --- dynamic index
  - It parametrizes the traditional static index, which allows the model's understanding of the document content to be a dynamic process that can be updated during training.
- Term-level features --- document-level features
  - It establishes a mapping from text to document identifier. Bridging the gap between terms and document identifiers can capture more document-level features which are essential for scoring the document.



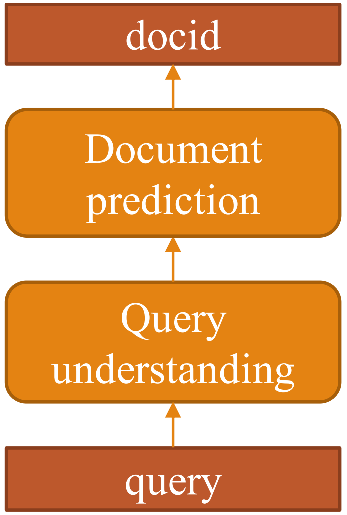
Workflow

Thinking

Q: What advantages the model-based IR system has for IR scenarios?

- Enhance multiple IR downstream tasks with one model
  - Document retrieval: query --- docid
  - Document summarization: docid --- text
  - Question answering: question --- answer
  - Related document retrieval: docid --- docid

- Zero- and Few-Shot Learning
- Response Generation
- Reasoning



Workflow

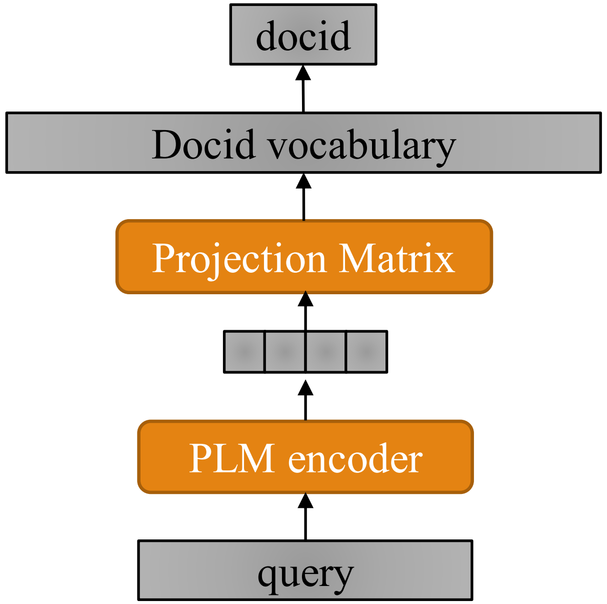
Model architecture

Query understanding

- Model queries in a fine-grained way to understand query intent
- Given a query $q = \{w_1, w_2 \ldots, w_n\}$
- $V^q = Transformer^{cls}([w_1, w_2 \ldots, w_n])$

Document prediction

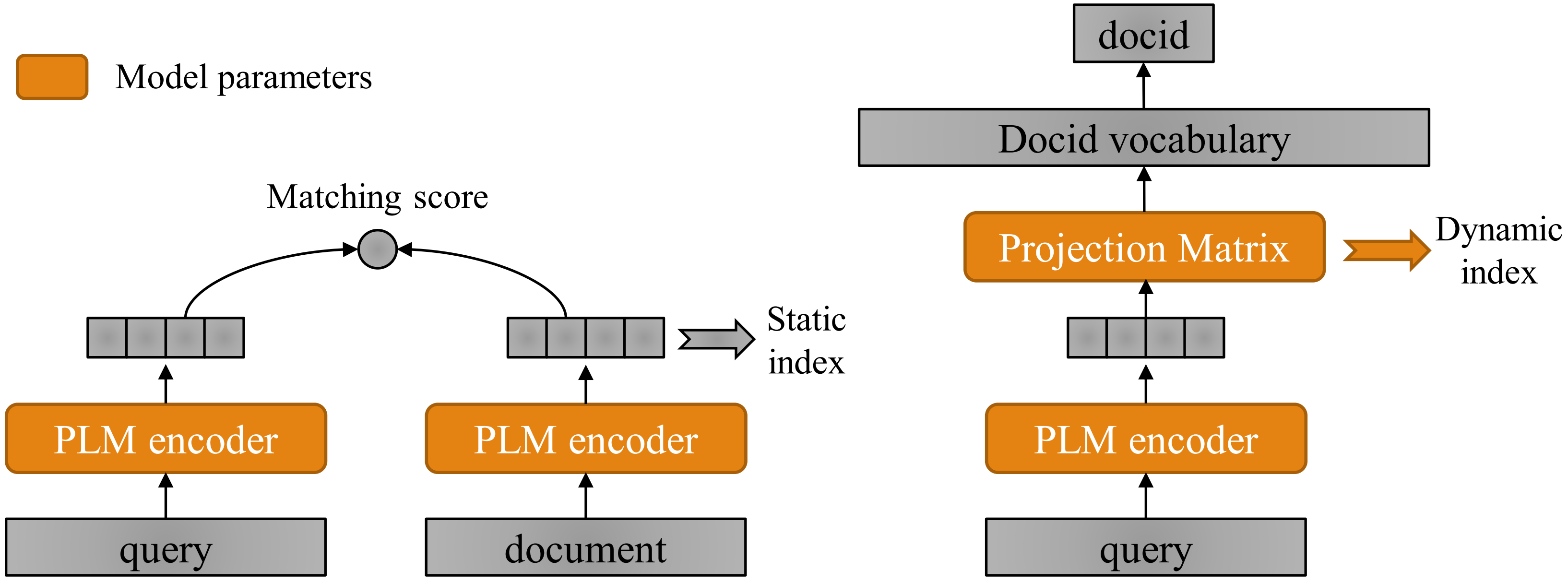
- Take the query representation to predict the most likely docid from the entire corpus *D*.
- $O^q = softmax\,(W_{doc}^T \cdot V^q)$, $W_{doc} \in R^{d_{model} \times |D|}$
- Retrieve the top-k documents by sorting the probability for the given query



DynamicRetriever

Model architecture

Comparison between dual encoder and DynamicRetriever



Dual Encoder

DynamicRetriever

Model training

Pre-trained Language models

- Pre-training on self-supervised data
  - learning the basic semantics of words and the semantic dependencies between words
- Fine-tuning on supervised data
  - enhancing the ability to handle specific tasks

Pre-trained model-based IR systems

- Pre-training
  - Memorizing the semantic of each docid in the model through multiple pre-training tasks
- Fine-tuning
  - Learning the matching relationships between queries and document identifiers
  - Capturing document-level meta information over term-level semantics

Vanilla model

Pre-training tasks

- Training with passage: (passage ----- docid)
  - Russia-Ukraine live news: Moscow launches full-scale invasion ----- (Russia-Ukraine live news, docA)
- Training with sampled terms: (sampled terms ----- docid)
  - Russia-Ukraine live news: Moscow launches full-scale invasion ----- (Russia-Ukraine Moscow invasion, docA)
- Training with n-gram: (N-gram ----- docids)
  - Russia-Ukraine live news: Moscow launches full-scale invasion ----- (live news, docA)
  - Russia Ukraine crisis live news: Russia declares war on Ukraine ----- (live news, docB)

Fine-tuning tasks

- Training with query-docid pairs: (query ----- docid)

Inferencing task

- Inferencing with query-docid pairs: (query ----- docid)

Experiments

Dataset: MS MARCO document ranking

Task: first-stage document retrieval

In order to control the amount of model parameters, we first tried 100k documents as a corpus to test the model performance.

Top 100k doc: rank all candidate documents based on click frequency and select the top 100k
- All docids are clicked on a query, and can be trained at the fine-tuning stage.

Random 100k doc: randomly sample 100k from the whole corpus
- 10% docids can be fine-tuned, 90% docids only can be learned during pre-training

Only consider the training queries and testing queries whose clicked documents exist in this set.

Results

model	Corpus size	Pre-train	Fine-tune	MRR
DynamicRetriever	Top 100k doc	passage	/	0.271
DynamicRetriever		passage + sampled terms	/	0.284
DynamicRetriever		passage	query-docid	0.557
DynamicRetriever		passage + sampled terms	query-docid	0.553
BM 25		/	/	0.238
Two-tower BERT		/	query-document	0.423

model	Corpus size	Pre-train	Fine-tune	MRR
DynamicRetriever	Random 100k doc	passage	/	0.505
DynamicRetriever		passage	query-docid	0.498
Two-tower BERT		/	query-document	0.546

problem

1、Lacking fine-tuning data

- Top 100k doc:
- 100k fine-tuning data, 500 inferencing data, overlap = 250
- Random 100k doc:
- 10k fine-tuning data, 150 inferencing data, overlap = 30

2、Poor generalizability of the model

- Since each document identifier is random, it is difficult for the model to infer the semantics of *doc123* by learning *doc456*.
- But for dense retrieval, if *doc123* and *doc456* have common words, they can be optimized together.

Dense-enhanced Model

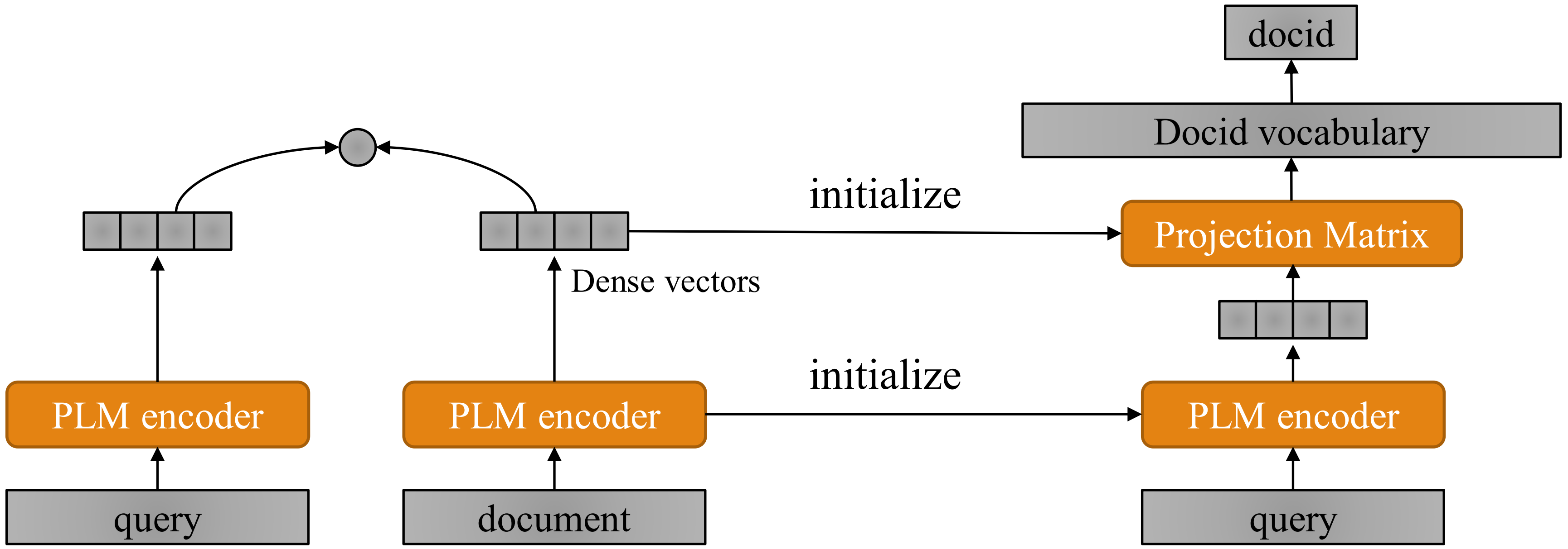
	DynamicRetriever	Dense retrieval model
Generalizability	poor	strong
Feature extraction	Document-level	Term-level
indexing	dynamic	static

Combining the advantages of dense retrieval and model-based IR system

- Strong generalizability
- Document-level features + term-level features
- Dynamic indexing

Integrating the advantages of dense retrieval model into our framework

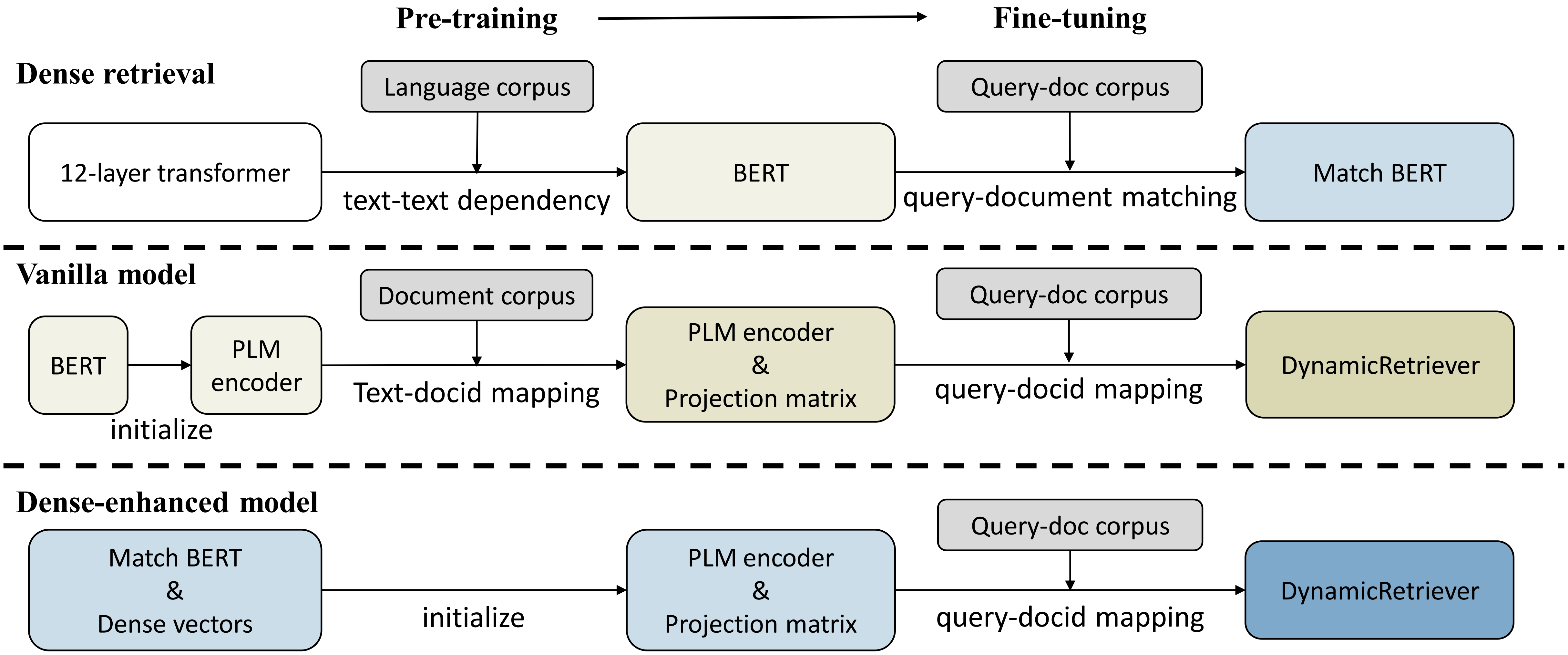
Dense-enhanced Model



Three steps to build dense-enhanced model

- Fine-tuning the two-tower BERT with query-document pairs
- Generating dense vectors to initialize the model parameters (encode the textual information into the model)
- Fine-tuning DynamicRetriever with query-docid pairs (focus on document-level features)

Framework



Pre-training
Fine-tuning
Dense retrieval
Language corpus
Query-doc corpus
12-layer transformer
text-text dependency
BERT
query-document matching
Match BERT
Vanilla model
Document corpus
Query-doc corpus
BERT
PLM encoder
initialize
Text-docid mapping
PLM encoder & Projection matrix
query-docid mapping
DynamicRetriever
Dense-enhanced model
Query-doc corpus
Match BERT & Dense vectors
initialize
PLM encoder & Projection matrix
query-docid mapping
DynamicRetriever

Results

Model	Corpus size	MRR
Two-tower BERT	Random 100k doc	0.5463
Vanilla model		0.4985
Dense-enhanced model		0.6443
Two-tower BERT	Top 100k doc	0.4238
Vanilla model		0.5576
Dense-enhanced model		0.5728

Vanilla model

- uses the pretraining tasks we design to learn the parameters for indexing documents

Dense-enhanced model

- uses finetuned two-tower BERT model to generate the document representations for initializing this part of the parameters.

Thinking

Users not clicking on a document is not just semantically irrelevant, but dislikes an author or a news site.

Query: Russia Ukraine

Document A:

- Russia-Ukraine live news: Moscow launches full-scale invasion (from BBC NEWS) √

Document B:

- Russia Ukraine crisis live news: Russia declares war on Ukraine (from WION) ×

Dense retrieval: Relying on the understanding of terms

- Russia Ukraine --- BBC NEWS √ Russia Ukraine --- WION ×

Dense-enhanced model:

- Russia Ukraine --- Document A √ Russia Ukraine --- Document B ×

Results

Model	Top 100k	Top 200k	Random 100k	Random 200k
Two-tower BERT	0.4238	0.401	0.5463	0.440
Vanilla model	0.5576	0.461	0.4985	/
Dense-enhanced model	0.5728	0.522	0.6443	0.495

Vanilla model

- With the increase of document corpus size, the difficulty of distinguishing between documents increases.

Dense-enhanced model

- Using dense vectors to alleviate this problem

Discussion

The number of documents increases ----- model parameters increase

How to scale the model to larger corpora?

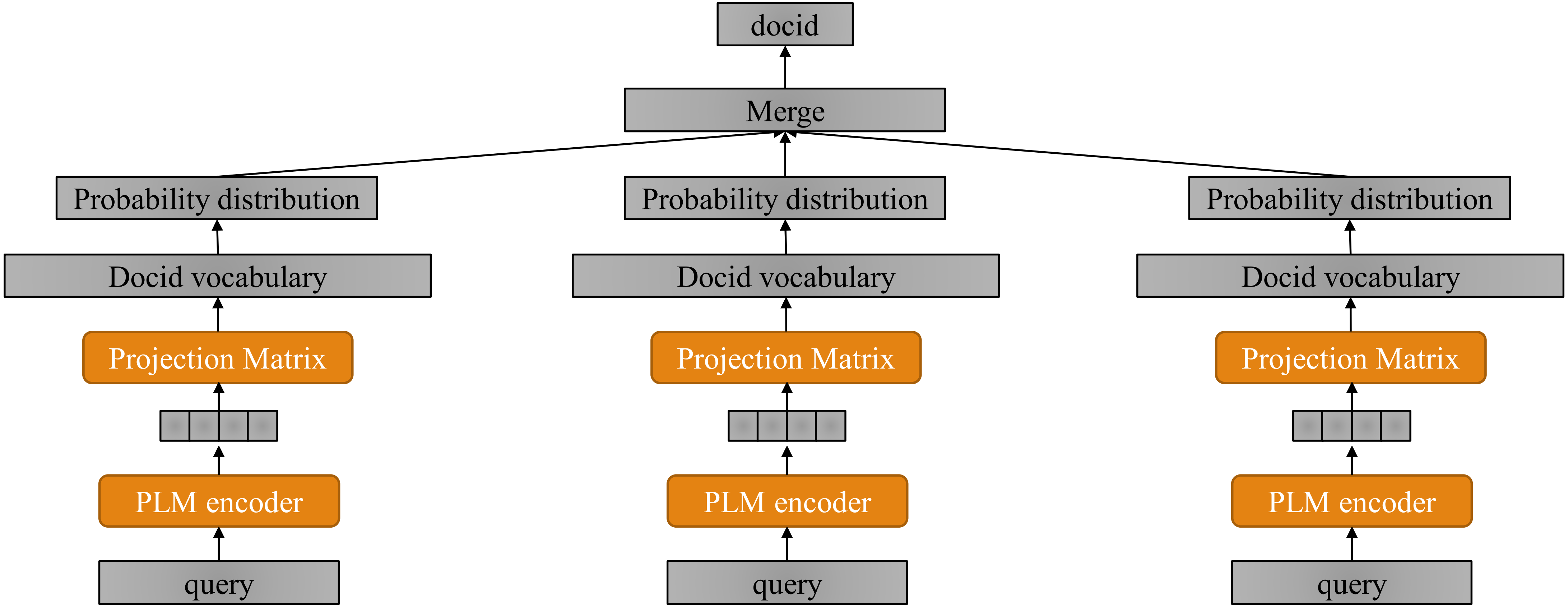
Distributed model

- We can train multiple sub-models distributedly, and then fuse their predictions to get the whole document ranking list.

Hierarchical model

- We can organize and categorize documents in a structured way, and then encode them into strings as docids by category.
- Represent docid by two or more parts, and predict them one by one, like a seq2seq process.

Distributed model



Model 1
doc1-doc10

Model 2
doc11-doc20

Model 3
doc21-doc30

Results

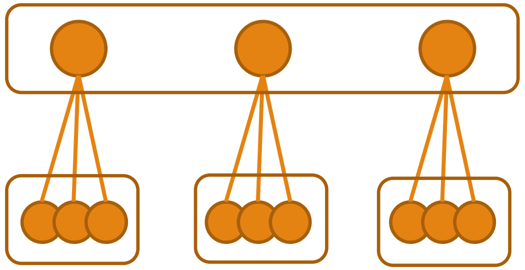
model	Corpus size	Pre-train	Fine-tune	MRR
Two-tower BERT	Top 300k doc	/	Query-document	0.417
3 Vanilla models		passage	/	0.239
3 Vanilla models		passage	Query-docid	0.453
Two-tower BERT	All		Query-document	0.282
30 Vanilla models		passage	/	0.130
30 Vanilla models		passage	Query-docid	0.118
30 Dense-enhanced models		Dense vectors	Query-docid	0.188

The ranking results after merging decrease sharply

- The scale of document scores between different sub-models trained independently are not consistent
- More suitable merge function is needed
- Add some common docids into different sub-models, to scale the space of each sub-model to the same level

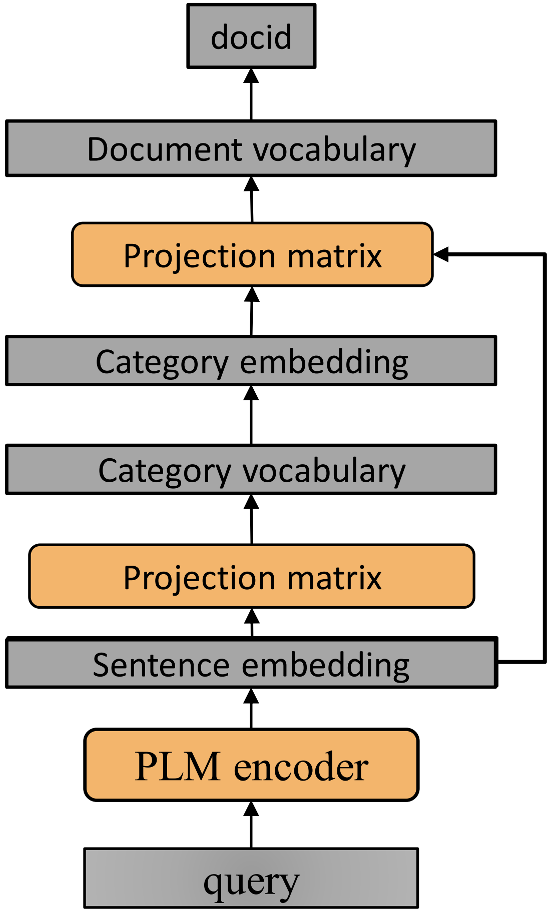
Hierarchical model

If we can divide the 9 documents equally into three categories, we can only use 3 ids to represent them. (11 12 13 21 22 23 31 32 33)



How to classify documents?

- Random
- Domain classifier
- Semantic classifier
- Hyperlink graph



Future work

Now:

Query ----- docid: search information

Future:

Text ----- Docid: adding references

Docid ----- Docid: finding related documents

Text ----- Text: question answering

Thinking: If the model can do all the above tasks well, do we still need to return the document list for the user to choose a relevant one?

Future work

WebBrain

- Now that the model has learned about all documents, can the model answer the query directly with references? (like Wikipedia)

Donald Trump

From Wikipedia, the free encyclopedia

For other uses, see Donald Trump (disambiguation).

Donald John Trump (born June 14, 1946) is an American politician, media personality, and businessman who served as the 45th president of the United States from 2017 to 2021.

Born and raised in Queens, New York City, Trump graduated from the Wharton School of the University of Pennsylvania with a bachelor's degree in 1968. He became president of his father Fred Trump's real estate business in 1971 and renamed it The Trump Organization. Trump expanded the company's operations to building and renovating skyscrapers, hotels, casinos, and golf courses. He later started various side ventures, mostly by licensing his name. From 2004 to 2015, he co-produced and hosted the reality television series *The Apprentice*. Trump and his businesses have been involved in more than 4,000 state and federal legal actions, including six bankruptcies.

Trump's political positions have been described as populist, protectionist, isolationist, and nationalist. He entered the 2016 presidential race as a Republican and was elected in an upset victory over Democratic nominee Hillary Clinton while losing the popular vote,[a] becoming the first U.S. president with no prior military or government service. The 2017–2019 special counsel investigation led by Robert Mueller established that Russia interfered in the 2016 election to benefit the Trump campaign, but not that members of the Trump campaign conspired or coordinated with Russian election interference activities. Trump's election and policies sparked numerous protests. Trump made many false and misleading statements during his campaigns and presidency, to a degree unprecedented in American politics, and promoted conspiracy theories. Many of his comments and actions have been characterized as racially charged or racist, and many as misogynistic.

Donald Trump

Official portrait, 2017

Thanks For Your Attention

Arxiv paper: coming soon …

E-mail: zhouyujia@ruc.edu.cn