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

Model Training

- Vanilla model
- OverDense model

Discussion

Experiments

- Experimental settings
- Experimental results

Future Work

Background

Sparse retrieval

- Building an inverted index, the key -- different terms and the value -- documents with this term.
- Retrieving relevant documents based on the matching between query terms and document terms
- Suffering from word mismatch

Dense retrieval

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

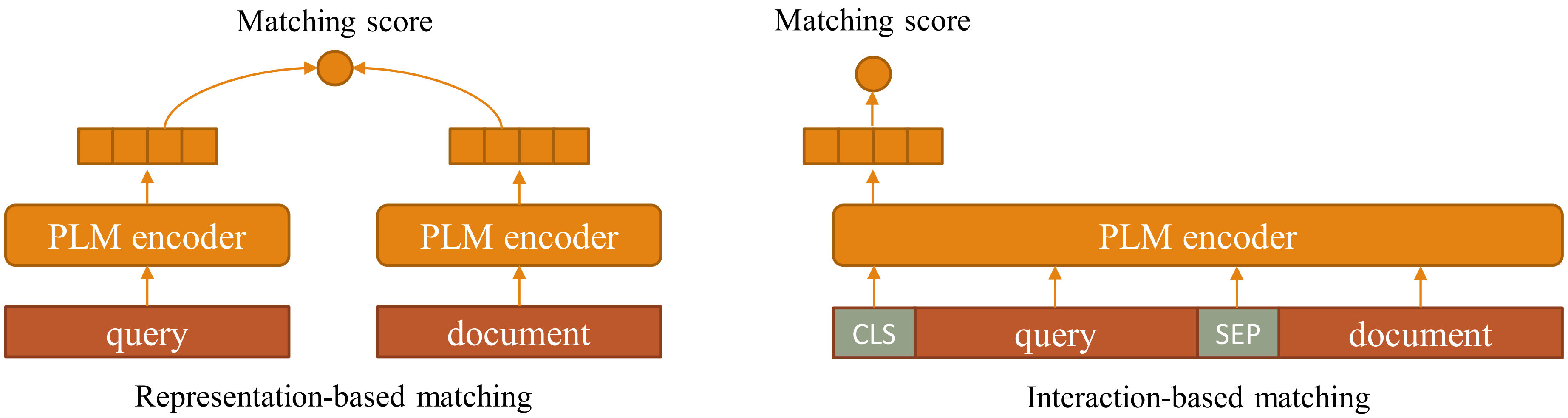
Background

Pre-trained language model (PLM)

- Considering contextual information to understand sentences

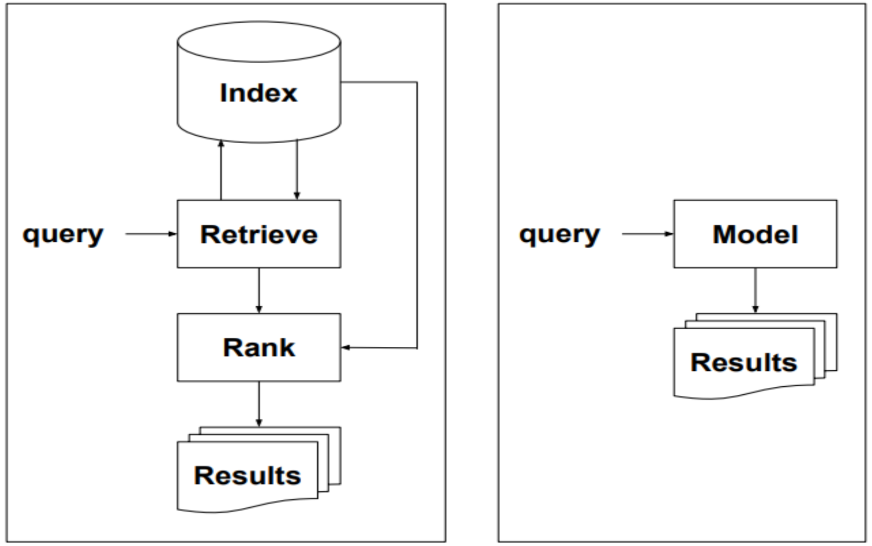
PLM for IR

- Retrieval (representation-based matching model)
- Re-ranking (Interaction-based matching model)

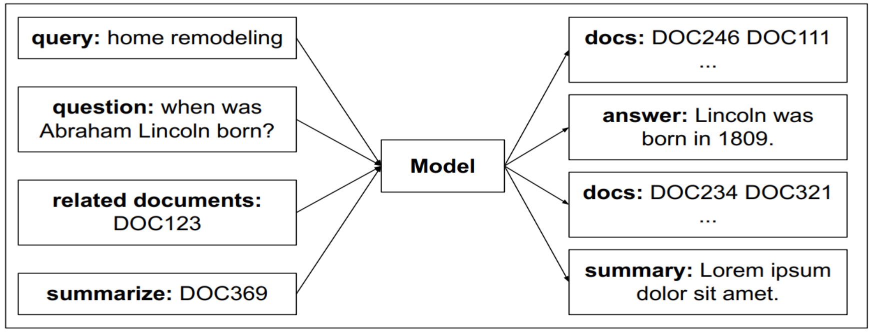


Motivation

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



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


query: home remodeling
question: when was Abraham Lincoln born?
related documents: DOC123
summarize: DOC369
Model
docs: DOC246 DOC111 ...
answer: Lincoln was born in 1809.
docs: DOC234 DOC321 ...
summary: Lorem ipsum dolor sit amet.

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

- Encoding both the semantic knowledge and document identifiers into the model, as well as document-level features such as authority.
- Converting the matching task between query and document representations to direct generation task of document identifiers.
- An end-to-end model ready for various IR tasks, including document retrieval, question answering, summarization, etc.

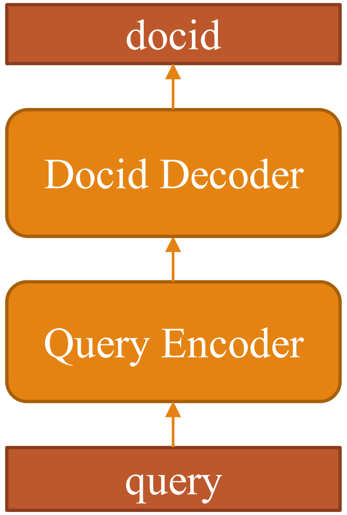
Motivation

A preliminary exploration on document retrieval -- DynamicRetriever

- Docid Decoder, with a projection matrix for all document identifiers.

Advantages of the model-based IR system for document retrieval

- 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
  - Bridging the gap between terms and document identifiers can capture more document-level features which are essential for scoring the document.



Workflow

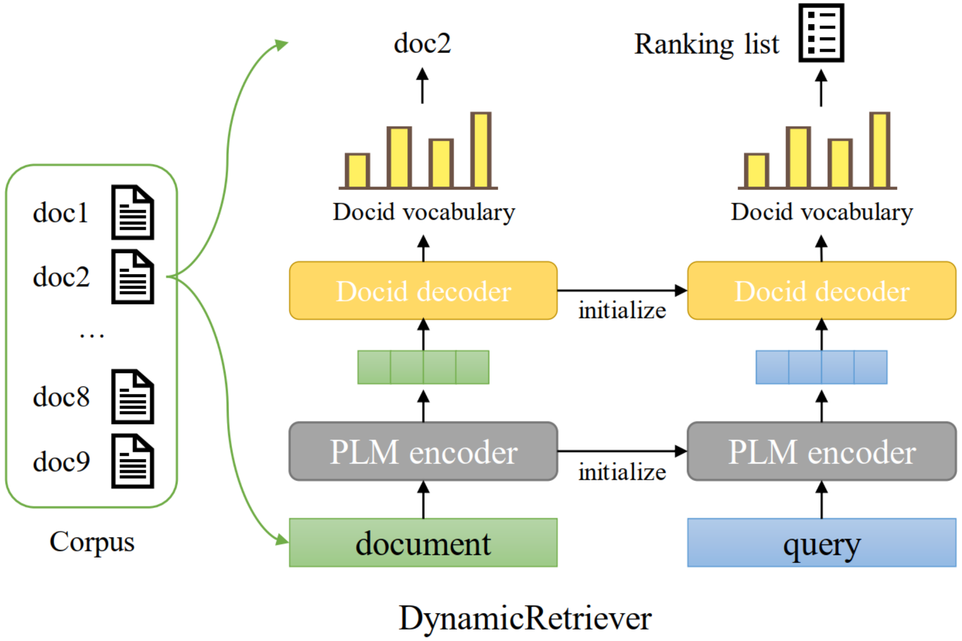
Model Architecture

PLM encoder

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

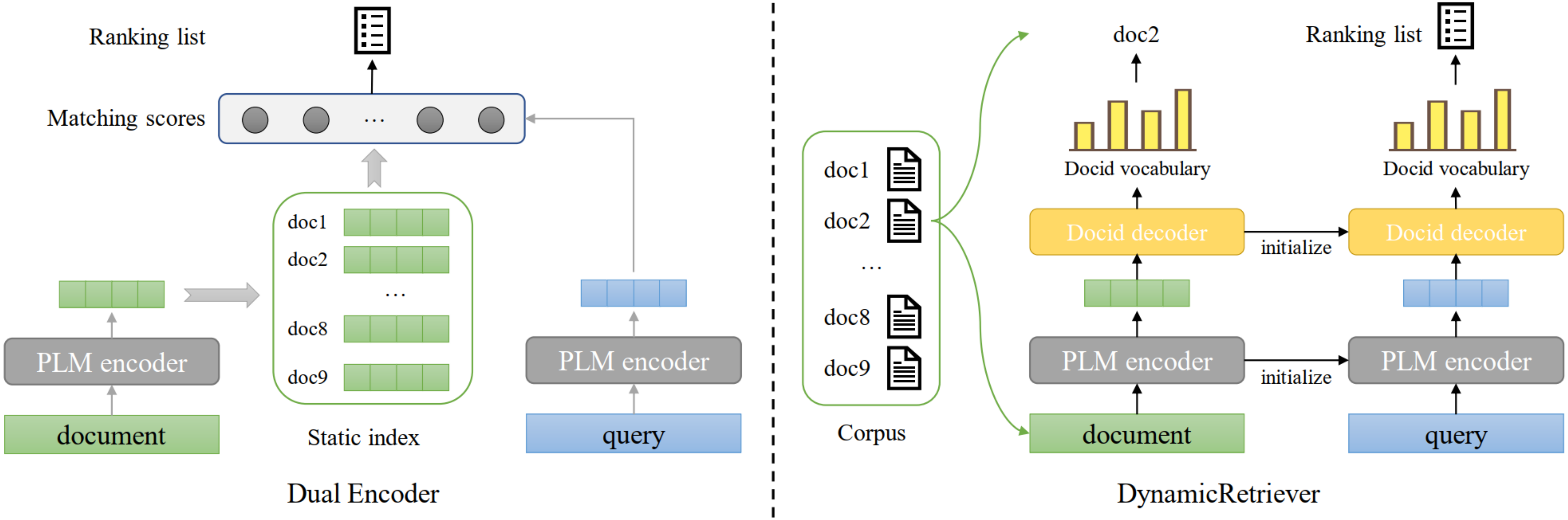
Docid decoder

- Take the query representation to predict the most relevant 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



Model Architecture

Comparison between dual encoder and 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

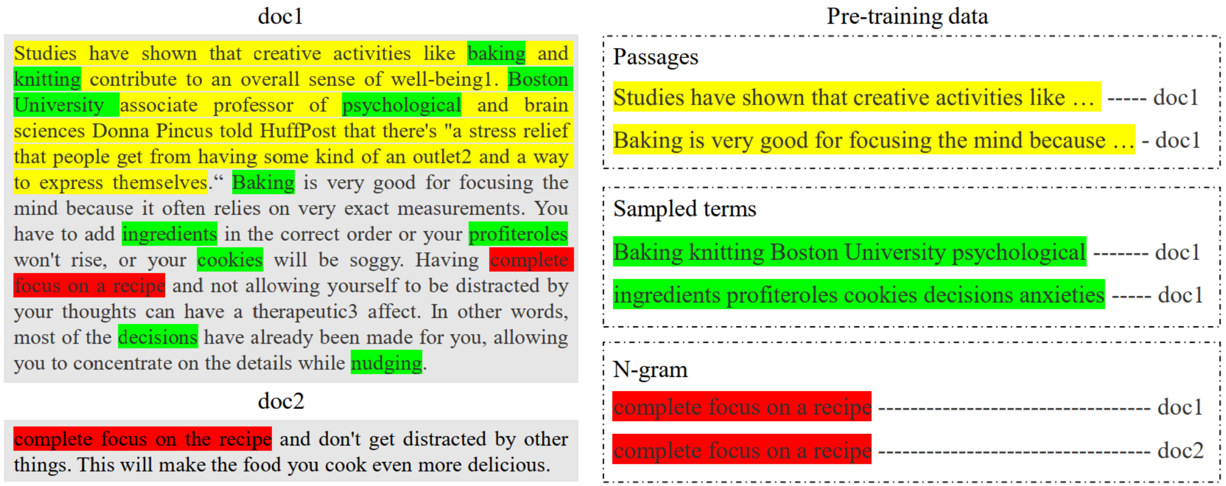
Pre-trained model-based IR systems —— Encoding document identifiers into the model

- Pre-training
  - Memorizing the content of each document identifier 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)
  - Passage-level semantic information
- Training with sampled terms: (term set ----- docid)
  - Important words to reflect the basic document content
- Training with n-gram: (N-gram ----- docids)
  - Semantic relatedness among multiple document identifiers



doc1

Studies have shown that creative activities like baking and knitting contribute to an overall sense of well-being1. Boston University associate professor of psychological and brain sciences Donna Pincus told HuffPost that there's "a stress relief that people get from having some kind of an outlet2 and a way to express themselves.“ Baking is very good for focusing the mind because it often relies on very exact measurements. You have to add ingredients in the correct order or your profiteroles won't rise, or your cookies will be soggy. Having complete focus on a recipe and not allowing yourself to be distracted by your thoughts can have a therapeutic3 affect. In other words, most of the decisions have already been made for you, allowing you to concentrate on the details while nudging.

doc2

complete focus on the recipe and don't get distracted by other things. This will make the food you cook even more delicious.

Pre-training data

Passages

Studies have shown that creative activities like … ----- doc1

Baking is very good for focusing the mind because … - doc1

Sampled terms

Baking knitting Boston University psychological ------- doc1

ingredients profiteroles cookies decisions anxieties ----- doc1

N-gram

complete focus on a recipe ---------------------------------- doc1

complete focus on a recipe ---------------------------------- doc2

Fine-tuning tasks

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

Inferencing task

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

Vanilla Model

Two shortcomings of the Vanilla model：

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 dependent, there may not be relatedness between two document identifiers even they share similar content.
- But for dense retrieval, if two documents have similar texts, their representation vectors will be closed.

OverDense Model

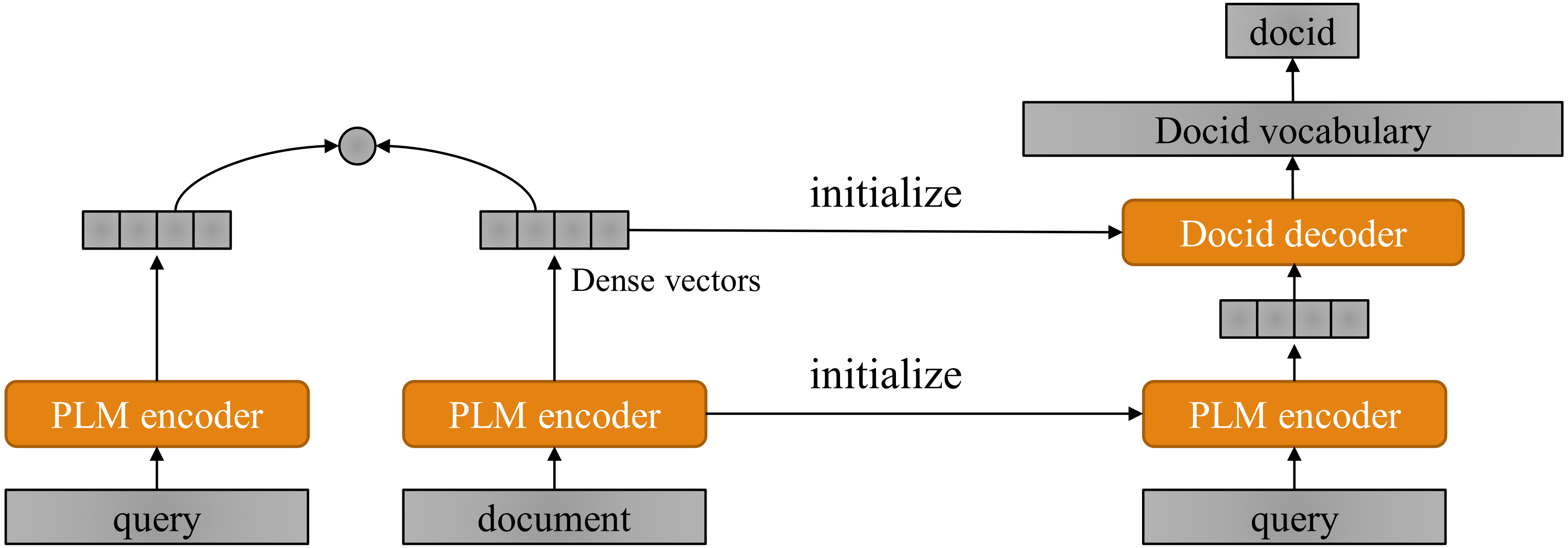
	DynamicRetriever	Dense retrieval model
Generalizability	poor	strong
Feature extraction	Document-level	Term-level
indexing	dynamic	static

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 -- OverDense Model

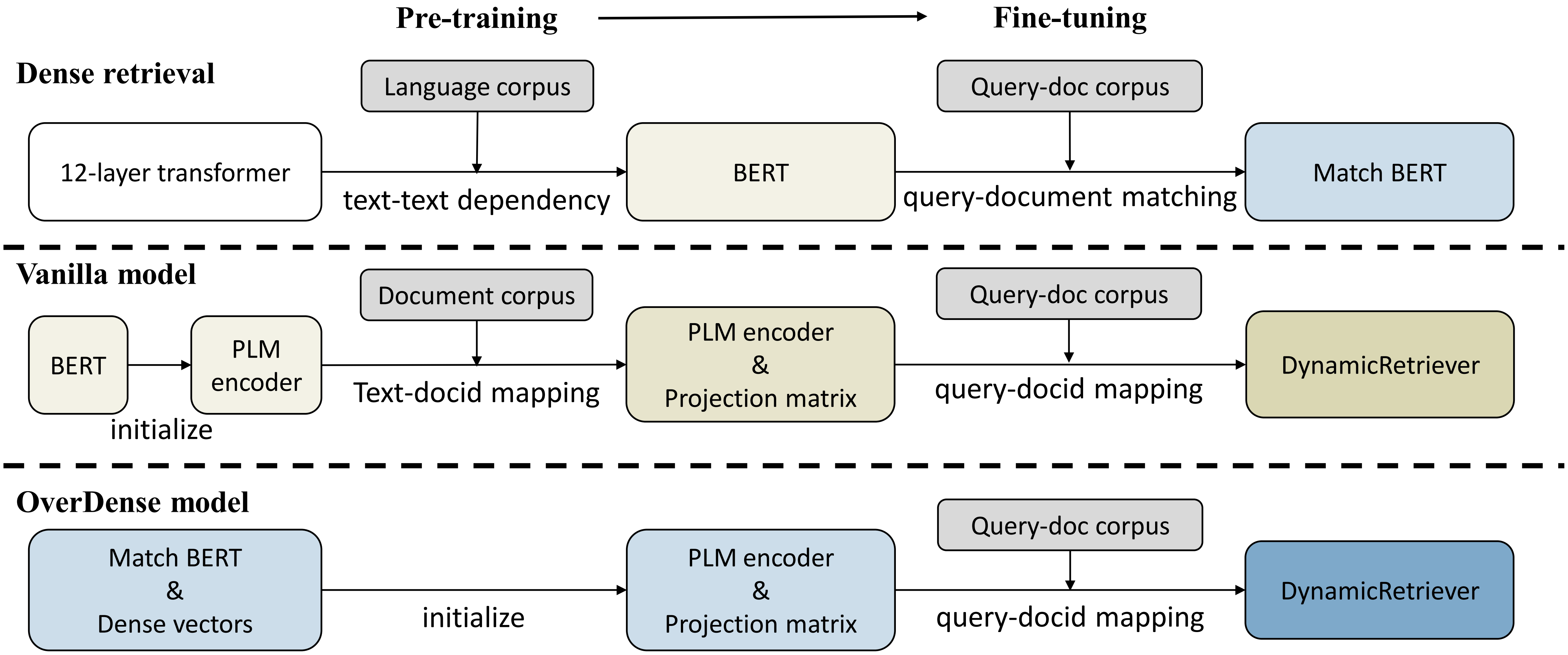
OverDense Model



Three steps to build OverDense 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)

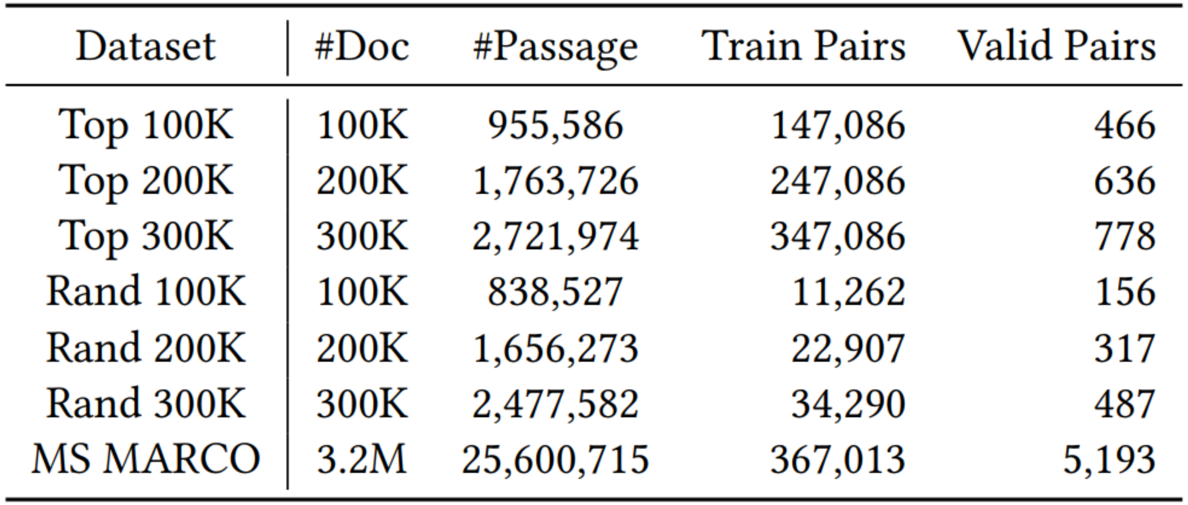
Framework


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
OverDense model
Query-doc corpus
Match BERT & Dense vectors
initialize
PLM encoder & Projection matrix
query-docid mapping
DynamicRetriever

Experimental Settings

Dataset: MS MARCO document ranking, various subsets with different scales and distributions.

- Top: documents with most clicks
- Rand: randomly sampled documents



Dataset	#Doc	#Passage	Train Pairs	Valid Pairs
Top 100K	100K	955,586	147,086	466
Top 200K	200K	1,763,726	247,086	636
Top 300K	300K	2,721,974	347,086	778
Rand 100K	100K	838,527	11,262	156
Rand 200K	200K	1,656,273	22,907	317
Rand 300K	300K	2,477,582	34,290	487
MS MARCO	3.2M	25,600,715	367,013	5,193

Task: first-stage document retrieval

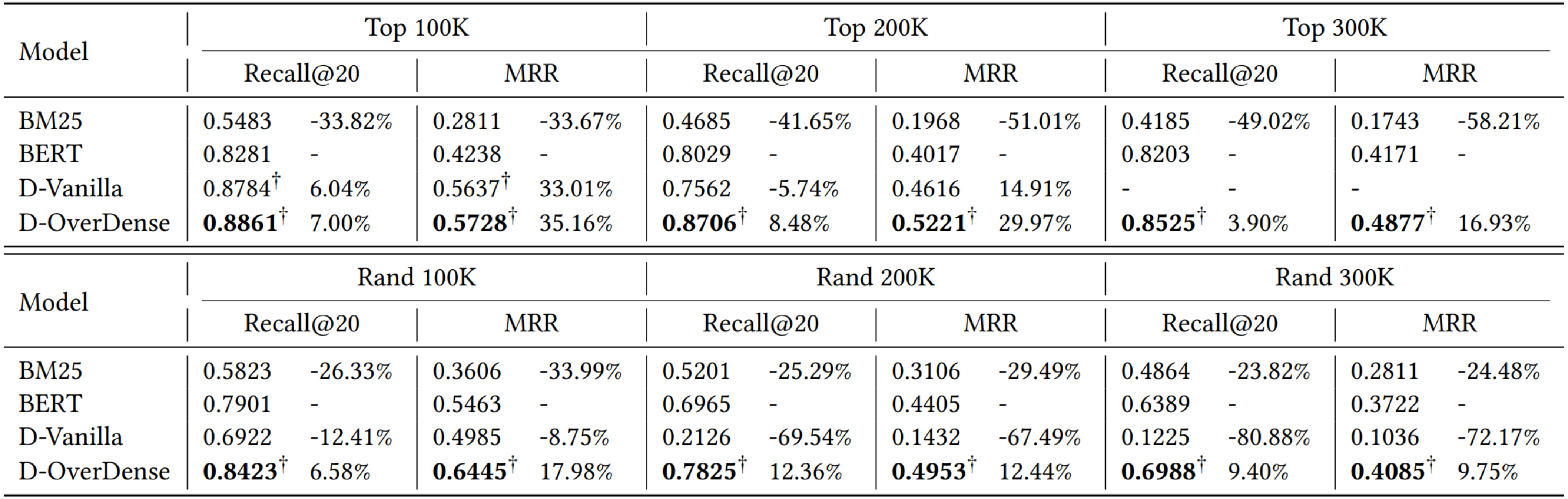
Evaluation Metrics:

- Recall@k for document retrieval
- MRR for ranking

Baselines:

- Sparse retrieval: BM25
- Dense retrieval: two-tower BERT

Overall Performance

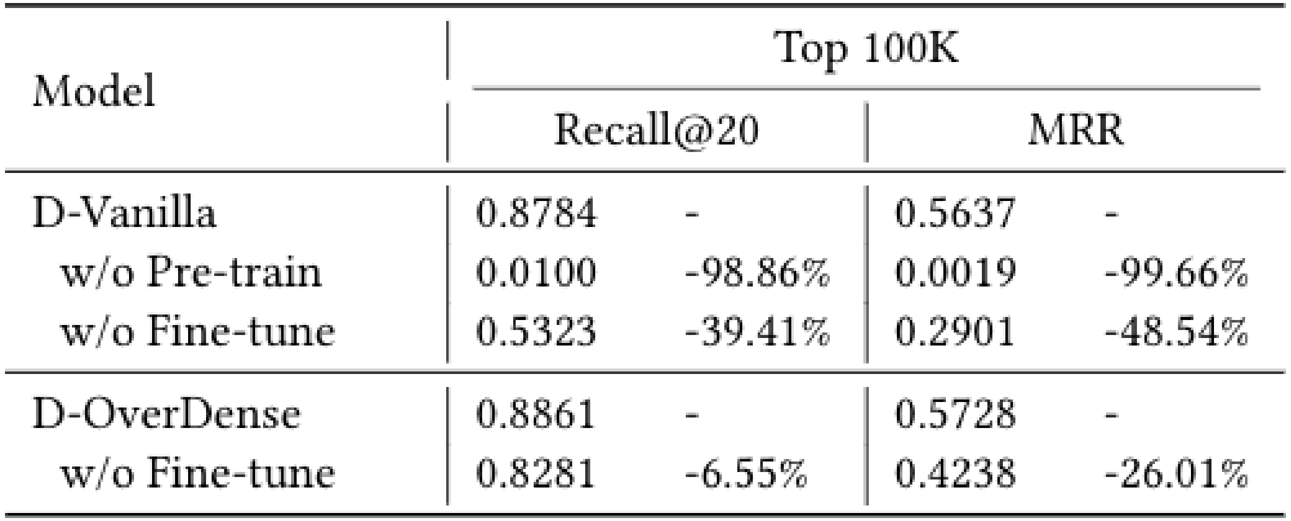


Model	Top 100K				Top 200K				Top 300K			
	Recall@20		MRR		Recall@20		MRR		Recall@20		MRR	
BM25	0.5483	-33.82%	0.2811	-33.67%	0.4685	-41.65%	0.1968	-51.01%	0.4185	-49.02%	0.1743	-58.21%
BERT	0.8281	-	0.4238	-	0.8029	-	0.4017	-	0.8203	-	0.4171	-
D-Vanilla	0.8784†	6.04%	0.5637†	33.01%	0.7562	-5.74%	0.4616	14.91%	-	-	-	
D-OverDense	**0.8861**†	7.00%	**0.5728**†	35.16%	**0.8706**†	8.48%	**0.5221**†	29.97%	**0.8525**†	3.90%	**0.4877**†	16.93%

Model	Rand 100K				Rand 200K				Rand 300K			
	Recall@20		MRR		Recall@20		MRR		Recall@20		MRR	
BM25	0.5823	-26.33%	0.3606	-33.99%	0.5201	-25.29%	0.3106	-29.49%	0.4864	-23.82%	0.2811	-24.48%
BERT	0.7901	-	0.5463	-	0.6965	-	0.4405	-	0.6389	-	0.3722	-
D-Vanilla	0.6922	-12.41%	0.4985	-8.75%	0.2126	-69.54%	0.1432	-67.49%	0.1225	-80.88%	0.1036	-72.17%
D-OverDense	**0.8423**†	6.58%	**0.6445**†	17.98%	**0.7825**†	12.36%	**0.4953**†	12.44%	**0.6988**†	9.40%	**0.4085**†	9.75%

- DynamicRetriever outperforms both the BM25 and two-tower BERT model, especially OverDense model.
- Vanilla model performs greatly on small subsets, but fails on larger datasets.
- OverDense model shows consistently better results than all baselines as the data scale increases.

Ablation Study



Model	Top 100K			
	Recall@20		MRR	
D-Vanilla	0.8784	-	0.5637	-
w/o Pre-train	0.0100	-98.86%	0.0019	-99.66%
w/o Fine-tune	0.5323	-39.41%	0.2901	-48.54%
D-OverDense	0.8861	-	0.5728	-
w/o Fine-tune	0.8281	-6.55%	0.4238	-26.01%

- Both the pre-training and fine-tuning tasks contribute to the model’s performance

Discussion

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

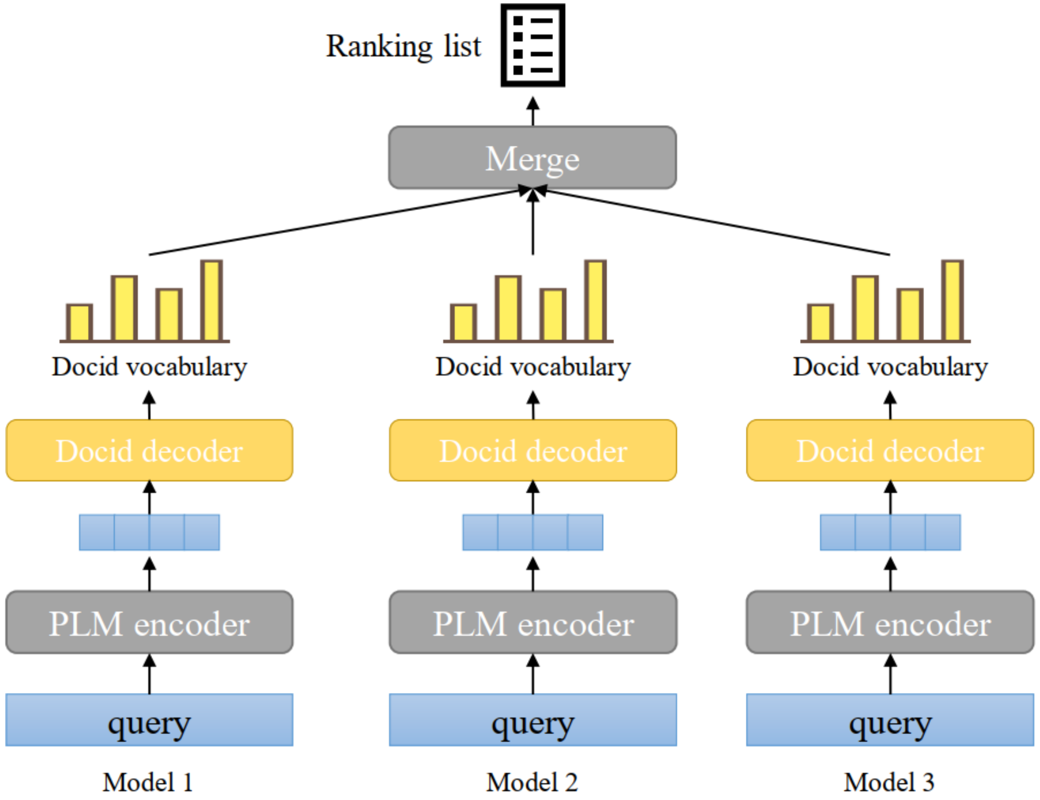
How to scale the model to larger corpora?

Two potential solutions:

- Distributed model
- Hierarchical model

Distributed model

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

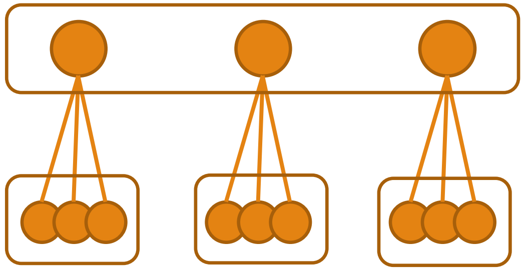


Model	MS MARCO		
	Recall@1	Recall@20	MRR
Each Group	0.5232	0.8423	0.6445
BERT	0.1665	0.6321	0.2817
Distributed Model	0.1011	0.4724	0.1895

- Different sub-models are trained independently, thus their document scores are not consistent to be directly merged.

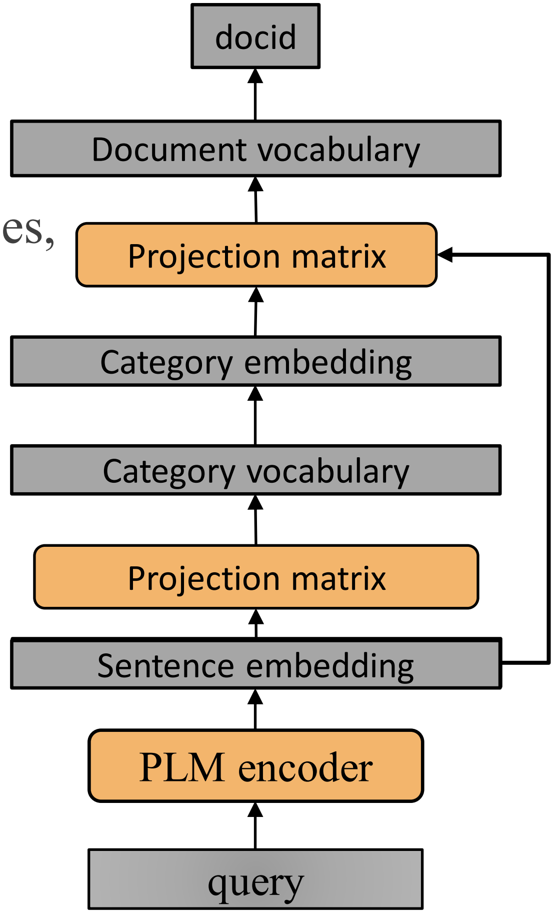
Hierarchical model

- Categorize and organize documents in a structured way, encode them into strings as docids by category, and then decode a docid one by on, like a sequence generation process.
- For example, 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:

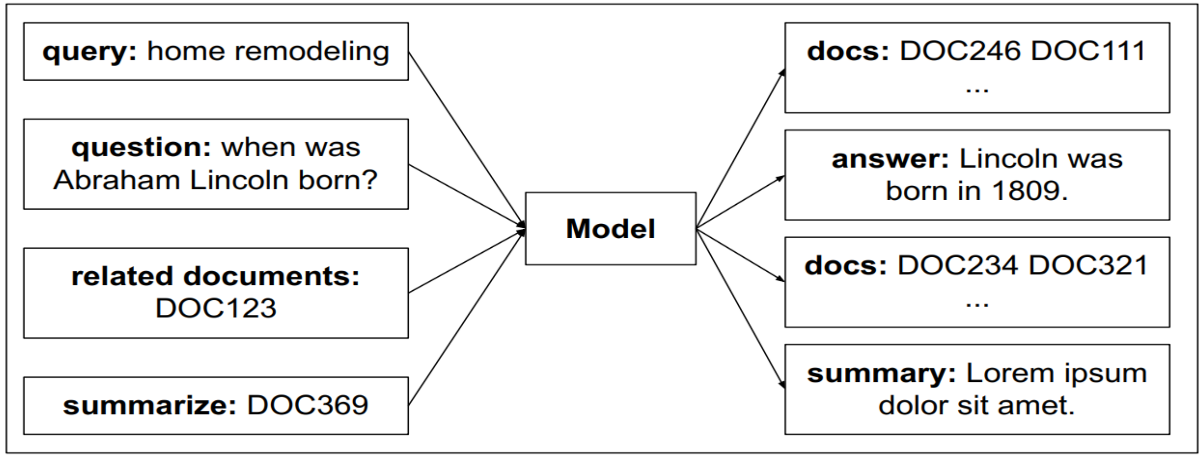
Query ----- docid: search information

Future:

Text ----- Docid: adding references

Docid ----- Docid: finding related documents

Text ----- Text: question answering

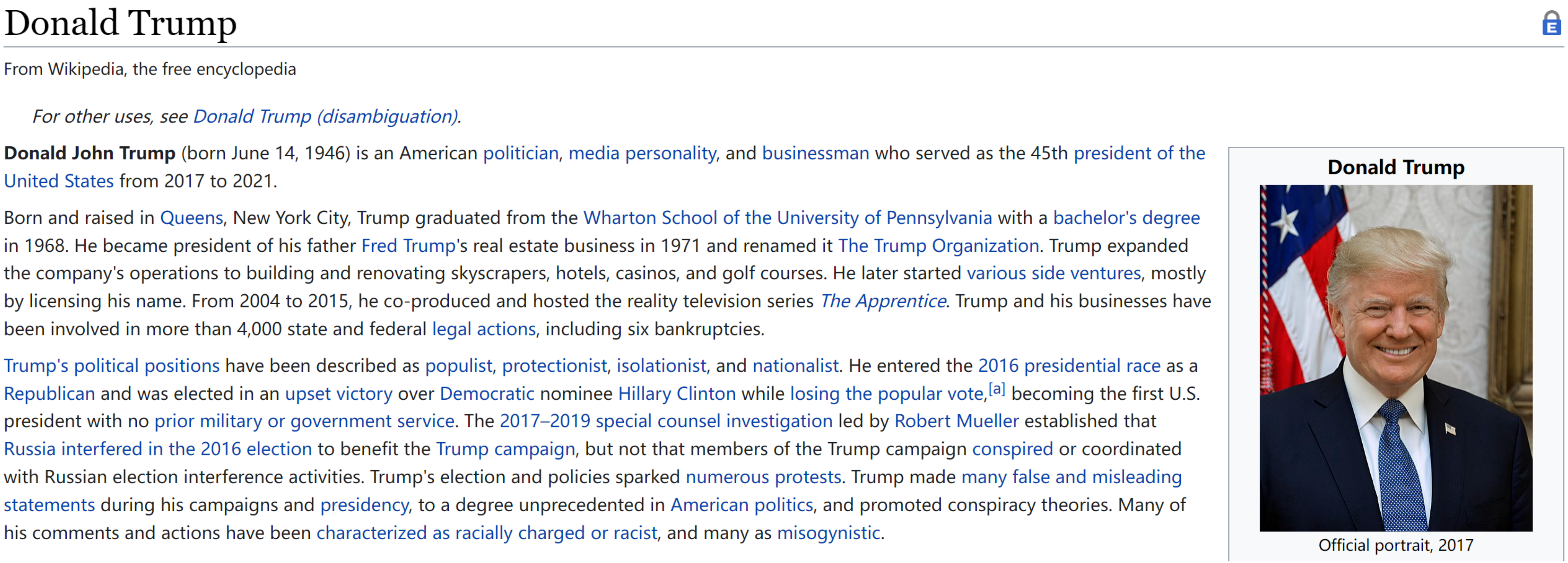

query: home remodeling
question: when was Abraham Lincoln born?
related documents: DOC123
summarize: DOC369
Model
docs: DOC246 DOC111 ...
answer: Lincoln was born in 1809.
docs: DOC234 DOC321 ...
summary: Lorem ipsum dolor sit amet.

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: https://arxiv.org/abs/2203.00537

E-mail: zhouyujia@ruc.edu.cn