SIGIR 2020 | 一文综述Learning to Match各种方法对比

深度传送门 2020-07-29

作者：坏星星是大脸猫
链接：https://zhuanlan.zhihu.com/p/163358322
编辑：深度传送门

这是一篇SIGIR 2020上的关于Learning to Match 方法的一些对比。文章实验很充分，对各种模型的对比也比较全面。是一篇好的Learning to Match 方法的总结。

A Comparison of Supervised Learning to Match Methods for Product Search

Fatemeh Sarvi[1] Nikos Voskarides[2] Lois Mooiman[3] Sebastian Schelter[2,4] Maarten de Rijke[2,4]
[1]AIRLab, University of Amsterdam [2]University of Amsterdam [3]Bol.com [4]Ahold Delhaize
[f.sarvi,n.voskarides,s.schelter,m.derijke]@uva.nl,lmooiman@bol.com

文章地址：https://arxiv.org/pdf/2007.10296.pdf
GitHub 地址：https://github.com/arezooSarvi/sigir2020-eComWorkshop-LTM-for-product-search

Vocabulary Gap 一直是信息检索领域的核心挑战，特别是在电商的搜索场景下，Vocabulary Gap的问题比网络搜索更加严重。本文对最近使用的Learning to Match 的方法进行了比较，进行比较的目的是为了更好的理解现有的流行方案并选择好的模型。

首先文章给出了如下结论：

1.一些短文本匹配的方式，例如 MV-LSTM 和DRMMTKS，仍然是最好的几个模型之一。如果兼顾时效性和准确性而言ARC-I 应该是首选的模型

2.最新的基于BERT的模型的效果中等，可能是BERT文本经过预训练的样本与产品搜索中的文本有很大不同。（这块我感觉的原因有如下几个1.bert 需要在搜索的预料进行再次的预训练 2.需要对bert进行fine-tune）

Implications of the vocabulary gap in product search.

在电商搜索中，query和title 的不匹配是常见的问题。虽然bm25仍然是比较常用的算法。但是现在越来越过的神经网络的工作通过有限维向量空间中表示查询和文档并 计算它们在该空间中的相似度，这些方法超过了原有的关键字匹配的方案。vocabulary gap 在电商搜索中挑战更加严峻 是由于 商品标题和query往往很短，并且title不一定是结构良好的句子，而是由短语或关键字的简单组合组成。

LEARNING TO MATCH METHODS

Learning to Match methods 一版分为如下几类 representation-based，interaction-based ，Hybrid Models。

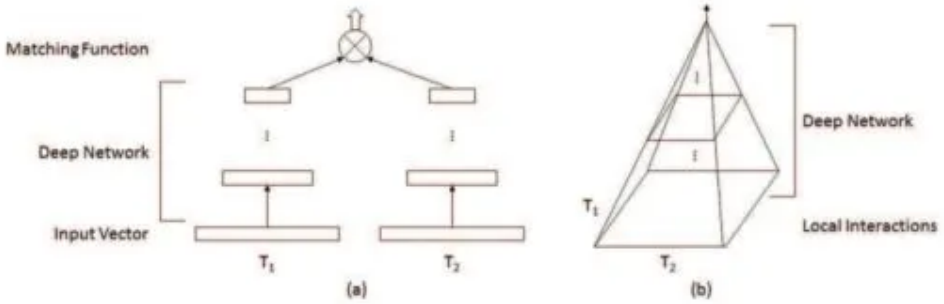
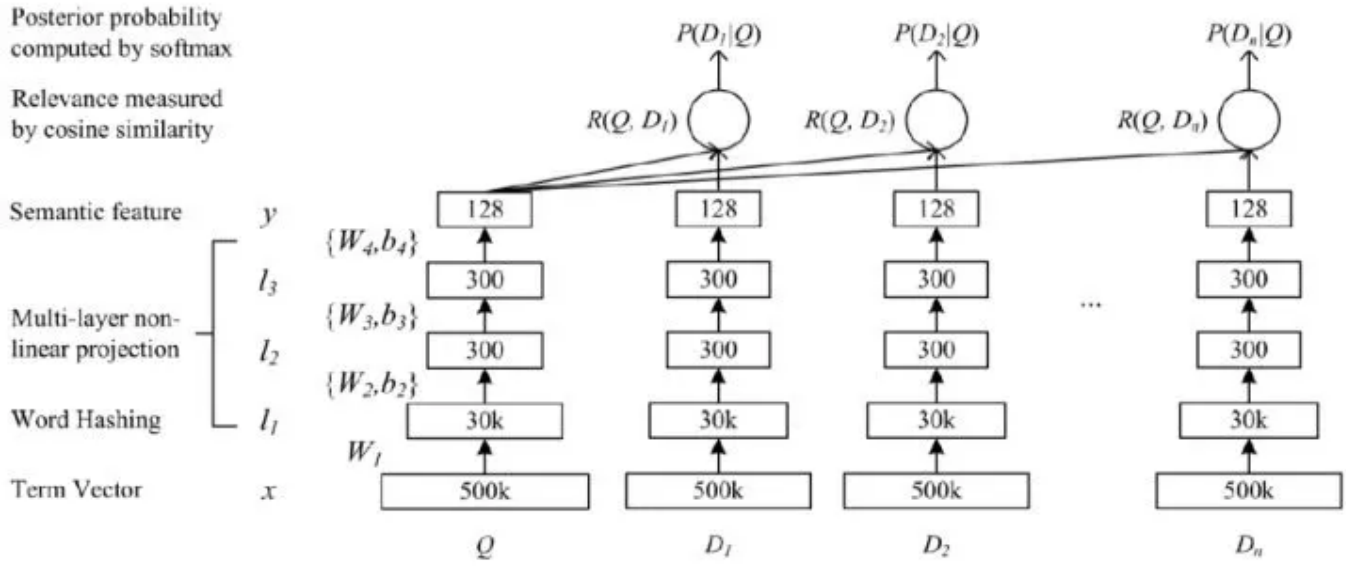


Figure 1: Two types of deep matching models: (a) Representation-focused models employ a Siamese (symmetric) architecture over the text inputs; (b) Interaction-focused models employ a hierarchical deep architecture over the local interaction matrix.

Representation-Based Models

representation 的model一版是通过分别学习query和doc的低维向量表示，然后通过一种匹配函数计算向量间的相似度。这种模型的优势在于模型简单，时效性好。

DSSM：DSSM是第一个提出深度语义匹配的模型。



CDSSM：将DSSM中的MLP换成了CNN。

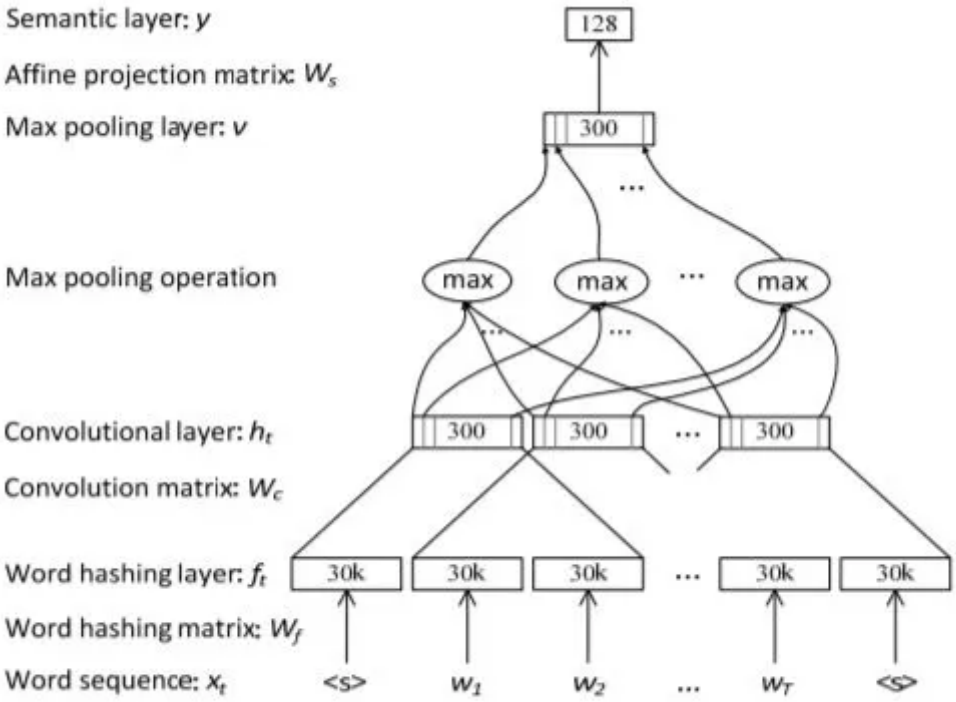


Figure 1: Illustration of the C-DSSM. A convolutional layer with the window size of three is illustrated.

MV-LSTM

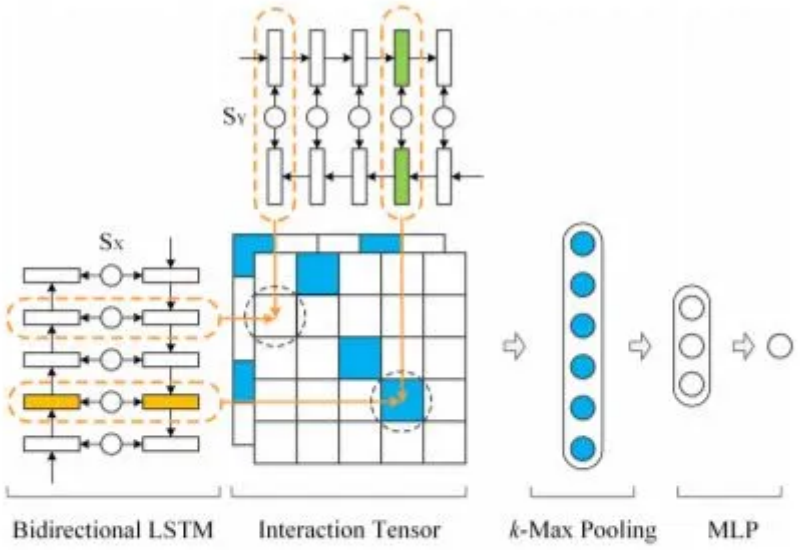


Figure 1: Illustration of MV-LSTM. S_X and S_Y are the input sentences. Positional sentence representations (denoted as the dashed orange box) are first obtained by a Bi-LSTM. k-Max pooling then selects the top k interactions from each interaction matrix (denoted as the blue grids in the graph). The matching score is finally computed through a multi-layer perceptron.

ARC-I.

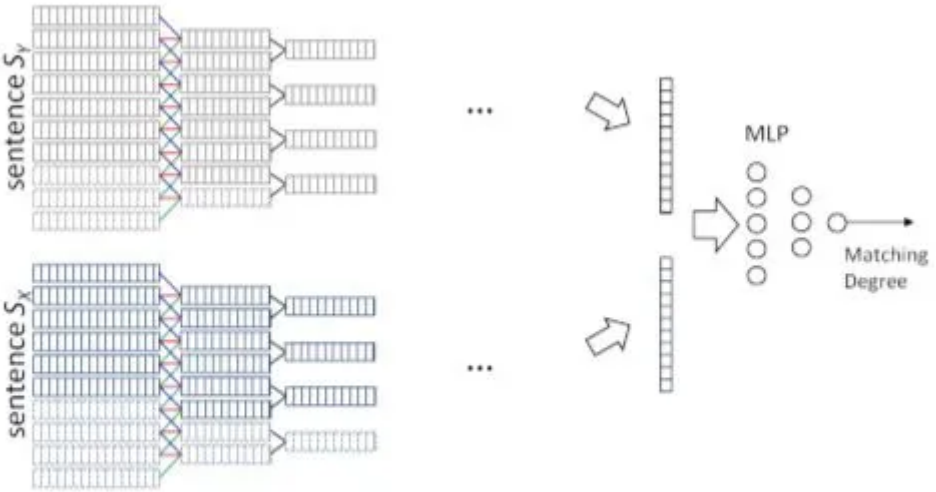


Figure 3: Architecture-I for matching two sentences.

Interaction-Based Models

Interaction model 一般是先对query和doc进行共同表示，然后在通过网络进行特征提取，输出相似度。

ARC-II.

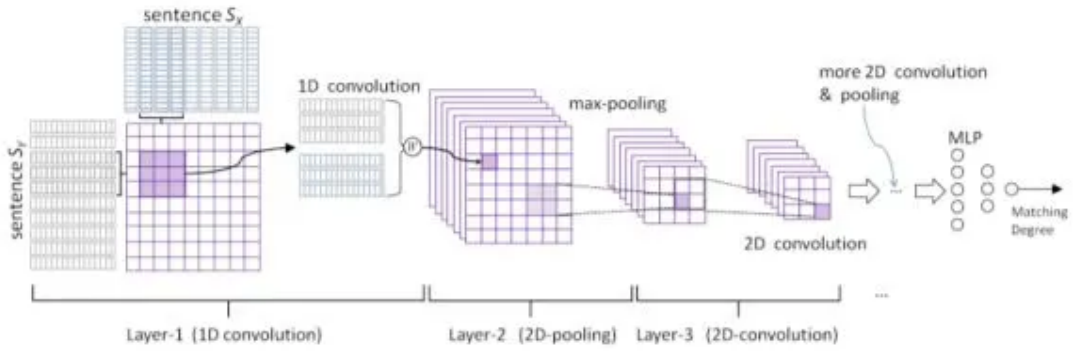


Figure 4: Architecture-II (ARC-II) of convolutional matching model

DRMM

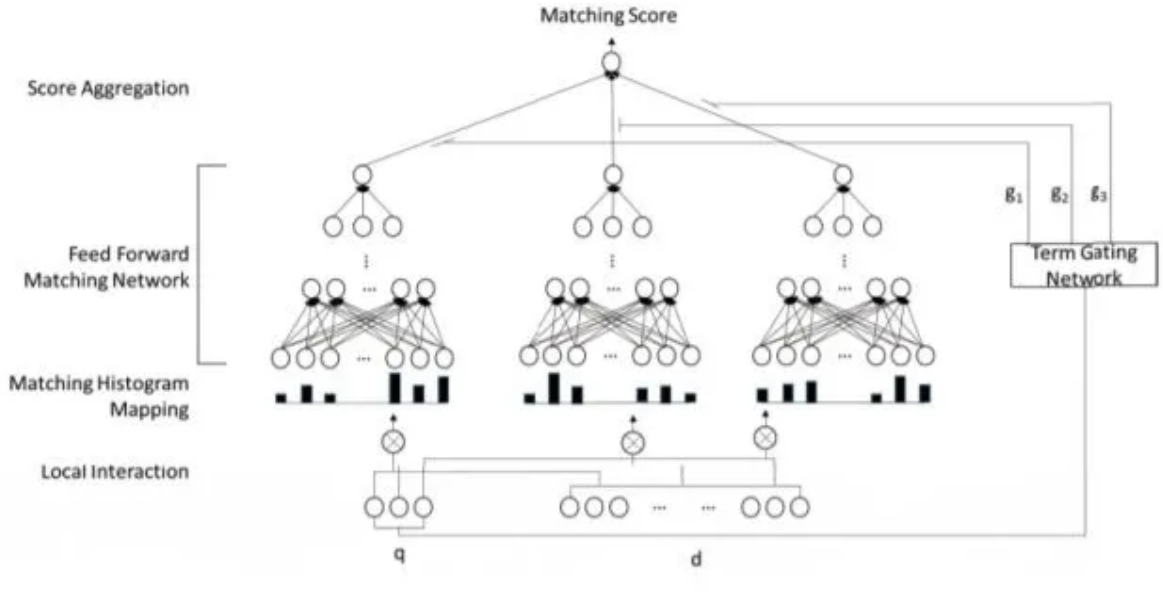


Figure 2: Architecture of the Deep Relevance Matching Model.

DRMMTKS

专用于短文本匹配，并将DRMM匹配的直方图替换为top-k最大池化层

MatchPyramid

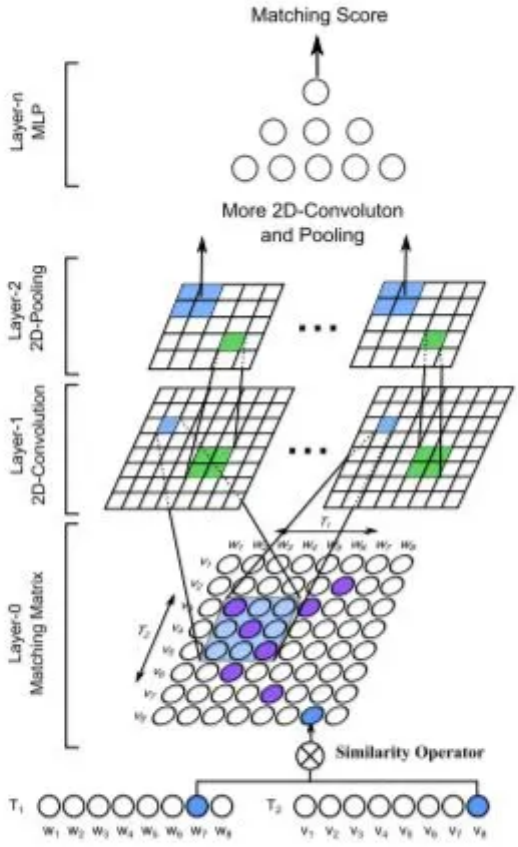
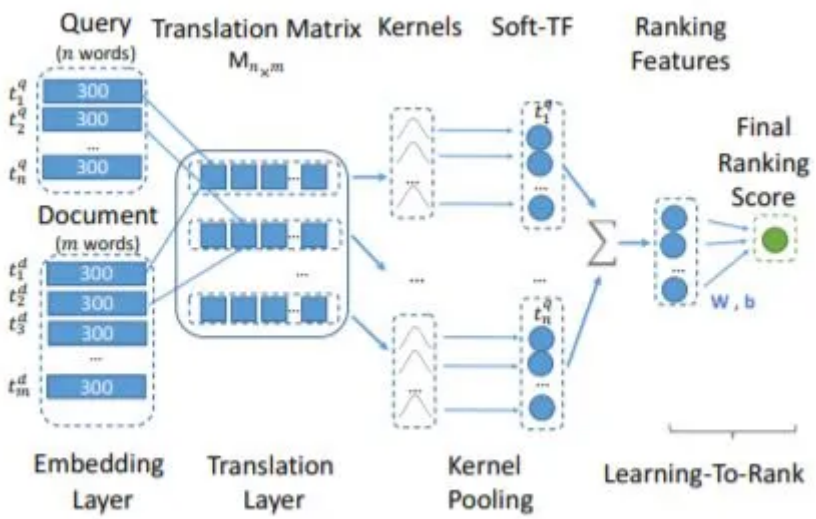
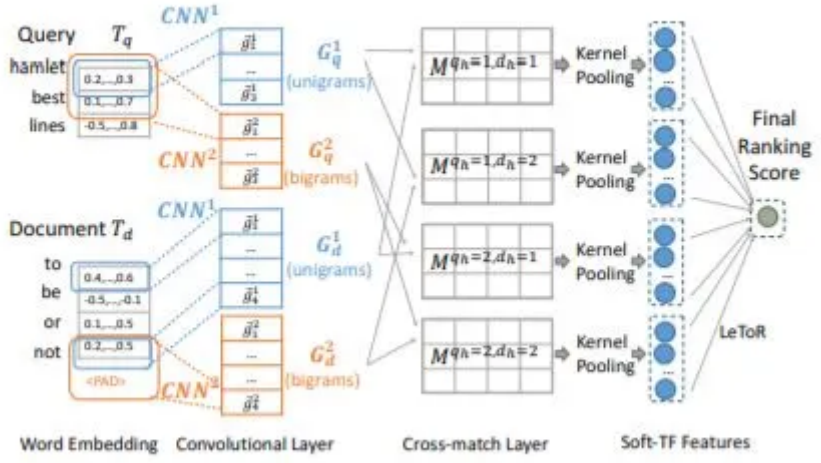


Figure 3: An overview of MatchPyramid on Text Matching.

K-NRM



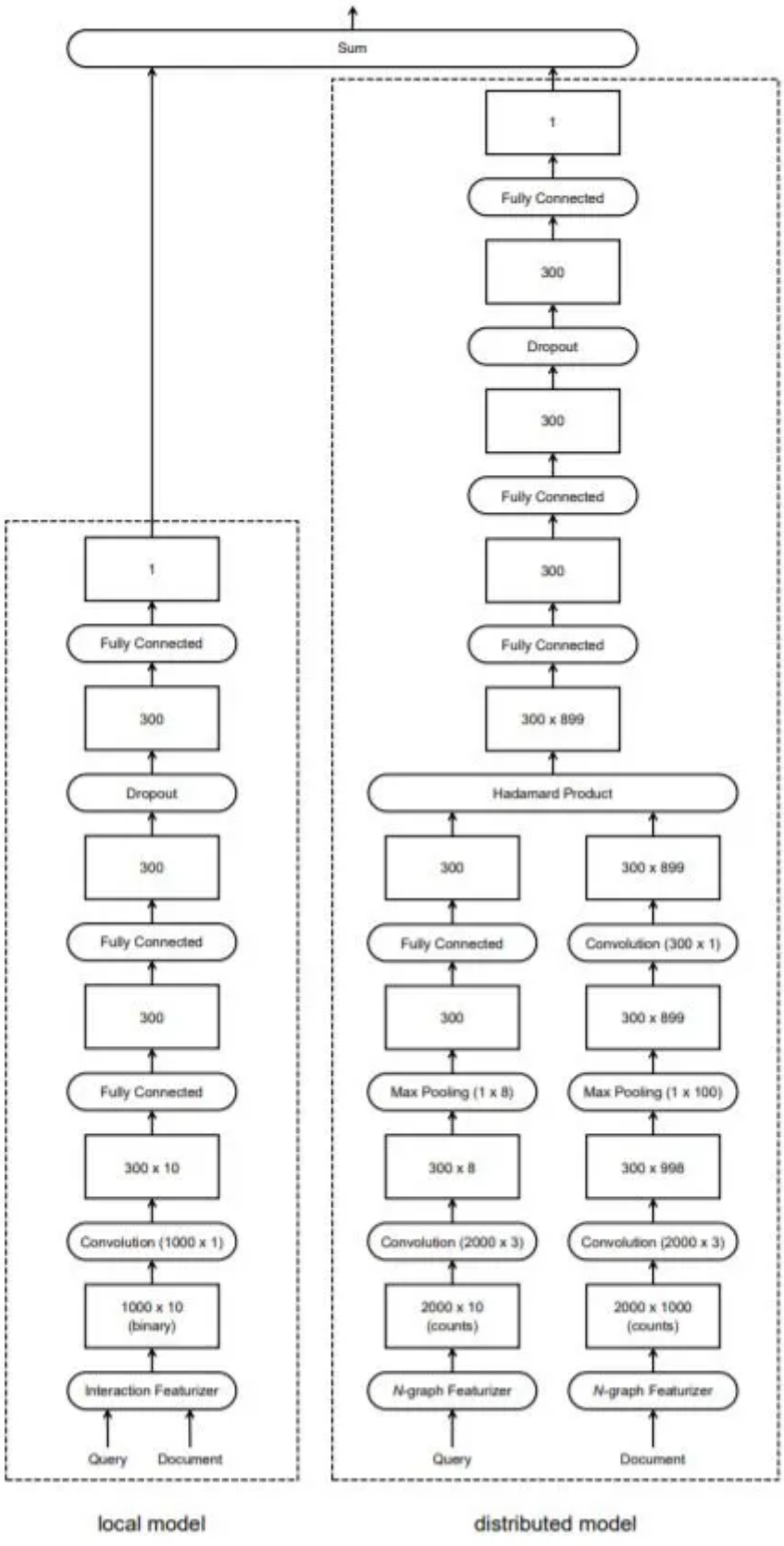
CONV-KNRM



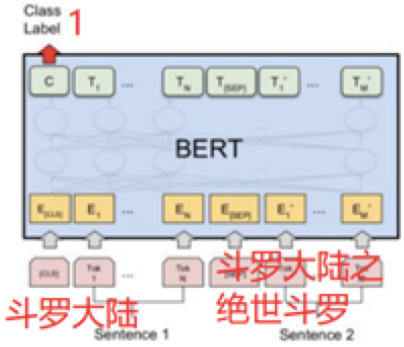
Hybrid Models

同时包含Representation和Interaction 的方式

DUET



BERT：通过预训的bert然后再采用cls作为向量表示信息，通过另一个线性变换层进行预测。



实验数据:

Table 1: Basic statistics of our datasets.

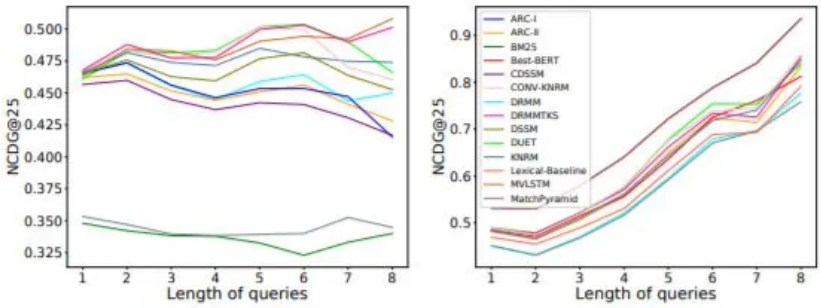
	CIKM 2016	Proprietary
#queries	51,888	53,474
#unique queries	26,137	40,125
#unique presented products	37,964	214,778
#clicks	36,814	63,859

模型的表现:

Table 2: Performance of MatchZoo models on both datasets in terms of NDCG at position 5 and 25.

Model	CIKM data		Proprietary data	
	NDCG@5	@25	@5	@25
Lexical	0.148	0.343	0.314	0.474
MatchPyramid	0.152	0.347	0.287	0.454
CDSSM	0.314	0.452	N/A	N/A
ARC-II	0.320	0.458	0.334	0.488
ARC-I	0.326	0.462	**0.408**	**0.549**
DRMM	0.331	0.464	0.288	0.455
DSSM	0.334	0.467	N/A	N/A
KNRM	0.341	0.472	0.337	0.490
DUET	0.345	0.473	0.350	0.500
MV-LSTM	0.342	0.474	**0.408**	**0.549**
CONV-KNRM	0.347	0.476	0.349	0.498
DRMMTKS	**0.347**	**0.477**	0.345	0.498
âĂŃBest-BERT	N/A	N/A	0.340	0.493

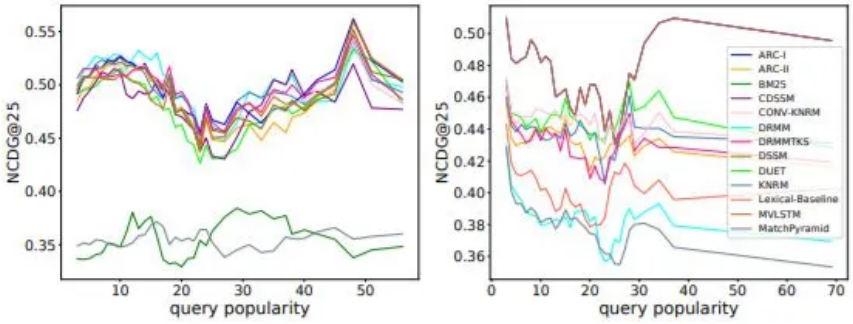
query长短对模型的影响:



(a) CIKM dataset **(b)** Proprietary dataset

Figure 1: Ranking performance for varying query length. On the X-axis we see the length of the query, and Y-axis indicated the average NDCG at position 25 per queries of a specified length.

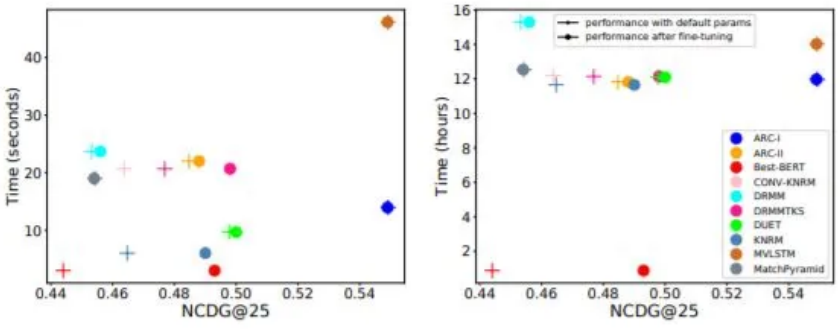
query的流行度对模型的影响：



(a) CIKM dataset **(b)** Proprietary dataset

Figure 3: The behavior of models based on query popularity. The flow is quite the same in all cases and all of the models tend to perform better for more frequently seen queries.

训练/推理时间与模型表现



(a) Performance vs. inference time **(b)** Performance vs. training time

Figure 5: Ranking performance in comparison to training and inference time for the proprietary dataset. Both ranking performances achieved from the default hyper-parameters and the fine-tuned ones are depicted in this figure.

总的来说，文章总结了12中Learning to Match的方法一些对比。实验还是很丰富，正好本人也正在做一些类似的事情在工业级的数据集上。我们的数据将会比文章数据规模大10倍以

上，同时对于bert 我们也会先进行一些fine-tune，另外我们也正在对比一些传统的模型。等实验完全做完之后。我们会放出一些实验记录情况。欢迎持续关注。

阅读原文