OpenMatch：开放域信息检索开源工具包

原创 TsinghuaNLP TsinghuaNLP 2020-08-06

清华大学自然语言处理与社会人文计算实验室近日开源了开放域信息检索工具包：OpenMatch,和神经网络信息检索必读论文集：NeuIRPapers。OpenMatch是清华大学计算机系与微软研究院团队联合完成的成果，基于Python和PyTorch开发，它具有两大亮点：一是为用户提供了开放域下信息检索的完整解决方案，并通过模块化处理，方便用户定制自己的检索系统。二是支持领域知识的迁移学习，包括融合外部知识图谱信息的知识增强模型以及筛选大规模数据的数据增强模型。

- **工具包地址：**https://github.com/thunlp/OpenMatch

- **论文集地址：**https://github.com/thunlp/NeuIRPapers

01 总体介绍

亮点一：提供开放域信息检索场景完整解决方案。

OpenMatch总体架构包括两大部分：一是相关文档检索，即根据用户检索词，从大规模文档集合中返回最相关的Top-K(K通常为100或1000)文档。二是文档重排序，即将各神经网络模型和非神经网络模型的排序特征整合，对Top-K文档重排序，进一步提升排序效果。总体架构如下图所示：

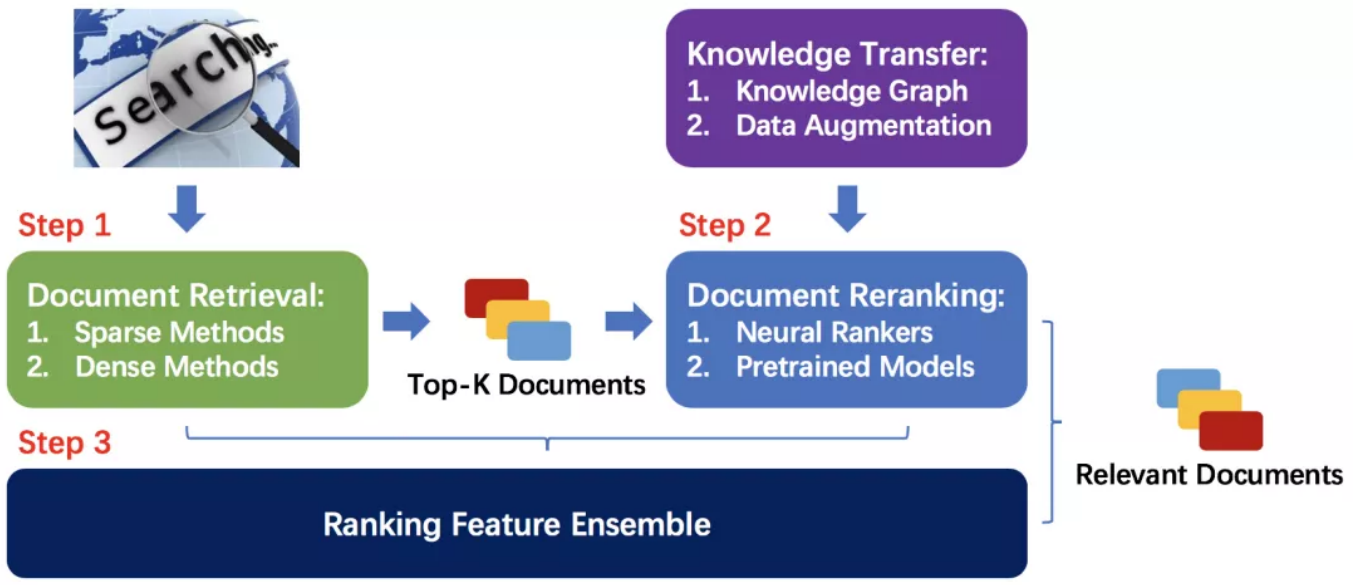

图1：OpenMatch总体架构

亮点二：支持领域知识迁移学习。

OpenMatch提供了融合外部知识图谱信息的知识增强模型，和筛选大规模数据的数据增强模型。如下表所示：

相关文档检索	文档重排序	领域知识迁移学习
BM25	K-NRM[1]	EDRM[3] (知识增强)
ANN	Conv-KNRM[2]	ReInfoSelect[8] (数据增强)
	TK[5]	
	Pretrained Models	
	Coor-Ascent	

表1：全模型表

其中，Pretrained Models基于huggingface’s Transformers实现。地址：https://github.com/huggingface/transformers

02 实现细节

一、数据处理模块(OpenMatch/data/)：支持多种数据格式和预训练词表，一键初始化，并行读入数据。

- Tokenizer：支持多种预训练词表进行初始化，可获得词表大小、词向量维度和矩阵等，用于对神经网络嵌入层初始化，也可仅读入非预训练词表，嵌入层可相应的自动随机初始化。支持对输入文本批量处理。定义基类，仅定义新的分词方法即可开发新的tokenizer。

- DataSet：支持多种文件格式(jsonl，trec等)的语料，支持pair-wise(ranking)和prediction(classification)两种训练方式。可配合DataLoader并行读取数据集。

二、神经网络模块(OpenMatch/modules/)：包含多种主流模块，方便快速搭建神经网络模型。

- Embedder：词嵌入层，可由Tokenizer是否返回词向量矩阵，决定随机初始化或词向量矩阵初始化。
- Attention：支持Multi-Head和Dot-Product两种注意力机制。
- Encoder：支持CNN和Transformer等编码方式。
- Matcher：支持Kernel Matcher。即对用户检索和文档的交互矩阵，采用kernel pooling得到排序特征。

所有神经网络模块如下表所示：

Embedder	Attention	Encoder	Matcher
Embedder	ScaledDotProductAttention	Conv1DEncoder	KernelMatcher
	MultiHeadAttention	FeedForwardEncoder	
		TransformerEncoder	

表2：神经网络模块表

神经网络模型可通过各种模块迅速搭建，如以下OpenMatch中已实现的模型：

- K-NRM：Embedder + KernelMatcher
- Conv-KNRM：Embedder + Conv1DEncoder + KernelMatcher
- TK：Embedder + TransformerEncoder + KernelMatcher

三、评测模块(OpenMatch/Metrics/)：支持多种官方指标，一键快捷评测。

- Metric：提供各种官方评测：如NDCG，MAP，MRR等，可根据用户设置的指标一键返回结果。

OpenMatch提供了快速入门代码，完整使用示例和详细的文档，方便用户快速掌握各工具的使用方法。如下图的快速入门：

```
tokenizer = om.data.tokenizers.WordTokenizer(pretrained="./data/glove.6B.300d.txt")
query_ids, query_masks = tokenizer.process(query, max_len=16)
doc_ids, doc_masks = tokenizer.process(doc, max_len=128)
model = om.models.KNRM(vocab_size=tokenizer.get_vocab_size(),
                       embed_dim=tokenizer.get_embed_dim(),
                       embed_matrix=tokenizer.get_embed_matrix())
ranking_score, ranking_features = model(torch.tensor(query_ids).unsqueeze(0),
                                        torch.tensor(query_masks).unsqueeze(0),
                                        torch.tensor(doc_ids).unsqueeze(0),
                                        torch.tensor(doc_masks).unsqueeze(0))
```

图2：快速入门示例

Tokenizer通过GloVe预训练词表初始化后，可一键获取输入文本的id和mask，并对神经网络模型初始化，一键获得排序分数和特征。更多样例可在GitHub查看。

03 实验结果

OpenMatch采用Robust04，ClueWeb09-B和ClueWeb12-B13作为基准数据。对各模型效果进行测试(仅对数据集进行5折交叉验证，长文档截断取第一段文本)，结果如下图：

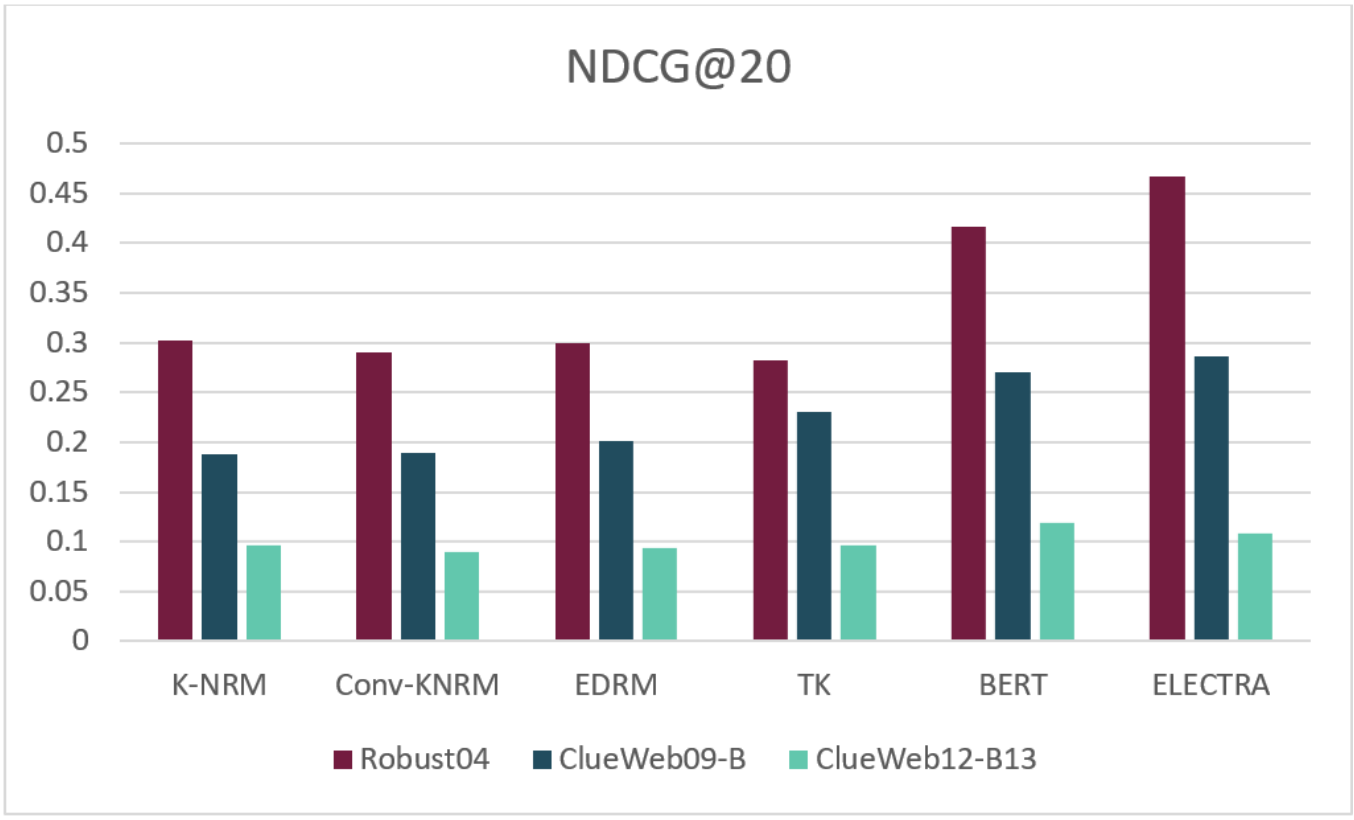

图3：各模型实验结果

我们(CMT: CMU, Microsoft and THU)在近期的TREC COVID (新冠肺炎信息检索)竞赛Round 2中取得了第一名，比赛官网、相关数据、模型checkpoint及复现方法已在GitHub放出。实验结果如下图所示：

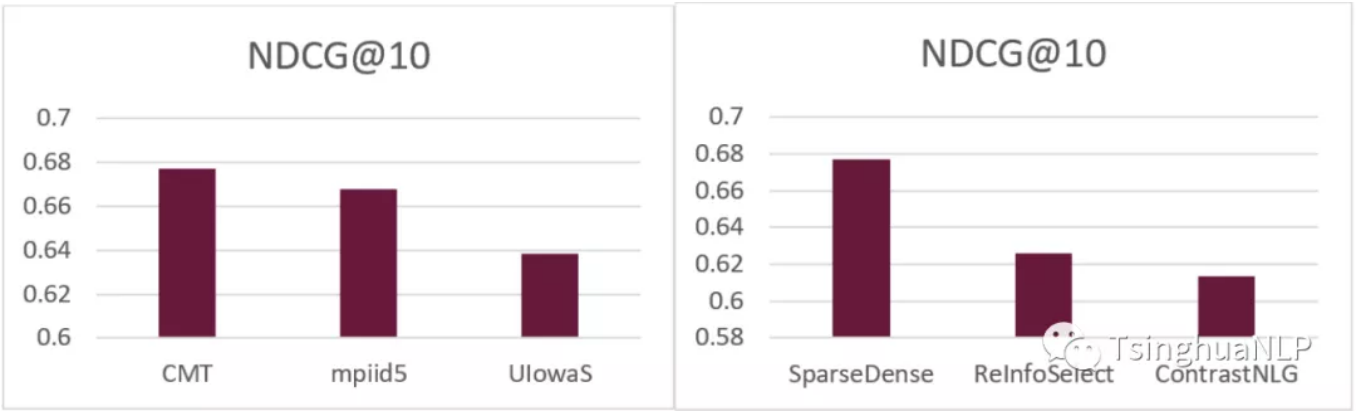

图4：新冠肺炎信息检索竞赛结果

左图为比赛前三名结果，右图是我们提交的三种方法：

- SparseDense：通过BM25(Sparse)+ANN(Dense)进行相关文档检索[9]，用medical marco数据训练的SciBERT[4]模型进行重排序。
- ReInfoSelect：通过BM25进行相关文档检索，用policy gradient筛选medical marco数据并训练SciBERT模型，进行重排序。
- ContrastNLG：通过BM25进行相关文档检索，用medical marco和生成的pseudo query[11]数据训练的SciBERT模型进行重排序。

可以看出，通过筛选数据的数据增强方法能够有效提升模型效果，通过BM25+ANN的文档检索方法可以大幅提升排序效果。另外，mpiid5的方法是通过BM25进行相关文档检索，使用上轮数据训练的ELECTRA[7]进行重排序，此方法也已在OpenMatch复现。

04 结语

OpenMatch项目将会长期维护及持续更新，我们欢迎大家使用OpenMatch作为信息检索领域学术研究和应用开发的工具，也期盼大家宝贵的意见和建议，或加入我们的队伍，共同开发，完善工具包。

05 开发团队

- 张凯韬，清华大学计算机系硕士生， https://github.com/zkt12
- 孙丝，清华大学电子系博士生， https://github.com/SunSiShining
- 刘正皓，清华大学计算机系博士生， http://nlp.csai.tsinghua.edu.cn/~lzh/
- 卢奥炜，清华大学计算机系本科生， https://github.com/LAW991224

06 指导老师

- 刘知远，清华大学计算机系副教授， http://nlp.csai.tsinghua.edu.cn/~lzy/
- 熊辰炎，微软研究院高级研究员， https://www.microsoft.com/en-us/research/people/cxiong/
- 孙茂松，清华大学计算机系教授， https://nlp.csai.tsinghua.edu.cn/staff/sms/

07 相关论文

[1] End-to-End Neural Ad-hoc Ranking with Kernel Pooling. Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, Russell Power. SIGIR 2017.

[2] Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. Zhuyun Dai, Chenyan Xiong, Jamie Callan, Zhiyuan Liu. WSDM 2018.

[3] Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. Zhenghao Liu, Chenyan Xiong, Maosong Sun, Zhiyuan Liu. ACL 2018.