一文读懂深度学习文本分类方法

AINLP 2019-06-07

以下文章来源于AI算法之心 ， 作者何从庆

本文系作者投稿，作者公众号：AI算法之心（id:AIHeartForYou），欢迎关注，点击文末"阅读原文"可直达原文链接，也欢迎大家投稿，AI、NLP相关即可。

近些天一直忙着毕业以及小论文投递的事情，没有及时更新公众号。在此表示抱歉。

最近有很多小伙伴想了解深度学习在文本分类的发展，因此，笔者整理最近几年比较经典的深度文本分类方法，希望帮助小伙伴们了解深度学习在文本分类中的应用。

笔者整理了近些年的相关深度文本分类论文，关注“AI算法之心”，后台回复“文本分类论文”即可下载。

Convolutional Neural Networks for Sentence Classification (EMNLP 2014)

Kim在EMNLP2014提出的TextCNN方法，在多个数据集上取得了很好的效果。由于其计算速度快以及可并行性，在产业界得到了广泛使用。TextCNN的模型示意图如下图所示。

TextCNN模型首先将文本映射成向量，然后利用多个滤波器来捕捉文本的局部语义信息，接着使用最大池化，捕捉最重要的特征。最近将这些特征输入到全连接层，得到标签的概率分布。

代码参考：

1) https://github.com/alexander-rakhlin/CNN-for-Sentence-Classification-in-Keras

2) https://github.com/brightmart/text_classification

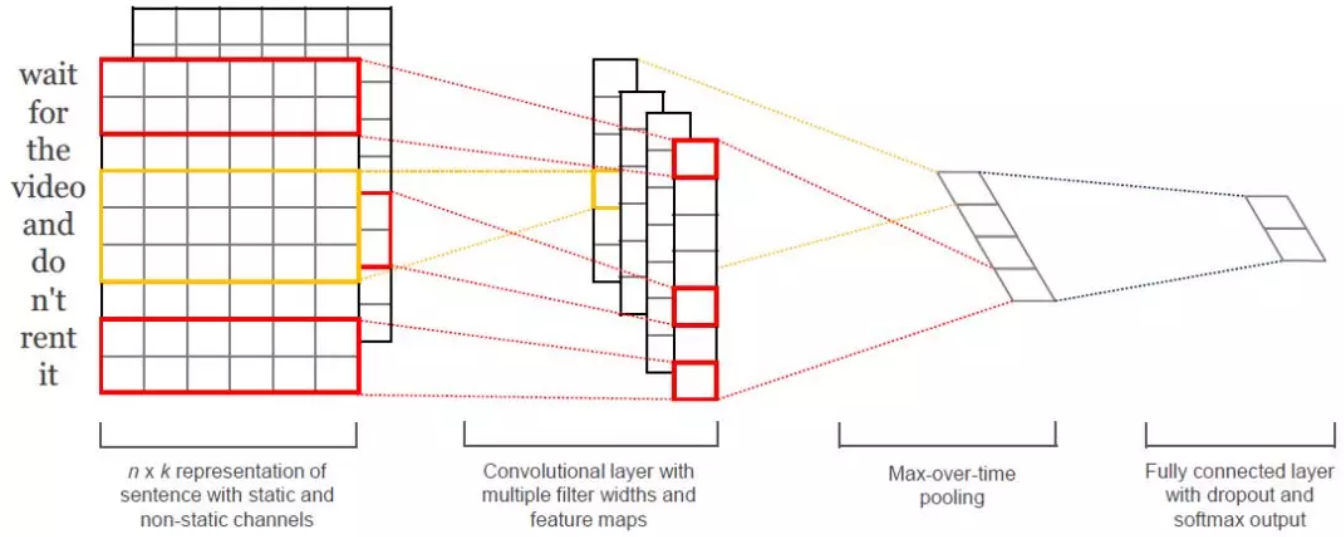


图1：TextCNN模型架构

Document Modeling with Gated Recurrent Neural Network for Sentiment Classification (EMNLP 2015)

Tang等人提出了一种利用GRU对文档进行建模的情感分类模型。模型如下图所示。

该模型首先将文本映射为向量，然后利用CNN/LSTM（论文中使用3个滤波器的CNN）进行句子表示。另外，为了捕获句子的全局语义表征，将其输送给平均池化层，再接入tanh激活函数。最后将整个句子的不同宽度卷积核的向量表示接入一个Average层，从而得到句子平均向量表示。

然后将得到的句子表示，输入到GRU中，得到文档向量表示。最后将文档向量输送给softmax层，得到标签的概率分布。

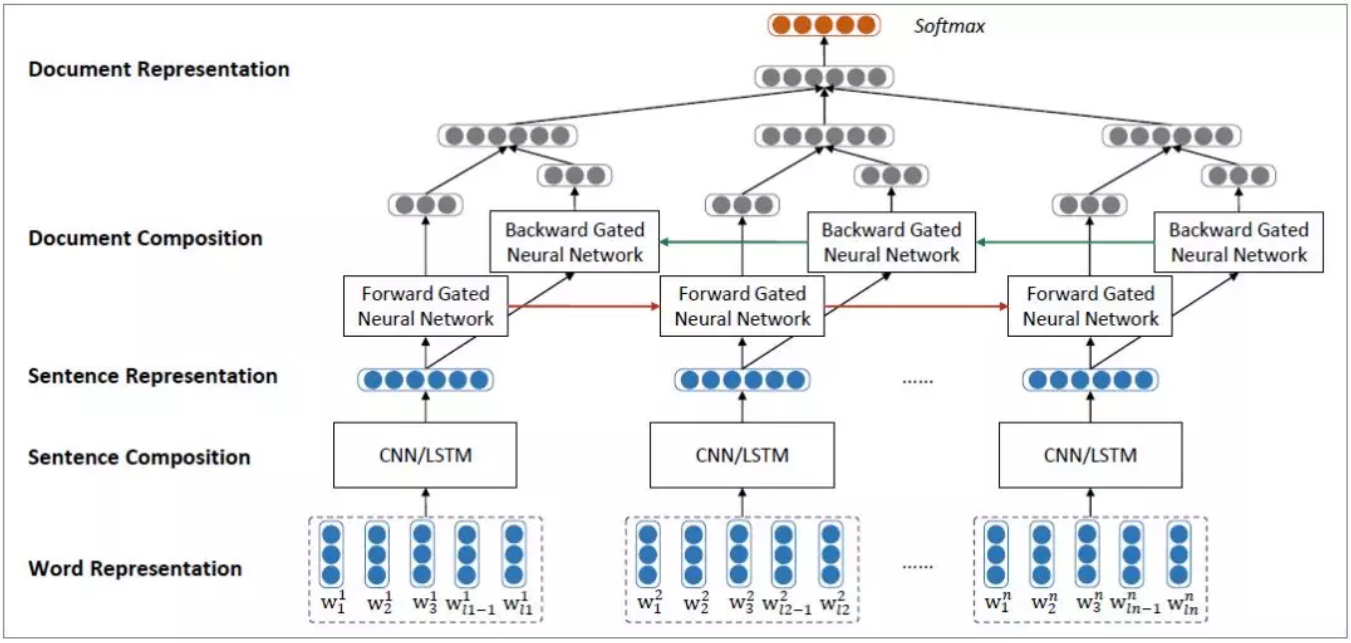


图2：文档级别情感分类的神经网络模型

Recurrent Convolutional Neural Networks for Text Classification (AAAI 2015)

Lai等人提出了一种无人工特征的循环卷积神经网络分类方法，简称RCNN。

RCNN首先利用Bi-RNN来捕捉前后的上下文表征，然后将其concat起来，接着使用滤波器filter_size=1的卷积层，并使用最大池化操作得到与文档最相关的向量表征，最后将这些向量输入到softmax层，得到标签的概率表征。

代码参考：

1）https://github.com/roomylee/rcnn-text-classification

2）https://github.com/brightmart/text_classification

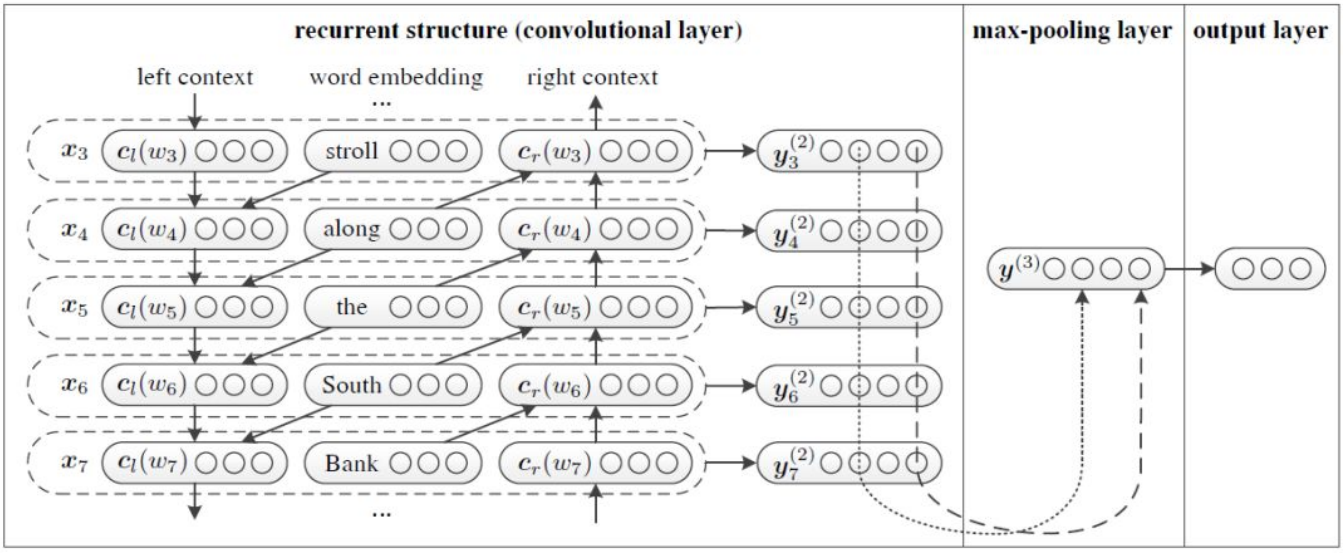


图3：RCNN的模型结构示意图

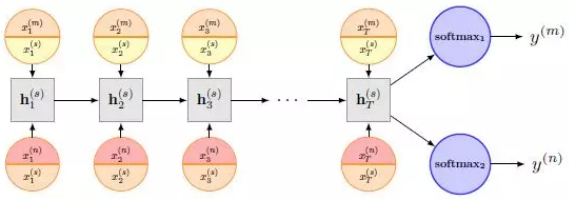
Recurrent Neural Network for Text Classification with Multi-Task Learning (IJCAI 2016)

Liu等人针对文本多分类任务，提出了基于RNN的三种不同的共享信息机制对具有特定任务和文本进行建模。

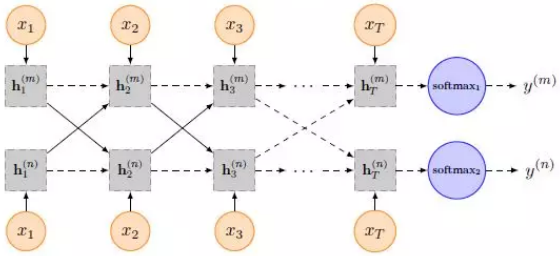
模型1(Uniform-Layer Architecture)：所有任务共享同一个LSTM层，并在每个特定任务后面拼接一个随机生成可训练的向量。LSTM层的最后一个时刻的隐藏层作为输入传入到softmax层。

模型2(Coupled-Layer Architecture)：每个任务具有自己独立的LSTM层，但是每一时刻所有任务的hidden state则会和下一时刻的character一起作为输入，最后一个时刻的hidden state进行分类。

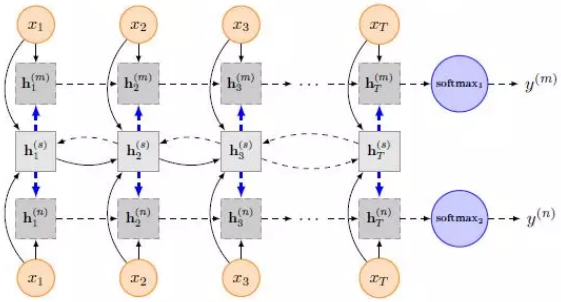
模型3(Shared-Layer Architecture)：除了一个共享的BI-LSTM层用于获取共享信息，每个任务有自己独立的LSTM层，LSTM的输入包括每一时刻的character和BI-LSTM的hidden state。



(a) Model-I: Uniform-Layer Architecture



(b) Model-II: Coupled-Layer Architecture



(c) Model-III: Shared-Layer Architecture

图4：三种架构进行多任务学习建模

Hierarchical Attention Networks for Document Classification (NAACL 2016)

Yang等人提出了一种用于文档分类的层次注意力机制网络，简称HAN。这篇文章和Tang等人都是针对于文档分类的问题，然而，这篇文章在句子级别以及文档级别提出了注意力机制，使得模型在构建文档时是能够赋予重要内容不同的权重，同时，也可以缓解RNN在捕捉文档的序列信息产生的梯度消失问题。HAN模型的模型示意图如下所示。

HAN模型首先利用Bi-GRU捕捉单词级别的上下文信息。由于句子中的每个单词对于句子表示并不是同等的贡献，因此，作者引入注意力机制来提取对句子表示有重要意义的词汇，并将这些信息词汇的表征聚合起来形成句子向量。具体的注意力机制的原理可以参考：

FEED-FORWARD NETWORKS WITH ATTENTION CAN SOLVE SOME LONG-TERM MEMORY PROBLEMS

然后，对于所有的句子向量输入到Bi-GRU中，捕捉句子级别的上下文信息，得到文档向量。同样地，为了奖励对文档进行正确分类的线索句，作者再次使用注意力机制，来衡量句子的重要性，得到文档向量。最后将文档向量均输入到softmax层，得到标签的概率分布。

代码参考：

1) https://github.com/richliao/textClassifier

2） https://github.com/brightmart/text_classification

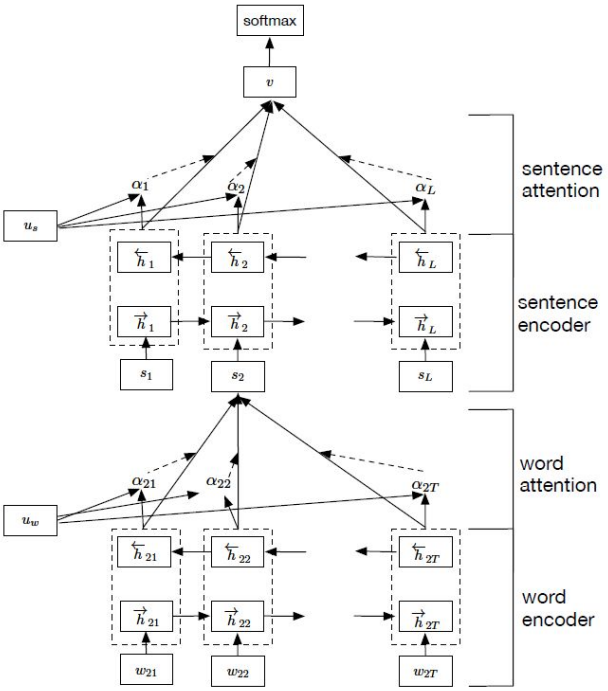


图3：HAN模型结构示意图

Bag of Tricks for Efficient Text Classification (EACL 2017)

Joulin等人提出了一种简单而又有效的文本分类模型，简称fastText。

fastText模型输入一个词序列（一段文本或者一句话），序列中的词与词组成特征向量，然后特征向量通过线性变换映射到中间层，中间层再映射到标签。输出这个词序列属于不同类别的概率。其中fastText在预测标签是使用了非线性激活函数，但在中间层不使用非线性激活函数。

代码参考：

1） https://github.com/facebookresearch/fastText

2） https://radimrehurek.com/gensim/models/fasttext.html

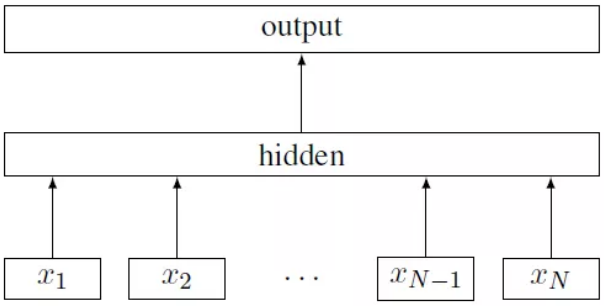


图4：fastText模型结构示意图

Deep Pyramid Convolutional Neural Networks for Text Categorization (ACL 2017)

Johnson 和Zhang 提出了一种单词级别的深层CNN模型，来捕捉文本的全局语义表征，该模型在不增加太多的计算开销的情况下，通过增加网络深度可以获得最佳的性能，简称DPCNN。模型结构示意图如下所示。

DPCNN模型首先利用“text region embedding”，将常用的word embedding 推广到包含一个或多个单词的文本区域的embedding，类似于增加一层卷积神经网络。

然后是卷积快的叠加（两个卷积层和一个shortcut连接，其中shortcut连接类似于残差连接），与步长为2的最大池化层进行下采样。最后使用一个最大池化层，得到每个文档的文档向量。

代码参考：
https://github.com/Cheneng/DPCNN

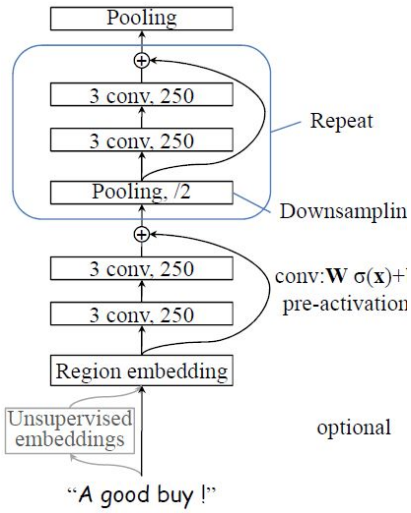


图4：DPCNN模型结构示意图

Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm (EMNLP 2017)

Felbo等人使用数以百万计的表情符号来学习任何领域的表情符号来检测情绪、情绪和讽刺，提出了DeepMoji模型，并取得了具有竞争性的效果。同时，DeepMoji模型在文本分类任务上也可以取得不错的结果。

DeepMoji模型首先使用embedding层将单词映射成向量，并将每个embedding维度使用双正切函数映射到[-1,1]。然后，作者使用两层的Bi-LSTM捕捉上下文特征。接着作者提出了一种新的注意力机制，分别将embeddding层以及2层的Bi-LSTM作为输入，得到文档的向量表征。最后，将向量输入到softmax层，得到标签的概率分布。

代码参考：

https://github.com/bfelbo/DeepMoji

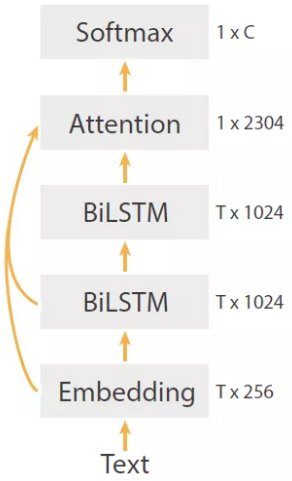


图5：DeepMoji模型结构示意图

Investigating Capsule Networks with Dynamic Routing for Text Classification (EMNLP 2018)

Zhao等人提出了一种基于胶囊网络的文本分类模型，并改进了Sabour等人提出的动态路由，提出了三种稳定动态路由。模型如下所示：

该模型首先利用标准的卷积网络，通过多个卷积滤波器提取句子的局部语义表征。然后将CNN的标量输出替换为向量输出胶囊，从而构建Primary Capsule层。接着输入到作者提出的改进的动态路由（共享机制的动态路由和非共享机制的动态路由），得到卷积胶囊层。最后将卷积胶囊层的胶囊压平，送入到全连接胶囊层，每个胶囊表示属于每个类别的概率。

代码参考：

https://github.com/andyweizhao/capsule_text_classification.

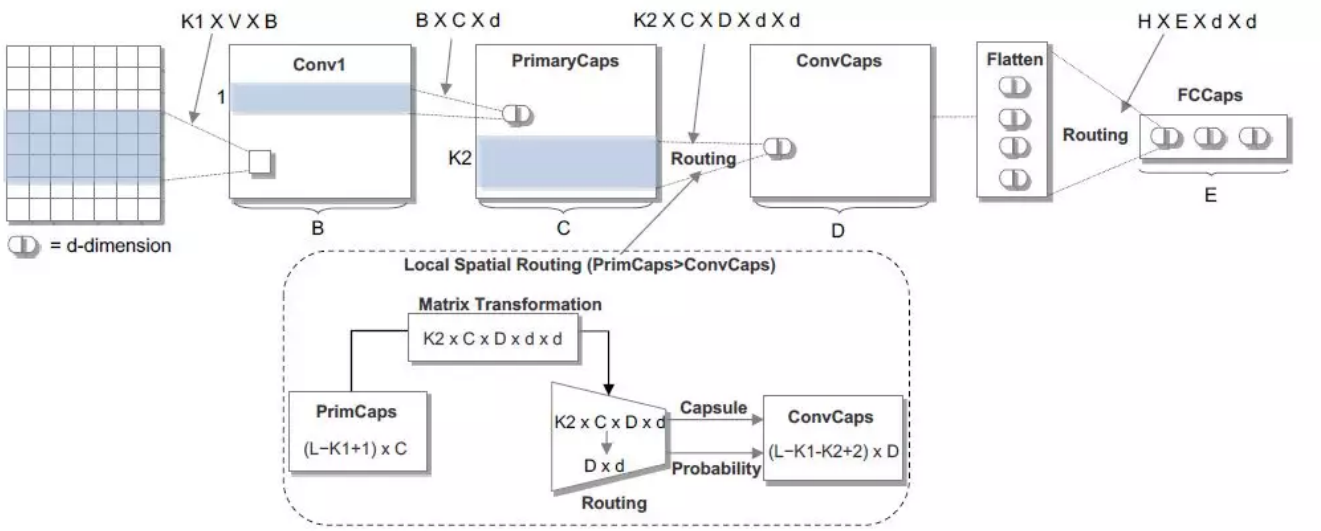


图6：文本分类的胶囊网络体系结构

Sentiment Analysis by Capsules (WWW 2018)

Wang等人提出了一种用于情感分类的RNN胶囊网络模型，简称RNN-Capsule。(这篇文章在可视化方面做的还是不错的)模型结构示意图如下所示。

RNN-Capsule首先使用RNN捕捉文本上下文信息，然后将其输入到capsule结构中，该capsule结构一共由三部分组成：representation module，probability module，和reconstruction module。具体地，首先用注意力机制计算capsule 表征；然后用capsule表征计算capsule状态的概率；最后用capsule表征以及capsule状态概率重构实例的表征。

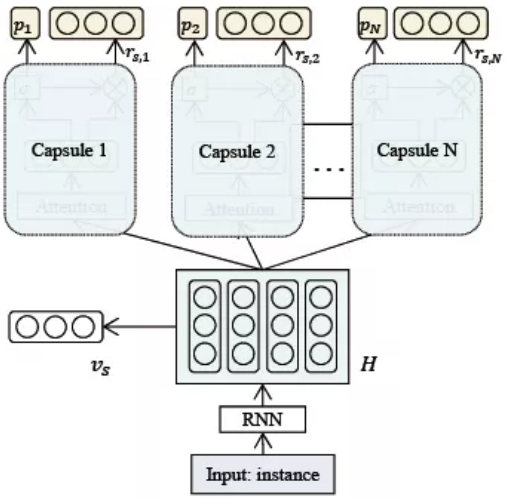


图7：RNN-Capsule模型结构示意图

Graph Convolutional Networks for Text Classification (AAAI 2019)

Yao等人提出了一种基于graph convolutional networks(GCN)进行文本分类。作者构建了一个包含word节点和document节点的大型异构文本图，显式地对全局word利用co-occurrence信息进行建模，然后将文本分类问题看作是node分类问题。

代码参考：

https://github.com/yao8839836/text_gcn

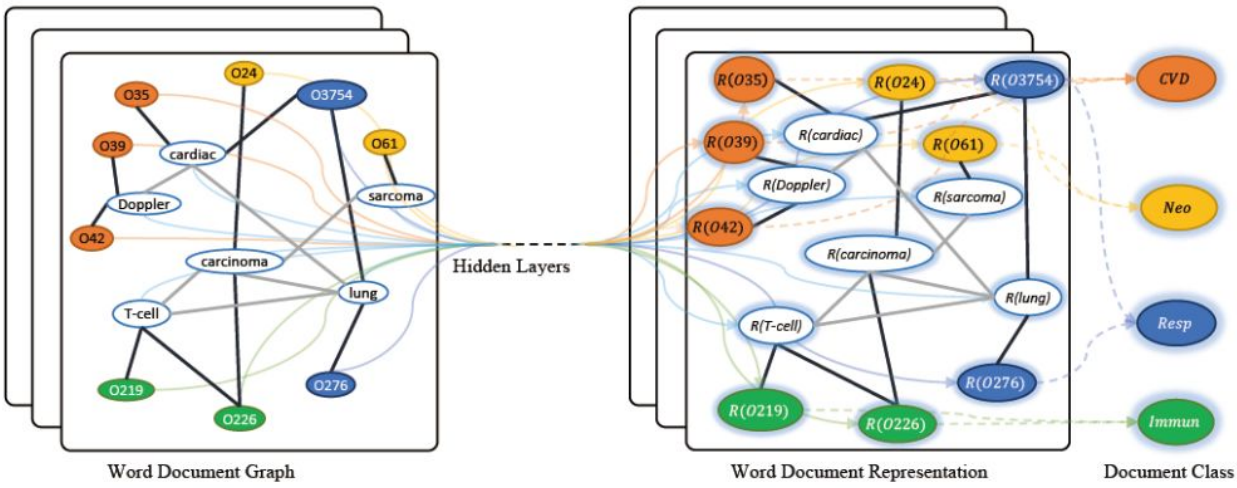


图7：Text GCN的模型结构

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (NAACL 2019)

Google提出的BERT模型，突破了静态词向量无法解决一词多义的问题。BERT是基于语言模型的动态词向量，在自然语言处理的多项任务中取得了最优秀的结果。笔者对BERT模型进行微调，在文本分类的多个领域，诸如法律、情感等，取得了非常有竞争性的性能。

BERT的模型架构是一个多层的双向Transformer编码器(Transformer的原理及细节可以参考Attention is all you need)。作者采用两套参数分别生成$\text{BERT}_{\text{BASE}}$模型和$\text{BERT}_{\text{LARGE}}$模型(细节描述可以参考原论文)，所有下游任务可以在这两套模型进行微调。

代码参考：

https://github.com/google-research/bert

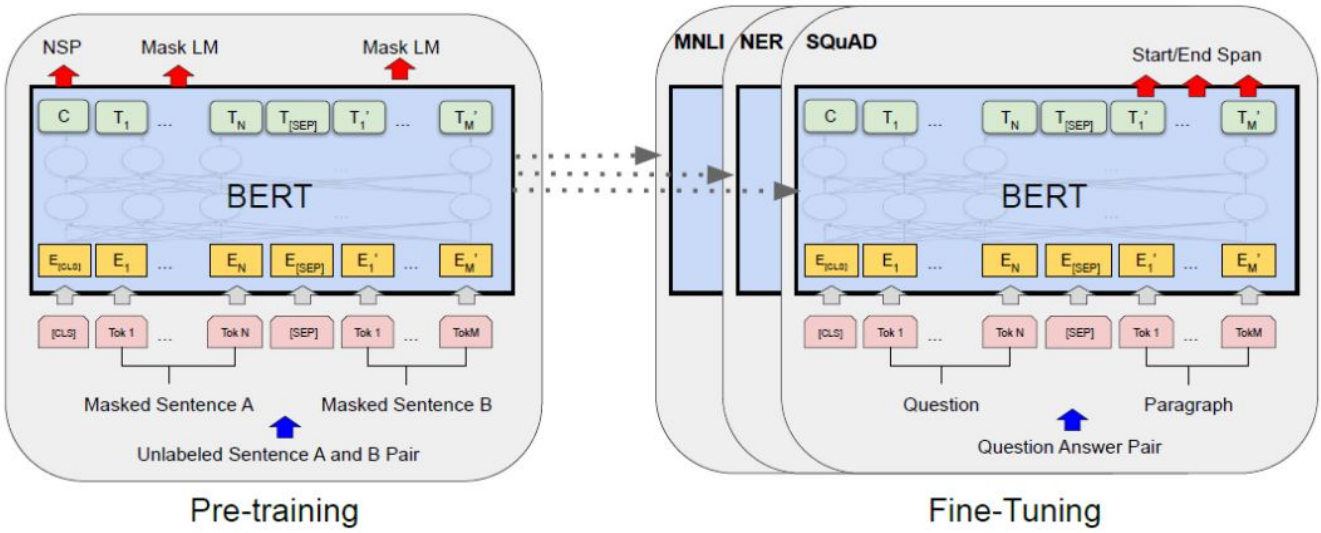


图8：BERT的Pre-training结构和Fine-Tuning结构

作者：何从庆，湖南大学计算机硕士，主要研究方向：机器学习与法律智能。

Github 主页：https://github.com/hecongqing

微信公众号：AI算法之心 ， 欢迎关注：

AI算法之心是一个介绍Python、PySpark、机器学习、自然语言处理、深度学习、算法竞赛的平台。不管你是刚入门的小白，还是资深的算法大佬，欢迎扫一扫下方的二维码与我们在AI的领域中一起学习成长！

阅读原文