

2021-2022学年秋季学期

自然语言处理

Natural Language Processing



授课教师：胡玥

助 教： 李运鹏

自然语言处理
Natural Language Processing

第 6 章 NLP基础任务

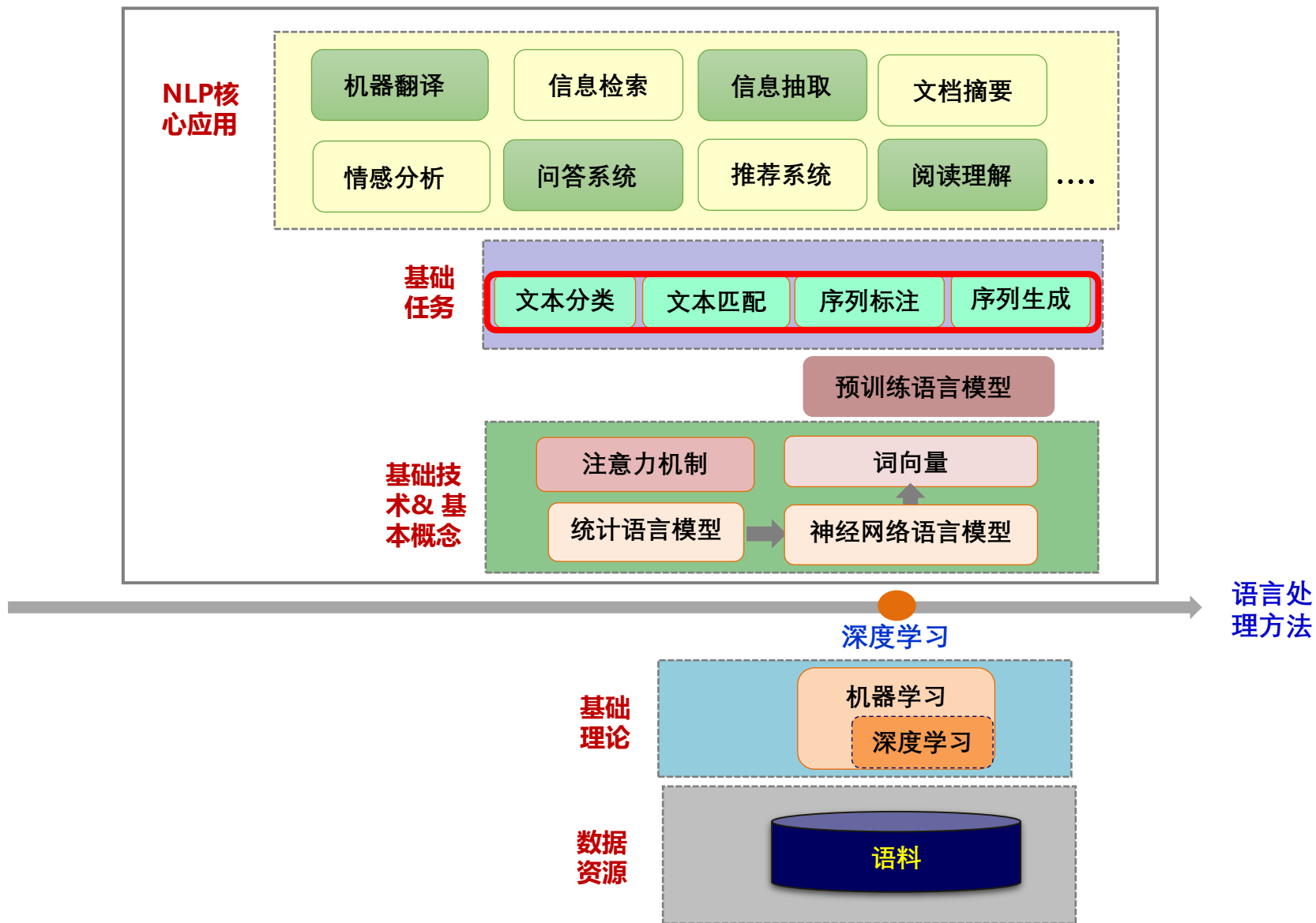
分类+匹配

1/3

授课教师：胡玥

授课时间：2021.10

基于深度学习的自然语言处理课程内容



内 容 提 要

6.1 文本分类

6.2 文本匹配

6.3 序列标注

6.4 序列生成

6.1 文本分类

■ 文本分类

本节内容:

1. 文本分类任务概述
2. 序列结构文本分类方法
3. 图结构文本分类方法
4. 文本分类评价指标
5. 常用数据集

1. 文本分类任务概述

■ 问题引入

如：关于电影《长津湖》有评论如下，问该影评的情感倾向？

《长津湖》中国影史上最大投资规模、参演人数最多的战争题材电影。

第一次生动塑造了以七连为代表的第九兵团这些鲜为人知的英雄群像，将这场气壮山河的战斗拍得惊心动魄感人肺腑。尤其是影片体现了更加彻底也更加现代战争观和历史观，达到了中国战争电影新高度。



文本分类问题定义： $f: X \rightarrow Y$

输入：X 句子/篇章

输出：X 所属类别 Y $Y \in \{\text{类别集合}\}$

1. 文本分类任务概述

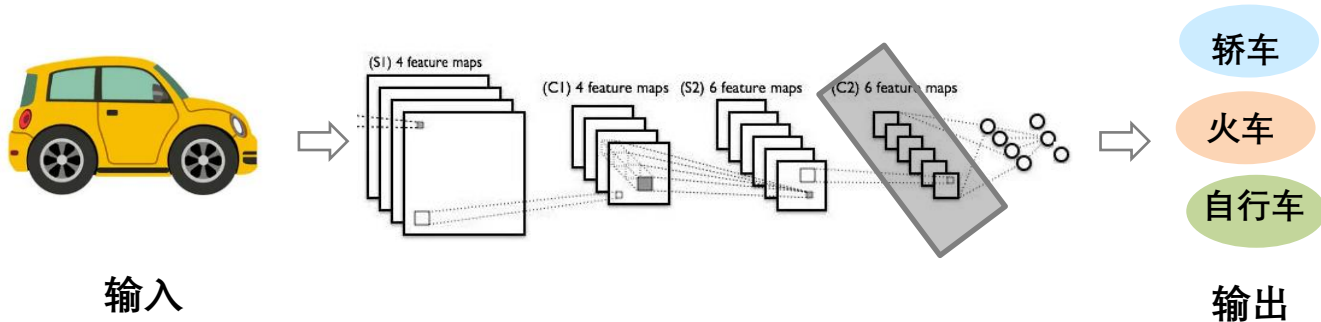
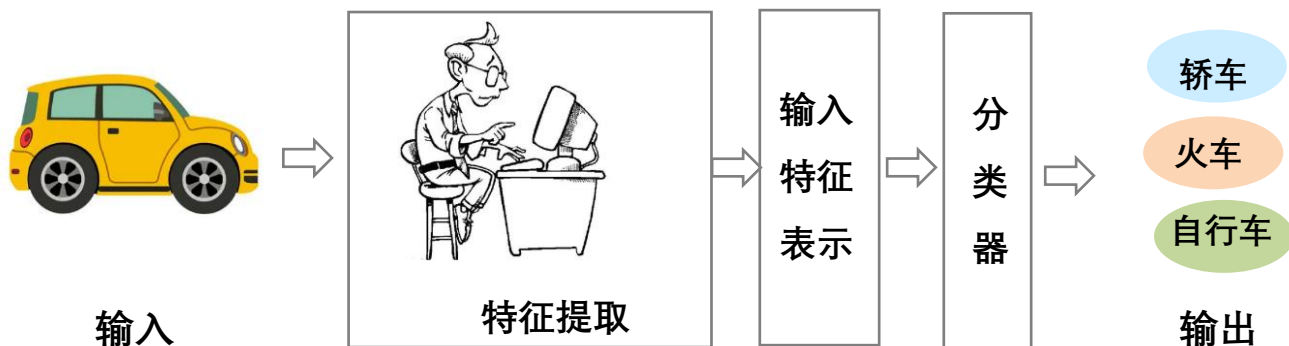
■ 文本分类

文本分类是NLP中的常见的重要任务之一，应用广泛，在很多领域发挥着重要作用，例如垃圾邮件过滤、舆情分析以及新闻分类等。



1. 文本分类任务概述

概率统计时代：特征工程+算法（Naive Bayes/SVM/LR/KNN……）



深度学习时代：自动获取特征（表示学习）端到端分类

1. 文本分类任务概述

■ 神经网络分类方法

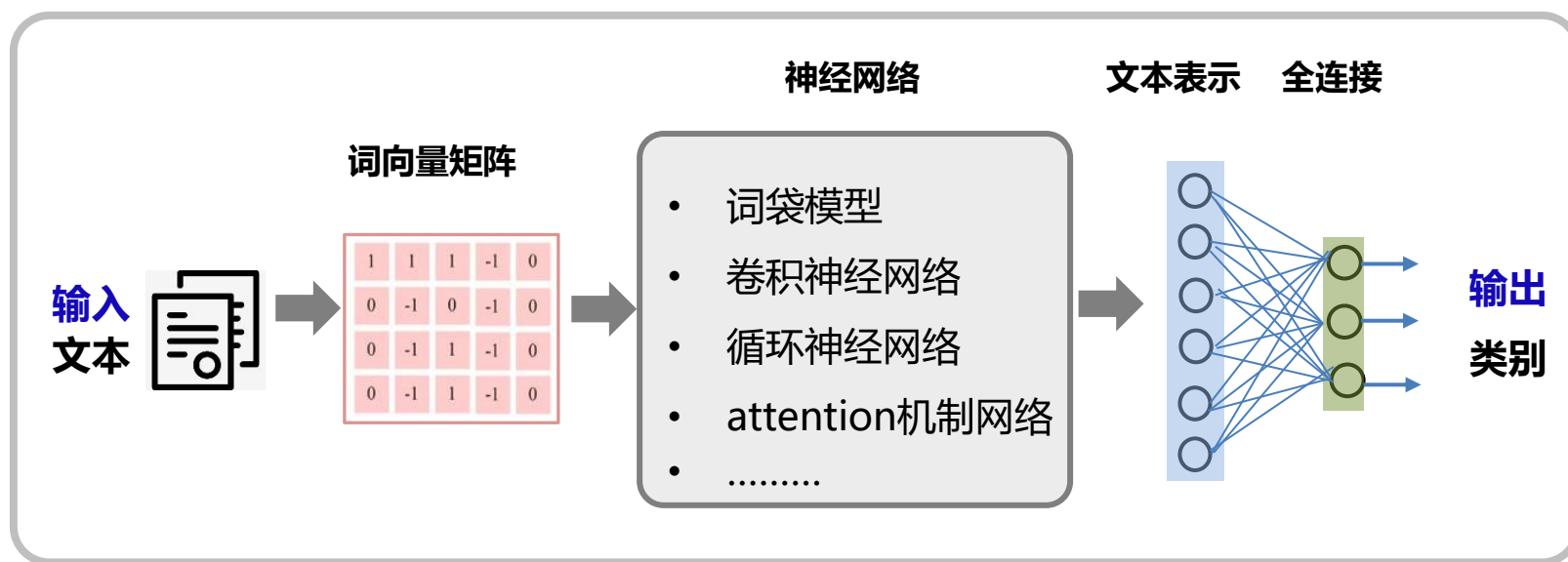
- ★ 基于词袋的文本分类
- ★ 基于卷积神经网络文本分类 (TextCNN/DPCNN/Char-CNN/VDCNN)
- ★ 基于循环神经网络文本分类 (TextRNN/TextRCNN)
- ★ 基于attention机制文本分类
- ★ 基于预训练模型的文本分类
- ★ 基于图卷积神经网络文本分类

也可以根据问题需要将上述方法结合形成混合模型

篇章级一般采用层次化的方法，先得到句子编码，然后以句子编码为输入，进一步得到篇章的表示

2. 序列结构文本分类方法

■ 序列结构文本分类框架

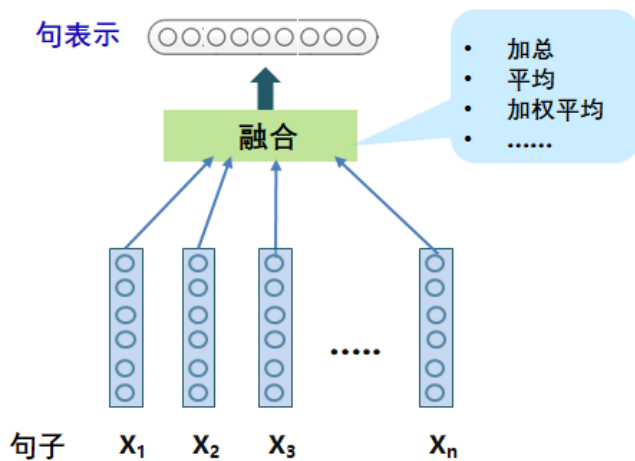


关键问题： 如何生成高质量的文本表示

2. 序列结构文本分类方法

★ 神经词袋模型 (Bag of words)

简单对文本序列中每个词嵌入进行平均/加总，作为整个序列的表示。
这种方法的缺点是丢失了词序信息。对于长文本，神经词袋模型比较有效。但是对于短文本，神经词袋模型很难捕获语义组合信息



2. 序列结构文本分类方法

FastText

Facebook提出了一种简单而有效的文本分类和表示学习方法可以在不到10分钟的时间内使用标准的多核CPU对超过10亿个单词进行快速文本训练，并在不到一分钟的时间内对312K类中的50万个句子进行分类。

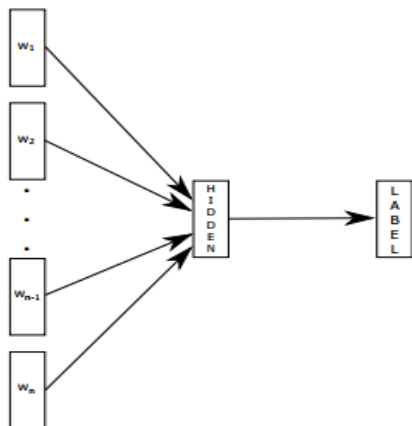


Figure 1: Model architecture for fast sentence classification.

输出层： 类别较少用softmax，
类别较多用hierarchical softmax

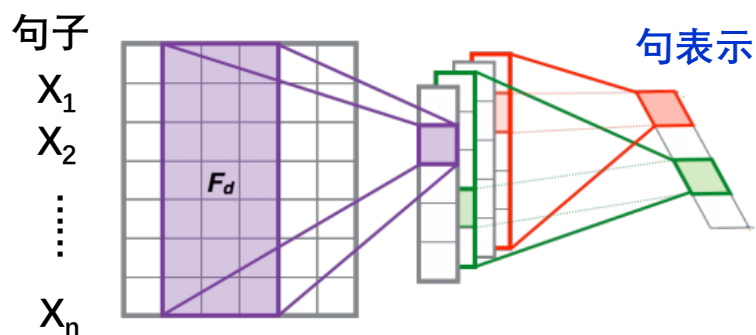
隐藏层： 将每个词向量相加取平均值

输入： Document中的每个词的词向量

2. 序列结构文本分类方法

★ 卷积神经网络模型 (Convolutional Neural Network)

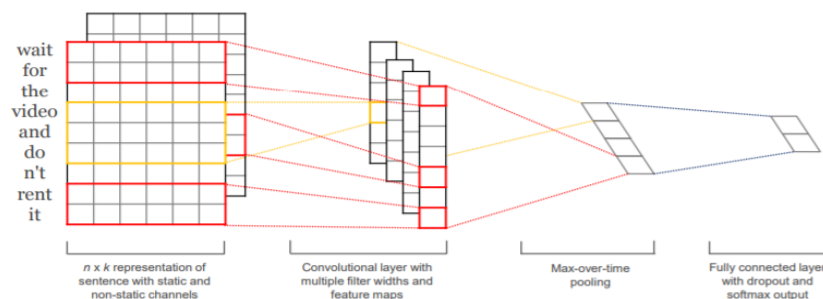
通过多个卷积层和子采样层，抽取序列的 n-gram 特征信息，最终将得到特征信息合并成一个固定长度的向量作为整个序列表示。



2. 序列结构文本分类方法

TextCNN

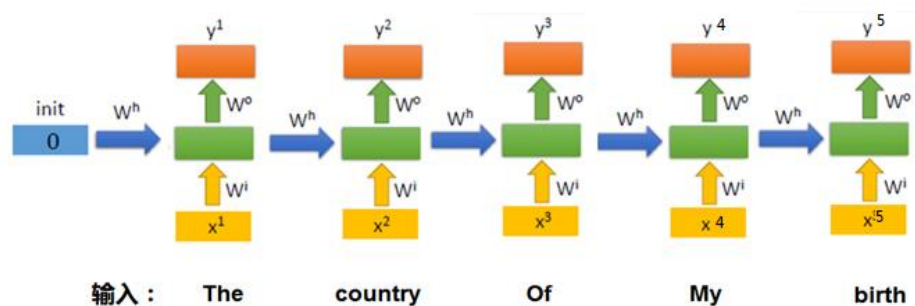
- 两套词向量构造出的句子矩阵作为两个通道
CNN-rand: 所有的词向量都随机初始化（模型训练时做参数进行训练）
CNN-static: 即用word2vec预训练好的向量（模型训练过程中不更新）
- 两个通道分别采用不同的卷积核进行卷积构建n-gram特征；然后将Pooling后的特征进行连接作为句表示；将生成的句表示做全连接分类
- 在训练时，只更新CNN-rand一组词向量，保持另外一组不变



2. 序列结构文本分类方法

★ 循环神经网络 (Recurrent Neural Network)

将文本序列看作时间序列，不断更新，最后得到整个序列的表示。这种表示中包含的是序列的顺序信息。

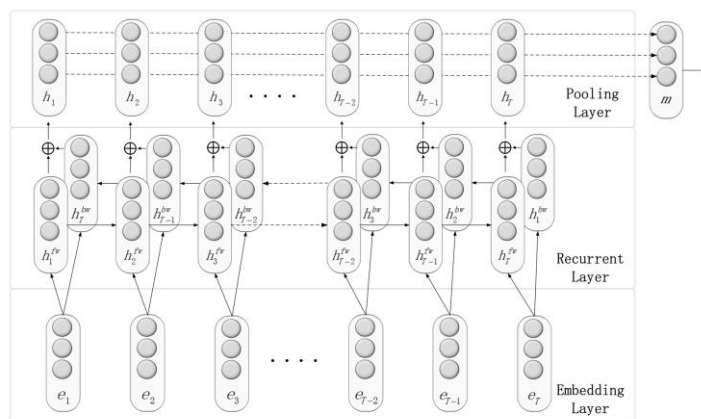


问题：RNN属于“biased model”，一个句子中越往后的词重要性越高，这有可能影响最后的分类结果，因为对句子分类影响最大的词可能处在句子任何位置。

2. 序列结构文本分类方法

BiLSTM

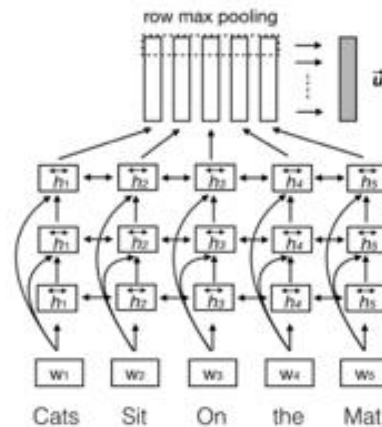
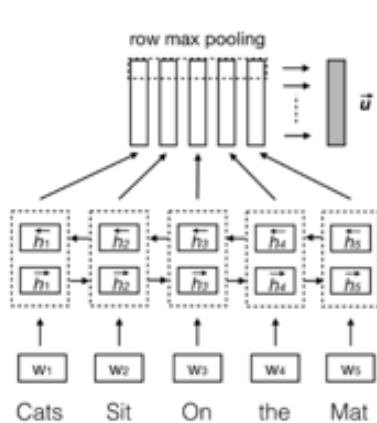
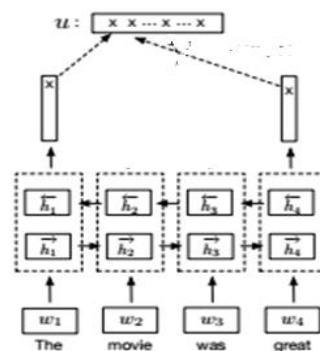
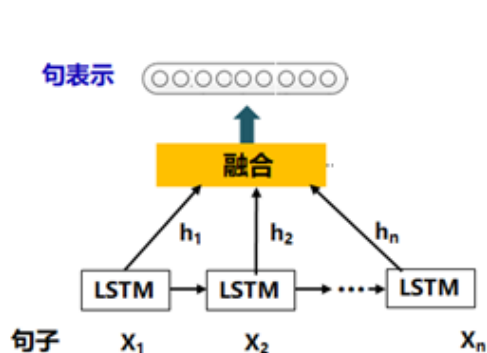
- 词向量, 通过word2vec算法在通用语料上预训练;
- 分别从两个方向上, 通过LSTM分别编码当前词的上文信息和下文信息;
- 然后通过求和操作, 得到融合上下文的词表示;
- 然后将pooling之后的向量作为句子表示, 用作全连接分类。



2. 序列结构文本分类方法

基于RNN的句表示

可根据需要形成各种基于RNN的句表示



2. 序列结构文本分类方法

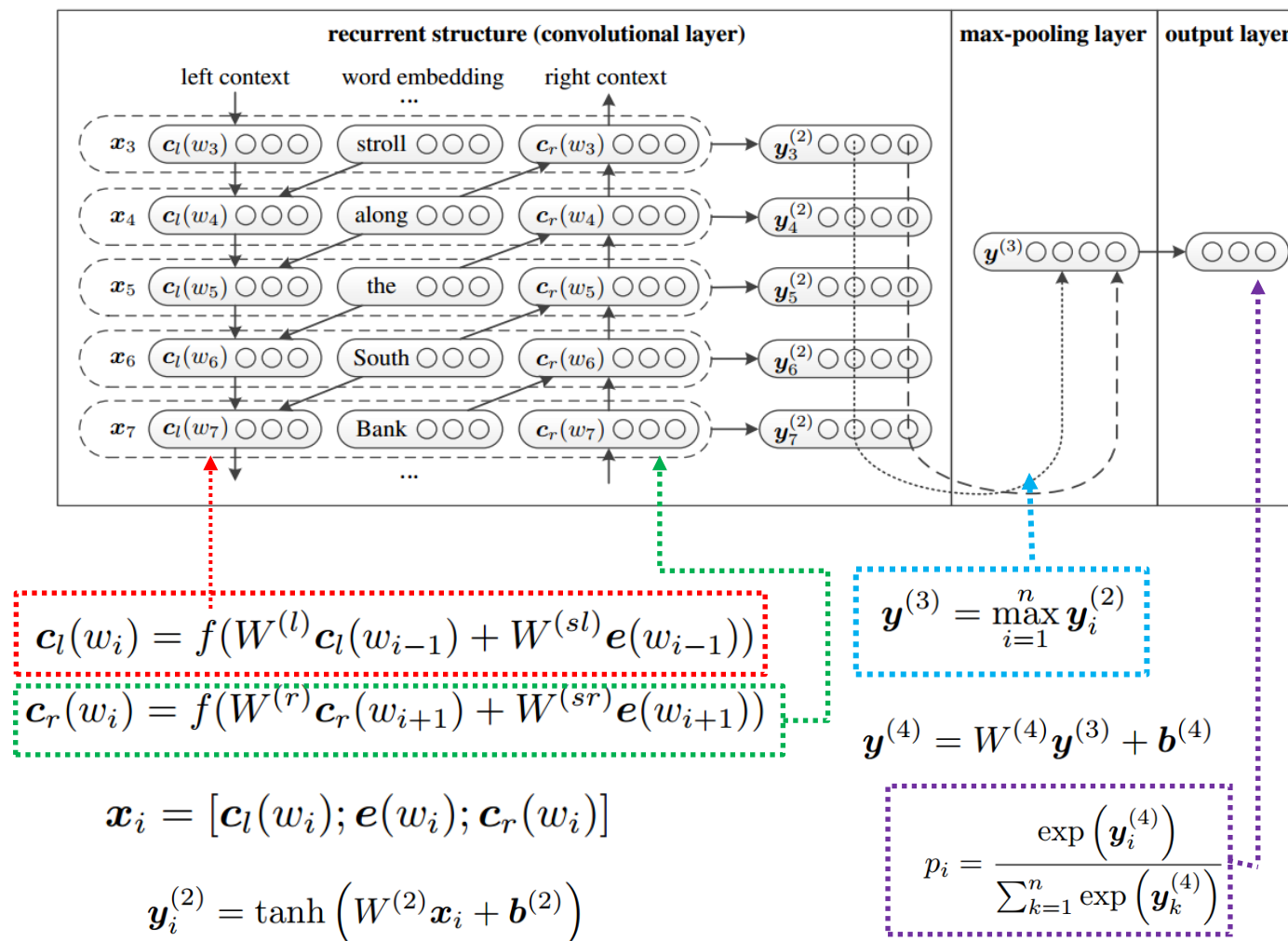
循环+卷积神经网络模型

- RNN擅长处理序列结构，能够考虑到句子的上下文信息。
- CNN属于无偏模型，能够通过最大池化获得最重要的特征。

结合二者的优势生成上下文窗口信息的卷积网络

2. 序列结构文本分类方法

Recurrent Convolutional Neural Networks



2. 序列结构文本分类方法

★ 注意力神经网络 (Attention Network)

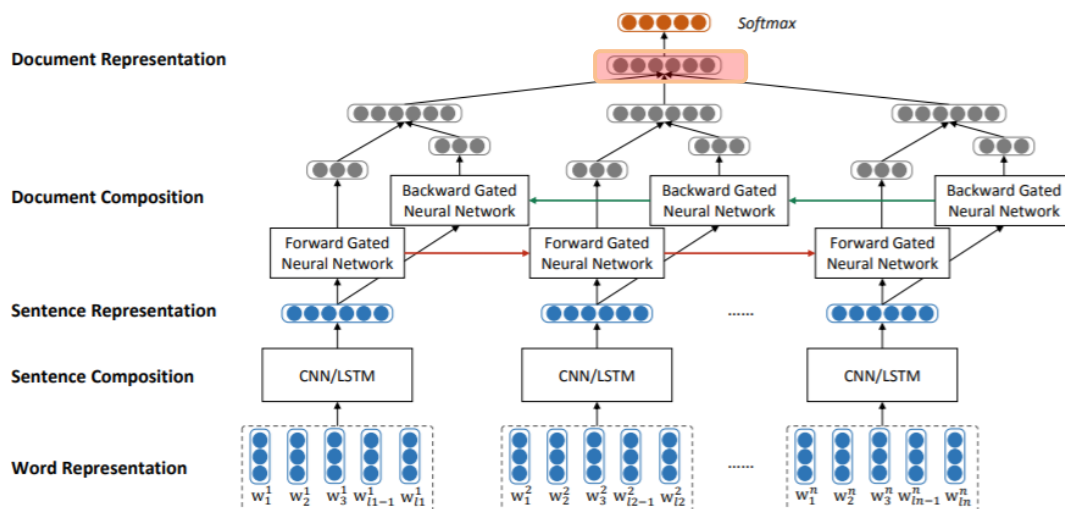
通过注意力机制对序列进行编码，最后得序列的表示，这种表示包含的是词与词之间的关联关系



2. 序列结构文本分类方法

LSTM/CNN-GRU (篇章级-混合模型)

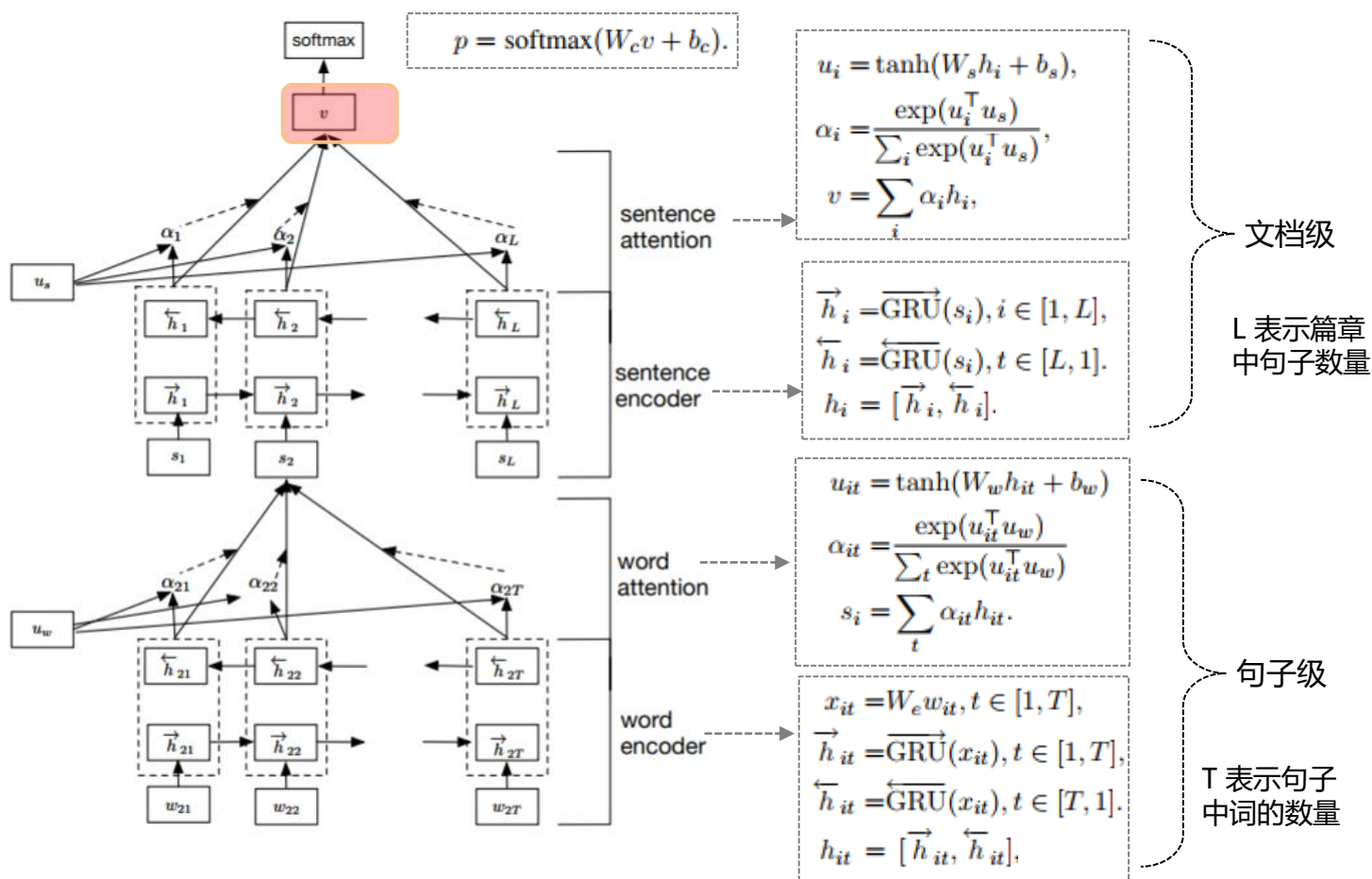
- 篇章中所有句子的词向量矩阵作为输入
- 用CNN/LSTM形成句子级向量表示
- 由句向量用双向RNN 形成每句的带有上下句子信息的句子表示
- 由句向量形成篇章级向量表示
- 用篇章级向量做分类



2. 序列结构文本分类方法

HAN (篇章级- Attention 模型)

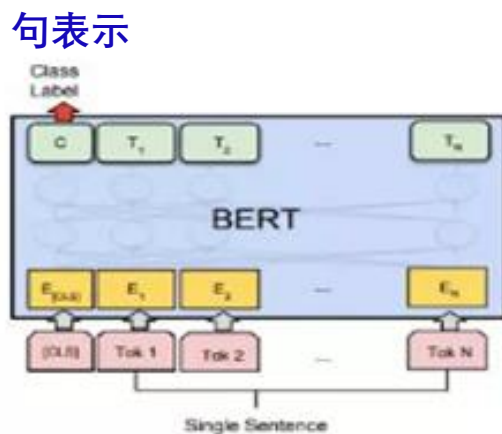
从句子级和文档级两个层次引入Attention机制，可识别分类决策的重要单词和句子



2. 序列结构文本分类方法

★ 基于预训练模型（Bert）

通过预训练模型形成句表示，然后将该句表示作为分类的输入



输入: 句子 $\{w_1, w_2, \dots, w_n\}$

3. 图结构文本分类方法

★ 图卷积神经网络文本分类

根据任务对原文本加入附加信息并构建原文本与附加信息的关系图（将附加的结构信息融入文本），然后利用图卷积的方法提取文本有效的特征表示

图卷积文本分类步骤：

1. Graph 构建

对原文本按照附加信息的不同构建不同的图结构。附加信息可以是词的近义信息，共现信息，先验知识信息等

2. 文本Graph节点特征表示

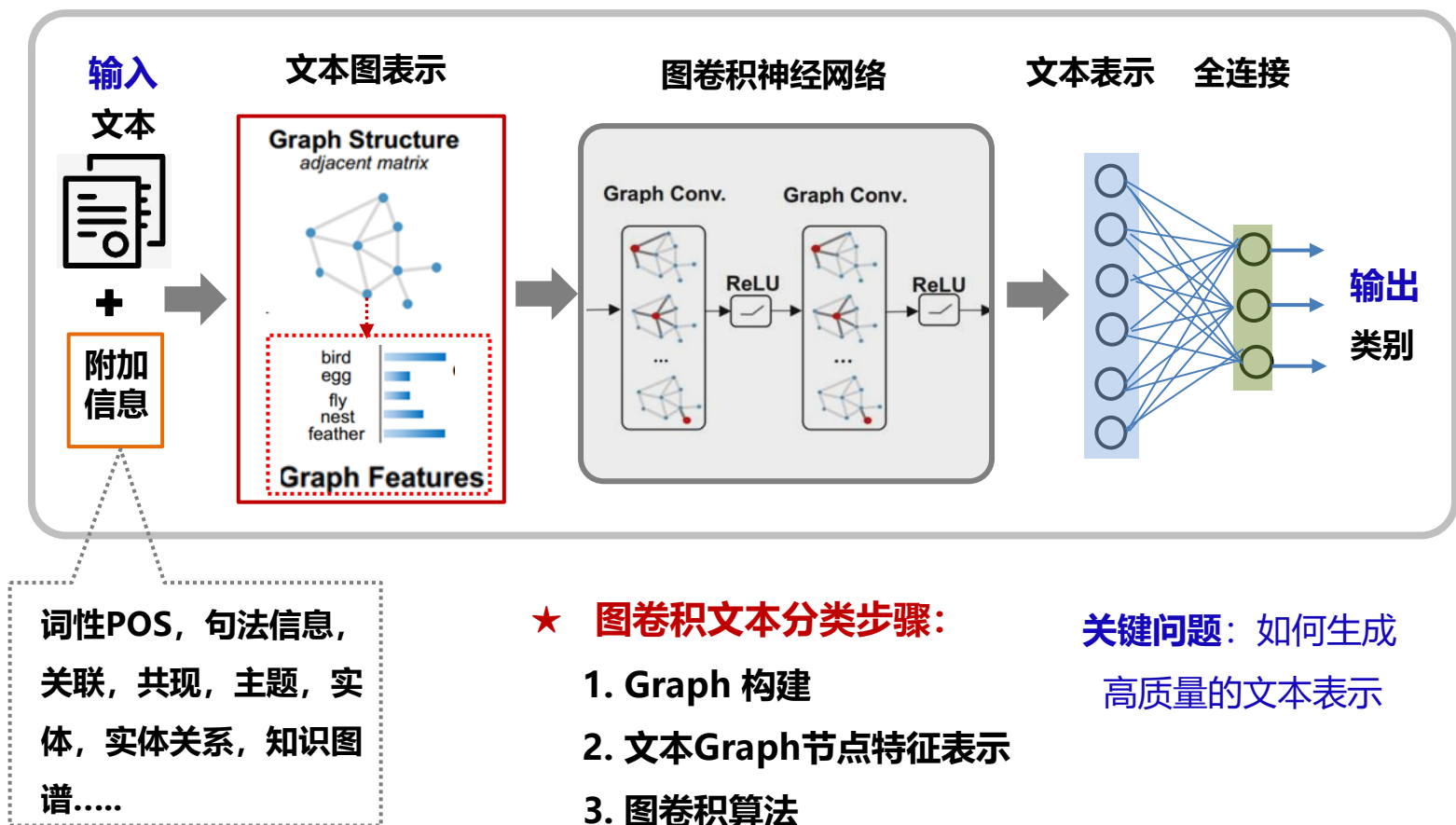
在图卷积中结点可以根据任务需要采用不同的结点表示方法。如，一般词向量，ELMO 词嵌入，Bi-LSTM 词向量嵌入，词袋词频等方法

3. 图卷积算法

构建好输入图和图上结点表示后，可以根据不同的任务构建不同的图卷积算法。如，一般图卷积，加入注意力机制的图卷积等

3. 图结构文本分类方法

■ 图文本分类框架



3. 图结构文本分类方法

例1. 对文本进行图卷积分类

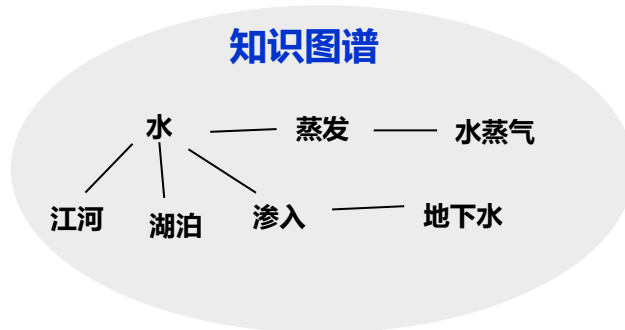
1. Graph 构建

文本

地面上的水不断蒸发变成了水蒸气，有的水在地面上汇成江河、湖泊，另外一些水渗入了地下，称为地下水。



查询图谱



地面上的水不断蒸发变成了水蒸气，有的水在地面上汇成江河、湖泊，另外一些水渗入了地下，称为地下水。



邻接表形式图结构

注：附加信息可以是词性POS，句法信息，关联，共现，主题，实体，实体关系，知识图谱 等

3. 图结构文本分类方法

2. 文本Graph节点特征表示

地面上的水不断蒸发变成了水蒸气，有的水在地面上汇成江河、湖泊，另外一些水渗入了地下，称为地下水。

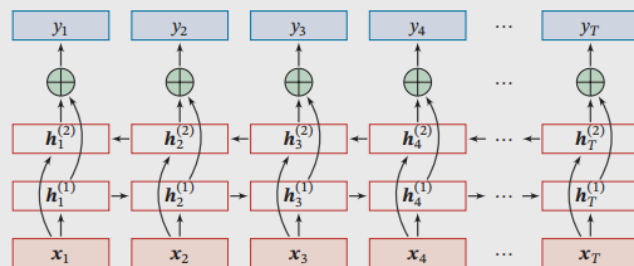
结点表示

表示方法

地面
水
不断
蒸发
变成
水蒸气
.....



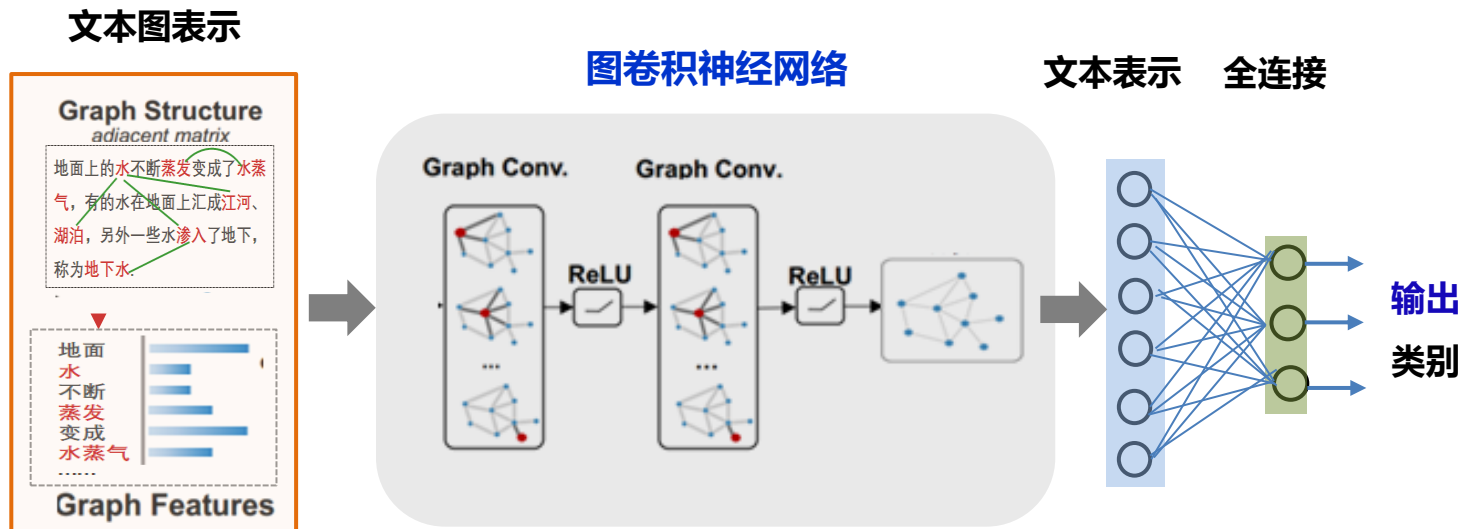
双向LSTM



注：结点表示可根据具体问题需要采用双向RNN，Elmo词嵌入等各种表示方法

3. 图结构文本分类方法

3. 图卷积算法



注：可以根据不同的任务构建不同的图卷积算法。如，图卷积网络文本分类算法，图卷积网络多层加权分类算法等

3. 图结构文本分类方法

例2. 短文本分类模型 Heterogeneous Graph Attention Networks (HGAT)

任务：对短文本进行分类

如：



问题：短文本包含信息量少，用一般神经网络提取特征效果有限

解决：引入附加信息增强短文本信息然后采用图卷积方法提取有效特征

3. 图结构文本分类方法

1. Graph 构建

附加信息：Topics 和 Entities

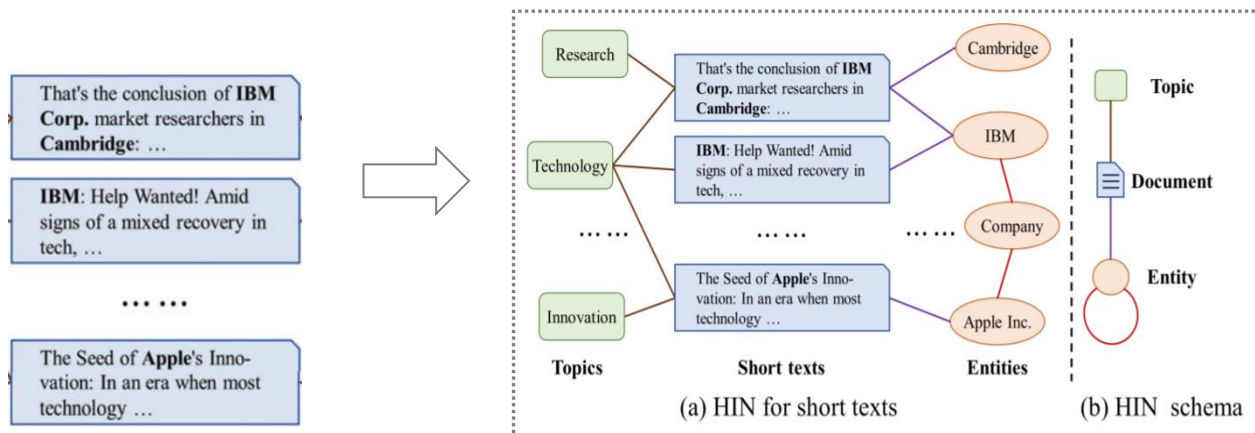
- Topics: 使用LDA的方式挖掘短文本的主题分布，然后选择 Top P
- Entities: 在文本中识别实体，然后使用实体链接工具TAGME同时考虑实体之间的关系

建立关联关系（构建图的边）：

- 如果某篇文档属于主题，则文档与主题词间建立连接
- 如果文档包含实体，则文档与实体间建立连接
- 实体之间如果距离（cosine similarity）在设定阈值内，建立实体间的连接

3. 图结构文本分类方法

1. Graph 构建

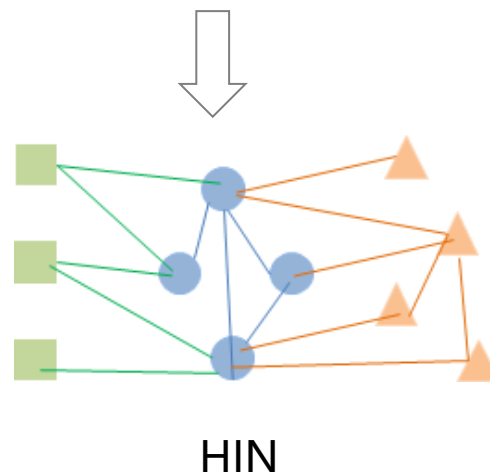


邻接表图结构 HIN (异质)

HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; \mathcal{V} 为结点, \mathcal{E} 为边

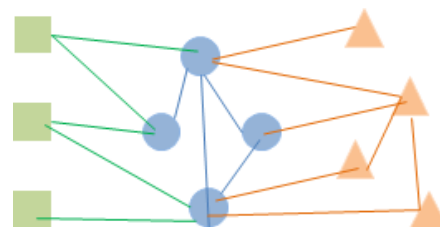
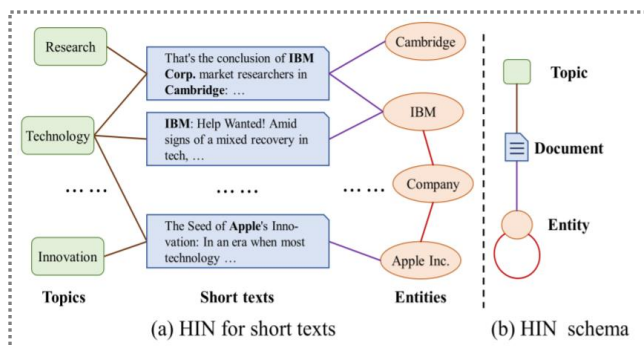
$\mathcal{V} = D \cup T \cup E$ (结点异构)

- short text $D = \{d_1, \dots, d_m\}$
- topics $T = \{t_1, \dots, t_K\}$
- entities $E = \{e_1, \dots, e_n\}$



3. 图结构文本分类方法

2. 文本Graph节点特征表示



HIN

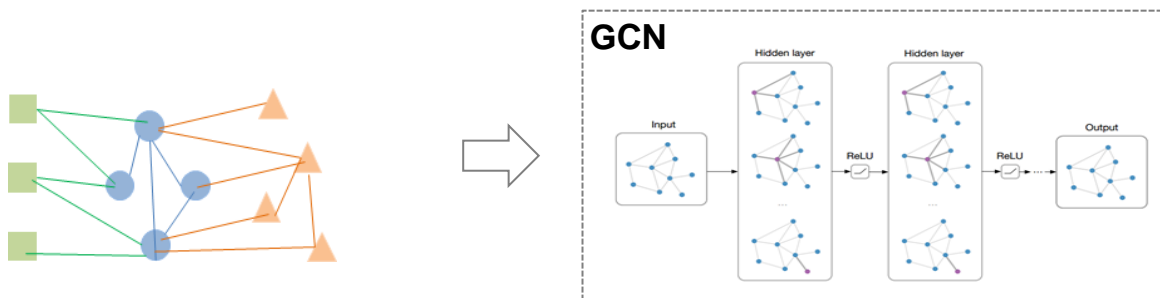
- short text D : 每个短文本 x_d 用TF-IDF 向量表示表示
- topics T : 每个主题用词分布 $x_t = \{\theta_i\}_{i=[1,w]}$ 表示
- entities E : 每个实体用其词嵌入 x_v 和它在Wikipedia中解释文本的 TF-IDF 向量连接表示

问题： 结点异质，表示不同，如何构建结点表示？

3. 图结构文本分类方法

3. 图卷积算法

方法一：将所有异质结点特征接连，形成同质特征结点,然后采用普通图卷积方法



HIN

$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \tilde{A}_{\tau} \cdot H_{\tau}^{(l)} \cdot W_{\tau}^{(l)}\right)$$

其中: $\tilde{A}_{\tau} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}_{\tau}|}$ 是邻接矩阵 \tilde{A} 的子矩阵,

行: 所有结点; 列: 类型为 τ 的邻接结点

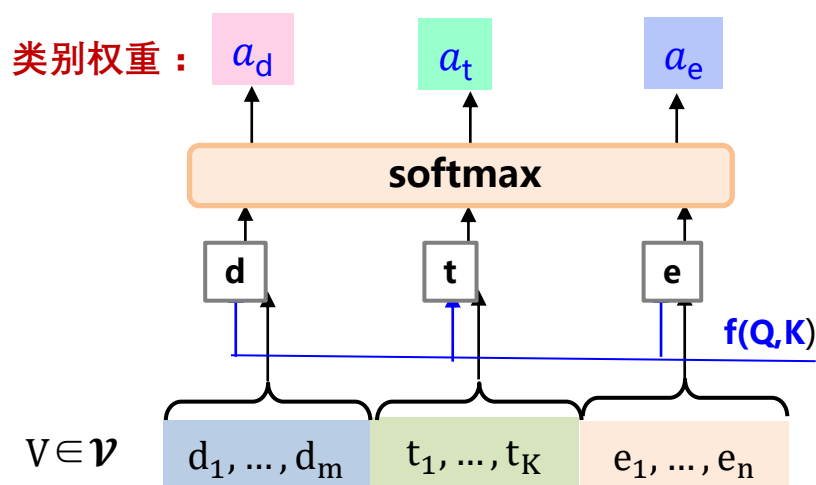
$W_{\tau}^{(l)} \in \mathbb{R}^{q^{(l)} \times q^{(l+1)}}$ 类型为 τ 的结点参数

问题: 无法区分不同信息之间的差异。 **改进:** 采用注意力机制

3. 图结构文本分类方法

方法二：采用双重注意力机制（类型级 + 结节点级）的异质图卷积方法

1. 类型级注意力：



Type level attention

Q: 结点 $V \in \mathcal{V}$

K: (与 v 邻接的 τ 类结点聚集)

$$h_\tau = \sum_{v'} \tilde{A}_{vv'} h_{v'}$$

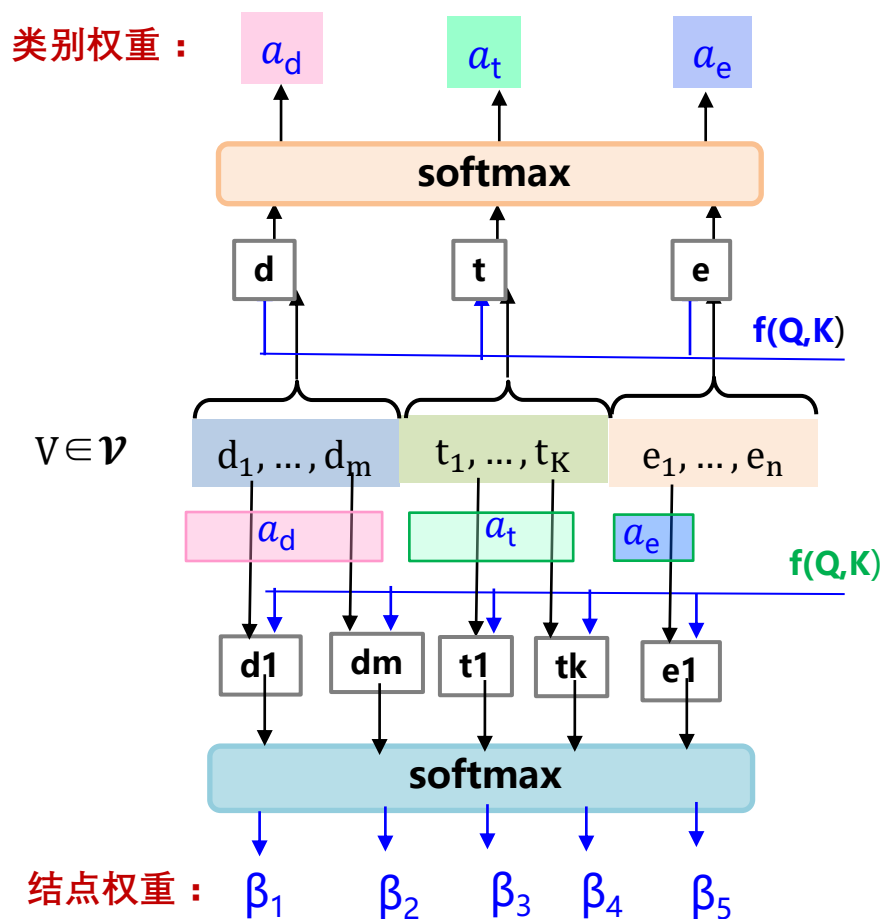
$$f(Q, K): a_\tau = \sigma(\mu_\tau^T \cdot [h_v || h_\tau])$$

$$\text{类型权重: } \alpha_\tau = \frac{\exp(a_\tau)}{\sum_{\tau' \in \mathcal{T}} \exp(a_{\tau'})}$$

μ_τ : attention vector for the type τ

3. 图结构文本分类方法

2. 结点级注意力:



Node level attention

Q: 结点 $V \in \mathcal{V}$

K: $v' \in \mathcal{N}_v$

f(Q,K): $b_{vv'} = \sigma(\nu^T \cdot \alpha_{\tau'}[h_v || h_{v'}])$

结点权重 : $\beta_{vv'} = \frac{\exp(b_{vv'})}{\sum_{i \in \mathcal{N}_v} \exp(b_{vi})}$

ν : attention vector for the node-level

3. 图结构文本分类方法

引入双重注意力机制（类型级 + 结节点级）后：

$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \tilde{A}_{\tau} \cdot H_{\tau}^{(l)} \cdot W_{\tau}^{(l)}\right)$$



$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \mathcal{B}_{\tau} \cdot H_{\tau}^{(l)} \cdot W_{\tau}^{(l)}\right)$$

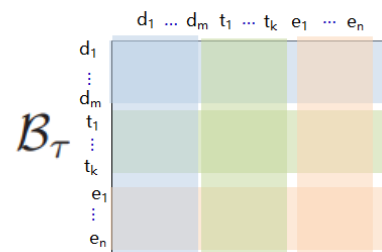
对于每个与 v 相邻的结点不但考虑了其类型的权重，还考虑了同一类型中各结点的权重

\mathcal{B}_{τ} 矩阵：注意力矩阵， v^{th} 行
 v'^{th} 列为 $\beta_{vv'}$

$$\alpha_{\tau} = \frac{\exp(a_{\tau})}{\sum_{\tau' \in \mathcal{T}} \exp(a_{\tau'})}$$

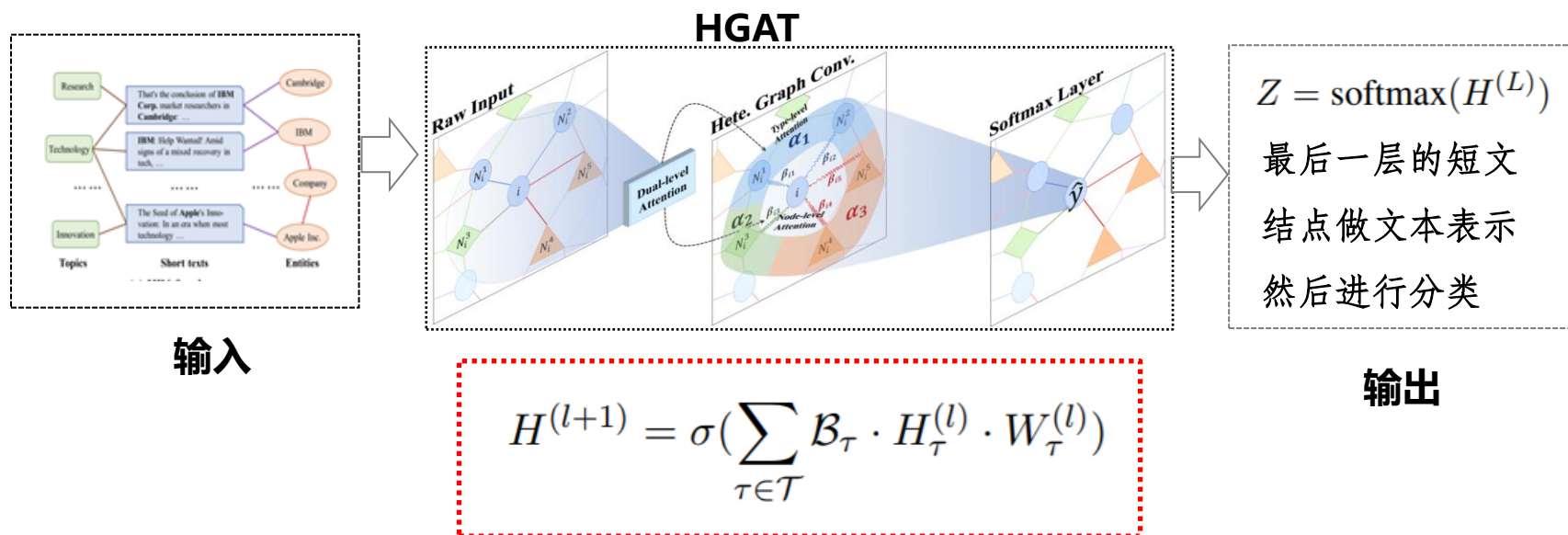
$$b_{vv'} = \sigma(\nu^T \cdot \alpha_{\tau'} [h_v || h_{v'}])$$

$$\beta_{vv'} = \frac{\exp(b_{vv'})}{\sum_{i \in \mathcal{N}_v} \exp(b_{vi})}$$



3. 图结构文本分类方法

双重注意力机制异质图卷积分类模型HGAT



HGAT输入: HIN

HGAT输出: 短文表示

HGAT参数: $W_{\tau}^{(l)}, \mu_{\tau}^T, \nu^T$

3. 图结构文本分类方法

模型学习:

优化目标: 最小化交叉熵损失函数

$$\mathcal{L} = - \sum_{i \in D_{\text{train}}} \sum_{j=1}^C Y_{ij} \cdot \log Z_{ij} + \eta \|\Theta\|_2$$

where C is the number of classes, D_{train} is the set of short text indices for training, Y is the corresponding label indicator matrix, Θ is model parameters, and η is regularization factor. For model optimization, we adopt the gradient descent algorithm.

3. 图结构文本分类方法

实验结果:

Dataset	SVM +TFIDF	SVM +LDA	CNN -rand	CNN -pretrain	LSTM -rand	LSTM -pretrain	PTE	TextGCN	HAN	HGAT
AGNews	57.73	65.16	32.65	67.24	31.24	66.28	36.00	<u>67.61</u>	62.64	72.10*
Snippets	63.85	63.91	48.34	77.09	26.38	75.89	63.10	<u>77.82</u>	58.38	82.36*
Ohsumed	41.47	31.26	35.25	32.92	19.87	28.70	36.63	<u>41.56</u>	36.97	42.68*
TagMyNews	42.90	21.88	28.76	57.12	25.52	<u>57.32</u>	40.32	54.28	42.18	61.72*
MR	56.67	54.69	54.85	58.32	52.62	<u>60.89</u>	54.74	59.12	57.11	62.75*
Twitter	54.39	50.42	52.58	56.34	54.80	<u>60.28</u>	54.24	60.15	53.75	63.21*

Test accuracy (%) of different models on six standard datasets. The second best results are underlined.

The note * means our model significantly outperforms the baselines based on t -test ($p < 0.01$).

Dataset	GCN -HIN	HGAT w/o ATT	HGAT -Type	HGAT -Node	HGAT
AGNews	70.87	70.97	71.54	71.76	72.10*
Snippets	76.69	80.42	81.68	81.93	82.36*
Ohsumed	40.25	41.31	41.95	42.17	42.68*
TagMyNews	56.33	59.41	60.78	61.29	61.72*
MR	60.81	62.13	62.27	62.31	62.75*
Twitter	61.59	62.35	62.95	62.45	63.21*

Test accuracy (%) of our variants.

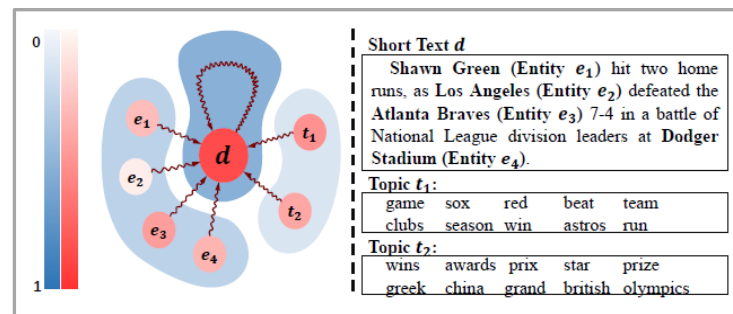


Figure 5: Visualization of the dual-level attention including node-level attention (shown in red) and type-level attention (shown in blue). Each topic t is represented by top 10 words with highest probabilities.

4. 文本分类评价指标

■ 二分类（以所关注的类为正类，其他类为负类）：

- TP——将正类预测为正类数；
- FP——将负类预测为正类数；
- FN——将正类预测为负类数；
- TN——将负类预测为负类数；

◆ 分类准确率（Accuracy）

- 对于给定的测试集，分类器正确分类的样本数/总样本数
- $Acc = (TP + TN) / (TP + FN + FP + TN)$

◆ 精确率（Precision）

- 预测正确的正例数据占预测为正例数据的比例
- $P = TP / (TP + FP)$

◆ 召回率（Recall）

- 预测正确的正例数据占实际为正例数据的比例
- $R = TP / (TP + FN)$

4. 文本分类评价指标

◆ 精确度和召回率的调和均值 (F-Score)

- $$F_{\beta} = \frac{(1+\beta^2)PR}{\beta^2P+R} = \frac{(1+\beta^2)TP}{(1+\beta^2)TP+\beta^2FP+FN}$$

β 取值反映了精确率和召回率在性能评估中的相对重要性。

- 通常情况下，取值为1 (F1) 。

F1又称平衡F分数，表明精确率和召回率一样重要。

$$F1 = 2TP/(2TP + FP + FN)$$

- 也可根据评估的重要性选取合适的 β ：

当 $\beta=2$ ，表明评估时，召回率的重要性比精确率高；

当 $\beta=0.5$ ，表明评估时，精确率的重要性比召回率高；

4. 文本分类评价指标

■ 多分类评价指标（假设有n个类别）

◆ 准确率（Accuracy）

对于给定的测试集，n个类预测正确的样本总数/总样本数

◆ 宏平均（Macro Averaged）

对所有类别的每一个统计指标值的算数平均值，分别称为宏精确率（Macro-Precision），宏召回率（Macro-Recall），宏F值（Macro-F Score）

$$\bullet \quad P_{marco} = \frac{1}{n} \sum_{i=1}^n P_i, \quad R_{marco} = \frac{1}{n} \sum_{i=1}^n R_i$$

$$\bullet \quad F_{marco} = \frac{1}{n} \sum_{i=1}^n F_i$$

4. 文本分类评价指标

◆ 微平均 (Mirco Averaged)

对每个样本进行P、R的统计，计算微精确率 (Micro-Precision) ，微召回率 (Micro-Recall) ，微F值 (Micro-F Score) 。

- $$P_{micro} = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i},$$
- $$R_{micro} = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i}$$
- $$F_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}}$$

Macro-averaging赋予每个类相同的权重，Micro-averaging赋予每个样本决策相同的权重

度量分类器对大类判别的有效性选择微平均，度量分类器对小类判别的有效性选择宏平均

5. 文本分类常用数据集

文本分类 (Sentiment Analysis)

数据集	No. L	Train	Dev	Test	链接
Yelp-2	2	650,000	-	50,000	https://www.kaggle.com/yelp-dataset/yelp-dataset
Yelp-5	5	560,000	-	38,000	
IMDb	2	25,000		25,000	www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
Movie Review	2	10,662	-	-	https://aclanthology.org/W02-1011/
SST-2	2	8,544	1,101	2,210	https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf
SST-5	5	6,920	872	1,821	
MPQA	2	10,606	-	-	https://www.cs.cornell.edu/home/cardie/papers/lre05withappendix.pdf
Amazon-2	2	3600000	-	400000	https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products
Amazon-5	5	3000000	-	650000	

5. 文本分类常用数据集

文本分类 (News/Topic Classification)

数据集	No. L	Train	Dev	Test	链接
AG News	4	120,000	-	7,600	http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
20 Newsgroups	20	18,821	-	-	http://qwone.com/~jason/20Newsgroups/
Reuters news	90	7,769	-	3,019	https://martin-thoma.com/nlp-reuters/
DBpedia	14	560,000	-	70,000	http://www.semantic-web-journal.net/system/files/swj558.pdf
EUR-Lex	3,956	19,314	-	-	https://link.springer.com/content/pdf/10.1007%2F978-3-540-87481-2_4.pdf
Ohsumed	23	7,400	-	-	https://davis.wpi.edu/xmdv/datasets/ohsumed.html

内 容 提 要

6.1 文本分类

6.2 文本匹配

6.3 序列标注

6.4 序列生成

6.2 文本匹配

■ 文本匹配

本节内容：

1. 文本匹配概述
2. 文本匹配方法

1. 文本匹配任务概述

■ 文本匹配

文本匹配是一个很宽泛的概念，很多自然语言处理的任务都会涉及文本匹配问题，根据自然语言处理任务的不同，“匹配”的定义可能非常不同。

例1：两个句子“感冒了是否要吃药”和“感冒了要吃什么药” 问：两个句子是否表达同样的意思？

该问题的句子**匹配**是计算二个句子相似度，可建模为二分类问题

例2：两个句子“我正在上海旅游”和“我正在八达岭长城” 问：这两句话是什么关系？

该问题属于“文本蕴含识别”问题，关系有三种（蕴含/矛盾/中立）

该**匹配**是判别二个句子之间的关系 是多分类问题

1. 文本匹配任务概述

这些任务的共性问题是研究两段文本之间关系的问题，因此我们广义的将研究两段文本间关系的问题定义为“文本匹配”问题，匹配的具体含义根据任务的不同有不同的定义。一般可建模为“分类”和“排位”二类问题

与文本匹配相关的NLP任务

Tasks	Text 1	Text 2	Objective
Paraphrase Identification	string 1	string 2	classification
Textual Entailment	text	hypothesis	classification
Question Answer	question	answer	classification/ranking
Conversation	dialog	response	classification/ranking
Information Retrieval	query	document	ranking

https://blog.csdn.net/Ding_xiaofei

1. 文本匹配任务概述

1.复述识别（paraphrase identification）

又称释义识别，是判断两段文本是不是表达了同样的语义，这一类场景一般建模成分类问题。

2.文本蕴含识别（Textual Entailment）

给定一个前提文本（text），根据这个前提去推断假说文本（hypothesis）与文本的关系，关系有：蕴含关系（entailment），矛盾关系（contradiction），蕴含关系（entailment）。这一类场景一般建模成多分类问题。

3.问答（QA）

根据Question在段落或文档中查找Answer，这类场景常常会被建模成分类问题；还有一类是根据Question从若干候选中找出正确答案，这类场景常常会被建模成排位（ranking）问题。

1. 文本匹配任务概述

4.对话 (Conversation)

与QA 类似，但是比QA更复杂，由于引入了历史轮对话，需要考虑在历史轮的限制下回复是否合理。一般建模为分类或排位问题。

5.信息检索 (IR)

信息检索是一个更为复杂的任务，往往会有Query—Tittle，Query—Document的形式（Query可能是一个Document）检索需要计算相似度和排序一般建模为排位问题。

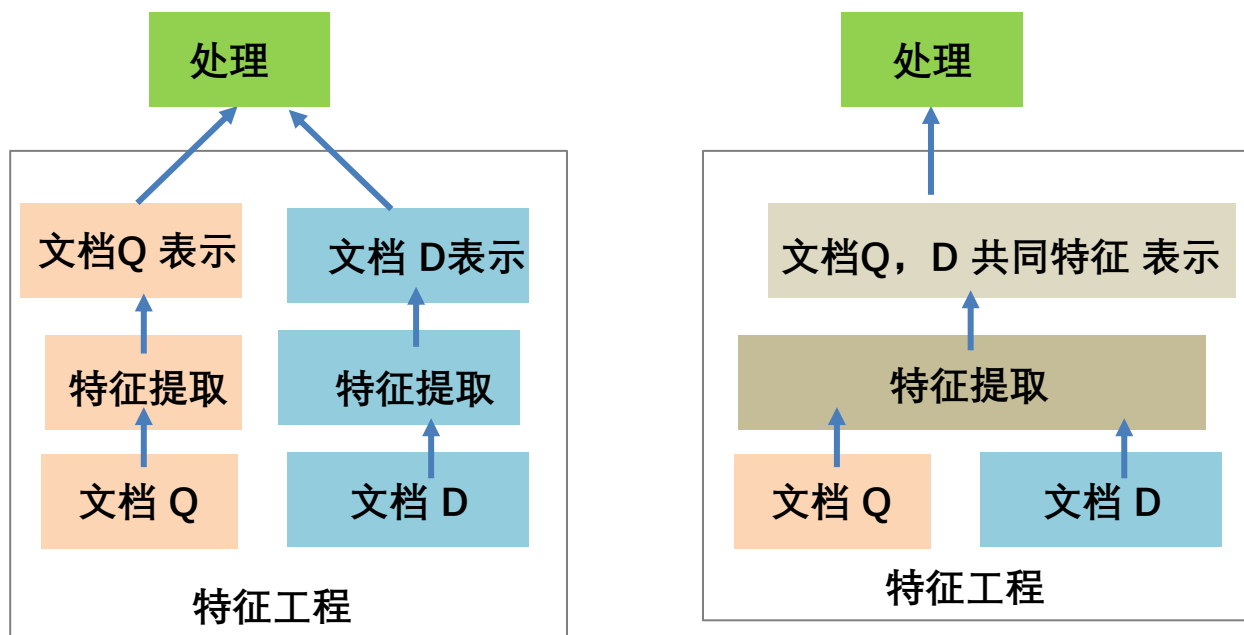
1. 文本匹配任务概述

■ 匹配方法

★ 规则方法：不同任务需要专门构建特征规则

★ 统计方法：特征工程+算法（PRanking / margin/ SVM/LR……）

以上二种传统文本匹配方法主要集中在人工定义特征之上的关系学习，焦点在于如何人工提取的特征和设置合适的文本匹配学习算法来学习到最优的匹配模

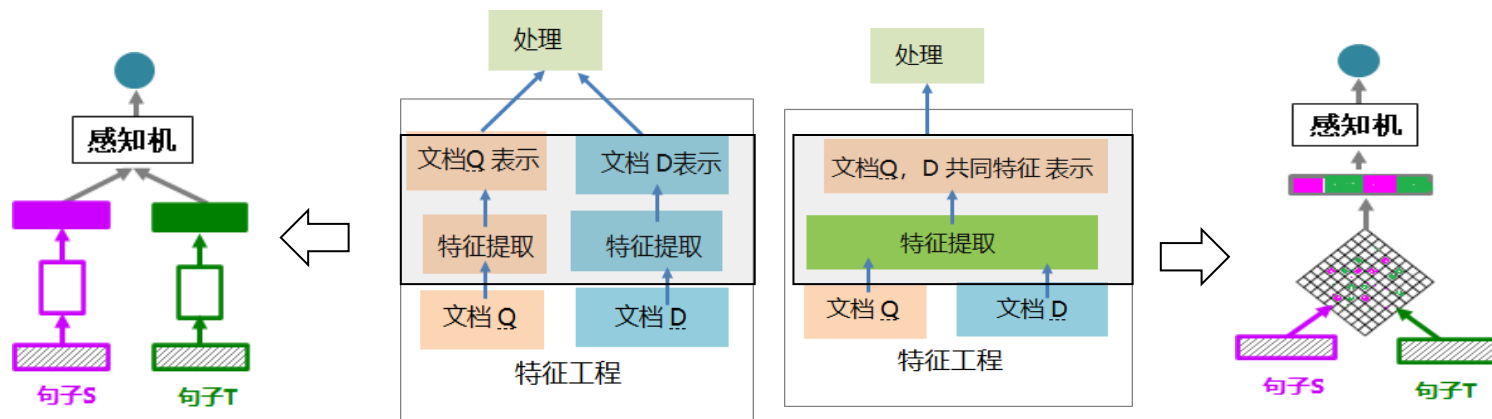


1. 文本匹配任务概述

★ 深度学习方法：

自动提取出词语之间的关系特征并结合短语匹配中的结构信息和文本匹配的层次化特性，更精细地描述文本匹配问题。

表示学习抽取有用特征



1. 文本匹配任务概述

深度学习匹配方法分为三类

◆ 基于单语义文档表达的深度学习模型（基于表示-孪生网络）

主要思路：首先将单个文本先表达成一个稠密向量（分布式表达）
然后直接计算两个向量间的相似度作为文本间的匹配度。

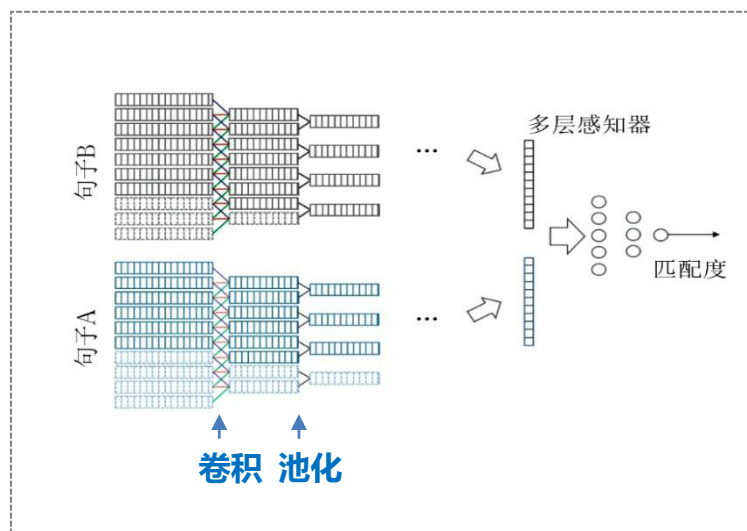
◆ 基于多语义文档表达的深度学习模型（基于交互-交互聚合）

主要思路：需要建立多语义表达，更早地让两段文本进行交互，然后挖掘文本交互后的模式特征，综合得到文本间的匹配度。

◆ 基于预训练语言模型BERT的模型

2. 文本匹配方法（编码型）

ARC-I (基于CNN)



输入：句子A和B

运算关系：

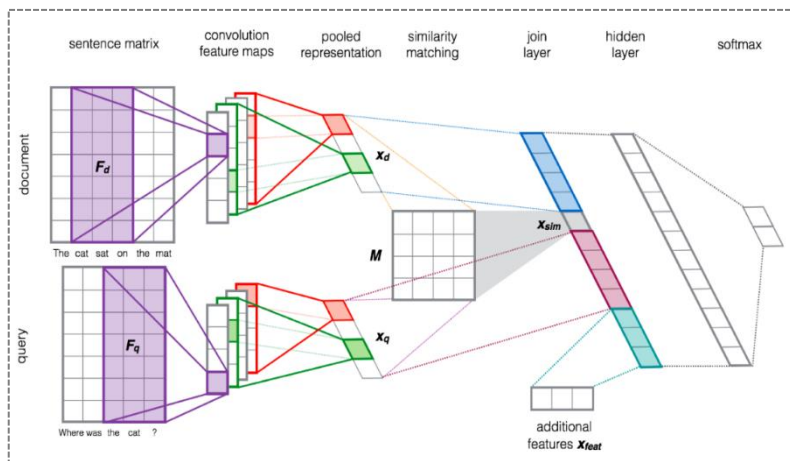
1. padding句子A/B到定长
2. 分别做多轮的卷积+池化运算
3. 拼接两个向量
4. 输入给多层感知机

输出：句子A和B的匹配度

特点： 在于将两个句子encode成句向量之后再用多层感知机进行分类，
没有体现出句子之间的交互操作

2. 文本匹配方法（编码型）

CDNN (基于CNN)



特点：引入相似矩阵

输入：句子d和q

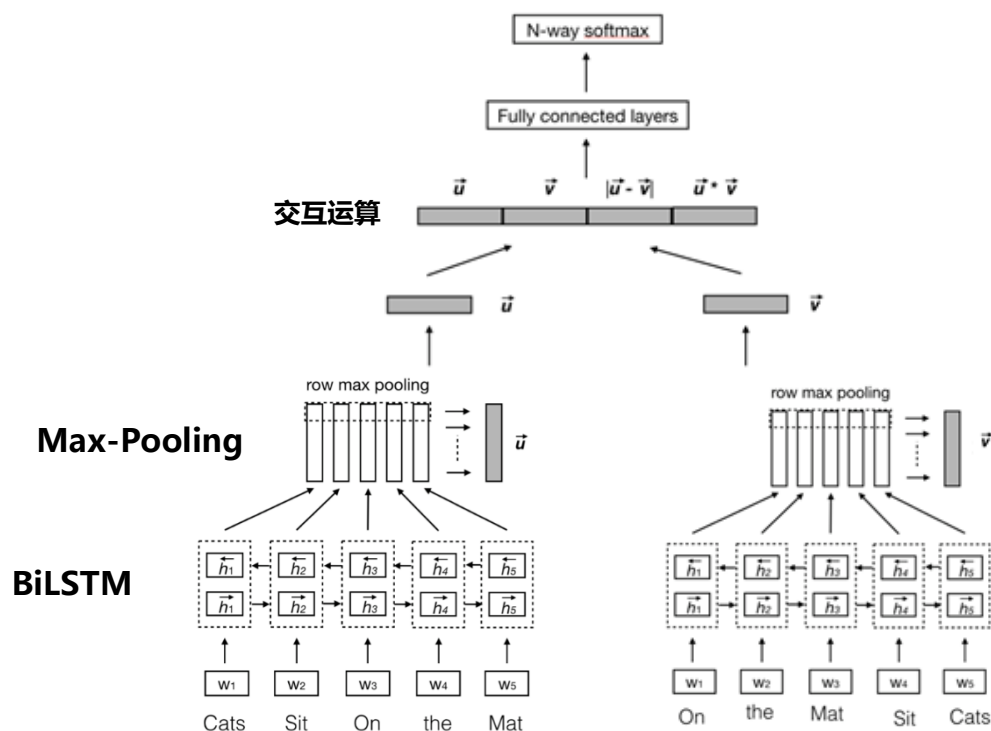
运算关系：

1. 分别做卷积+池化运算，得到句向量
2. 计算相似度 $X_{sim} = X_q^T M X_d$
3. 计算单词重叠数等其他特征
4. 拼接句向量、相似度和其他特征
5. 输入给多层感知机

输出：句子d和q的匹配度

2. 文本匹配方法（编码型）

InferSent (基于RNN)



输入：句子A和B

运算关系：

1. 将A，B分别通过BiLSTM-Max表示成句向量
2. 将2个句向量交互运算输入给多层感知机

输出：句子A和B的匹配度

2. 文本匹配方法（交互聚合型）

ABCNN (基于CNN 和 Attention)

论文提出了2类交互方法：

1. BCNN: Basic Bi-CNN（仅有CNN的基础模型BCNN）

2. ABCNN: Attention-Based BCNN（添加了attention）

① ABCNN-1

通过对输入句子的向量表示进行 attention 操作，从而影响卷积网络

② ABCNN-2

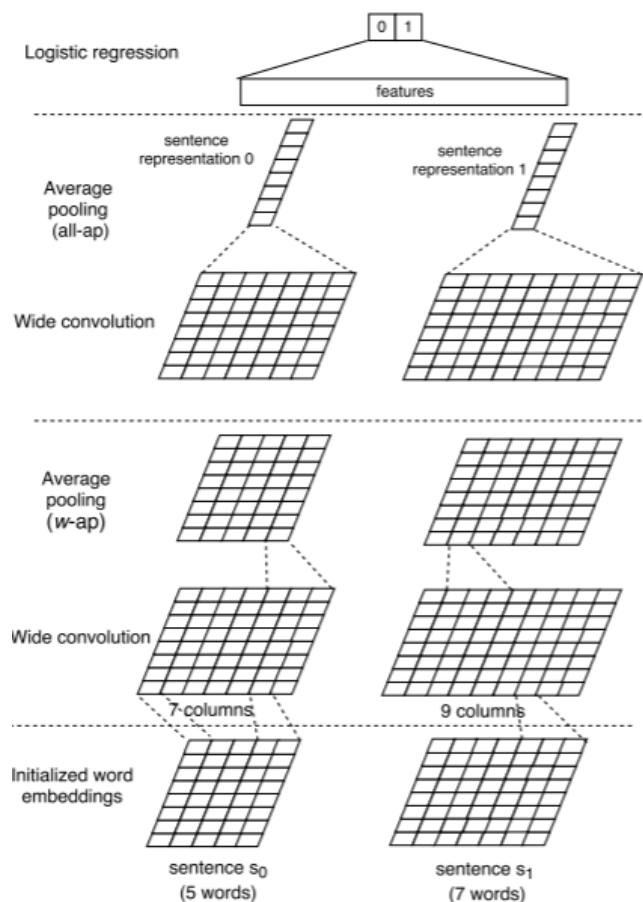
对 conv 层的输出进行 attention，从而对卷积层的输出结果进行加权

③ ABCNN-3

将 ABCNN-1 和 ABCNN-2结构进行叠加

2. 文本匹配方法（交互聚合型）

1. BCNN: Basic Bi-CNN



输入层：

词嵌入：可用word2vec，glove，elom等方法

卷积层

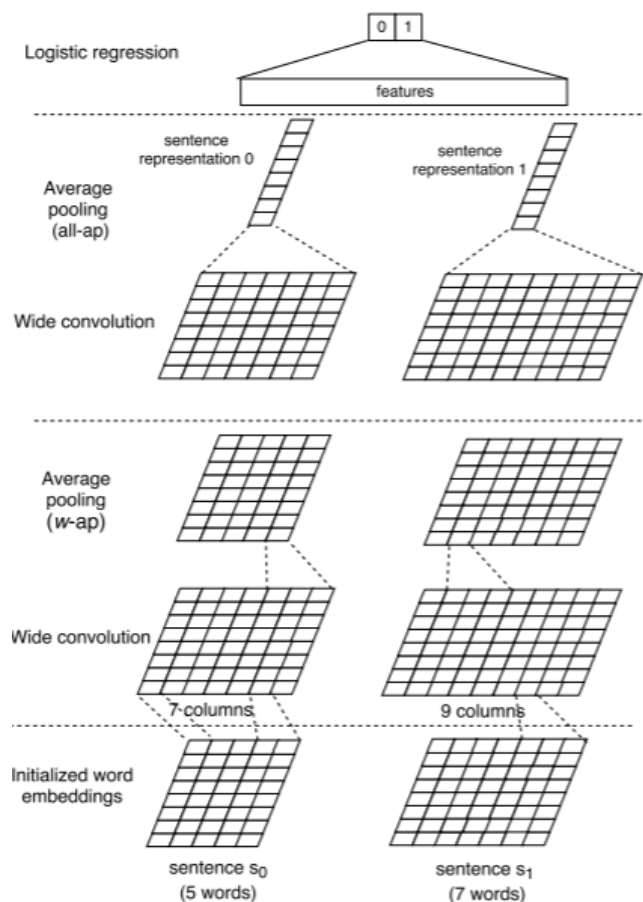
采用宽卷积得到一个长度为

$$sent_length + w_s - 1$$

的向量，（w为窗口宽度）

2. 文本匹配方法（交互聚合型）

1. BCNN: Basic Bi-CNN



Pooling 层（平均池化层）

- 中间的池化层是w-ap：
使用滑动窗口的形式，以窗口宽度 w 对卷积输出进行 Average Pooling
pooling 层后仍然会变回 $sent_length$
可以使 conv-pooling 层就无限叠加
- 最后层是all-ap（大小与输入相同，保证输出维度相同）

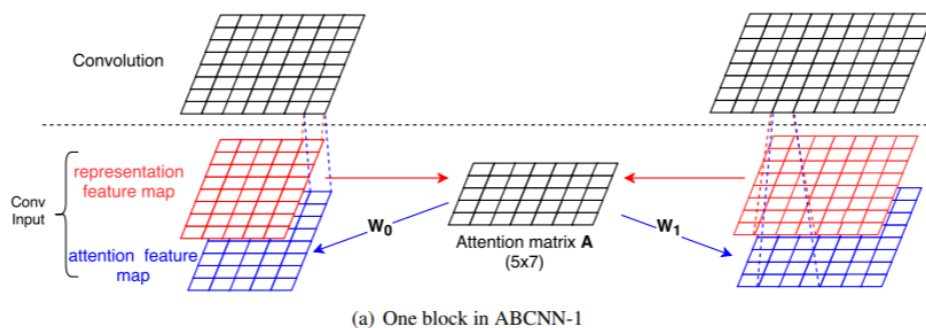
输出层

简单的二分类

2. 文本匹配方法（交互聚合型）

2. ABCNN: Attention-Based BCNN ① ABCNN-1

通过对输入句子的向量表示进行 attention 操作，从而影响卷积网络

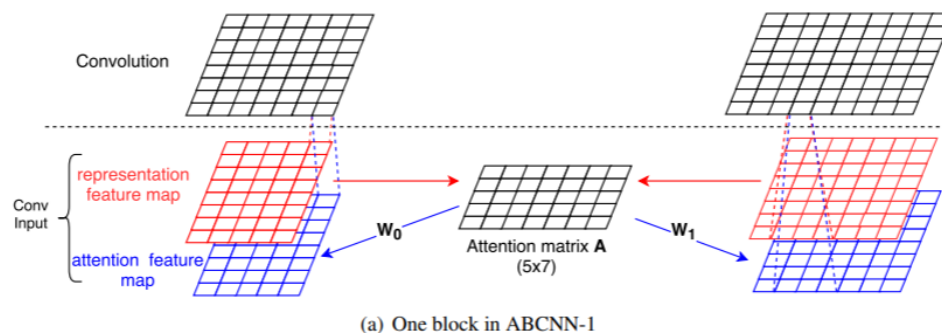


- 针对两个不同的序列，生成attention matrix A

$$A_{i,j} = match_score(F_{0,r}[:, i], F_{1,r}[:, j]) \quad F_{i,r} \in R^{d \times s} \text{ 表示句子的向量表示}$$

$$\text{match-score: } \frac{1}{1 + |x - y|} \quad (\text{可以用多种方式进行计算})$$

2. 文本匹配方法（交互聚合型）



- 得到了attention矩阵A，则可以计算句子的attention特征

$$F_{0,a} = W_0 \cdot A^T \quad F_{1,a} = W_1 \cdot A$$

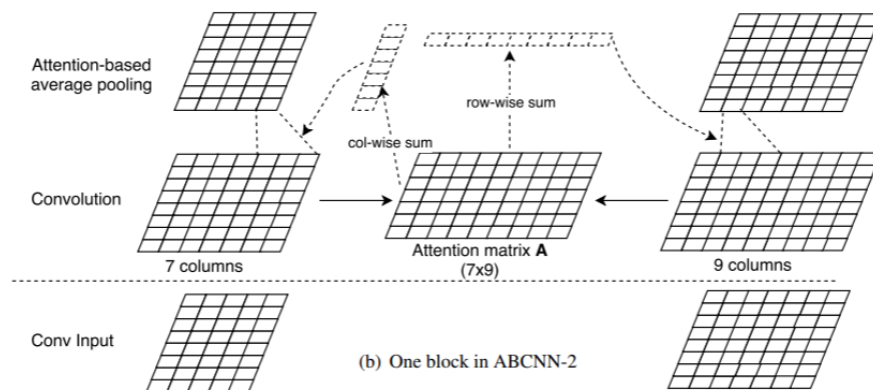
其中： $W_0 \in R^{d \times s}$, $W_1 \in R^{d \times s}$ 是模型参数

- 将原始的句子向量 $F_{i,r}$ 和attention 特征向量 $F_{i,a}$ 进行叠加
作为卷积层的输入向量

2. 文本匹配方法（交互聚合型）

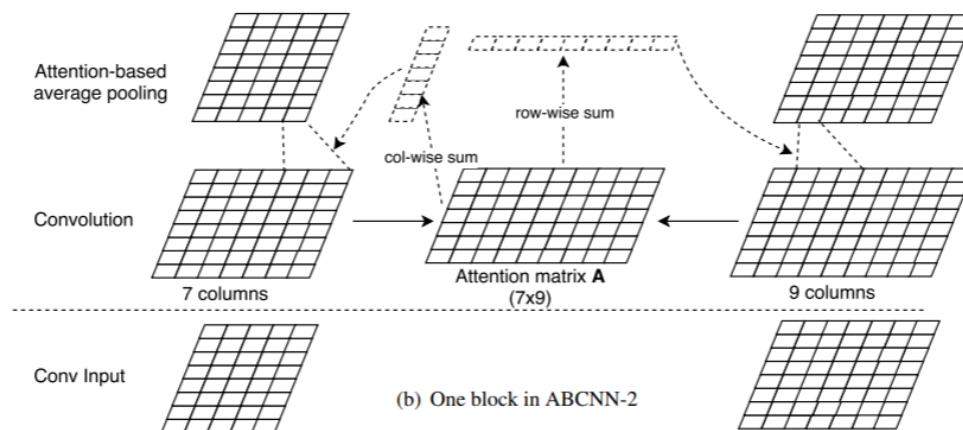
2. ABCNN: Attention-Based BCNN ② ABCNN-2

对 conv 层的输出进行 attention，从而对卷积层的输出结果进行加权



- 针对卷积后两个不同的序列，生成attention matrix A（方法同ABCNN-1）
分别为两个句子计算它们的 conv 输出和 attention 矩阵 Average Pooling

2. 文本匹配方法（交互聚合型）



- pooling : 根据计算出的 Attention 权重向量来计算得到

$$a_{0,j} = \sum A[j, :] \quad a_{1,j} = \sum A[:, j]$$

利用这个 attention 值对卷积层的输出进行加

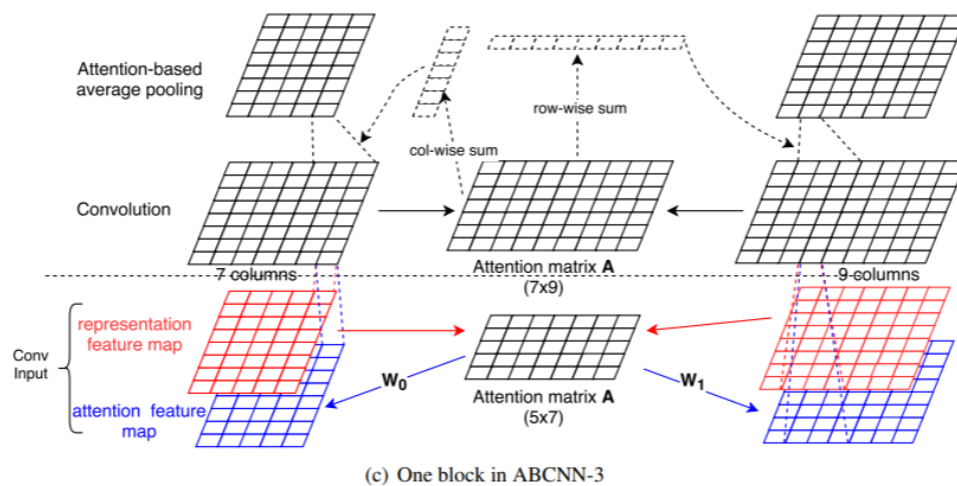
$$F_{i,r}^p[:, j] = \sum_{k=j:j+w} a_{i,k} \cdot F_{i,r}^c[:, k], j = 1 \dots s_i$$

最后池化后的结果维度和输入维度是相同的

2. 文本匹配方法（交互聚合型）

2. ABCNN: Attention-Based BCNN ③ ABCNN-3

ABCNN-3 是 ABCNN-1 和 ABCNN-2 结构进行叠加



2. 文本匹配方法（交互聚合型）

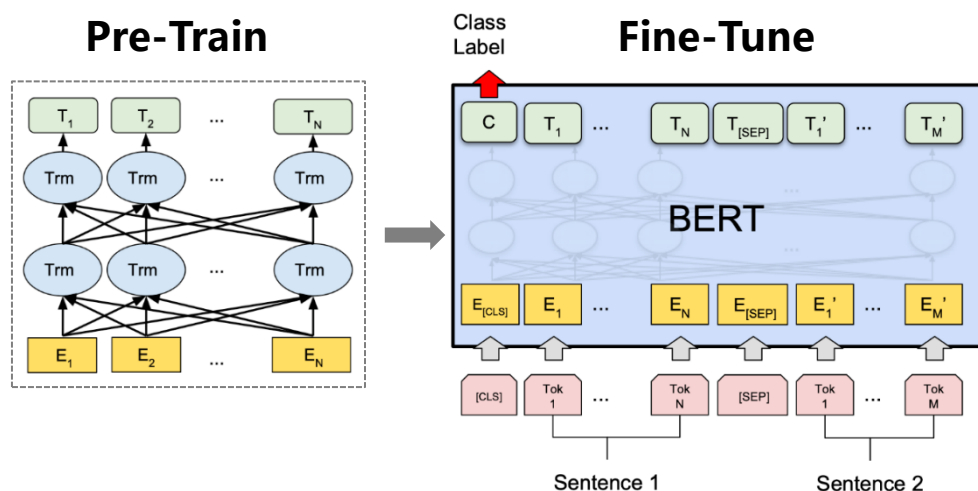
实验结果：

	method	MAP	MRR
Baselines	WordCnt	0.4891	0.4924
	WgtWordCnt	0.5099	0.5132
	CNN-Cnt	<u>0.6520</u>	<u>0.6652</u>
	Addition	0.5021	0.5069
	Addition(+)	0.5888	0.5929
	A-LSTM	0.5347	0.5483
	A-LSTM(+)	0.6381	0.6537
BCNN	one-conv	0.6629	0.6813
	two-conv	0.6593	0.6738
ABCNN-1	one-conv	0.6810*	0.6979*
	two-conv	0.6855*	0.7023*
ABCNN-2	one-conv	0.6885*	0.7054*
	two-conv	0.6879*	0.7068*
ABCNN-3	one-conv	0.6914*	0.7127*
	two-conv	0.6921*	0.7108*

Table 3: Results on WikiQA. Best result per column is bold. Significant improvements over state-of-the-art baselines (underlined) are marked with * (t -test, $p < .05$).

2. 文本匹配方法（基于预训练语言模型）

利用BERT



输入：句子X和Y

运算关系：

1. 预训练双向Transformer语言模型BERT
2. 取BERT最后一层[CLS]的表示作为句子关系的表示
3. 输入分类器

输出：句子X和Y的匹配度

优势：更深层的语义表达更好的通用语义表示

推荐工具:

- MatchZoo 是一个通用的文本匹配工具包，旨在方便大家快速的实现、比较、以及分享最新的深度文本匹配模型。

Keras Version: <https://github.com/NTMC-Community/MatchZoo>

PyTorch Version: <https://github.com/NTMC-Community/MatchZoo-py>



- Transformer相关模型的工具包

All Version: <https://github.com/huggingface/transformers>



build passing license Apache-2.0 website online release v2.0.0

State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

参考文献:

- grave et, al. Bag of Tricks for Efficient Text Classification, 2017
- Kim et, al. Convolutional Neural Networks for Sentence Classification, 2014
- Lai et al. Recurrent Convolutional Neural Networks for Text Classification, 2015
- Yang et al. Hierarchical Attention Networks for Document Classification., 2016
- Hu et al. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification 2019
- <https://www.jiqizhixin.com/articles/2019-10-18-14>
- <https://www.cnblogs.com/ZhangHT97/p/13391689.html>
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. EMNLP 2018.
- Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. ACL 2017.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT, 2018.
- https://blog.csdn.net/weixin_38526306/article/details/88425997

在此表示感谢!

谢谢各位！



Q&A