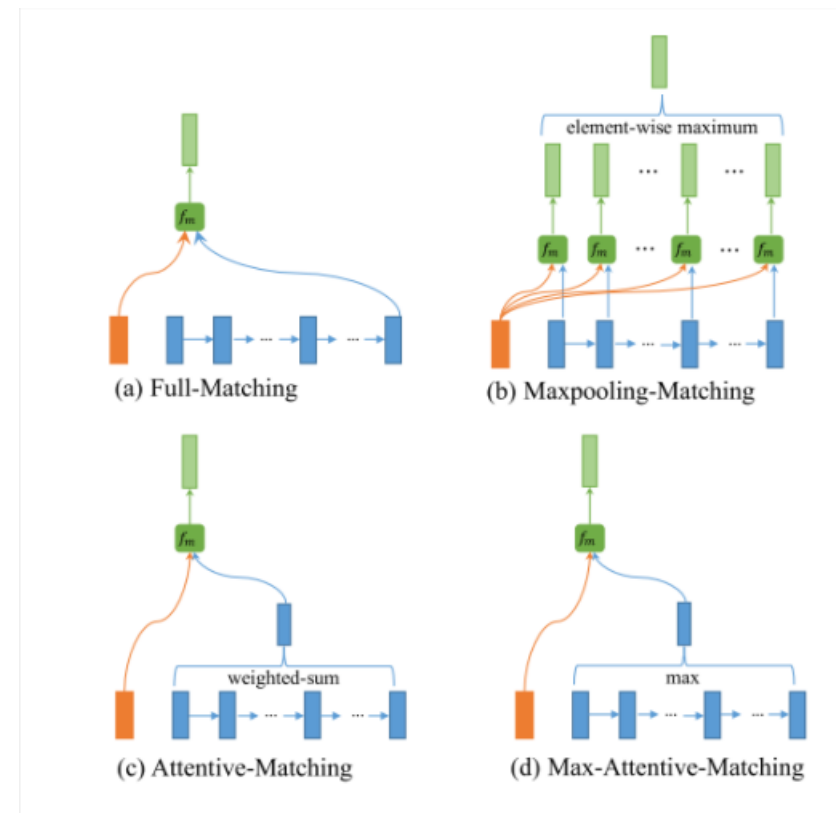


# 《中文深度学习分词是否有必要？》 论文报告

张佳敏

# 实验：Sentence Matching

- 实验所基于的模型是BiMPM，论文中引用了王志国研究员的论文，故粗略读了一下论文，了解了一下模型。
  - 除此之外，模型还给了四种匹配策略，分别是full-matching、maxpooling-matching、attentive-matching和max-attentive-matching，利用多视角余弦匹配函数可比较两个向量。



# 实验：Sentence Matching

- 论文当中提到两个中文数据集是：BQ和LCQMC，于是尝试去找数据集，找到了LCQMC。
  - 与BQ数据集比较两个句子是否具有相同语义含义不同的是，LCQMC偏向于两个句子是否具有相同的意图。
  - 下载下来的数据集是三个txt文件，以1和0来代表两个句子含义或目的相同与不同。

杭州哪里好玩	杭州哪里好玩点	1
这是什么乌龟值钱吗	这是什么乌龟！值钱嘛？	1
心各有所属是什么意思？	心有所属是什么意思？	0
什么东西越热爬得越高	什么东西越热爬得很高	1
世界杯哪位球员进球最多	世界杯单界进球最多是哪位球员	0
韭菜多吃什么好处	多吃韭菜有什么好处	1
云赚钱怎么样	怎么才能赚钱	0
何灵结婚了嘛	何灵结婚了么	1
长的清新是什么意思	小清新的意思是什么	0
我们可以结婚了吗？	在熙结婚了吗？	0
想买男人酒补肾壮阳酒哪里有啊	哪里有男人酒补肾壮阳酒	1
淘宝上怎么用信用卡分期付款	淘宝怎么分期付款，没有信用卡	0
最近有没有什么好看的韩剧	最近有什么好看的韩剧	1
《校花的贴身高手》中的林逸	校花贴身高手	1
叔叔是什么人	我是叔叔的什么人	0
这姑娘漂亮不	我姑娘漂亮吧	0
在淘宝网买手机可靠吗？	在淘宝网上买手机可靠吗？	1
山楂干怎么吃好吃？	山楂怎么做好吃	0
时间都去哪怕了歌谱	时间煮雨歌谱	0
苏州哪里能买到这个衣服	苏州哪里有买大号衣服的？	0
最好玩的手机网游	好玩的手机网游	1
石榴是什么时候成熟的？	成熟的石榴像什么？	0
刘诗诗杨幂谁漂亮	刘诗诗和杨幂谁漂亮	1
微信号怎么二次修改	怎么再二次修改微信号	1
什么牌子的精油皂好	什么牌子的精油好？	0
刚出生的小野鸡怎么养	刚抓来的野鸡怎么养殖	0
如何入侵他人手机	如何入侵别人的手机	1
红米刷什么系统好	红米可以刷什么系统	1
这叫什么高跟鞋	这种高跟鞋叫什么呀	1
汇理财怎么样	怎么样去理财？	0
什么是刷屏	什么叫刷屏？	1

LCQMC数据集：[https://github.com/pengming617/bert\\_textMatching/tree/master/data](https://github.com/pengming617/bert_textMatching/tree/master/data)

# 实验：Sentence Matching

- 实验所基于的模型是BiMPPM，论文中引用了王志国研究员的论文，故粗略读了一下论文，了解了一下模型。
  - 架构：对所要比的两句话进行匹配预测处理，将匹配结果聚合到一个向量中，在最后再将所得的结果进行最终处理。主要核心在句子之间的匹配。
  - 模型框架如图所示。
  - 其中模型匹配层为模型的核心。这一层的目的是用其中一句话的time step的上下文向量去匹配其中另一句话的上下文向量。

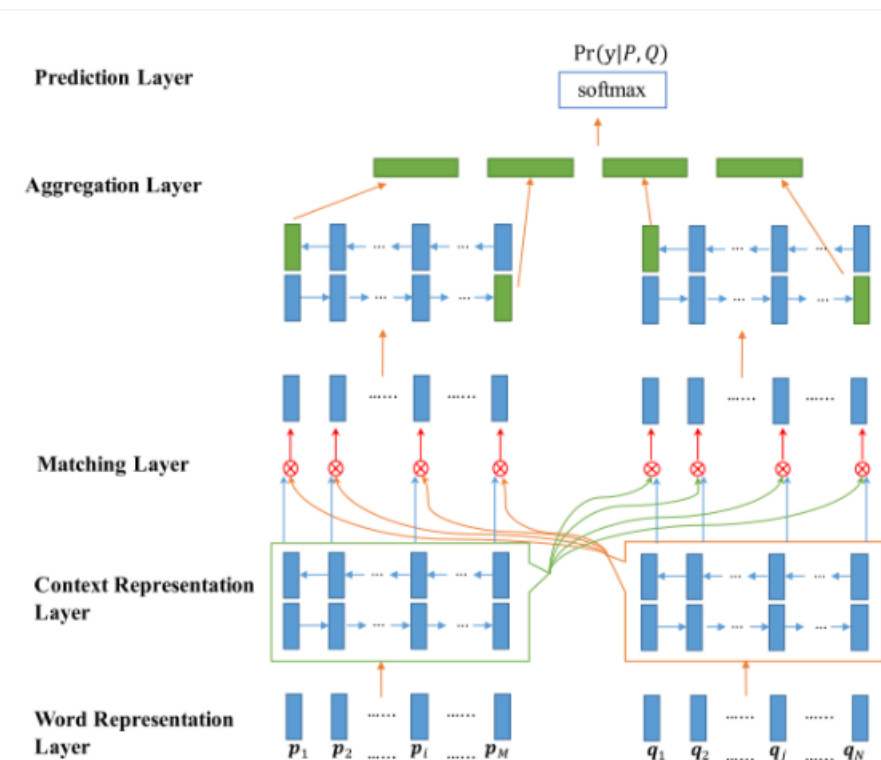
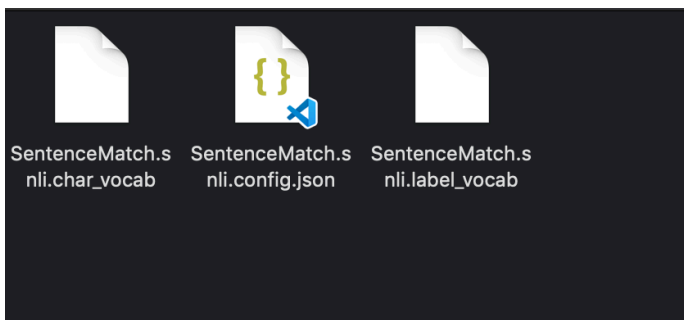


Figure 1: Architecture for Bilateral Multi-Perspective Matching (BiMPPM) Model, where ⊗ is the multi-perspective matching operation described in sub-section 3.2.

# 实验：Sentence Matching

- 在Github上找到了王志国研究员关于实现Sentence Matching的代码<sup>[1]</sup>，于是尝试去跑
  - 编辑好配置文件后使用数据集去训练，在训练中生成Testing所要用的log文件。

```
"train_path": "/Users/zhangjiamin/Desktop/BiMPM-master/snli/train.tsv",  
"dev_path": "/Users/zhangjiamin/Desktop/BiMPM-master/snli/dev.tsv",  
"word_vec_path": "/Users/zhangjiamin/Desktop/BiMPM-master/snli/wordvec.txt",  
"model_dir": "/Users/zhangjiamin/Desktop/BiMPM-master/snli/logs",  
"suffix": "snli",
```



【1】 代码地址：<https://github.com/zhiguowang/BiMPM>

# 实验：Sentence Matching

- 在Github上找到了王志国研究员关于实现Sentence Matching的代码<sup>[1]</sup>，于是尝试去跑
  - 在测试过程中，由于训练过程可能是配置和路径没设置好，开始浪费了一些时间，最后test的时候还是没有成功实现，最后提示没有注册的内核，是系统原因未用GPU用的CPU跑的，之后会继续换台电脑跑一遍。

```
File "/Users/zhangjiamin/Library/Python/2.7/lib/python/site-packages/tensorflow/python/client/session.py", line 1370, in _do_call
    raise type(e)(node_def, op, message)
tensorflow.python.framework.errors_impl.InvalidArgumentError: No OpKernel was registered to support Op 'CudnnRNNCanoicalToParams' used by
bi_lstm/CudnnRNNCanoicalToParams (defined at /Desktop/BiMPM-master/src/layer_utils.py:21) with these attrs: [num_params=16, T=DT_FLOAT, is
"lstm", seed2=0, seed=0, dropout=0]
Registered devices: [CPU]
Registered kernels:
<no registered kernels>

[[Model/char_lstm/char_lstm_cudnn_bi_lstm/char_lstm_cudnn_bi_lstm/CudnnRNNCanoicalToParams]]
```

【1】 代码地址：<https://github.com/zhiguowang/BiMPM>

# 实验：Sentence Matching

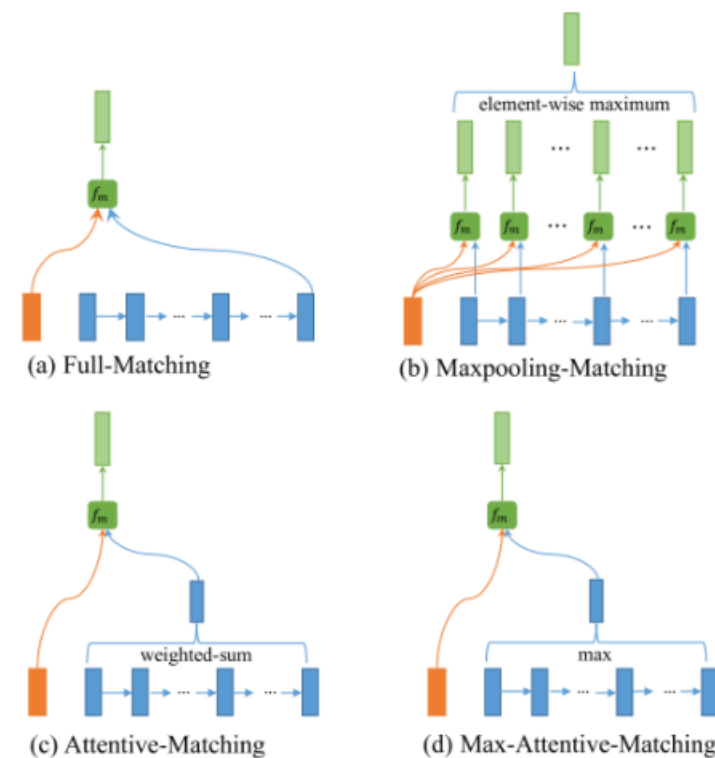
- 之后又细看了BiMPPM模型中匹配层的四种匹配策略。

- Full-Matching全匹配

- 图中画了一个方向的P -> Q, p在上层LSTM输出的隐藏层每一个 $h_i$ 与q在上层LSTM最后输入的 $h_N$ 做计算。匹配矩阵因为双向故也有两个。

$$\vec{m}_i^{full} = f_m(\vec{h}_i^p, \vec{h}_N^q; W^1)$$

$$\overleftarrow{m}_i^{full} = f_m(\overleftarrow{h}_i^p, \overleftarrow{h}_1^q; W^2)$$



# 实验：Sentence Matching

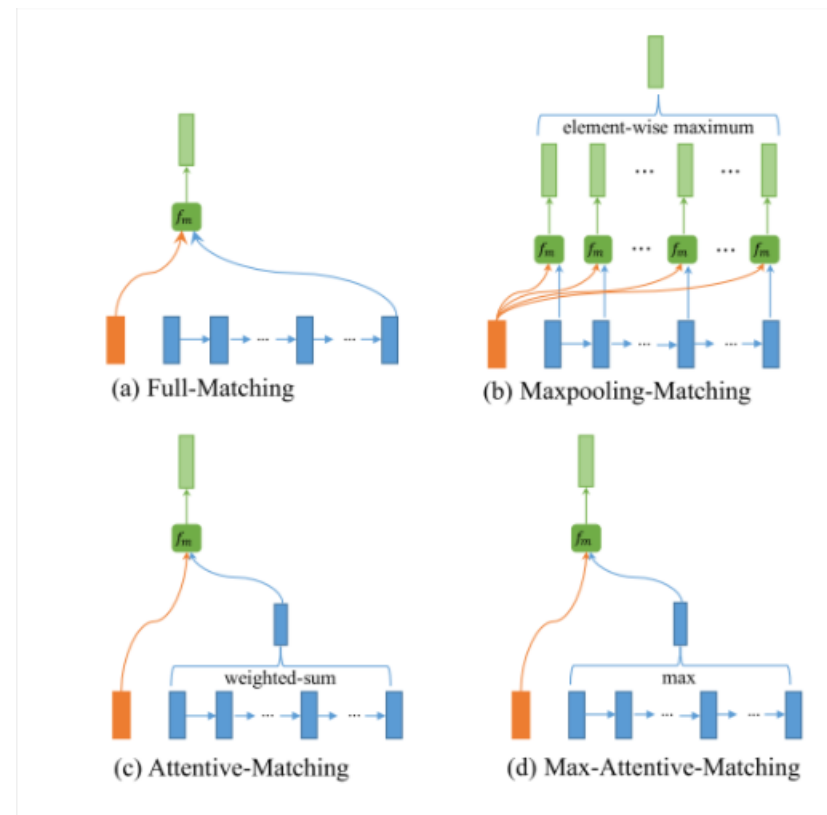
- BiMPPM模型中匹配层的四种匹配策略。
  - Maxpooling-Matching
    - 与全匹配模式不同，在这个模式里p与q在每一个时刻的上一层的隐藏层输出做相应计算，并且在每一维度上都取了max。

$$\vec{m}_i^{max} = \max_{j \in (1 \dots N)} f_m(\vec{h}_i^p, \vec{h}_j^q; W^3)$$

$$\overleftarrow{m}_i^{max} = \max_{j \in (1 \dots N)} f_m(\overleftarrow{h}_i^p, \overleftarrow{h}_j^q; W^4)$$

where  $\max_{j \in (1 \dots N)}$  is element-wise maximum.

上图为计算公式





# 实验：Sentence Matching

- BiMPM模型中匹配层的四种匹配策略。
  - Attentive-Matching

- 如图所示，在这个模式下，先计算p与q之间的注意力权重即注意力机制，再对q进行调整，最后进行一个类似full-matching的计算

$$\vec{\alpha}_{i,j} = \text{cosine}(\vec{h}_i^p, \vec{h}_j^q) \quad j = 1, \dots, N$$

$$\overleftarrow{\alpha}_{i,j} = \text{cosine}(\overleftarrow{h}_i^p, \overleftarrow{h}_j^q) \quad j = 1, \dots, N$$

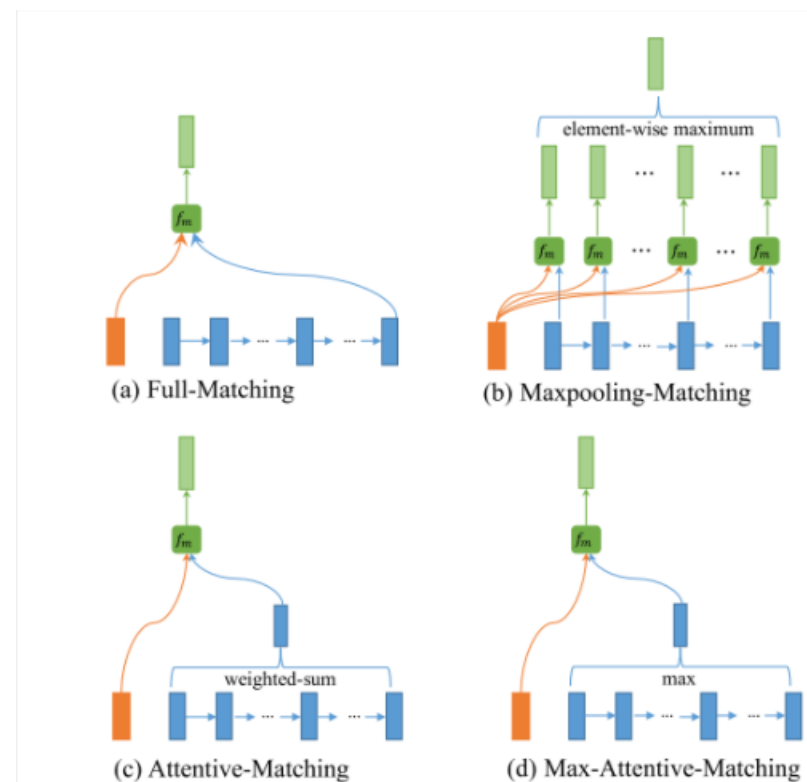
- 在注意力矩阵计算好之后，重新调整q的表示
  - 最后再开始最后的计算

$$\vec{m}_i^{\text{att}} = f_m(\vec{h}_i^p, \vec{h}_i^{\text{mean}}, \mathbf{W}^5)$$

$$\overleftarrow{m}_i^{\text{att}} = f_m(\overleftarrow{h}_i^p, \overleftarrow{h}_i^{\text{mean}}, \mathbf{W}^6)$$

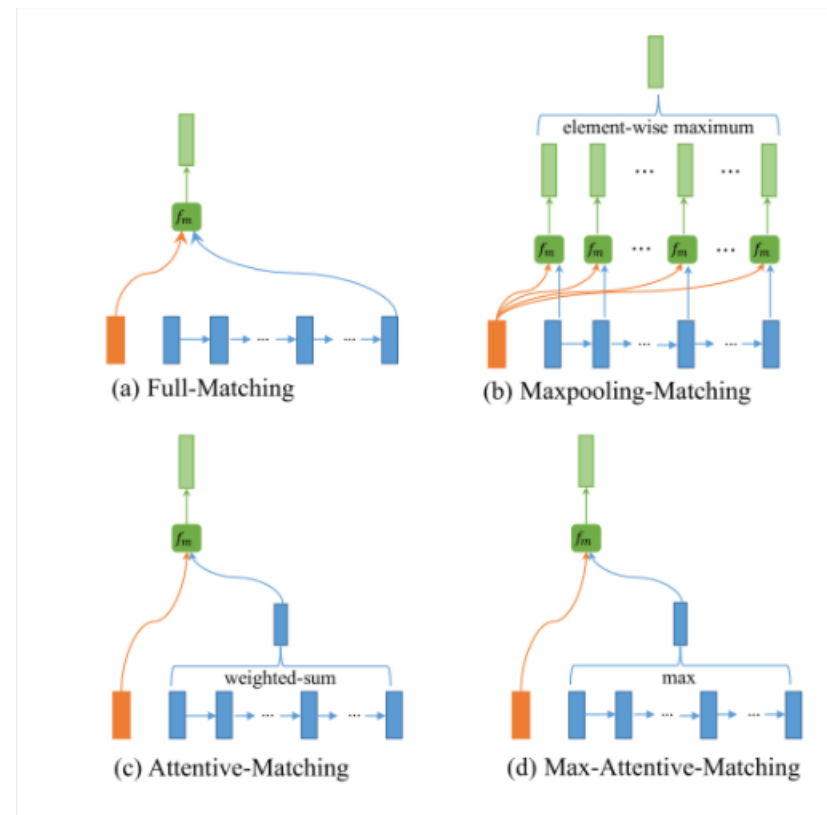
$$\vec{h}_i^{\text{mean}} = \frac{\sum_{j=1}^N \vec{\alpha}_{i,j} \cdot \vec{h}_j^q}{\sum_{j=1}^N \vec{\alpha}_{i,j}}$$

$$\overleftarrow{h}_i^{\text{mean}} = \frac{\sum_{j=1}^N \overleftarrow{\alpha}_{i,j} \cdot \overleftarrow{h}_j^q}{\sum_{j=1}^N \overleftarrow{\alpha}_{i,j}}$$



# 实验：Sentence Matching

- BiMPM模型中匹配层的四种匹配策略。
  - Max-Attentive-Matching
    - 与Attentive-Matching比较相似，不同的是权重的使用并不一样，Attentive-Matching是利用注意力权重相乘，这里就是hard-attention。



感谢漆老师的观看！