附：开发平台

操作系统	Ubuntu 16.04
内存容量	128 G
硬盘容量	8 T
显卡型号	GTX 1080Ti * 4
开发框架	Keras / Tensorflow (Python)

一、术语解释

【词向量】

“词向量”（词嵌入）是将一类将词的语义映射到向量空间中去的自然语言处理技术。即将一个词用特定的向量来表示，向量之间的距离（例如，任意两个向量之间的L2范式距离或更常用的余弦距离）一定程度上表征了的词之间的语义关系。由这些向量形成的几何空间被称为一个嵌入空间。

例如，“椰子”和“北极熊”是语义上完全不同的词，所以它们的词向量在一个合理的嵌入空间的距离将会非常遥远。但“厨房”和“晚餐”是相关的话，所以它们的词向量之间的距离会相对小。

理想的情况下，在一个良好的嵌入空间里，从“厨房”向量到“晚餐”向量的“路径”向量会精确地捕捉这两个概念之间的语义关系。在这种情况下，“路径”向量表示的是“发生的地点”，所以你会期望“厨房”向量 - “晚餐"向量（两个词向量的差异）捕捉到“发生的地点”这样的语义关系。基本上，我们应该有向量等式：晚餐 + 发生的地点 = 厨房（至少接近）。如果真的是这样的话，那么我们可以使用这样的关系向量来回答某些问题。例如，应用这种语义关系到一个新的向量，比如“工作”，我们应该得到一个有意义的等式，工作+ 发生的地点 = 办公室，来回答“工作发生在哪里？”。

词向量通过降维技术表征文本数据集中的词的共现信息。方法包括神经网络(“Word2vec”技术)，或矩阵分解。

以“第六届泰迪杯数据挖掘挑战赛”为例，生成一个词向量矩阵具体步骤：

①分词：

```
Out[1]: ['第六届 泰迪杯 数据挖掘 挑战赛']
```

②tokenizer：

```
Out[2]: {'挑战赛': 4, '数据挖掘': 3, '泰迪杯': 2, '第六届': 1}
```

③squences：

```
Out[3]: [[2, 3], [3, 4]]
```

④padding：

```
Out[4]: array([[0, 0, 0, 0, 0, 0, 0, 0, 2, 3],
               [0, 0, 0, 0, 0, 0, 0, 0, 3, 4]], dtype=int32)
```

【MRR(Mean Reciprocal Rank)】

是一个国际上通用的对搜索算法进行评价的机制，即第一个结果匹配，分数为1，第二个匹配分数为0.5，第n个匹配分数为1/n，如果没有匹配的句子分数为0。最终的分数为所有得分之和。

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Query	Results	Correct Response	Rank	Reciprocal Rank
cat	catten, cati, **cats**	cats	3	1 / 3
torus	torii, **tori**, toruses	tori	2	1 / 2
virus	**viruse**, virii, viri	viruses	1	1 / 1
MRR = (1 / 3 + 1 / 2 + 1 / 1) / 3 = 11 / 18				

【MAP(Mean Average Precision)】

Mean average precision for a set of queries is the mean of the average precision scores for each query. 一组查询的平均准确率是每个查询的平均精度分数的平均值。MAP 是反映系统在全部相关文档上性能的单值指标。

$$\mathbf{MAP} = \frac{\sum_{q=1}^{Q}\mathbf{AveP(q)}}{Q}$$

Rank	Correct	Score
1	wrong	0
2	right	1 / 2
3	right	2 / 3
MAP = (0 + 1 / 2 + 2 / 3) / 3 = 5 / 18		

【TOP-1】

对于输出的未知答案，如果概率最大的是正确答案，才认为正确。

【各项指标】

①accuacy: 在训练集上的准确率。

②val_accuacy: 在验证集上的准确率。

③loss: 在训练集上的损失。

④val_loss: 在验证集上的损失。

二、模型设计

（一）基于 LSTM 模型

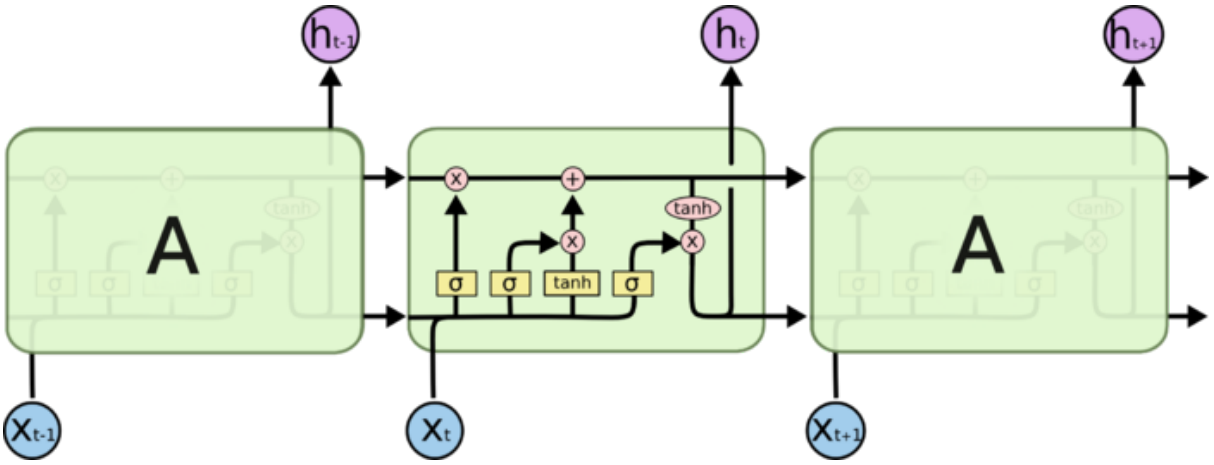
近两年深度学习在自然语言处理领域取得了非常好的效果。深度学习模型可以直接进行端到端的训练，而无须进行传统的特征工程过程。在自然语言处理方面，主要的深度学习模型是 RNN，以及在 RNN 之上扩展出来的 LSTM。

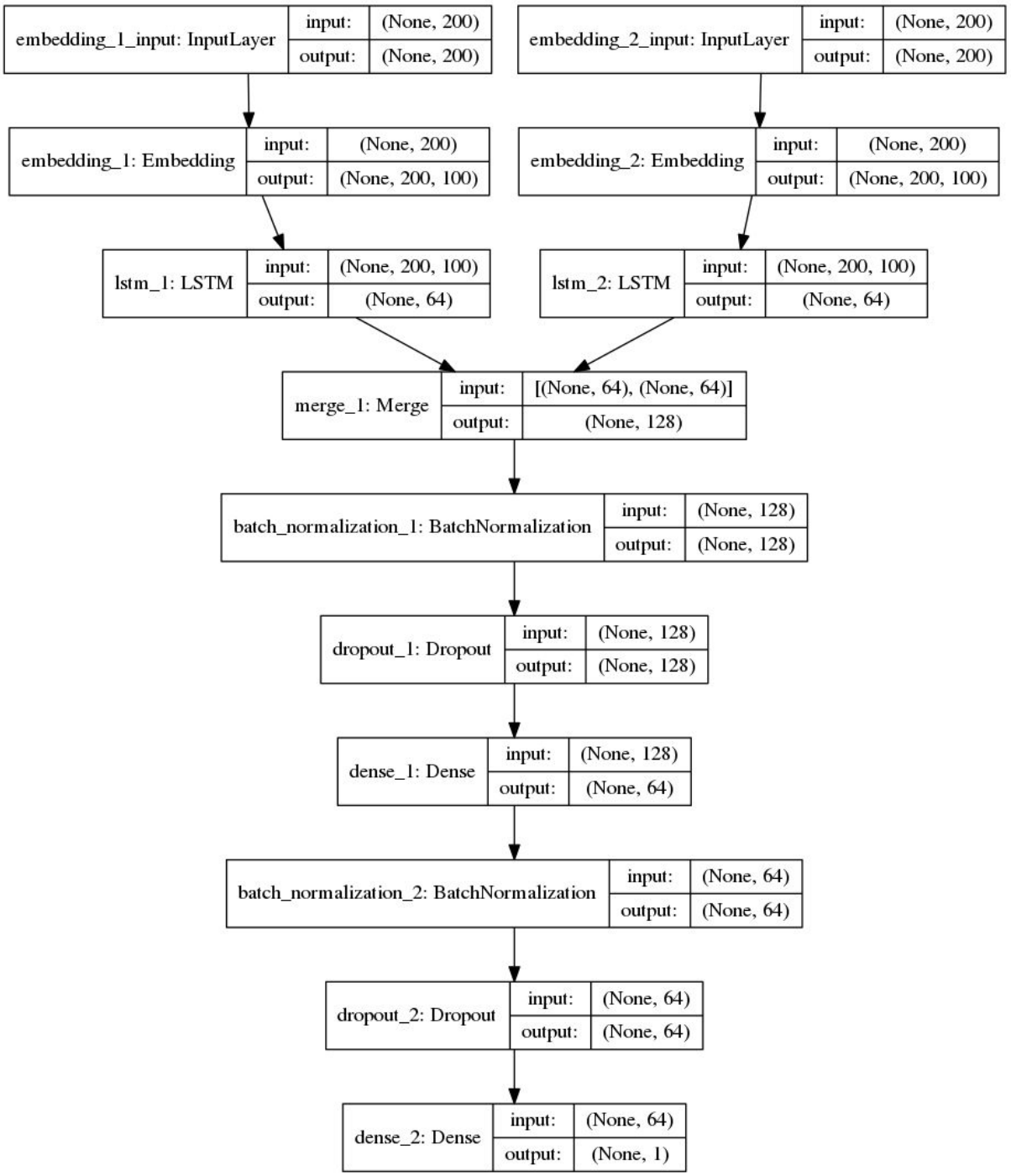
LSTM 是一种带有选择性记忆功能的 RNN，它可以有效地解决长时间依赖问题，并能学习到之前的关键信息。它增加了一条状态线，以记住从之前的输入学到的信息，另外增加三个门(gate)来控制该状态，分别为忘记门、输入门和输出门。

忘记门的作用是选择性地将之前不重要的信息丢掉，以便存储新信息。

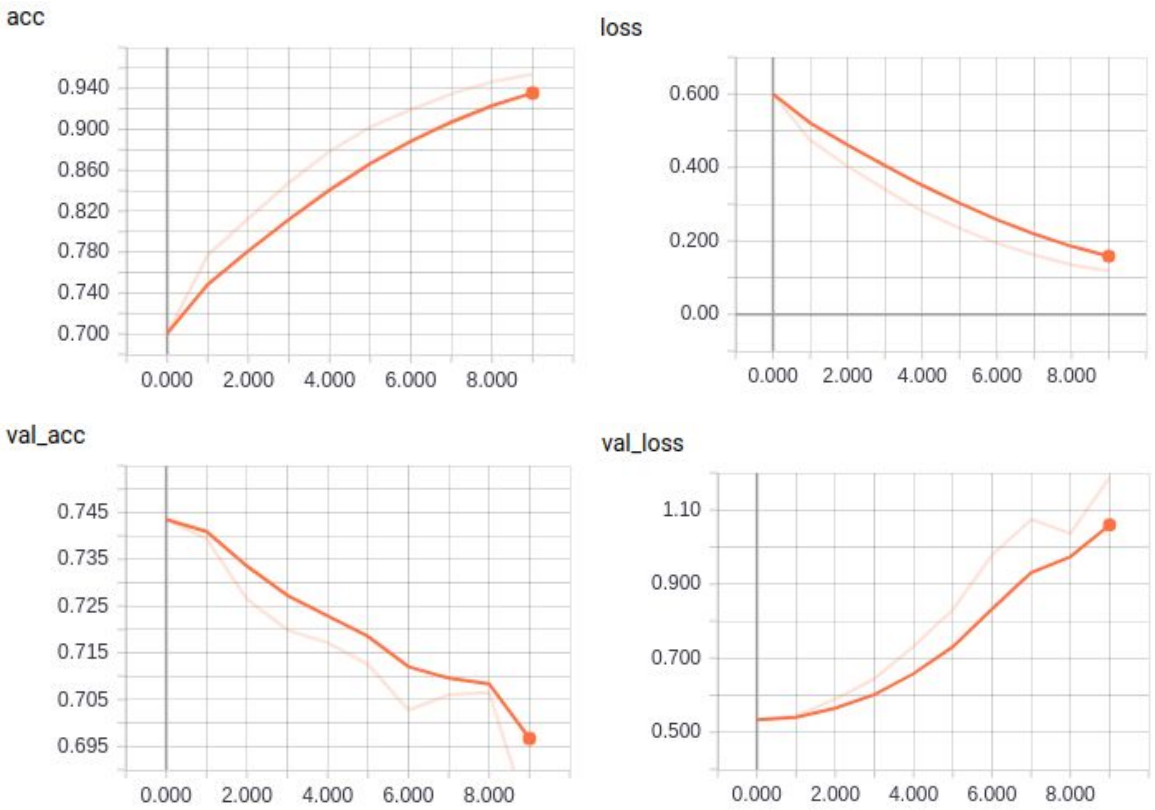
输入门的作用是根据当前输入学习到新信息，更新当前状态。

输出门的作用是根据当前输入和当前状态得到一个输出，该输出除了作为基本的输出外，还会作为下一个时刻的输入。


h_{t-1}
h_t
h_{t+1}
A
A
tanh
σ
σ
tanh
σ
X_{t-1}
X_t
X_{t+1}


embedding_1_input: InputLayer
input: (None, 200)
output: (None, 200)
embedding_2_input: InputLayer
input: (None, 200)
output: (None, 200)
embedding_1: Embedding
input: (None, 200)
output: (None, 200, 100)
embedding_2: Embedding
input: (None, 200)
output: (None, 200, 100)
lstm_1: LSTM
input: (None, 200, 100)
output: (None, 64)
lstm_2: LSTM
input: (None, 200, 100)
output: (None, 64)
merge_1: Merge
input: [(None, 64), (None, 64)]
output: (None, 128)
batch_normalization_1: BatchNormalization
input: (None, 128)
output: (None, 128)
dropout_1: Dropout
input: (None, 128)
output: (None, 128)
dense_1: Dense
input: (None, 128)
output: (None, 64)
batch_normalization_2: BatchNormalization
input: (None, 64)
output: (None, 64)
dropout_2: Dropout
input: (None, 64)
output: (None, 64)
dense_2: Dense
input: (None, 64)
output: (None, 1)

【LSTM 模型】



【实验数据】

指标：

	训练ACC	验证ACC	测试ACC	MRR	MAP	TOP-1
准确率	0.9540	0.7435	0.7201	0.6487	0.5550	0.4600

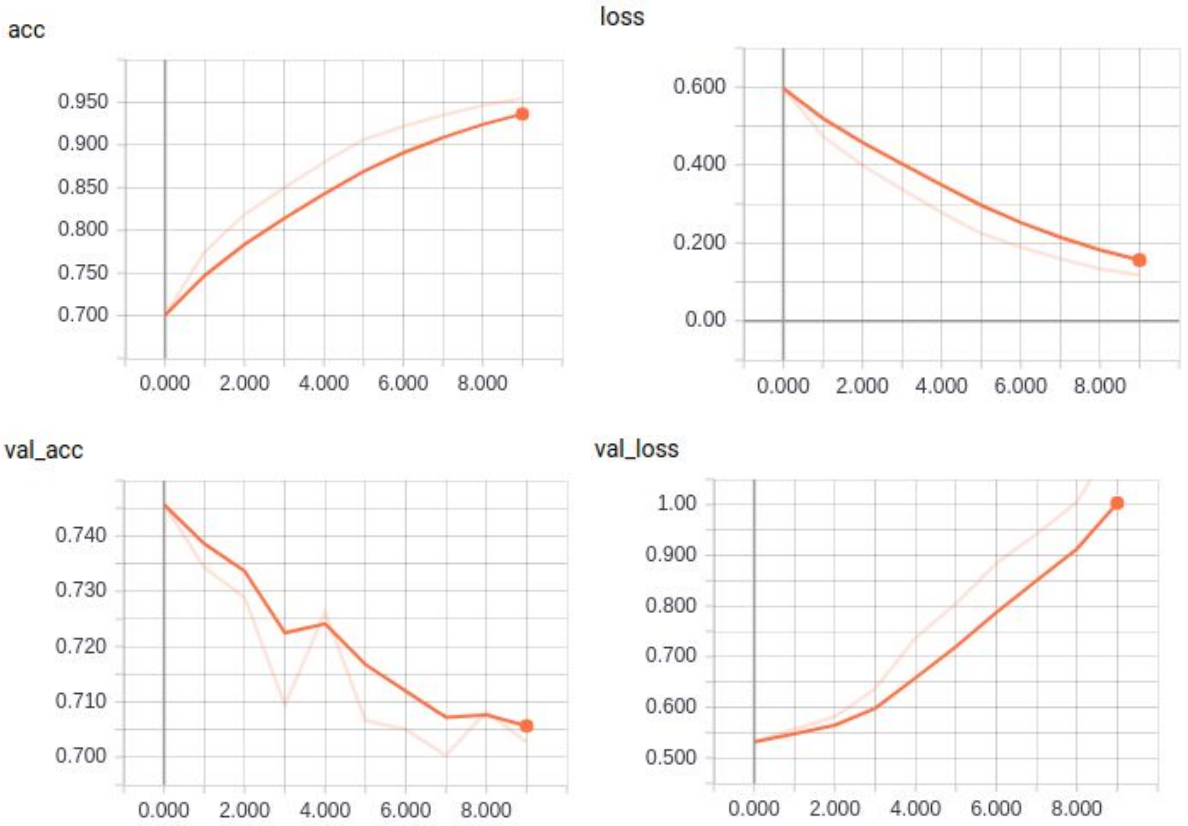
结论：模型比较简单，容易过拟合，导致训练准确率和验证准确率两极化。

（二）基于 Bi-LSTM 模型

单向 LSTM，根据前面的信息推出后面的信息，但有时候只看前面的信息是不够的。**例如：今天天气__，风刮在脸上仿佛刀割一样。**

如果根据“天气”，可能推出“晴朗”、“暖和”、“寒冷”等。但是如果加上后面的形容，能选择的范围就变小了，“晴朗”、“暖和”不可能选，而“寒冷”被选择的概率更大。

LSTM 虽然解决了长期依赖问题，但是无法利用文本的下文信息。Bi-LSTM 同时考虑文本的上下文信息，将时序相反的两个 LSTM 网络连接到同一个输出。前向 LSTM 可以获取输入序列的上文信息，后向 LSTM 可以获取输入序列的下文信息，模型准确率得到大大提升。



【实验数据】

指标：

	训练ACC	验证ACC	测试ACC	MRR	MAP	TOP-1
准确率	0.9545	0.7457	0.7301	0.6578	0.5691	0.4850

结论：训练效果同 LSTM ，模型还是比较简单，容易过拟合，导致训练准确率和验证准确率两极化。

（三）基于 Bi-LSTM + Word2Vec 模型

Word2Vec，为一群用来产生词向量的相关模型。这些模型为浅而双层的神经网络，用来训练以重新建构语言学之词文本。网络以词表现，并且需猜测相邻位置的输入词，在Word2Vec中词袋模型假设下，词的顺序是不重要的。训练完成之后，Word2Vec 模型可用来映射每个词到一个向量，可用来表示词对词之间的关系。该向量为神经网络之隐藏层。

Word2Vec 采用 CBOW 和 Skip-Gram 来建立神经网络词嵌入。CBOW 是已知当前词的上下文，预测当前词。而 Skip-Gram 相反，是在已知当前词，预测当前词的上下文。

实验使用维基百科中文语料生成 Word2Vec 模型，大致流程如下：

1、**下载语料**

https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

2、**提取正文**，将 xml 格式的 wiki 数据转换为 text 格式。

3、**繁简转换**，如果抽取中文的话需要将繁体转化为简体(维基百科的中文数据是繁简混杂的，里面包含大陆简体、台湾繁体、港澳繁体等多种不同的数据)。可以使用opencc进行转换，也可以使用其它繁简转换工具。

4、**编码转换**，由于后续的分词需要使用utf-8格式的字符，而上述简体字中可能存在非utf-8的字符集，避免在分词时候进行到一半而出现错误，因此先进行字符格式转换。使用iconv命令将文件转换成utf-8编码。

5、**分词处理**，使用 jieba 分词工具。

6、**训练**

```
In [ ]: from gensim.models import word2vec
        import logging

        input = './wiki.zh.text.jian.utf8.seg'
        output = './wiki.vector'
        logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
        sentences = word2vec.LineSentence(input)
        model = word2vec.Word2Vec(sentences, size=100, window=5, min_count=5, workers=4)
        model.wv.save_word2vec_format(output, binary=False)
```

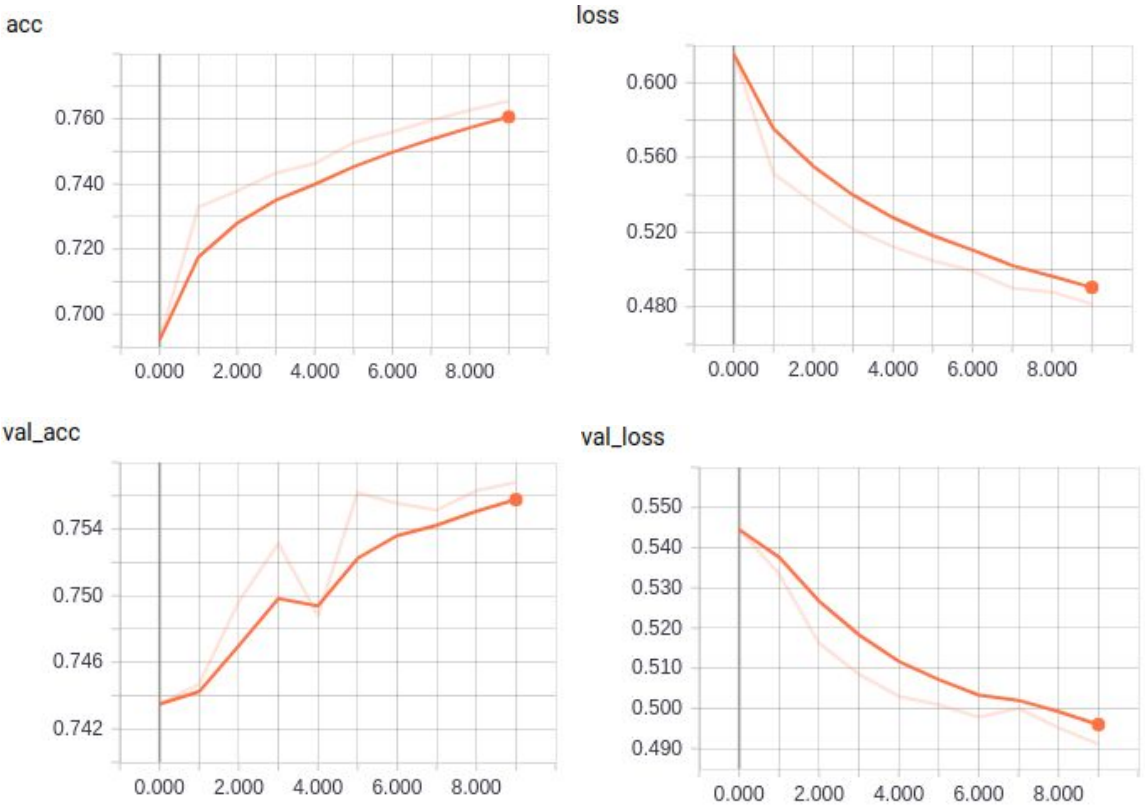
7、**测试。**

```
In [1]: from gensim.models.keyedvectors import KeyedVectors
        model = KeyedVectors.load_word2vec_format('./wiki.vector', binary=False)

In [2]: # model['女人'] + model['国王'] - model['男人'] = model['皇后']
        model.most_similar(positive=['woman', 'king'], negative=['man'])

Out[2]: [('queen', 0.7293198108673096),
         ('bride', 0.683803915977478),
         ('mistress', 0.6707652807235718),
         ('prince', 0.6648019552230835),
         ('wives', 0.6588137149810791),
         ('princess', 0.6529775857925415),
         ('queens', 0.6459839344024658),
         ('daughters', 0.6443694829940796),
         ('mother', 0.6377485990524292),
         ('godmother', 0.6289682388305664)]

In [3]: model.similarity('woman', 'man')

Out[3]: 0.6361447854063201

In [4]: model.similarity('queen', 'king')

Out[4]: 0.6681771014772736
```

可以看到，在 Word2Vec 词向量模型中：

1、“女人” + “国王” - “男人” ≈ “皇后”。

2、“女人”和“男人”（或“皇后”和“国王”）在空间向量上接近。



【实验数据】

指标：

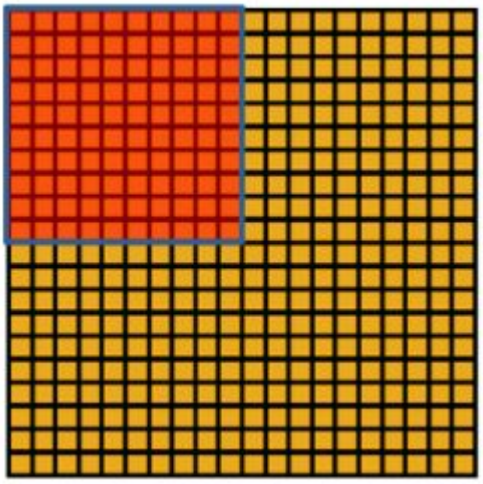
	训练ACC	验证ACC	测试ACC	MRR	MAP	TOP-1
准确率	0.7655	0.7568	0.7390	0.6868	0.5994	0.5300

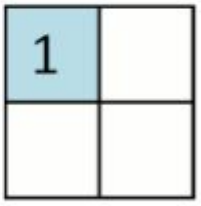
结论：使用预训练的词向量（相当于增加数据量），防止过拟合，保证训练准确率和验证准确率正常增长，同时显著提升 MRR、MAP、TOP-1 指标。

（四）基于 Bi-LSTM + Word2Vec + CNN 模型

1、卷积层

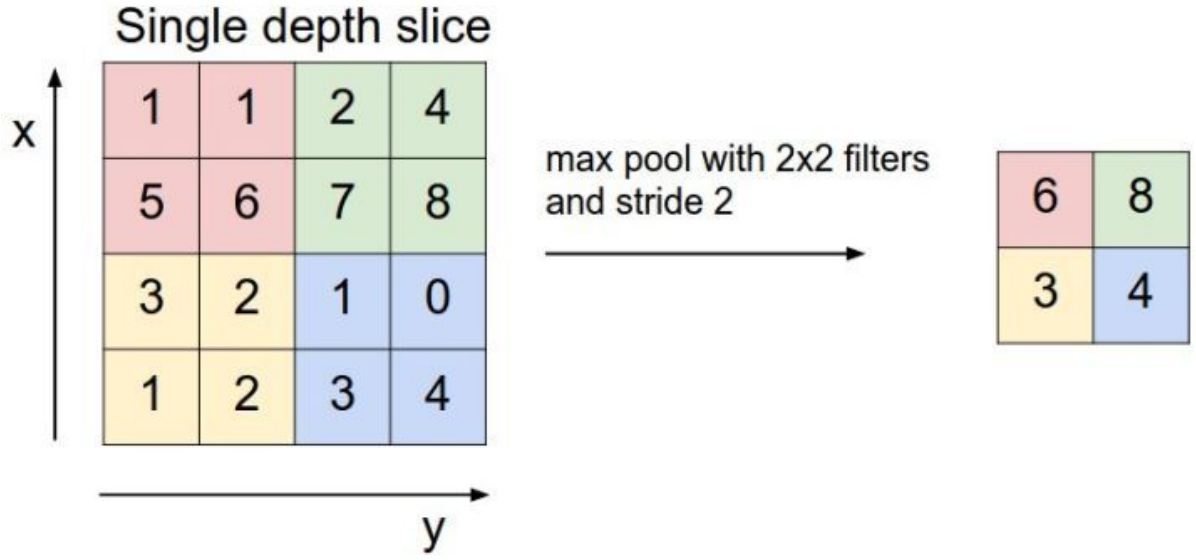
卷积神经网络（Convolutional Neural Network）最早是应用在计算机视觉当中，而如今 CNN 也早已应用于自然语言处理（Natural Language Processing）的各种任务。在图像中卷积核通常是对图像的一小块区域进行计算，而在文本中，一句话所构成的词向量作为输入。

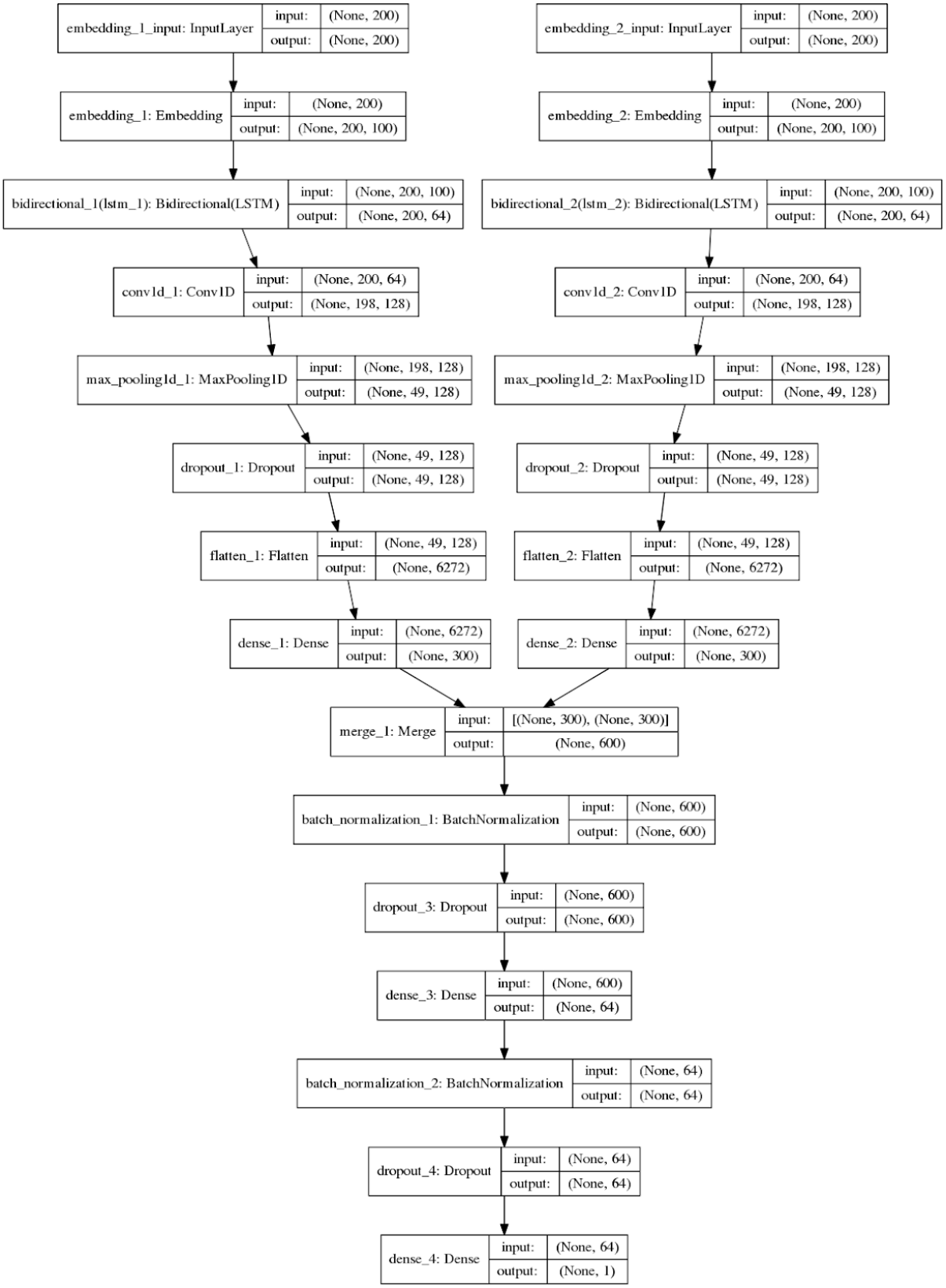




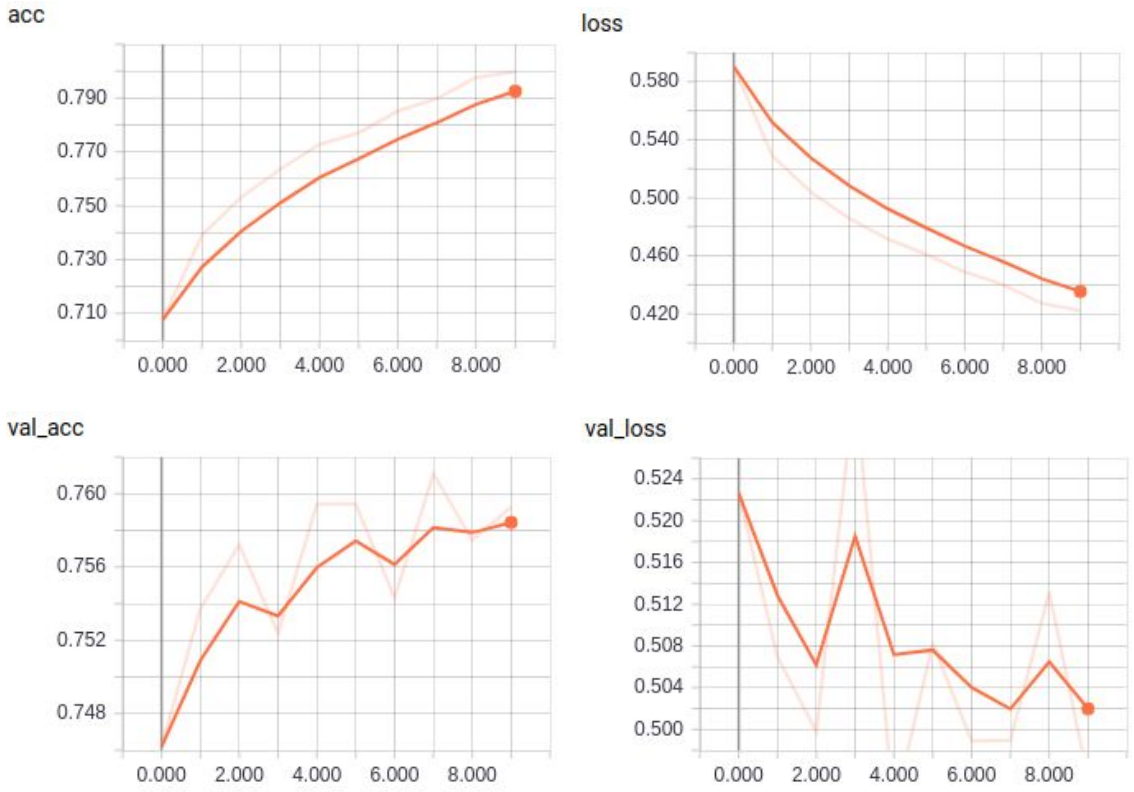
2、池化层

卷积神经网络的一个重要概念就是池化层，一般是在卷积层之后。池化层对输入做降采样。池化的过程实际上是对卷积层分区域求最大值或者对每个卷积层求最大值。





【Bi-LSTM + Word2Vec + CNN 模型】



【实验数据】

指标：

	训练ACC	验证ACC	测试ACC	MRR	MAP	TOP-1
准确率	0.7998	0.7611	0.7383	0.6952	0.6116	0.5450

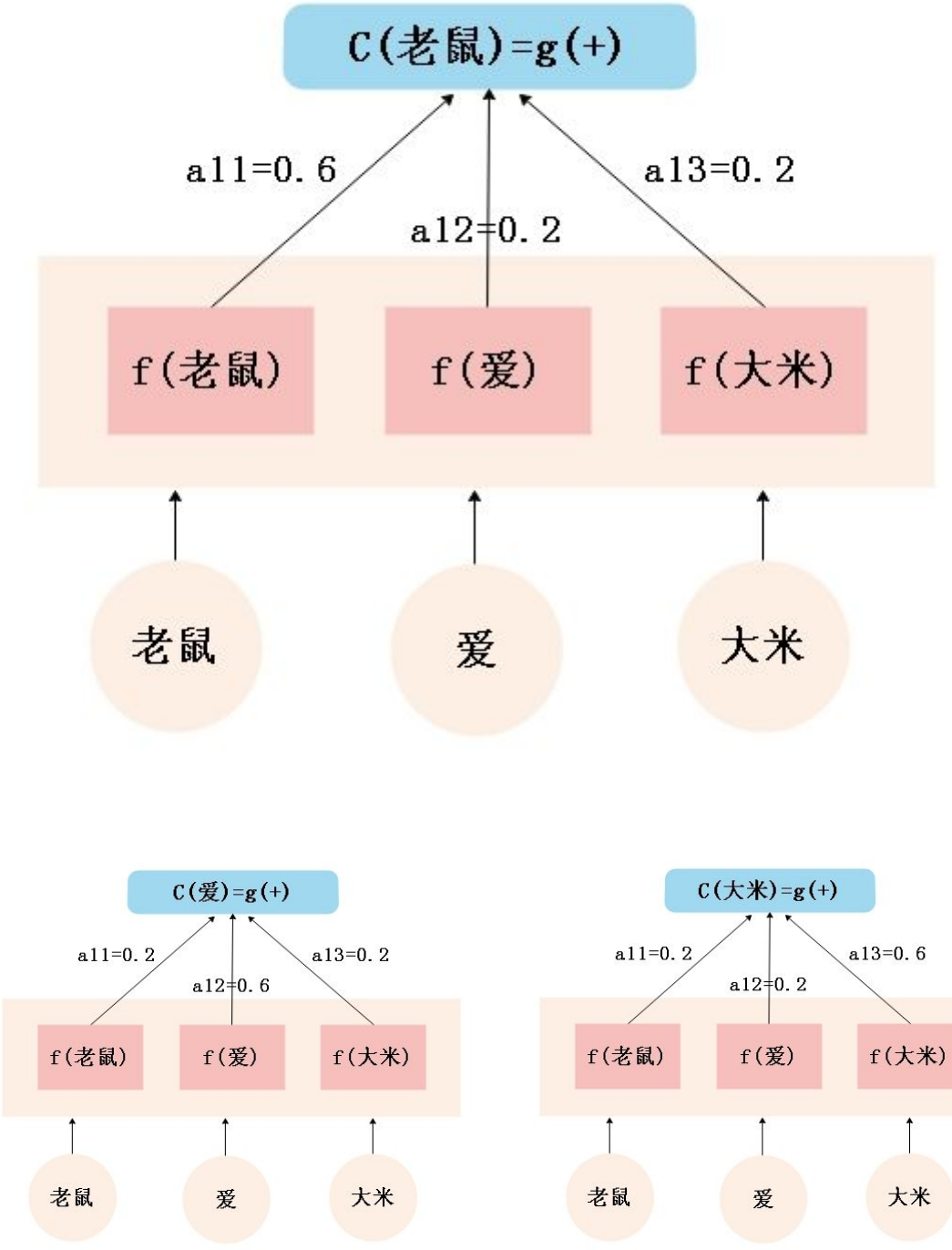
结论：进一步提升准确率，同时显著提升 MRR、MAP、TOP-1 指标。

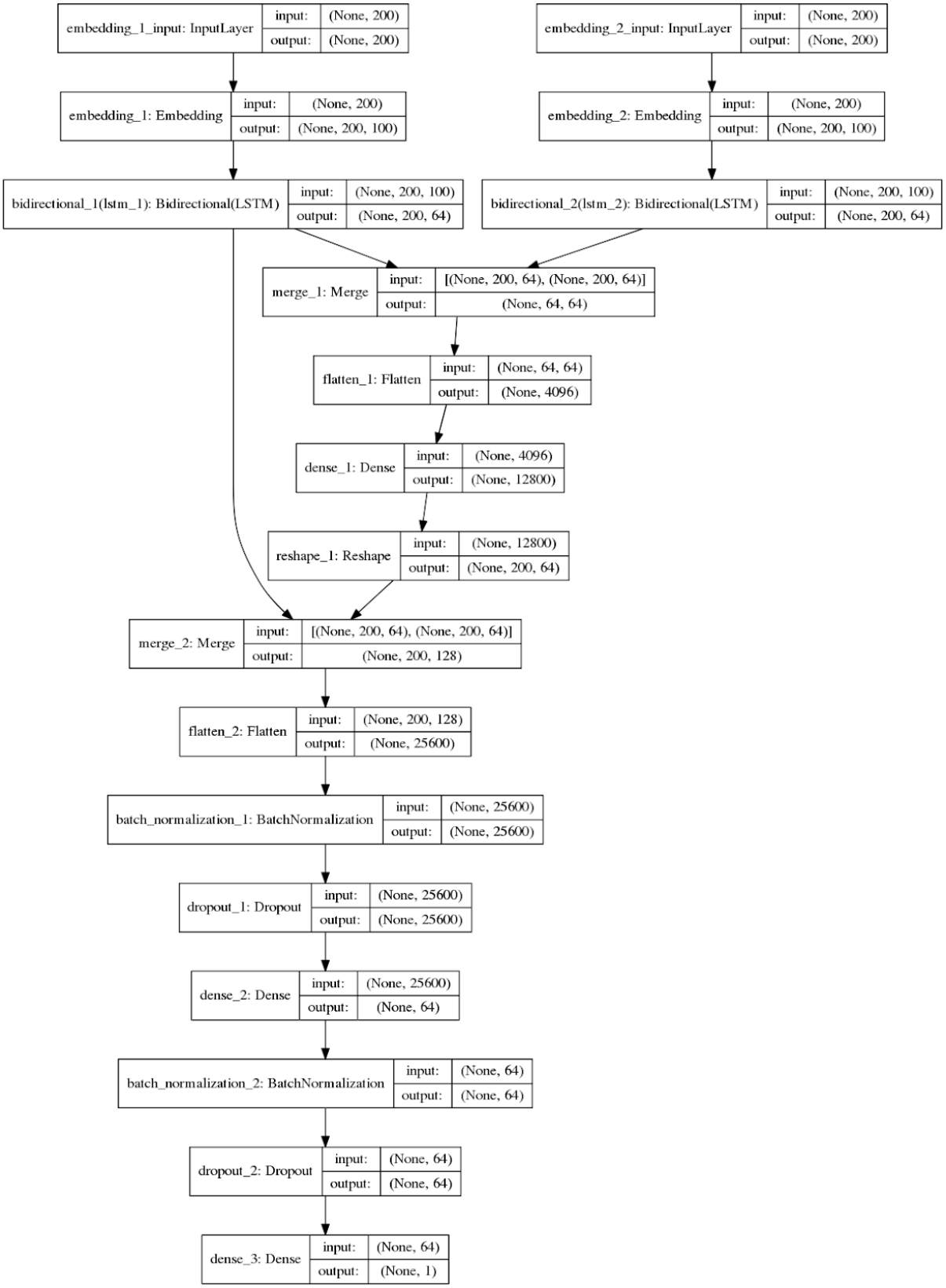
（五）基于 Bi-LSTM + Word2Vec + Attention 模型

基于深度学习的 NLP 研究方法，基本上都是先将句子分词，然后每个词转化为对应的词向量序列。第一个思路是 RNN 层，第二个思路是 CNN 层，而 Google 的大作《Attention Is All You Need》提供了第三个思路：Attention，注意力机制！

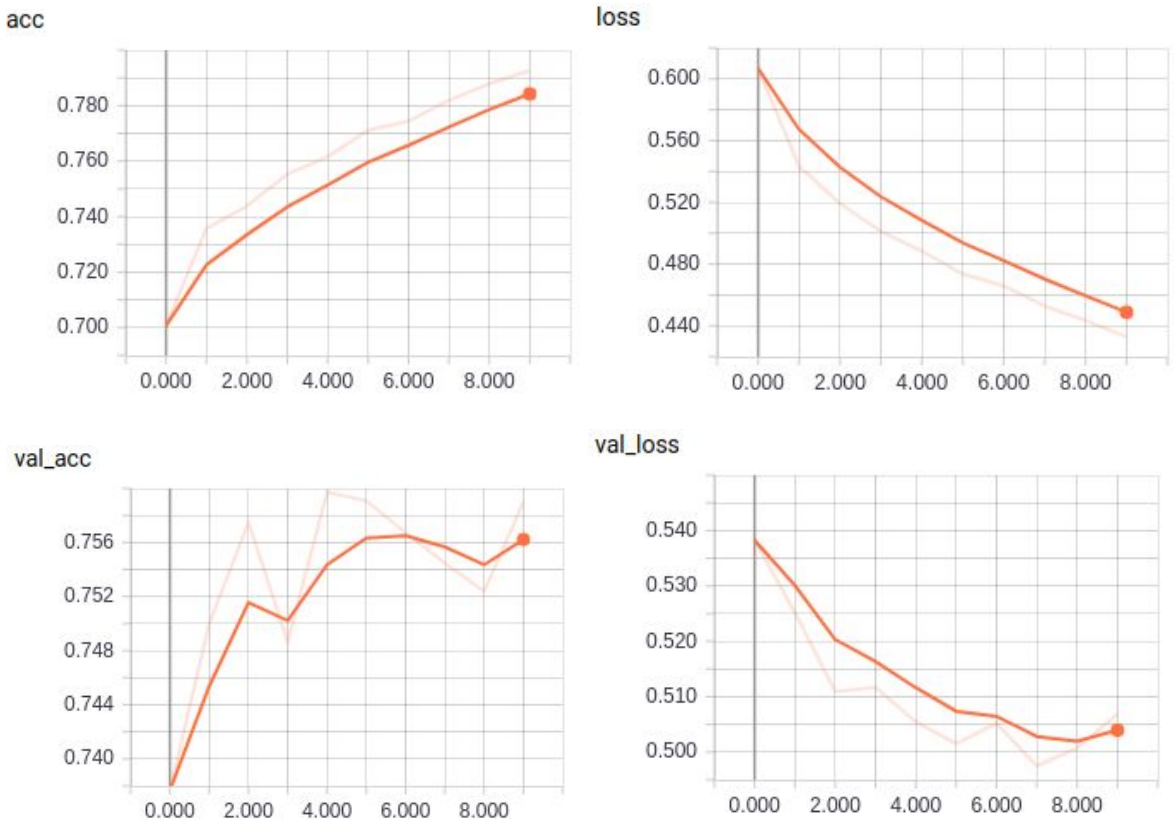
注意力机制来源于人脑，我们在阅读的时候，注意力通常不会平均分配在文本中的每个词。如果直接将每个时刻的输出向量相加再平均，就等于认为每个输入词对于文本表示的贡献是相等的。但实际情况往往不是这样，比如在情感分析中，文本中地名、人名这些词应该占有更小的权重，而情感类词汇应该享有更大的权重。所以在合并这些输出向量时，希望可以将注意力集中在那些对当前任务更重要的向量上。也就是给他们都分配一个权值，将所有的输出向量加权平均。

注意力机制的思路是：原先都是相同的中间语义表示 C 会替换成根据当前生成单词而不断变化的 Ci，每个 Ci 对应着不同单词的注意力分配概率分布。f 函数代表 Encoder 对单词的某种变换函数，g 函数代表 Encoder 根据单词的中间表示合成整个句子中间语义表示的变换函数，一般来说，g 函数是 f 函数加权求和。





【Bi-LSTM + Word2Vec + Attention 模型】



【实验数据】

指标：

	训练ACC	验证ACC	测试acc	MRR	MAP	TOP-1
准确率	0.7926	0.7597	0.7415	0.7048	0.6216	0.5600

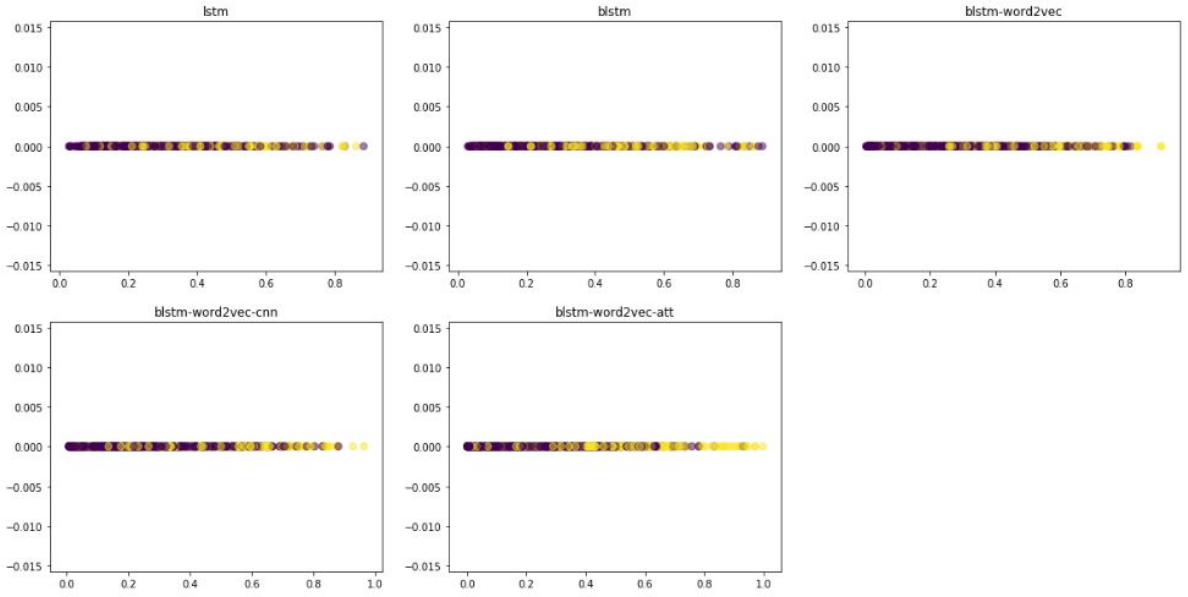
结论：进一步提升准确率，同时显著提升 MRR、MAP、TOP-1 指标。

（五）总结

模型	训练ACC	验证ACC	测试ACC	MRR	MAP	TOP-1
lstm	0.9540	0.7435	0.7201	0.6487	0.5550	0.4600
blstm	0.9545	0.7457	0.7301	0.6578	0.5691	0.4850
blstm + word2vec	0.7655	0.7568	0.7390	0.6868	0.5994	0.5300
blstm +	0.7998	0.7611	0.7383	0.6952	0.6116	0.5450

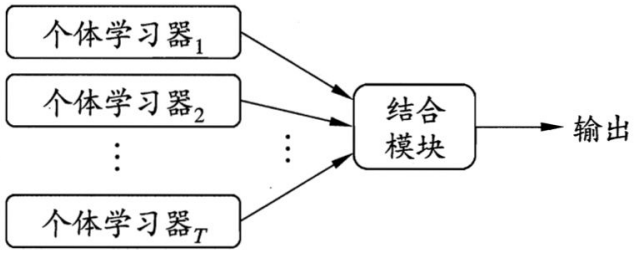
word2vec + cnn						
blstm + word2vec + att	0.7926	0.7597	0.7415	0.7048	0.6216	0.5600

下图为在测试数据集中，真实答案和预测答案的 概率分布-散点图：
其中紫色为 0 (false)、黄色为 1 (true)，对应横坐标为预测概率。



三、模型融合

使用单一模型的风险是很大的。首先，模型可能对数据集存在片面性，不能考虑到所有影响因素。其次，模型可能对数据集存在依赖性，容易造成过拟合。如果采用集成学习，通过将多个学习器进行结合，通常可以获得比单一学习器显著优越的泛化性能。



周志华的《机器学习》中提到，要获得好的集成，个体学习器通常应该“好而不同”，即个体学习器要有一定的“准确性”，并且要有“多样性”，即学习器间具有差异。

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集成	√	√	√

(a) 集成提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集成	√	√	×

(b) 集成不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集成	×	×	×

(c) 集成起负作用

假设存在三个模型（正确率均为 70%），以“投票集成”模型为例，通过简单的数学运算，研究集成学习如何降低错误率：

全部正确	0.7 * 0.7 * 0.7 = 0.3429
两个正确	0.7 * 0.7 * 0.3 + 0.7 * 0.3 * 0.7 + 0.3 * 0.7 * 0.7 = 0.4409
一个正确	0.3 * 0.3 * 0.7 + 0.3 * 0.7 * 0.3 + 0.7 * 0.3 * 0.3 = 0.189
全部错误	0.3 * 0.3 * 0.3 = 0.027
最终正确率 = 0.3429 + 0.4409 = 0.7838 ≈ 78% 说明集成学习理论上有利于提高准确率。	

我们选取以下方案进行测试：

方案	投票	平均(1:1:1:1:1)	权重 (1:1:2:3:3)
说明	即少数服从多数原则，分类得票数超过一半的作为预测结果。	将所有预测结果相加取平均值。	将所有预测结果按照权重计算。
准确率	0.7440	0.7415	0.7503
其中，“权重集成”模型提高了 1 个百分点 说明集成学习实际上也有利于提高准确率。			

四、泛化能力

1、定义：泛化能力（generalization ability）是指机器学习算法对新鲜样本的适应能力。学习的目的是学到隐含在数据对背后的规律，对具有同一规律的学习集以外的数据，经过训练的网络也能给出合适的输出，该能力称为泛化能力。

2、性质：通常期望经训练样本训练的网络具有较强的泛化能力，也就是对新输入给出合理响应的能力。应当指出并非训练的次数越多越能得到正确的输入输出映射关系。网络的性能主要用它的泛化能力来衡量。

3、实验：使用百度开源数据集 WebQA，该数据集含有 44 万条问答记录。

将 WebQA 数据集转换为赛题数据集格式：

answer 对应赛题数据集的 label，其中 no_answer 代表 0，其余回答代表 1。

【转换前】


```
{'evidences': {'Q_TRN_005637#00': {'answer': ['no_answer'],
   'evidence': '1、十月革命胜利,世界上出现了第一个社会主义国家.一个崭新的社会主义报刊体系在苏俄确立形成.<e>2、二战结束后,又有欧、亚、拉美一系列国家脱离了资本主义体系,走社会主义道路,社会主义报业得到很大发展.<e>3、“苏东”剧变后,这些国家的报业结构和性质发生了重大变化.<e>十六、苏联时期报刊体制的主要特征是怎样的?<e>1、苏联的报刊,都属于国家所有,是党和国家机构的重要组成部分；其基本职能是集体的宣传员、集体的鼓动员和集体的组织者.<e>2、苏联的各级报刊绝对服从于各级党委的领导.<e>3、苏联报纸信息来源单一,言论高度集中.<e>4、苏联报刊在建设时期是社会主义建设的工具.<e>十七、发展中国家报业又何共同特点?<e>1、早期报刊、尤其是报业发端较早的国家的早期报刊,大多是殖民者创办的；<e>2、随着反殖民主义反封建斗争的开展,这些国家的民族报刊逐步发展起来,并推动了反殖民主义反封建斗争的进程；<e>3、民族解放运动胜利后,大多数报业获得了前所未有的发展,但也有的国家报业重新陷入本国独裁者的控制之下.<e>十八、新闻通讯社是在怎样的背景下诞生的?它的功能与作用如何?'},
  'Q_TRN_005637#01': {'answer': ['no_answer'],
   'evidence': '1566年,世界最早的印刷报纸《威尼斯新闻》诞生于1566年的意大利威尼斯'},
  'Q_TRN_005637#02': {'answer': ['no_answer'],
   'evidence': '世界上最早的报纸诞生在1609年。'},
  'Q_TRN_005637#03': {'answer': ['中国'],
   'evidence': '北宋末年(公元11,12世纪)出现的印刷报纸,不仅是中国新闻史上最早的印刷报纸,也是世界新闻史上最早的印刷报纸.中国新闻事业历史的悠久,内容的丰富,是任何西方国家都难以比肩的.<e>中国古代的报纸产生于中国的封建社会时期,是封建地主阶级及其政治代表占统治地位的封建自然经济通过新闻手段的反映.在漫长的封建社会时期,中国古代的报纸,不论是官方的邸报,还是民办的小报和京报,都必然要和当时的封建统治者保持一定的联系,受他们的制约.官方的邸报固然是封建统治阶级的喉舌和御用的宣传工具,民办的小报和京报也只能在封建统治阶级的控制下活动,不能越雷池一步.封建统治者绝不允许可以自由报道一切消息和自由发表一切意见的报纸存在.中国古代的报纸在为当时的读者提供朝野政治和社会信息方面确实起过一定的作用,但始终没有摆脱统治阶级的掌握.中国古代报纸的历史,基本上是一部封建统治阶级掌握传播媒介,控制舆论工具,限制言论出版自由的历史.<e>中国古代的邸报有1200年左右的历史.小报有近千年的历史.民间报房出版的邸报,京报有近400年的历史.它们从诞生到结束,持续的时间都不算短,但发展不快,形式内容的变化不大.'},
  'Q_TRN_005637#04': {'answer': ['no_answer'],
   'evidence': '因此，一般认为，世界上最早的报纸诞生在1609年。'},
  'Q_TRN_005637#05': {'answer': ['中国'],
   'evidence': '报纸从诞生到今天已经走过了漫长的历史，公元前60年，古罗马政治家恺撒把罗马市以及国家发生的时间书写在白色的木板上，告示市民。这便是世界上最古老的报纸。中国在7世纪，唐朝宫廷内就发行过手写的传阅版，这应该算是中国最早的报纸。'},
  'Q_TRN_005637#06': {'answer': ['中国'],
   'evidence': '最早的写在纸上的报纸和印刷在纸上的报纸都诞生于中国.唐玄宗开元年间(公元713年--742年)出现的开元杂报,不仅是中国新闻史上最早的报纸,也是世界新闻史上最早的报纸.'},
  'Q_TRN_005637#07': {'answer': ['no_answer'],
   'evidence': '由于邮件一般是定期到达，最初是每周到达一次，因而最早出现的定期刊物多是周刊。后来随着社会经济的发展，各地经济联系的加强，邮件传递的次数日益增多，由每周一次逐渐地改为每周二次、每周三次，以至每天一次，于是新闻印刷品的刊期也逐渐缩短，由周刊改为周二刊，最后出现了日报。<e>从时间上看，世界上最早的定期刊物诞生于德国。早在1597年2月，萨穆埃尔·迪尔鲍姆就在奥格斯堡创办了一份类似于编年表的月刊。1609年，又有一家出版了19年的不定期报纸改为定期出版，名为《观察周刊》，每期一张，仅一条新闻。'},
  'Q_TRN_005637#08': {'answer': ['no_answer'],
   'evidence': '答：1566年,世界最早的印刷报纸《威尼斯新闻》诞生于1566年的意大利威尼斯16世纪，意大利港口城市威尼斯出现了资本主义萌芽。'},
  'Q_TRN_005637#09': {'answer': ['no_answer'],
   'evidence': '答：1566年,世界最早的印刷报纸《威尼斯新闻》诞生于1566年的意大利威尼斯邸报》是我国在世界上发行最早，时间最久的报纸。'},
  'Q_TRN_005637#10': {'answer': ['中国'],
   'evidence': '综上所述,有必要对中国古代报纸作一个历史评价:<e>中国是一个历史悠久的文明古国.中国的先民们曾经为世界物质和精神文明的发展做出过杰出的贡献,和现代新闻事业有着密切关系的造纸术和印刷术,就首先发明于中国.世界新闻事业(特别是其中的报刊部分)得以发展的物质条件是中国人首先提供的.<e>世界新闻事业史上最早的报纸也出于中国.最早的写在纸上的报纸和印刷在纸上的报纸都诞生于中国.'}},
  'question': '世界上最早的报纸诞生于'}
```


【转换后】

```
{'item_id': 5637,
 'passages': [{'content': '1、十月革命胜利,世界上出现了第一个社会主义国家.一个崭新的社会主义报刊体系在苏俄确立形成.<e>2、二战结束后,又有欧、亚、拉美一系列国家脱离了资本主义体系,走社会主义道路,社会主义报业得到很大发展.<e>3、“苏东”剧变后,这些国家的报业结构和性质发生了重大变化.<e>十六、苏联时期报刊体制的主要特征是怎样的?<e>1、苏联的报刊,都属于国家所有,是党和国家机构的重要组成部分;其基本职能是集体的宣传员、集体的鼓动员和集体的组织者.<e>2、苏联的各级报刊绝对服从于各级党委的领导.<e>3、苏联报纸信息来源单一,言论高度集中.<e>4、苏联报刊在建设时期是社会主义建设的工具.<e>十七、发展中国家报业又何共同特点?<e>1、早期报刊、尤其是报业发端较早的国家的早期报刊,大多是殖民者创办的;<e>2、随着反殖民主义反封建斗争的开展,这些国家的民族报刊逐步发展起来,并推动了反殖民主义反封建斗争的进程;<e>3、民族解放运动胜利后,大多数报业获得了前所未有的发展,但也有的国家报业重新陷入本国独裁者的控制之下.<e>十八、新闻通讯社是在怎样的背景下诞生的?它的功能与作用如何?',
   'label': 0,
   'passage_id': 563700},
  {'content': '1566年,世界最早的印刷报纸《威尼斯新闻》诞生于1566年的意大利威尼斯',
   'label': 0,
   'passage_id': 563701},
  {'content': '世界上最早的报纸诞生在1609年。', 'label': 0, 'passage_id': 563702},
  {'content': '北宋末年(公元11,12世纪)出现的印刷报纸,不仅是中国新闻史上最早的印刷报纸,也是世界新闻史上最早的印刷报纸.中国新闻事业历史的悠久,内容的丰富,是任何西方国家都难以比肩的.<e>中国古代的报纸产生于中国的封建社会时期,是封建地主阶级及其政治代表占统治地位的封建自然经济通过新闻手段的反映.在漫长的封建社会时期,中国古代的报纸,不论是官方的邸报,还是民办的小报和京报,都必然要和当时的封建统治者保持一定的联系,受他们的制约.官方的邸报固然是封建统治阶级的喉舌和御用的宣传工具,民办的小报和京报也只能在封建统治阶级的控制下活动,不能越雷池一步.封建统治者绝不允许可以自由报道一切消息和自由发表一切意见的报纸存在.中国古代的报纸在为当时的读者提供朝野政治和社会信息方面确实起过一定的作用,但始终没有摆脱统治阶级的掌握.中国古代报纸的历史,基本上是一部封建统治阶级掌握传播媒介,控制舆论工具,限制言论出版自由的历史.<e>中国古代的邸报有1200年左右的历史.小报有近千年的历史.民间报房出版的邸报,京报有近400年的历史.它们从诞生到结束,持续的时间都不算短,但发展不快,形式内容的变化不大.',
   'label': 1,
   'passage_id': 563703},
  {'content': '因此,一般认为,世界上最早的报纸诞生在1609年。', 'label': 0, 'passage_id': 563704},
  {'content': '报纸从诞生到今天已经走过了漫长的历史,公元前60年,古罗马政治家恺撒把罗马市以及国家发生的时间书写在白色的木板上,告示市民。这便是世界上最古老的报纸。中国在7世纪,唐朝宫廷内就发行过手写的传阅版,这应该算是中国最早的报纸。',
   'label': 1,
   'passage_id': 563705},
  {'content': '最早的写在纸上的报纸和印刷在纸上的报纸都诞生于中国.唐玄宗开元年间(公元713年--742年)出现的开元杂报,不仅是中国新闻史上最早的报纸,也是世界新闻史上最早的报纸.',
   'label': 1,
   'passage_id': 563706},
  {'content': '由于邮件一般是定期到达,最初是每周到达一次,因而最早出现的定期刊物多是周刊。后来随着社会经济的发展,各地经济联系的加强,邮件传递的次数日益增多,由每周一次逐渐地改为每周二次、每周三次,以至每天一次,于是新闻印刷品的刊期也逐渐缩短,由周刊改为周二刊,最后出现了日报。<e>从时间上看,世界上最早的定期刊物诞生于德国。早在1597年2月,萨穆埃尔·迪尔鲍姆就在奥格斯堡创办了一份类似于编年表的月刊。1609年,又有一家出版了19年的不定期报纸改为定期出版,名为《观察周刊》,每期一张,仅一条新闻。',
   'label': 0,
   'passage_id': 563707},
  {'content': '答:1566年,世界最早的印刷报纸《威尼斯新闻》诞生于1566年的意大利威尼斯16世纪,意大利港口城市威尼斯出现了资本主义萌芽。',
   'label': 0,
   'passage_id': 563708},
  {'content': '答:1566年,世界最早的印刷报纸《威尼斯新闻》诞生于1566年的意大利威尼斯邸报》是我国在世界上发行最早,时间最久的报纸。',
   'label': 0,
   'passage_id': 563709},
  {'content': '综上所述,有必要对中国古代报纸作一个历史评价:<e>中国是一个历史悠久的文明古国.中国的先民们曾经为世界物质和精神文明的发展做出过杰出的贡献,和现代新闻事业有着密切关系的造纸术和印刷术,就首先发明于中国.世界新闻事业(特别是其中的报刊部分)得以发展的物质条件是中国人首先提供的.<e>世界新闻事业史上最早的报纸也出于中国.最早的写在纸上的报纸和印刷在纸上的报纸都诞生于中国.',
   'label': 1,
   'passage_id': 563710}],
 'question': '世界上最早的报纸诞生于'}
```


4、单个模型泛化评估

模型	测试ACC	MRR	MAP	TOP-1
lstm	0.6899	0.6809	0.5539	0.5112
blstm	0.6876	0.6754	0.5491	0.5053
blstm + word2vec	0.7072	0.6987	0.5765	0.5362
blstm + word2vec + cnn	0.7103	0.7121	0.5918	0.5547
blstm + word2vec + att	0.6553	0.6688	0.5586	0.5000

5、模型融合泛化评估：

方案	投票	平均 (1:1:1:1:1)	权重 (1:1:2:3:3)
准确率	0.7030	0.7050	0.7055

6、总结：

单个模型泛化能力不显著，甚至远低于平均水平。而模型融合能够在大数据下继续保持泛化能力，表明我们的模型设计合理。

参考文献、参考链接：

Embedding:

https://keras-cn-docs.readthedocs.io/zh_CN/latest/blog/word_embedding/

MRR、MAP:

https://en.wikipedia.org/wiki/Information_retrieval

LSTM:《LONG SHORT-TERM MEMORY Technical Report》

http://www.bioinf.jku.at/publications/older/2604.pdf

Bi-LSTM:《Bidirectional LSTM-CRF Models for Sequence Tagging》

https://arxiv.org/pdf/1508.01991v1

Word2Vec:《Distributed Representations of Words and Phrases and their Compositionality》

https://arxiv.org/abs/1310.4546

CNN:

http://cs231n.github.io/convolutional-networks/

Attention:《Attention is All You Need》

https://arxiv.org/abs/1706.03762

WebQA: https://www.spaces.ac.cn/archives/4338

《**机器学习**》，周志华，清华大学出版社

《**深度学习**》，[美]Ian,Goodfellow,[加]Yoshua,Bengio,[加]Aaron,Courville，人民邮电出版社