**报告题目：基于BiLSTM模型的中文分词（seq2seq）及相关研究**

**报告提交时间：2020年12月18日星期五**

**中文摘要**

随着时代的发展，人工智能已经在越来越多的领域取得令人十分瞩目的应用。直至目前为止，有相当数量的研究者将人工智能研究的众多成果之一，神经网络模型，引入了自然语言处理这个领域中。分词作为中文自然语言处理当中最为重要的基础工作，已经运用神经网络模型取得了一些成果。

从机器学习的角度看，中文分词可以抽象为序列标注任务。序列标注任务指的是：在给定的标签集合中，为观察序列里的每一个元素标注特定的标签。传统的序列标注方法需要人工构造特征，并不具有广泛的适用性，因此，本文借用神经网络模型在学习特征方面的优势，利用神经网络模型技术对中文分词任务进行研究。

在已有工作基础上，本文提出了通过利用BiLSTM模型预测“4-标注”体系以实现中文分词的方法。具体工作如下：

（1）相比RNN、BiRNN等模型，本文选用的BiLSTM模型在解决自然语言处理中的序列标注问题时存在着很大的优势，它能够利用双向LSTM神经网络，既保存了文本序列中前文信息，又保存了后文信息，较好地符合序列标注任务的要求；（2）本文评估了BiLSTM模型在直接标注序列，与BiLSTM结合维特比算法标注序列之间的精度值（Precision）、回归值（Recall）和f1值的差异；（3）本文简单探索了训练时期（Epoch）与泛化性能（Generalization performance）之间的关系。

本文利用BiLSTM模型，在sighan2005公开数据集上进行了训练以及实验。最终的结果表明：该方法可以有效地完成中文自然语言处理中的分词任务，并对歧义句、人名等有较好的处理结果。

**关键词：**BiLSTM模型； 中文分词； Keras； GPU加速

**Abstract**

With the development of the times, artificial intelligence has been applied in more and more fields. Up to now, a considerable number of researchers have introduced neural network model, one of the achievements of artificial intelligence, into the field of NLP. As the most important basic work in Chinese natural language processing, word segmentation has achieved some results by using neural network model.

From the perspective of machine learning, Chinese word segmentation can be abstracted as a sequential tagging task. Sequence tagging task refers to: in a given tag set, each element in the observation sequence is labeled with a specific tag. Traditional sequence tagging methods need to construct features manually, which is not widely applicable. Therefore, this paper uses the advantages of neural network model in learning features, and uses neural network model technology to study Chinese word segmentation task.

On the basis of existing work, this paper proposes a method of Chinese word segmentation by using BiLSTM model to predict the "4-tagging" system. The specific work is as follows:

(1) Compared with RNN, BiRNN and other models, the BiLSTM model selected in this paper has great advantages in solving the problem of sequence annotation in NLP. It can use the bidirectional LSTM to save both the preceding information and the subsequent information in the text sequence, which meets the requirements of the sequence annotation task;

(2) This paper evaluates the differences of precision, recall and F1 values between bilstm and bilstm combined with Viterbi algorithm in labeling sequences directly;

(3) the relationship between epoch and generalization performance is briefly explored.

In this paper, BiLSTM model is used to train and experiment on the open data set of SIGHAN2005. The final results show that this method can effectively complete the word segmentation task in Chinese natural language processing, and has good processing results for ambiguous sentences and names.

**Keywords:** BiLSTM model; Chinese Word Segmention; Python Keras; CUDA

**1.选题背景及意义**

互联网的迅猛发展使得当前信息增长速度爆炸，如此海量的文本信息通过人工处理及分类已经不再具有可行性。我们需要利用NLP相关技术来分析转化例如文本信息这样大量非结构化的数据。中文分词作为中文自然语言处理领域最重要的基础工作之一，是其它中文自然语言处理任务的必要且关键的环节，其性能、处理效果的好坏，对接下来的工作而言尤为重要。

自从2012年Geoffrey Hinton和他的学生改良了卷积神经网络（CNN）后，深度学习便在各个领域掀起了属于它的热潮。自然语言处理也不例外。

在NLP发展初期，分词任务主要基于词典的匹配方法，又叫机械分词法。它是按照一定的策略将待分析的汉字串与一个“充分大”的机器词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功（识别出一个词）。常用的字符串匹配方法有如下几种：正向最大匹配法（从左到右）；逆向最大匹配法（从右到左）；最小切分（每一句中切出的词数最小）；双向最大匹配（进行从左到右、从右到左两次扫描）。这类算法的优点是速度快，时间复杂度可以保持在 O(n)，实现简单，效果尚可；但对歧义和未登录词处理效果不佳。

第二类方法是基于统计的分词方法。基于统计的分词方法是在给定大量已经分词的文本的前提下，利用统计机器学习模型学习词语切分的规律（称为训练），从而实现对未知文本的切分。例如最大概率分词方法和最大熵分词方法等。随着大规模语料库的建立，统计机器学习方法的研究和发展，基于统计的中文分词方法渐渐成为了主流方法。主要的统计模型有：N元文法模型（N-gram），隐马尔可夫模型（Hidden Markov Model，HMM），最大熵模型（ME），条件随机场模型（Conditional Random Fields，CRF）等。在实际的应用中，基于统计的分词系统仍然需要使用分词词典来进行字符串匹配分词，同时使用统计方法识别一些新词，即将字符串频率统计和字符串匹配结合起来，既发挥匹配分词切分速度快、效率高的特点，又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。

然而随着深度学习的兴起，基于神经网络的特征学习方法为自然语言处理提供了一种崭新的思路。

深度学习方法在完成自然语言处理任务方面具有很大优势：1、基于深度学习的方法构建模型，可以自动捕捉到自然语言处理任务中所需要的特征；2、在自然语言处理领域，已标记的数据集十分稀少珍贵，更有甚者需要通过商业途径获取，极大提高了研究的门槛。深度学习相较于传统方法可以利用较低的成本处理无标记的数据；3、自然语言处理中许多任务是密切相关的，例如分词与词性标注与命名实体识别间的关系正是如上所述。传统方法往往将其看做各自独立的任务而忽略了它们之间的关系，而基于深度学习的方法可以在特征提取级别对这些任务统一建模。这不仅使得模型泛化性能好，也能在各任务间互为补充。

**2.研究现状分析**

本节主要大概地介绍深度学习的简单理论基础以及发展现状，包括神经元概念，模型结构，模型训练等，为之后介绍本文研究内容做简单铺垫。

**2.0** 神经元与神经网络模型及激活函数

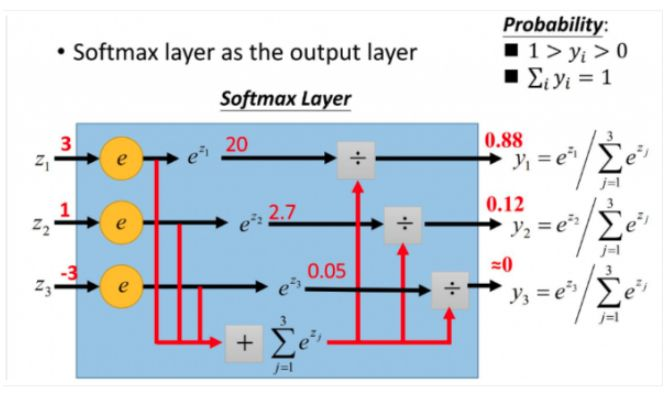
神经元是具有标量输入和标量输入的计算单元，每个输入都有与其相关联的权重。神经元将输入乘以与其对应的权重后相加，通过一个非线性函数f，再将结果传递给输出。

这个非线性函数f我们将其称之为激活函数，这个函数主要作用是为神经网络提供非线性的建模能力。如果没有激活函数，该神经网络仅能表达线性映射。即使该网络有再多的隐藏层，其与单层神经网络也是等价的，这会让拟合效果大打折扣。可以说，这是神经网络模型中十分重要的一部分。

激活函数根据其不同作用可分为许多种。在本文所建立的模型中，使用的是Softmax作为激活函数。举一个简单的例子：设一个数组n，ni表示数组n中的第i个元素。则该元素的softmax值即是：

***Si = ei/Σjej***

下图可以帮助更好理解softmax的工作方式。

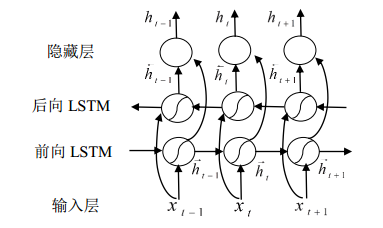


图（1） softmax工作方式

可以看到，输入分别为3、1、-3，经过softmax函数映射到[0, 1]之间，映射后的值满足概率的性质（即映射后的值之和为1）。那么便可以选取其中概率最大者作为输出结点，也就是最终的的预测目标。

**2.1** LSTM与BiLSTM

LSTM的全称是Long Short-Term Memory，它是循环神经网络（Recurrent Neural Network，RNN）的一种。LSTM由于其设计，非常适合用于对时序数据的建模，如文本数据。BiLSTM是Bi-directional Long Short-Term Memory的缩写，是由前向LSTM与后向LSTM组合而成。两者在自然语言处理任务中都常被用来建模上下文信息。



图（2） 在时间上展开的BiLSTM模型

**2.2** BiLSTM的特点及选用原因

相较LSTM，BiLSTM具有前后双种方向捕捉语义依赖的功能，这样的特点能让其更好地注意到前后双向信息间的交互。如“我是广州人……”放在文章的开头，而“我在\_\_\_工作”放在文章的结尾，BiLSTM能对空格中的内容作出良好的预测表现。

**2.3** 标注体系

在中文分词中，对字进行标注是一个对字进行抽象的过程，也是必不可少的处理。这意味着将成千上万个机器无法理解的汉字，映射到了机器能够理解的标签上。这样的过程需要通过2.0小节所展示的方式（神经网络与激活函数），结合2.1与2.2小节中所介绍的模型（BiLSTM模型）来完成。

本文使用的“4-标注”体系，标签（label）包括“s”、“b”、“m”与“e”。它们分别代表:

singl（独自成词的单字)；

begin（位于词开头的字）；

middle（位于三字以上的词的内部的字）；

end（位于词结尾的字）。

例如，在“我明天就交作业”这句话中，其标注应为“sbessbe”。四标注不是唯一的标注方式，类似的标注方式还有六标注等。理论上来说，标注越多，结果也就越精细，歧义亦会越少。然而标注过多会造成维度太高，样本不足等麻烦。故而常见的标注方式还是以四标注居多。

**2.4** 中文分词的评估指标

中文分词的评估指标同机器学习中的评估指标类似，通常由三个值来判断：分别是精确率（precision，又叫查准率），召回率（recall，又叫查全率）以及fscore值。f1代表精确率和召回率同样重要。

对于一个长度为n的字符串，分词结果是一系列的词语。设各个词在文中的起止区间可以记做[i, j]。标准答案中，所有的词的起止区间构成一个集合A，字符串n分词结果中，所的词的起止区间构成一个集合B，那么我们可得精确率和召回率的计算公式如下：

***Precision = |A∩B|/|B|, Recall = |A∩B|/|A|***

举个例子，比如“结合成分子时”一句：

标准分词为[‘结合’,‘成’,‘分子’,‘时’]；

分词区间[(1, 2), (3, 3), (4, 5), (6, 6)]构成集合A；

算法分词为[‘结合’, ‘成分’, ‘子时’];

分词区间为[(1, 2), (3, 4), (5, 6)]构成集合B。

根据上文所述的计算方法，我们可知该种分词方式的P值 = 1/3， R值 = 1/4， f1 值 = 2/7.

**3.本文算法**

**3.1** 算法概述与框架图

本节的主要内容是：研究基于BiLSTM模型实现seq2seq（序列到序列）的中文分词算法。值得注意的一点是，本研究是在BiLSTM模型结合维特比算法已得到较好处理效果的情况下，尝试去除维特比算法而使用BiLSTM模型直接输出预测序列，原因如下：

1、BiLSTM+维特比算法的本质仍是n-gram模型，只是将以往已经被尝试过的方法，如之前学习过的HMM模型+维特比算法，中的预测概率的部分变为了神经网络；

2、维特比算法会降低代码可读性，增加复杂度，尤其是面对语料库巨大的情景；

3、BiLSTM本身即可做到seq2seq（序列到序列）的输出，将输出结合softmax，便很容易得到概率大小；

4、维特比算法的核心原理在于动态规划寻找最大概率和的路径，然而，根据学习经验来看，期望的预测标签往往概率值与别的值差别极大，这意味着前文累加的过程中影响极小，导致概率最大者往往就是正确的。在后文将给出实践证明。

设想的算法工作原理为：

1. 输入待切分的序列；
2. 拟合训练后的模型，对待切分的序列进行标注；
3. 依据序列的位置进行分词处理。

研究主题所使用的逻辑框架图如下所示：

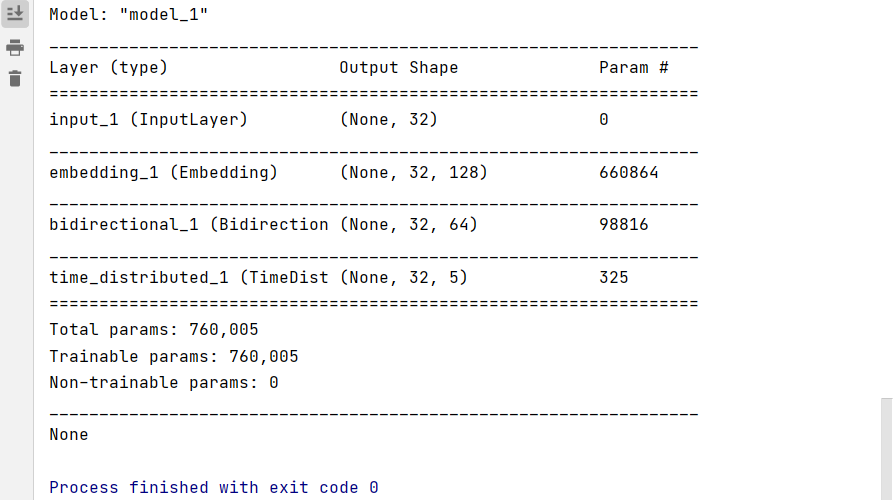
图（3）逻辑框架图

首先需要建立BiLSTM模型，这部分由代码中的lstm\_model.py完成；其次需要数据集对建立好的模型进行训练，这部分由代码train.py完成；最后我们可以选择分词（val.py）或者对其进行性能评估（test.py）。

**3.2** 算法各模块流程图与公式文字描述

3.2.1 建模

在工作的开始，我们需要借助keras建立一个BiLSTM神经网络的模型。我们对这个模型的期望输出是为字典，其中key表示标签，值表示各标签的预测概率。这个模型主要包括了下列所述的几个部分，由model.summary()我们可见：

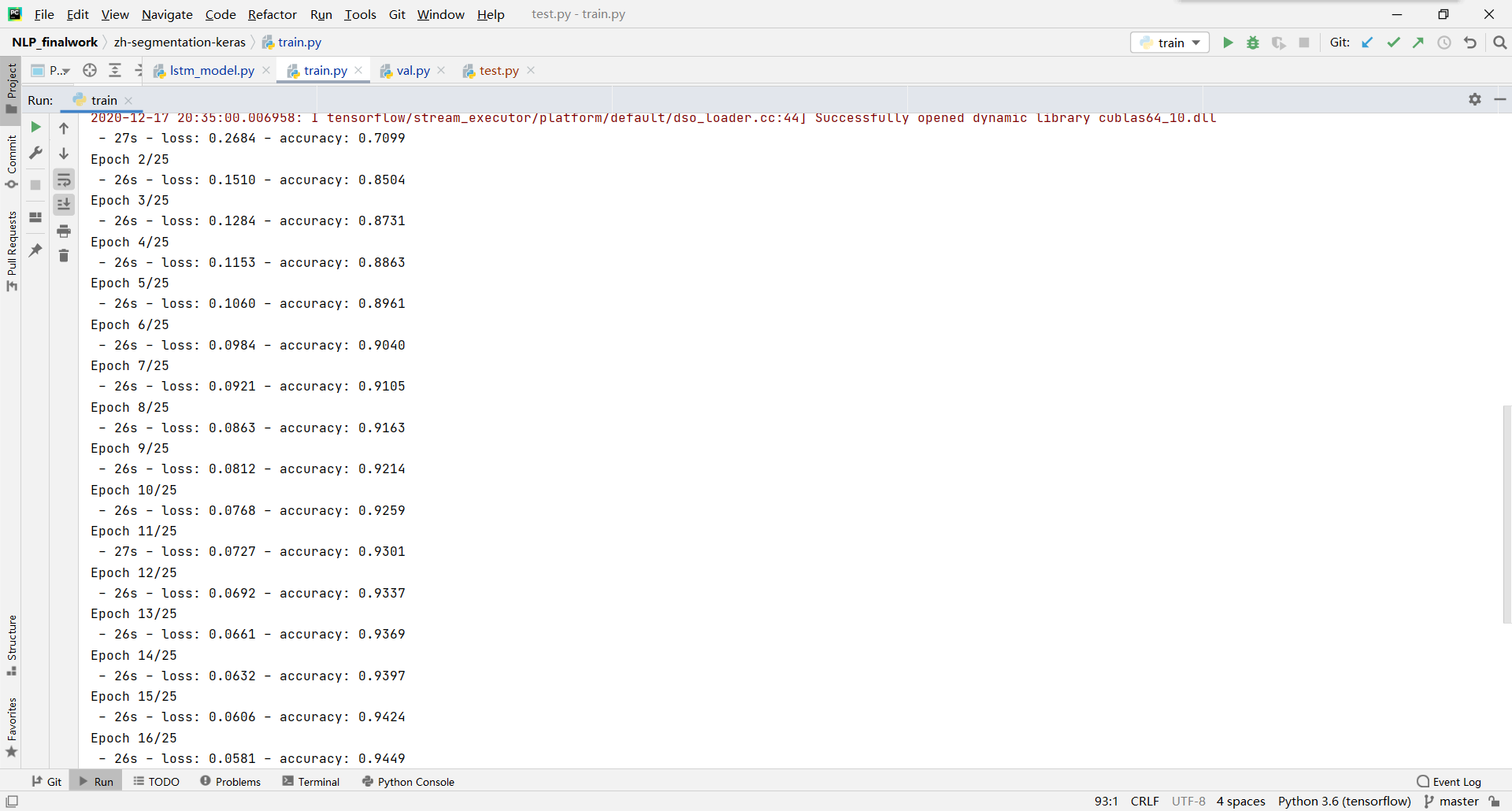


图（4） 建立好的BiLSTM模型

其中我们可以知道，模型对应的输入是（32， 128），共有约76万个参数。

3.2.2 训练过程

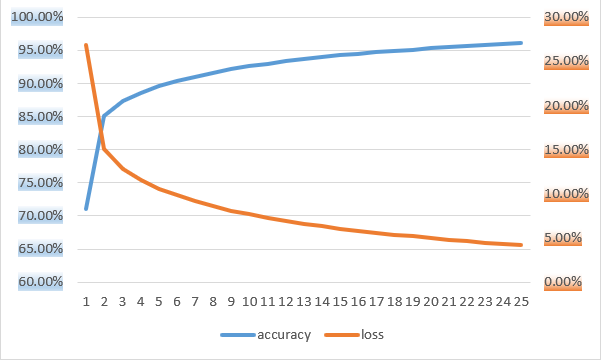
我们使用的训练集是bakeoff2005语料中MSR（微软亚洲研究院）所提供的、带有四标注的部分。将数据预处理，规范化之后，便可以开始训练了。训练过程中，使用了GPU进行加速，出于自身设备性能考虑，maxlearn设置为32。



图（5） 将epoch设置为25时，每个epoch的指标

Epoch可译为“轮次”，一个epoch对应的就是网络的一轮更新。每一轮更新中网络更新的次数可以随意，但通常会设置为遍历一遍数据集。因此一个epoch的含义是模型完整的看了一遍数据集。 设置epoch的主要作用是把模型的训练的整个训练过程分为若干个段，这样我们可以更好的观察和调整模型的训练。当指定了验证集时，每个epoch执行完后都会运行一次验证集以确定模型的性能。另外，我们可以使用回调函数在每个epoch的训练前后执行一些操作，如调整学习率，打印目前模型的一些信息等。

然而在选择训练轮次时，我们往往会担忧过拟合的问题，为了避免这种情况的发生（尽管使用的训练集足够大），我们检查了Accuracy在每个轮次中的情况：



图（6） Accuracy和loss与epoch间的关系。横坐标为epoch

幸运的是，我们的模型在25个epoch内的Accuracy始终稳定上升，表示没有出现过拟合的情况。而在25epoch时Accuracy达到了96.12%，表现较好。

3.2.3 分词与评估

3.2.3.1分词

在训练好模型之后，便可以使用保存好的权重对待分割的句子进行序列标注（预测）。具体过程为：

1. 将输入的文本按行分割，一行作为一次predict的对象；
2. 将model.predict的返回值字典化（使用标签作为key）；
3. 遍历字典，将预测标签为“s”与“m”的词在其后断开，完成分词。

3.2.3.2评估

与分词不同，评估用到的大容量的测试样本决定了需要特别为其设计方法。我们选择的评估体系在2.4节中已经简要介绍过，并且值得一提的是，我们设计了一个结合维特比算法横向比对的环节，以研究两者在中文分词任务处理方面的性能优劣。步骤如下：

① 用于评估的函数需要两个参数：A、算法（BiLSTM模型/BiLSTM模型+维特比算法）对测试集给出的分词结果与B、gold集（标准分词）的分词结果。两者的类型皆为链表；

② 获得B的长度，意为测试集中句子条数。

③ 利用列表推导式，查找集合A与集合B中相同不相同的数目，并且计数；

1. 根据correctPred与A、B长度，计算P值、R值与f1值，并返回。

**3.3** 算法细节

**3.3.1** 中间值展示与分析

以句子“结合成分子时。结婚的和尚未结婚的。”为例。分词结果如下：



图（7） 例子分词结果

可以看到分词结果表现良好，未受到歧义的影响。下一步我们查看预测标签在其中各个汉字上的概率（每行对应一个字，空行表示句子与句子间分隔）：

[{'s': -2.262371, 'b': -0.11025185, 'm': -12.143046, 'e': -8.173215},

{'s': -6.170763, 'b': -4.072777, 'm': -4.6322136, 'e': -0.029280744},

{'s': -0.7691913, 'b': -3.000892, 'm': -3.57915, 'e': -0.77877337},

{'s': -7.188862, 'b': -0.29343805, 'm': -1.3726169, 'e': -9.14826},

{'s': -6.64331, 'b': -16.539183, 'm': -9.358721, 'e': -0.0013899922},

{'s': -3.0518044e-05, 'b': -14.437579, 'm': -14.418417, 'e': -10.43521}]

[{'s': -10.004602, 'b': -4.5300556e-05, 'm': -16.616095, 'e': -20.048607},

{'s': -5.3633633, 'b': -18.0418, 'm': -13.1594925, 'e': -0.00469809},

{'s': -1.4305125e-06, 'b': -14.426297, 'm': -15.694129, 'e': -14.1902895},

{'s': -0.0015478185, 'b': -6.4800415, 'm': -15.9714155, 'e': -11.271168},

{'s': -3.5285428, 'b': -0.33923778, 'm': -5.5185914, 'e': -1.3691552},

{'s': -0.98048, 'b': -10.45185, 'm': -6.0142207, 'e': -0.47417933},

{'s': -7.7457504, 'b': -0.0006849727, 'm': -8.310634, 'e': -11.976471},

{'s': -4.717671, 'b': -16.925915, 'm': -12.670527, 'e': -0.008979401},

{'s': -2.3841887e-06, 'b': -17.254147, 'm': -18.917187, 'e': -12.936456}]

稍加观察，在“结合成分子时。结婚的和尚未结婚的。”的第一个结字当中，标签为b的概率（取对数处理）远远大于其余三者，按照预测标签为最大概率标签，第一个“结”字的预测标签是为b。剩余字对其做同样处理，得到输出序列“besbes。bessbebes。”按照该输出序列划分的分词结果为['结合', '成', '分子', '时', '。', '结婚', '的', '和', '尚未', '结婚', '的', '。']，符和正确划分答案。

在该例中，P、R、f1三值皆为1，分词表现优秀，验证了3.1节中陈述的理由。

**4.实验结果或系统展示**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | f1 |
| epoch=10； mode= DirectPred； test1； | 0.8387681159420289 | 0.8063392546151167 | 0.822234061445569 |
| epoch=10； mode= DirectPred； test2； | 0.8571428571428571 | 0.7923713566030947 | 0.8234854151084517 |
| epoch=10 mode= DirectPred test3； | 0.844796828543112 | 0.8185135394661033 | 0.8314475224346469 |
| epoch=10 mode= vtbi test3； | 0.8482887123663823 | 0.8306126368350297 | 0.8393576245694047 |
| epoch=10 mode= vtbi test2； | 0.8626923076923076 | 0.8071248650593739 | 0.8339840118981222 |
| epoch=10 mode= vtbi test1； | 0.8461263834344876 | 0.8254963427377221 | 0.8356840620592385 |

表（1） 相同条件下（epoch与测试集）BiLSTM与BiLSTM+维特比算法的表现情况

表中结果证明了，相同条件下BiLSTM与BiLSTM+维特比算法的表现情况大体一致，标准差小于

**5.问题的讨论与分析**

**5.1** 分词情况不理想的例子

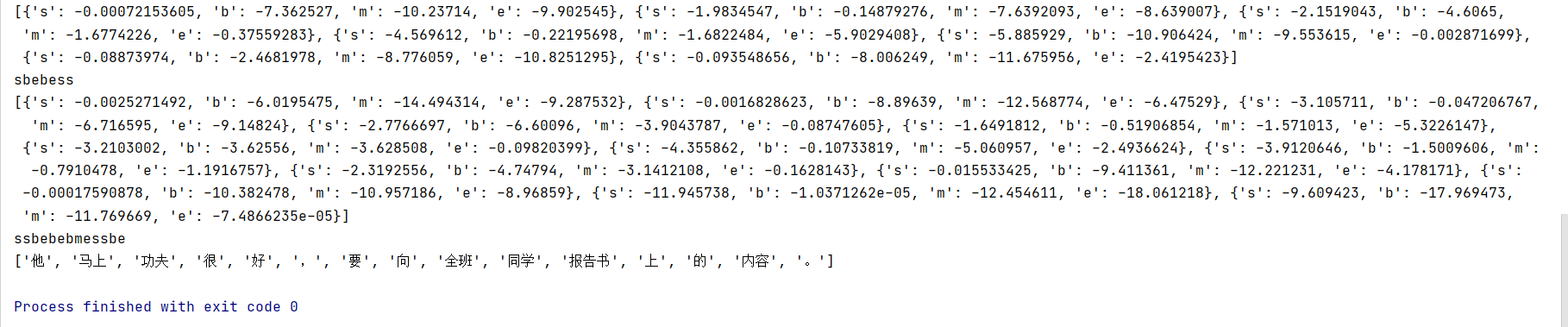
在本次作业中，我自然也遇到了不理想的分词情况。这里以“他马上功夫很好，要向全班同学报告书上的内容。”为例进行分析。

理想情况下，分词结果应为[他 马 上 功夫 很 好 ， 要 向 全班 同学 报告 书 上 的 内容 。 ]，关键的歧义点在“马上”以及“报告书”二词。

“马上”既可以理解为很快，直接（right now），还可以理解为在马背上（on a horse）。在“他马上功夫很好”这句话中，正确的理解应为在马背上，意为“他的骑术很好。”

“报告书”既可以理解为报告（report）， 也可以理解为报告书上的内容（report the connent in book）。在“要向全班同学报告书上的内容”这一句话中，正确的理解显然应该是后者。

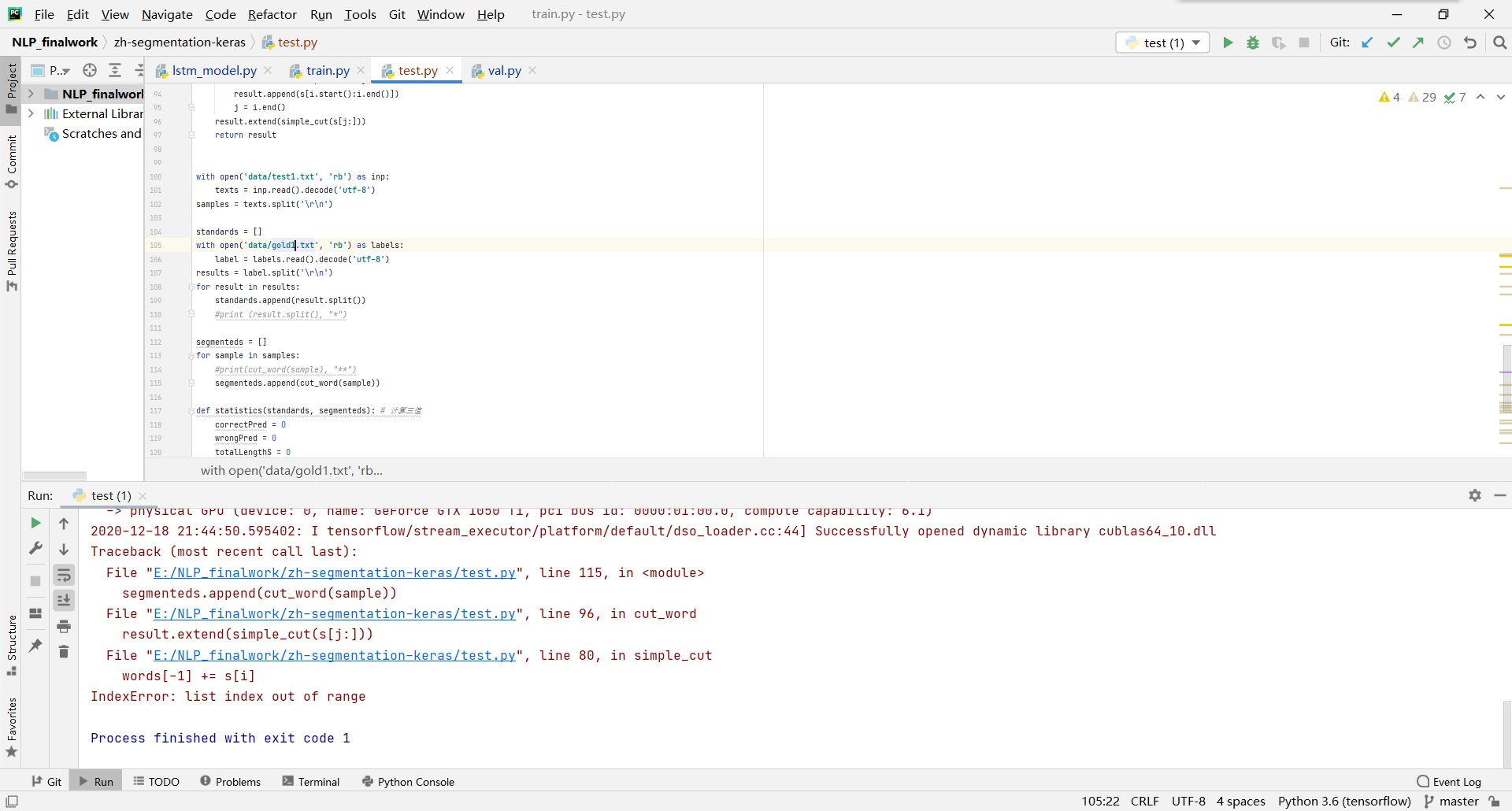
当我们print出各个词的概率时，我们可以发现，毫无疑问地都是第一种理解占有比例极大。



图（8） 例子分词结果

将print的内容进行分析，如果想要完整理解句子的意思，在现有体系下（依赖于最大概率预测标签）想要解决问题，必须增加权值。我的初步设想是加入注意力机制，这个想法来源于 ***Improving Chinese Word Segmentation with Wordhood Memory Networks*** 这篇论文。

**5.2** 评估相关问题

在计算相关的指标时，我设法直接使用‘pku\_test’与‘pku\_test\_gold’当做3.2.3.2节当中的输入与输出，但是往往会报错。情况如下：

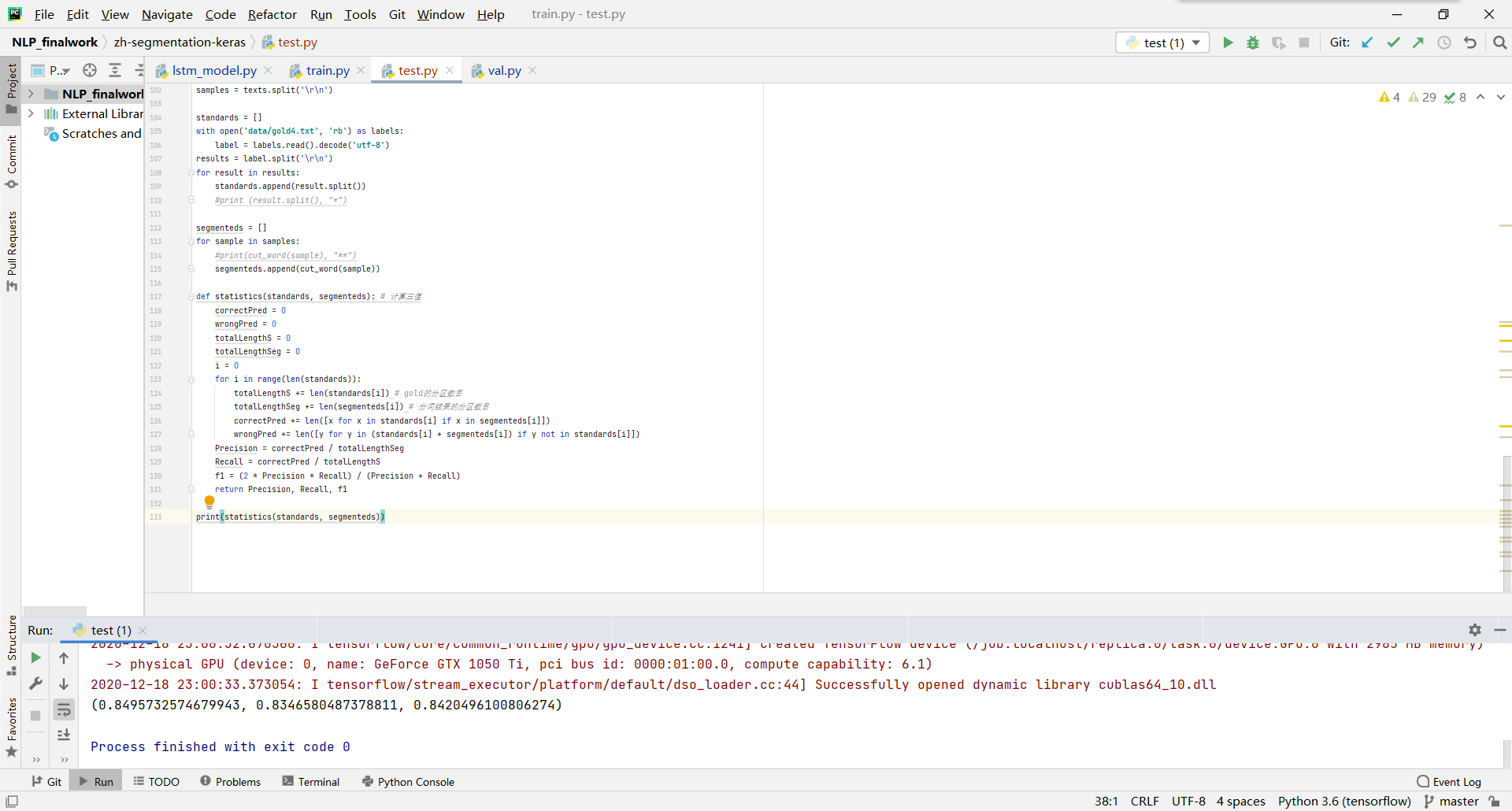
图（9） 报错情况

起初我认为出现了bug，然而在经过排查后，发现如果测试集较小的话则能够正常运行，并且给出p值r值f1值。这说明并不是index溢出导致的。

因此我将目光锁定在了内存方面。解决尝试如下：

1. 试着缩小测试集的大小；
2. 将建模时的maxlen由32改为了96，然而这么做除了导致内存占用变多与训练时间大大加长（由26s/epoch->80s/epoch）之外，并没有什么用。
3. 将评估模块中的maxlen由32改为了96；
4. 由于每次训练时间较长，检测成本大，我也尝试过保存多种权重的模型，不过收效甚微。

最后的解决方案为1+3，并且在‘pku\_test’中随机选择了多个长度的文段用作测试集，最后成果如下：



图（10） 解决方案

**6.结论**

BiLSTM实现seq2seq的预测序列是在相当程度上可行的方案，由表（1）多组测试数据可知，BiLSTM模型于BiLSTM+维特比算法的差距不大。

**7.学习体会和建议**

这次报告时间非常地紧张，最后的三值相较现在成熟的模型也并不好看。然而从选题，到配置环境（一定要选择合适版本的依赖！），到分析、解决问题，再到撰写文档，在一周左右的时间兼顾其他学科的作业同时能够达到甚至略为超出自己定下的底线完成度，我觉得这也是一次相当丰富充实的体验。

从我自己的眼界来评价，这次的作业尚有许多开展不完善的地方。例如表格（1）本应当有不同epoch下，两种模型在3分测试集下的表现对比，然而由于自己时间规划大大欠妥不得已省去。又比如最关键的评估，一来没有使用完整的数据集，二来分值在整篇文档中的地位不够突显，这也暴露出自己代码能力不足与组织文档能力欠缺的毛病。

这就是我初次应用神经网络的经验与体会，虽然结果不完美，但我已经尽力了。