同济大学
计算机科学与技术系

中文信息处理实验报告

学　　号 **1852694**

姓　　名 吴其平

专　　业 计算机科学与技术

授课老师 卫志华

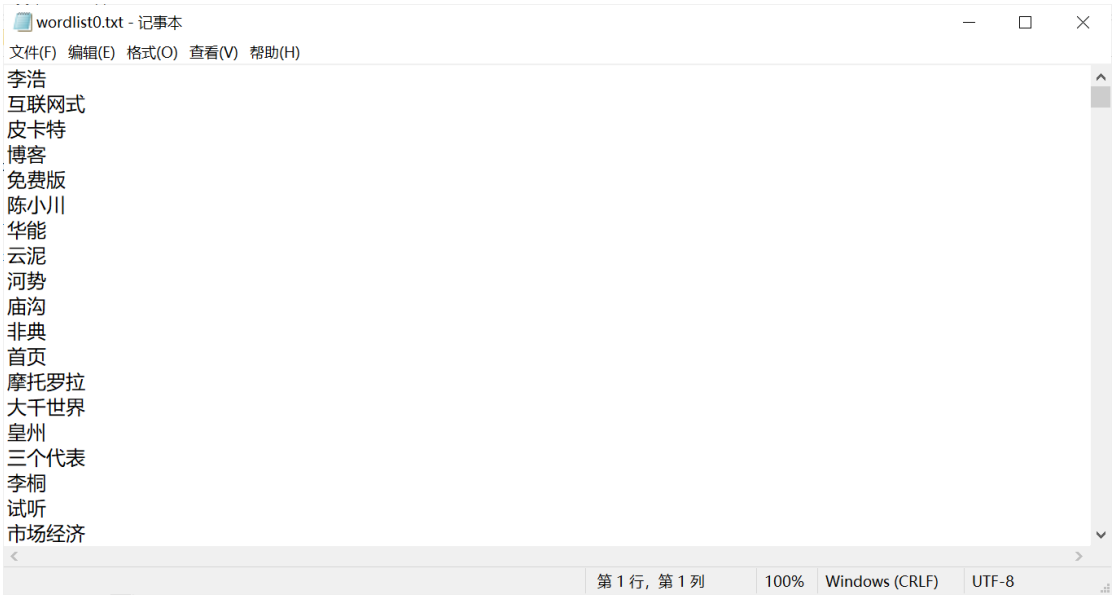
一、实验内容

对中文句子进行词级别的划分。

二、数据说明

本实验使用的数据集包括词典和语料库。

词典保存在 txt 文件中，其格式如下图所示。

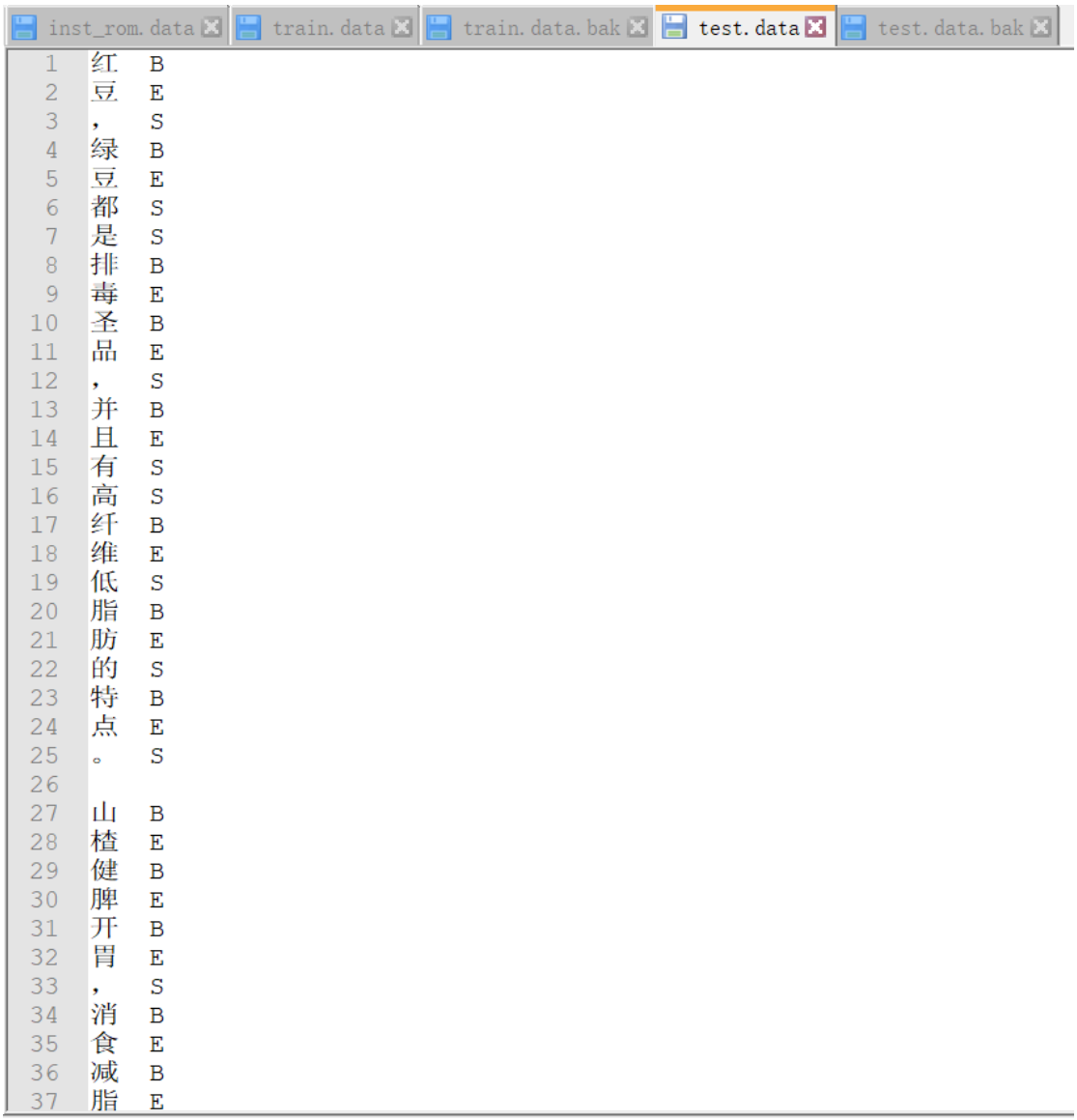


语料库为人民日报预料库 2014，使用 4-tag 标签，即：S（单字），B（词首），M（词中），E（词尾）。其中 train.data 为训练集，test.data 为测试集，其格式如下图所示。

inst_rom.data | train.data | train.data.bak | test.data | test.data.bak

```
在 S
广 S
工 S
龙 B
洞 E
校 B
区 E
附 B
近 E
的 S
出 B
租 M
屋 E
， S
董 B
云 M
为 E
瘫 B
痪 E
在 S
床 S
的 S
母 B
亲 E
擦 B
脸 E

白 B
菜 E
清 B
炖 E
白 B
萝 M
卜 E
是 S
什 B
么 E
```



三、算法说明

本实验共使用了 3 种不同的分词算法，分别为机械分词、自分割分词和基于 LSTM 分词。

1. 机械分词

机械分词需要词典但不需要语料库，分为正向最大匹配算法、逆向最大匹配算法、正向最小匹配算法、逆向最小匹配算法。

正向最大匹配：用 MAXL 表示最大词长，按照从左到右的顺序，首先从汉字串中取长度为 MAXL 的子串查词典。若词典中存在这个词，则切分出该

子串，指针后移 MAXL 个汉字后继续切分，否则，子串长度减一，再与词典匹配。若长度为 2 的子串还不能在词典中查到，则取当前汉字为词（单字词），指针后移一个汉字继续匹配。

逆向最大匹配：与正向最大匹配的区别在于抽取顺序，从汉字串尾端开始抽取。

正向最小匹配：按照从左到右的顺序，首先从汉字串中取长度为 2 的子串查词典。若词典中存在这个词，则切分出该子串，指针后移 2 个汉字，否则，子串长度逐次加一继续匹配。若一直到长度为 MAXL 的子串仍无法匹配，则切分出当前汉字。

逆向最小匹配：与正向最小匹配的区别在于抽取顺序，从汉字串尾端开始抽取

2. 自分割分词

自分割分词既不需要词典也不需要语料库。

（1）对输入串 S 先以数字、标点符号为断点进行一次切分，剩余的中文文本则以断点为边界形成一组字串块。

（2）在字串块中找到出现两次以上的 n-gram（n>2）项，将得到的 n-gram 项和其出现的次数保存起来，在每个字串块，找到在字串块内出现的 n-gram 项作为候选 n-gram 项。

（3）按照长度和出现的次数递减的原则对候选 n-gram 逐步进行如下测试：如果被测试的 n-gram 的出现次数比由它形成的两个(n-1)-gram 项少，则不采纳该 n-gram 项，反之则不采纳(n-1)-gram 项，全部测试完成后则得到全部被采纳的 n-gram 项。

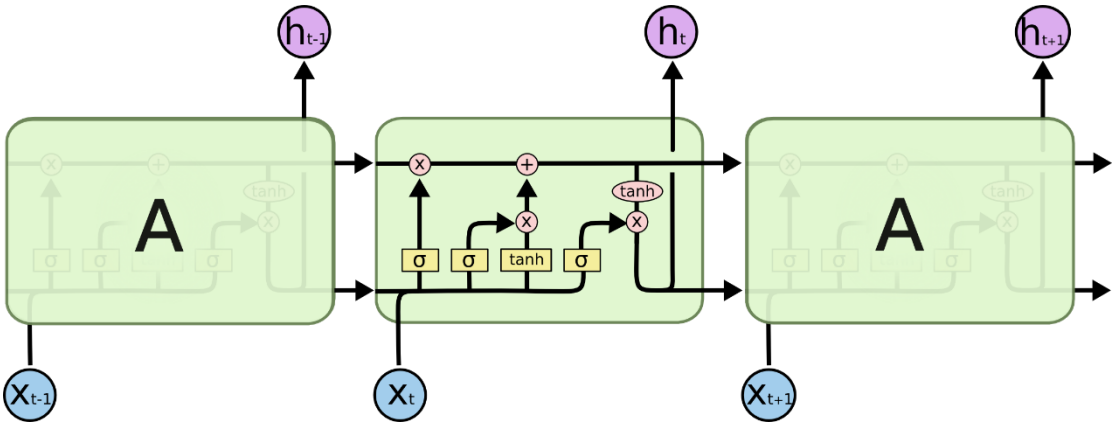
（4）按照步骤（3）得到的 n-gram 项对每个字串块进行划分。

（5）如果有在所有字串块中出现两次以上的划分词条，把它加入到一个初步词典中，按照长度和出现次数递减的规则类似获取被采纳的 n-gram 项的步骤处理整个初步词典，得到新的词条和新的切分。如果结果符合此条件则重复该步骤。

3. 基于 LSTM 分词

RNN 具有循环的网络结构，具备保持信息的能力。RNN 中的循环网络模块将信息从网络的上一层传输到下一层，网络模块的隐含层每个时刻的输出都依赖于以往时刻的信息。

LSTM 相比 RNN 作了改进，它增加了一种携带信息跨越多个时间步的方法，能够保存信息以便后面使用，从而防止较早期的信号在处理过程中逐渐消失。



神经网络不能够直接接收文本，需要将文本处理成数值张量作为神经网络的输入。本实验使用词嵌入的方法，先将字转为其在字典中的序号，然后再通过 Embedding 转为 128 维的向量。对于标签，使用 one-hot 编码方式，即

tag	one-hot 编码
S	$(1,0,0,0)^T$
B	$(0,1,0,0)^T$
M	$(0,0,1,0)^T$
E	$(0,0,0,1)^T$

经过 Embedding 层和 LSTM 层后，再通过全连接层完成各类别概率的计算，取概率值最大的类别为最终结果。

模型的结构如下：

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 128)         675072
_________________________________________________________________
lstm_1 (LSTM)                (None, 256)               394240
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 1028
=================================================================
Total params: 1,070,340
Trainable params: 1,070,340
Non-trainable params: 0
```

四、函数说明

本实验使用的语言为 Python，使用的库有 keras、numpy 和 matplotlib。

1. 机械分词

正向最小匹配：forwardMin(s,dic,maxL)

参数	类型	含义
s	string	需要分割的中文句子
dic	list	词典
maxL	int	最大词长

返回值	类型	含义
res_dic	list	分词结果

正向最大匹配：forwardMax(s,dic,maxL)

参数	类型	含义
s	string	需要分割的中文句子
dic	list	词典
maxL	int	最大词长

返回值	类型	含义
res_dic	list	分词结果

逆向最小匹配：backwardMin(s,dic,maxL)

参数	类型	含义
s	string	需要分割的中文句子
dic	list	词典
maxL	int	最大词长

返回值	类型	含义
res_dic	list	分词结果

逆向最大匹配：backwardMax(s,dic,maxL)

参数	类型	含义
s	string	需要分割的中文句子
dic	list	词典
maxL	int	最大词长

返回值	类型	含义
res_dic	list	分词结果

2. 自分割分词

计算每个词出现的次数：buildCount(s,n)

参数	类型	含义
s	string	中文句子
n	int	最大词长

返回值	类型	含义
count	dict	从词到次数的映射

自分割分词：autoSplit(s,n,split_to_single=False)

参数	类型	含义
s	string	需要分割的中文句子
n	int	最大词长
split_to_single	bool	对没有采纳的长度大于 2 的词是否切分到单字，默认 False

返回值	类型	含义
s_list	list	分词结果

3. 基于 LSTM 分词

将字转为其在字典中的序号：toIndex(s,dic)

参数	类型	含义
s	string	中文句子
dic	dict	字到序号的映射

返回值	类型	含义
x	list	对应的序号列表

将序号转为对应的中文句子：toCharacter(x,character_list)

参数	类型	含义
x	list	序号列表
character_list	list	序号到字的映射

返回值	类型	含义
res	string	对应的中文句子

将预测结果转为对应的标签：toTag(preds,length)

参数	类型	含义

preds	list	预测列表
length	int	预测的有效长度

返回值	类型	含义
res	list	对应的预测标签列表

根据预测结果进行分词：toResult(s,preds)

参数	类型	含义
s	string	中文句子
preds	list	预测列表

返回值	类型	含义
res	list	分词结果

读入训练数据：readData(file_path)

参数	类型	含义
file_path	string	文件路径

返回值	类型	含义
train_X	list	训练集特征列表
train_y	list	训练集标签列表
maxL	int	中文句子的最大长度
dic	dict	字到序号的映射
character_list	list	序号到字
data	list	中文句子列表
labels	list	one-hot 标签列表

读入测试数据：readTestData(file_path)

参数	类型	含义

file_path	string	文件路径

返回值	类型	含义
test_X	list	测试集特征列表
test_y	list	测试集标签列表
data	list	中文句子列表
labels	list	one-hot 标签列表

计算 Precision：P(y_pred,y)

参数	类型	含义
y_pred	list	预测标签列表
y	list	真实标签列表

返回值	类型	含义
res	list	每个类别的 Precision 值

计算 Precision：P(y_pred,y)

参数	类型	含义
y_pred	list	预测标签列表
y	list	真实标签列表

返回值	类型	含义
res	list	每个类别的 Precision 值

计算 Recall：R(y_pred,y)

参数	类型	含义
y_pred	list	预测标签列表
y	list	真实标签列表

返回值	类型	含义
res	list	每个类别的 Recall 值

计算 F 值：F(P,R,alpha=1)

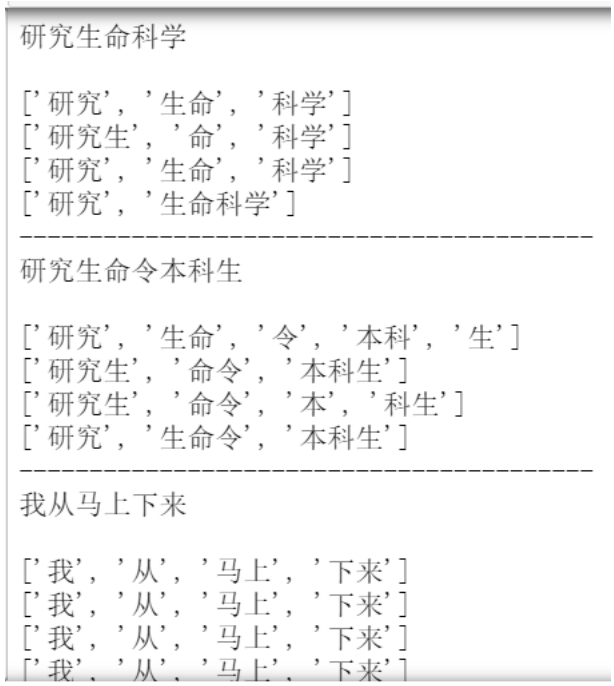
参数	类型	含义
P	list	每个类别的 Precision 值
R	list	每个类别的 Recall 值
alpha	float	P 和 R 的调和权重，默认 1.0

返回值	类型	含义
res	list	每个类别的 F 值

五、结果分析与改进

1. 机械分词

分别使用正向最小匹配、正向最大匹配、逆向最小匹配、逆向最大匹配对 test.txt 中的句子进行分词，部分结果如下。



```
我马上下来

['我', '马上', '下来']
['我', '马上', '下来']
['我', '马上', '下来']
['我', '马上', '下来']
----------------------------------------
北京大学生喝进口红酒

['北京', '大学', '生', '喝进', '口红', '酒']
['北京大学', '生', '喝进', '口红', '酒']
['北京', '大', '学生', '喝', '进口', '红酒']
['北京', '大学生', '喝', '进口', '红酒']
----------------------------------------
在北京大学生活区喝进口红酒

['在', '北京', '大学', '生活', '区', '喝进', '口红', '酒']
['在', '北京大学', '生活区', '喝进', '口红', '酒']
['在', '北京', '大学', '生活区', '喝', '进口', '红酒']
['在', '北京大学', '生活区', '喝', '进口', '红酒']
```

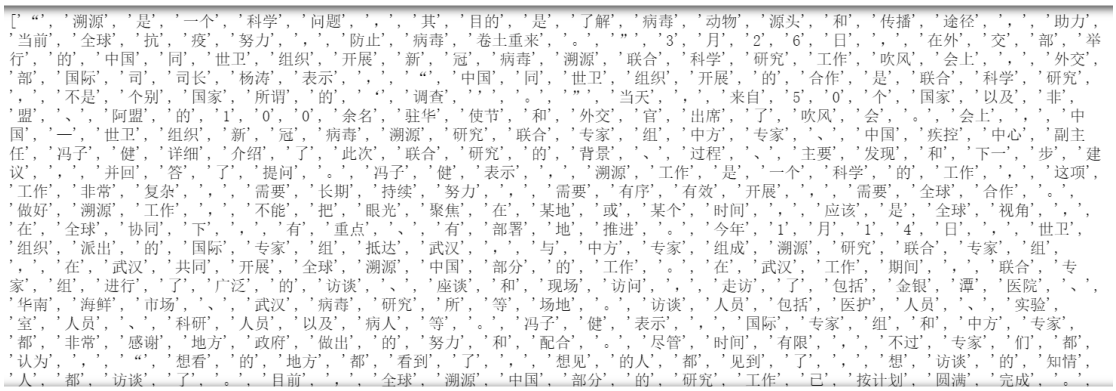
```
从小学电脑

['从小', '学', '电脑']
['从小', '学', '电脑']
['从', '小学', '电脑']
['从', '小学', '电脑']
----------------------------------------
从小学毕业

['从小', '学', '毕业']
['从小', '学', '毕业']
['从', '小学', '毕业']
['从', '小学', '毕业']
----------------------------------------
美军中将竟公然说

['美军', '中将', '竟', '公然', '说']
['美军', '中将', '竟', '公然', '说']
['美军', '中将', '竟', '公然', '说']
['美军', '中将', '竟', '公然', '说']
```

对 test2.txt 中的文章进行分词，部分结果如下。



机械分词的优点是效率高，不需要语料库，缺点是正确率低。

对于未登录词，可以人工将它们保存入词典，即可正确分词，如下图。

```
if '吴其平' in dic:
    dic.remove('吴其平')
s='吴其平常常写代码'
print(forwardMin(s,dic,maxL))
print(forwardMax(s,dic,maxL))
print(backwardMin(s,dic,maxL))
print(backwardMax(s,dic,maxL))
```

```
['吴', '其', '平常', '常', '写', '代码']
['吴', '其', '平常', '常', '写', '代码']
['吴', '其', '平', '常常', '写', '代码']
['吴', '其', '平', '常常', '写', '代码']
```

```
dic.append('吴其平')
s='吴其平常常写代码'
print(forwardMin(s,dic,maxL))
print(forwardMax(s,dic,maxL))
print(backwardMin(s,dic,maxL))
print(backwardMax(s,dic,maxL))
```

```
['吴其平', '常常', '写', '代码']
['吴其平', '常常', '写', '代码']
['吴其平', '常常', '写', '代码']
['吴其平', '常常', '写', '代码']
```

对于有歧义句子的部分测试结果如下图。

```
s='结婚的和尚未结婚的'
print(forwardMin(s,dic,maxL))
print(forwardMax(s,dic,maxL))
print(backwardMin(s,dic,maxL))
print(backwardMax(s,dic,maxL))
```

```
['结婚', '的', '和尚', '未结', '婚', '的']
['结婚', '的', '和尚', '未结', '婚', '的']
['结婚', '的', '和', '尚未', '结婚', '的']
['结婚', '的', '和', '尚未', '结婚', '的']
```

```
s='原子结合成分子'
print(forwardMin(s,dic,maxL))
print(forwardMax(s,dic,maxL))
print(backwardMin(s,dic,maxL))
print(backwardMax(s,dic,maxL))
```

```
['原子', '结合', '成分', '子']
['原子', '结合', '成分', '子']
['原子', '结', '合成', '分子']
['原子', '结', '合成', '分子']
```

对于歧义没有太好的改进方法，这个问题是机械分词算法本身的不足。

2. 自分割分词

对 test.txt 中的句子进行分词，部分结果如下。

```
这样的人才能经受住考验

['这样的人才能经受住考验']
----------------------------------------
他俩儿谈恋爱是从头年元月开始的

['他俩儿谈恋爱是从头年元月开始的']
----------------------------------------
在这些企业中国有企业有十个

['在这些', '企业', '中国有', '企业', '有十个']
----------------------------------------
结婚的和尚未结婚的

['结婚的', '和尚未', '结婚的']
----------------------------------------
```

对 test2.txt 中的文章进行分词，结果如下。

['“', '溯源', '是一个科学', '问题', '，', '其目的是了解', '病毒', '动物源头和传播途径', '，', '助力当前', '全球抗疫', '努力', '，', '防止', '病毒', '卷土重来', '。', '”', '，', '3', '月', '26', '日', '，', '在', '外交部', '举', '行的', '中国同世卫组织开展', '新冠病毒', '溯源', '联合科学研究', '工作', '吹风会', '上', '，', '外交部', '国际', '司司长杨涛表示', '，', '“', '中国同世卫组织开展', '的', '合作', '是', '联合科学研究', '，', '不是个别', '国家', '所谓的“调查”', '。', '”', '当天', '，', '来自', '50', '个', '国家', '以及', '非盟、阿盟的', '100', '余名驻华使节和外交官出席了', '吹风会', '。', '会上', '，', '中国', '一', '世卫组织', '新冠病毒', '溯源研究联合专家组', '中方专家', '、', '中国', '疾控中心副主任冯子健详细介绍了此次', '联合', '研究', '的背景、过程、主要发现和下一步建议', '，', '并回答了提问', '。', '冯子健表示', '，', '溯源工作', '是一个科学', '的', '工作', '，', '这项', '工作', '非常', '复杂', '，', '需要', '长期持续', '努力', '，', '需要', '有序有效', '开展', '，', '需要', '全球', '合作', '。', '做好', '溯源工作', '，', '不能把眼光聚焦在某地或某个', '时间', '，', '应该是', '全球', '视角', '，', '在', '全球', '协同下', '，', '有重点、有部署地推进', '。', '今年', '1', '月', '14', '日', '，', '世卫组织', '派', '出的', '国际专家组', '抵达', '武汉', '，', '与', '中方专家', '组成', '溯源研究联合专家组', '，', '在武汉', '共同', '开展', '全球溯源中国部分的', '工作', '。', '在武汉', '工作', '期间', '，', '联合', '专家', '组', '进行', '了广泛的', '访谈', '、座谈和现场访问', '，', '走访了', '包括', '金银潭医院、华南海鲜市场、', '武汉', '病毒', '研究', '所等场地', '。', '访谈', '人员', '包括', '医护', '人员、', '实验室', '人员、', '科研', '人员', '以及', '病人等', '。', '冯子健表示', '，', '国际专家组', '和', '中方专家', '都', '非常', '感谢', '地方', '政府做', '出的', '努力', '和配合', '。', '尽管', '时间', '有限', '，', '不过', '专家', '们都认为', '，', '“', '想看的', '地方', '都看', '到了', '，', '想见的', '人都', '见', '到了', '，', '想', '访谈', '的知情', '人都', '访谈', '了', '。', '目前', '，', '全球溯源中国部分的', '研究', '工作', '已按计划圆满', '完成', '。', '”', '“', '中国的', '抗疫', '工作', '是在阳光下', '进行', '的', '。', '”', '杨涛说', '，', '中国', '积极', '克服困难与', '世卫组织', '专家', '联合', '进行', '病毒溯源', '研究', '，', '体现了', '中国的', '开放、透明', '。', '中国', '在自身', '疫情防控', '面临较大压力的情况下', '，', '率先', '同', '世卫组织', '开展', '溯源', '合作', '，', '是为', '全球抗疫', '事业作', '出的', '积极', '贡献', '。', '目前', '，', '全球', '还处', '在新冠', '肺炎', '疫情', '大流行', '期间', '，', '各国首要任务依然是控制', '疫情', '。', '杨涛说', '：', '“', '中国', '率先', '宣布', '，', '在新冠', '疫苗研发', '完成', '并投入使用后', '，', '将作为', '全球', '公共产品', '。', '中方愿继续本着团结、', '合作', '、科学、公正的原则', '，', '同', '国际', '社会', '开展', '疫情防控', '合作', '，', '践行人类卫生健康', '共同', '体理念', '。', '”']

自分割分词的优点是效率高，不需要语料库且不需要词典，缺点是对于短句子效果不佳。如果句子中没有重复出现的词，那么自分割就无法切分出任何词。所以自分割对长文章分词的正确率会远高于对短句分词的正确率。

考虑到其特点，可以对自分割分词的算法稍作改进。原始的算法中要求出现次数大于 2 的 n-gram 才被采纳，可以改为不小于 2 的 n-gram 就采纳。此外，由于中文的单字词和二字词最多，因此可以将最后未切分的较长的字串切分到单字，以提高分词的正确率。

3. 基于 LSTM 分词

由于算力有限，本实验仅在训练集上进行了一轮训练。使用 binary_crossentropy 作为 loss 函数，RMSprop 作为优化器。训练 loss 为 0.3522，accuracy 为 0.8383。

部分训练集上的分词结果如下图。

['去年', '董云', '以539分', '考', '上', '了', '[', '广', '东', '工业', '大学', '，', '“', '学', '业', '还', '是', '妈', '妈', '？', '”',
'曾', '让', '董云', '难', '以', '抉择', '，', '最终', '她', '选择', '了', '带着', '妈', '妈', '上', '大学', '，', '举家', '迁', '来', '广州', '，', '边', '读', '书', '，', '边', '照', '顾妈', '妈', '。']
['“', '董云', '，', '热', '水袋', '充好', '了', '吗', '？', '”', '妈', '妈', '在', '房', '间', '擦着', '眼泪', '说', '，', '桌', '上',
'的', '电', '热', '水袋', '刚', '充', '了', '[5分', '钟', '电', '，', '董云', '正', '抽', '空', '在', '客', '厅', '啃', '着', '隔夜', '的',
'凉', '包子', '。']
['摸', '摸', '热', '水袋', '温度', '差', '不多', '了', '，', '董云', '小心', '翼', '翼', '地', '放', '到', '妈', '妈', '脖子', '下', '，',
'此时', '已', '是', '8点', '03分', '，', '她', '背起', '书', '包', '飞', '奔着', '下楼', '了', '。']
['董云', '妈', '妈', '告', '诉', '记者', '，', '每逢', '天', '气', '变化', '，', '她', '就', '会', '难', '受', '，', '“', '手', '一', '直
抖', '，', '还', '会', '流', '眼泪', '，', '这时', '我', '要', '用', '热', '水袋', '垫', '着', '脖子', '。']
['”', '妈', '妈', '略', '带愧', '疚', '地', '说', '：', '“', '董云', '每天', '早', '上', '8点', '15分', '上', '课', '，', '但', '时',
'常', '要', '帮', '我', '收拾', '到', '[8点', '才', '能', '出', '门', '。']

在 test.data 上进行测试，得到评估指标如下。

	S	B	M	E
Precision	0.8778	0.6219	0.7611	0.6219
Recall	0.7103	0.7154	0.3169	0.6984
F1	0.7590	0.6517	0.3027	0.6439

部分测试集上的分词结果如下图。

```
8
['温馨', '提示', '：', '具体', '的', '克', '数', '去', '买', '的', '时', '候', '请', '人', '家', '帮', '你', '称', '下', '，', '看', '看',
'[100克', '大', '概有', '多', '少', '，', '，', '[', '回来', '就', '好', '办', '了', '，', '有', '点', '误', '差', '不', '要', '紧', '的', '。']
----------------------------------------

9
['做', '法', '：', '将', '所有', '材', '料', '共放', '在', '锅', '中', '，', '，', '加', '[1000', '毫升', '冷', '水', '，', '，', '煎到', '豆烂', '即']
----------------------------------------

10
['Tips', '：', '红', '豆', '和', '绿', '豆', '煮', '之前', '最好', '用', '冷', '水', '泡', '一个', '小时', '，', '，', '会', '比', '较容', '易',
'煮烂', '的', '。']
----------------------------------------

11
['《', '[', '新天', '龙', '八', '部', '》', '新旧', '版', '雷人', '剧情', '造', '型', '钟', '汉', '良', '大', '引', '争议', '段', '誉',
'娘', '气']
----------------------------------------

12
['【', '[天', '龙', '八', '部', '滑雪', '板', '雷人', '[', '新天', '龙', '八', '部频', '现', '重口', '味', '木', '婉', '清蕾', '丝', '内裤',
'蒙纱', '】', '由', '钟', '汉', '良', '、', '金', '基', '范', '等', '主演', '的', '电视', '剧', '《', '[', '新天', '龙', '八', '部', '》',
'昨晚', '首播', '。']
----------------------------------------

13
['作', '为', '已', '是', '[', '第五', '度', '翻', '拍', '的', '《', '[天', '龙', '八', '部', '》', '，', '，', '新', '版', '无疑', '是', '颠', '覆
性', '最', '大', '的', '，', '，', '也', '注', '定', '会', '是', '槽', '点', '最多', '的', '一', '部', '剧', '。']
----------------------------------------
```

基于 LSTM 分词的优点是正确率较高，缺点是效率低，需要语料库，需要大量数据来训练模型。

基于 LSTM 分词的总体正确率对于机械分词和自分割分词要高，但是依然存在一些问题，如预测的结果存在 EE、BB、EM 等的情况，原因是 LSTM 能学习到输入的字之间的依赖，但无法学习到输出的标签之间的关系，因此需要使用 CRF 来对输出标签的特征进行约束。通常可以在 LSTM 后添加一层 CRF 再连接全连接层，以进一步提高学习能力。