1. 利用已经学习过的理论方法和北京大学标注的《人民日报》分词和词性标注语料，设计实现至少两种不同的汉语词语自动切分方法，进行性能测试和分析。然后利用不同类型的网络文本测试分词系统，
2. 对比分析分词方法和不同测试样本的性能变化。在得到的分词结果的基础上，实现子词压缩。

需要提交实现代码以及实验报告，不需要提交数据集。

文件结构

`max_matching` 目录下是基于匹配的分词方法代码，`bilstm_crf` 目录下是基于神经网络的分词方法代码。

程序运行

基于匹配的方法

```
cd max_matching
python fmm_main.py  #训练集
python fmm_test.py  #测试集

python bmm_main.py  #训练集
python bmm_test.py  #测试集

python bimm_main.py  #训练集
python bimm_test.py  #测试集
```

基于神经网络的方法

```
cd bilstm_crf/data
python create_data.py  #构建数据
cd ..
python train.py  #训练模型
python inference.py  #推理新的句子
```

数据准备

代码分别位于两种算法目录下的 `data` 目录中：

词典构建

```
def load_vocab():

    with open('ChineseCorpus199801.txt', 'r', encoding='gb2312', errors='ignore') as f:
        corpus = f.readlines()

    # ['19980131-04-004-004/m', '这/r', '就/d', '是/v', '[江/j', '峡/j', '大道/n]ns', '。/w'] 64

    vocab = []
    for c in corpus:
        print("="*10)
        print(c)
        word_list = c.split(' ')
        word_list = [w for w in word_list if w != '\n']
        word_list = word_list[1:]
        word_list = [w.split('/')[0] for w in word_list]
        word_list = [w.split('[')[1] if '[' in w else w for w in word_list ]
        word_list = [w for w in word_list if len(w)]
        if not len(word_list):
            continue
        print(word_list)
        vocab.extend(word_list)
    vocab = list(set(vocab))

    return vocab
```

语句构建

```
def load_gt_sentences():

    with open('ChineseCorpus199801.txt', 'r', encoding='gb2312', errors='ignore') as f:
        corpus = f.readlines()

    gt_sentences = []
    for c in corpus:
        word_list = c.split(' ')
```

```
        word_list = [w for w in word_list if len(w)]
        word_list = [w for w in word_list if w != '\n']
        word_list = word_list[1:]
        word_list = [w.split('/')[0] for w in word_list]
        word_list = [w.split('[')[1] if '[' in w else w for w in word_list ]
        if not len(word_list):
            continue
        gt_sentences.append(''.join(word_list))

    return gt_sentences
```

分词标签构建

```
def load_gt_partition():

    with open('ChineseCorpus199801.txt', 'r', encoding='gb2312',
errors='ignore') as f:
        corpus = f.readlines()

    gt_partition = []
    for c in corpus:
        word_list = c.split(' ')
        word_list = [w for w in word_list if w != '\n']
        word_list = word_list[1:]
        word_list = [w.split('/')[0] for w in word_list]
        word_list = [w.split('[')[1] if '[' in w else w for w in word_list ]
        word_list = [w for w in word_list if len(w)]
        if not len(word_list):
            continue
        gt_partition.append(word_list)

    return gt_partition
```

神经网络标签构建

```
def prepare_nn_gt():
    gt_sentences = load_gt_sentences()
    gt_partitions = load_gt_partition()
    assert len(gt_sentences) == len(gt_partitions)
    gt_labels = []
    for partition in gt_partitions:
        gt_label = []
        for word in partition:
            n_char = len(word)
            if n_char == 1:
                gt_label.append('S')
            elif n_char == 2:
                gt_label.extend(['B', 'E'])
            else:
                gt_label.extend(['B'] + ['M'] * (n_char - 2) + ['E'])
        gt_labels.append(gt_label)
    return gt_sentences, gt_labels


```

```python
def get_list(input_str: str) -> list:
    """Transform single word to tag sequence.

    Args:
        input_str (str): Single word.

    Returns:
        list: The tag sequence of word `input_str`.
    """
      # tag sequence to be returned

    if len(input_str) == 1:  # "我"
        output_str = [tag2id['S']]
    elif len(input_str) == 2:  # "喜欢"
        output_str = [tag2id['B'], tag2id['E']]
    else:  # "太平洋"
        output_str = []
        M_num = len(input_str) - 2
        M_list = [tag2id['M']] * M_num
        output_str += [tag2id['B']]
        output_str += M_list
        output_str += [tag2id['E']]

    return output_str
```

汉语分词方法

最大匹配法

这个方法是一种有词典的分词方法，也是一种基于规则的分词方法。

正向最大匹配算法

伪代码：（根据课件第八章第 30 页转化而来）

```python
def Partition(S, vocab):
    """
    Args:
        S (str): Input string with Chinese characters.
        vocab (list): The vocabulary that store the words.
    """
    results = []

    i = 0
    while True:
        n = len(S) - i
        if n == 1:
            return results
        m = get_max_len_of_words(vocab)
        if n < m:
            m = n

        w = S[i:i+m]
        while w not in vocab:
```

```
            if len(w) > 1:
                w = w[:-1]
            elif len(w) == 1:
                vocab.append(w)
                break
        results.append(w)
        i = i + len(w)
        if i == len(S):
            return results
```

其实上面的代码再加上一个从词表中获取最长单词的长度的函数就构成了正向最大匹配算法的代码。完整代码位于 `fmm.py` 中。

代码示例结果为：

```
S = "他是研究生物化学的一位科学家"
vocab = ["他", "是", "研究", "研究生", "生物", "化学", "的", "一位", "科学", "科学家"]
print(partition(S, vocab))

# Output
['他', '是', '研究生', '物', '化学', '的', '一位', '科学家']
```

从这个结果中可以看出：

- 优点：
  - 程序简单易行，开发周期短；
  - 仅需要很少的语言资源（词表），不需要任何词法、句法、语义资源；
- 弱点：
  - 歧义消解的能力差；
  - 切分正确率不高，一般在95%左右；

逆向最大匹配算法

逆向最大匹配算法的算法思想类似于正向最大匹配算法，算法的实现代码位于 `bmm.py` 中。

代码示例结果为：

```
S = "他是研究生物化学的一位科学家"
vocab = ["他", "是", "研究", "研究生", "生物", "化学", "的", "一位", "科学", "科学家"]
print(partition(S, vocab))

# Output
['他', '是', '研究', '生物', '化学', '的', '一位', '科学家']
```

双向最大匹配算法

双向最大匹配算法的原理就是将正向最大匹配算法和逆向最大匹配算法进行比较，从而选择正确的分词方式。

比较原则与步骤：

比较两种匹配算法的结果:

- 如果分词数量结果不同，则选择数量较少的那个
- 如果分词数量结果相同
    - 分词结果相同，返回任意一个
    - 分词结果不同，返回单字数较少的一个
    - 若单字数也相同，任意返回一个

双向最大匹配算法的代码位于 `bidirectionalmm.py` 中。

代码示例结果为:

```
S = "他是研究生物化学的一位科学家"
vocab = ["他", "是", "研究", "研究生", "生物", "化学", "的", "一位", "科学", "科学家"]
print(partition(S, vocab))

# Output
['他', '是', '研究', '生物', '化学', '的', '一位', '科学家']
```

结果比较

```
+--------+-----------+--------+-------+------------+-----------+--------+
| Method | Precision | Recall |   F1  | OOV_Recall | IV_Recall |  Time  |
+--------+-----------+--------+-------+------------+-----------+--------+
|  F-MM  |   97.78   | 97.05  | 97.41 |    0.0     |   97.05   | 23945s |
|  B-MM  |   97.98   | 97.24  | 97.61 |    0.0     |   97.24   | 24092s |
| Bi-MM  |   98.11   | 97.32  | 97.71 |    0.0     |   97.32   | 46618s |
+--------+-----------+--------+-------+------------+-----------+--------+

+--------+---------+-------+-------+---------+---------+-----+-------+
| Method |    TP   |   FN  |   FP  |    IV   |   IV_R  | OOV | OOV_R |
+--------+---------+-------+-------+---------+---------+-----+-------+
|  F-MM  | 1088304 | 33138 | 24709 | 1121442 | 1088304 |  0  |   0   |
|  B-MM  | 1090496 | 30946 | 22490 | 1121442 | 1090496 |  0  |   0   |
| Bi-MM  | 1091406 | 30036 | 21059 | 1121442 | 1091406 |  0  |   0   |
+--------+---------+-------+-------+---------+---------+-----+-------+
```

基于神经网络的分词方法

基于神经网络的分词方法就是把分词任务看作序列标注任务，它的输入输出均为序列，即输入为 n 个字的语句，输出为 m 个词语，是 $n:m$ 的对应关系。

首先对句子中的每个字都转换为分布式向量表示，即嵌入。然后把这些向量送入循环神经网络 LSTM 或 RNN。最后取每一个 LSTM 的输出再经过 CRF 或者 Softmax 算法得到每一个字的位置类别：B、M、E、S，分别表示词的开始、词的中间、词的结束位置和汉字单独成词。

> 例如：“门把手/损坏/了”这个分词对应的标签就是“[B, M, E, B, E, S]”。

在基于神经网络的分词方法中，我使用了 LSTM + CRF 模型。评价模型性能的方式就是记录模型的预测输出与真值标签之间的重合情况，求得预测的准确率、召回率等指标。

基于 LSTM + CRF 模型的结果指标如下:

```
precision:  0.9454079939735807
recall:  0.9369134376110914
fscore:  0.9411415486387923
```

LSTM 的输出经过一个全连接层，这一层后直接接上 Softmax 模型就变为了没有结合 CRF 的深度学习方法，但 Bi-LSTM 虽然学习到了上下文的信息，不过在预测过程中输出序列之间并没有互相的影响，仅仅是挑出每一个字最大概率的 label，通过引入 CRF 加入了对 label 之间顺序性的考虑，因此效果更好。

> 例如，LSTM 的预测会是 ['B', 'B', 'S', 'E', 'M'] 这种，但实际的标签一定不会出现 ['E', 'B', 'M'] 这种明显颠倒的顺序。

测试结果

正向最大匹配算法

```
================================================================================
========================
这个仅仅是一个小测试
['这个', '仅仅', '是', '一个', '小', '测试']
================================================================================
========================
这仅仅是一个小测试
['这', '仅仅', '是', '一个', '小', '测试']
================================================================================
========================
李小福是创新办主任也是云计算方面的专家
['李', '小', '福', '是', '创新', '办', '主任', '也', '是', '云', '计算', '方面',
'的', '专家']
================================================================================
========================
实现祖国的完全统一是海内外全体中国人的共同心愿
['实现', '祖国', '的', '完全', '统一', '是', '海内外', '全体', '中国', '人', '的',
'共同', '心愿']
================================================================================
========================
南京市长江大桥
['南京市', '长江', '大桥']
================================================================================
========================
中文分词在中文信息处理中是最最基础的，无论机器翻译亦或信息检索还是其他相关应用，如果涉及中文，都离不开中文分词，因此中文分词具有极高的地位。
['中文', '分', '词', '在', '中文', '信息', '处理', '中', '是', '最最', '基础',
'的', '，', '无论', '机器', '翻译', '亦', '或', '信息', '检索', '还是', '其他', '相
关', '应用', '，', '如果', '涉及', '中文', '，', '都', '离不开', '中文', '分',
'词', '，', '因此', '中文', '分', '词', '具有', '极', '高', '的', '地位', '。']
================================================================================
========================
蔡英文和特朗普通话
['蔡', '英文', '和', '特', '朗', '普通话']
================================================================================
========================
研究生命的起源
['研究生', '命', '的', '起源']
```

```
==========
他从马上下来
['他', '从', '马上', '下来']
==========
老人家身体不错
['老人家', '身体', '不错']
==========
老人家中很干净
['老人家', '中', '很', '干净']
==========
这的确定不下来
['这', '的确', '定', '不', '下来']
==========
乒乓球拍卖完了
['乒乓球', '拍卖', '完了']
==========
香港中文大学将来合肥一中进行招生宣传今年在皖招8人万家热线安徽第一门户
['香港', '中文', '大学', '将来', '合肥', '一中', '进行', '招生', '宣传', '今年', '在', '皖', '招', '8', '人', '万', '家', '热线', '安徽', '第一', '门户']
==========
在伦敦奥运会上将可能有一位沙特阿拉伯的女子
['在', '伦敦', '奥运会', '上将', '可能', '有', '一', '位', '沙特阿拉伯', '的', '女子']
==========
美军中将竟公然说
['美军', '中将', '竟', '公然', '说']
==========
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']
==========
将信息技术应用于教学实践
['将', '信息', '技术', '应用', '于', '教学', '实践']
==========
天真的你
['天真', '的', '你']
==========
我们中出了一个叛徒
['我们', '中', '出', '了', '一个', '叛徒']
```

逆向最大匹配算法

```
==========================================================================================
这个仅仅是一个小测试
['这个', '仅仅', '是', '一个', '小', '测试']
==========================================================================================
这仅仅是一个小测试
['这', '仅仅', '是', '一个', '小', '测试']
==========================================================================================
李小福是创新办主任也是云计算方面的专家
['李', '小', '福', '是', '创新', '办', '主任', '也', '是', '云', '计算', '方', '面的', '专家']
==========================================================================================
实现祖国的完全统一是海内外全体中国人的共同心愿
['实现', '祖国', '的', '完全', '统一', '是', '海内外', '全体', '中', '国人', '的', '共同', '心愿']
==========================================================================================
南京市长江大桥
['南京市', '长江', '大桥']
==========================================================================================
中文分词在中文信息处理中是最最基础的，无论机器翻译亦或信息检索还是其他相关应用，如果涉及中文，都离不开中文分词，因此中文分词具有极高的地位。
['中文', '分', '词', '在', '中文', '信息', '处理', '中', '是', '最最', '基础', '的', '，', '无论', '机器', '翻译', '亦', '或', '信息', '检索', '还是', '其他', '相关', '应用', '，', '如果', '涉及', '中文', '，', '都', '离不开', '中文', '分', '词', '，', '因此', '中文', '分', '词', '具有', '极', '高', '的', '地位', '。']
==========================================================================================
蔡英文和特朗普通话
['蔡', '英文', '和', '特', '朗', '普通话']
==========================================================================================
研究生命的起源
['研究', '生命', '的', '起源']
==========================================================================================
他从马上下来
['他', '从', '马上', '下来']
==========================================================================================
老人家身体不错
['老人家', '身体', '不错']
==========================================================================================
老人家中很干净
['老人', '家中', '很', '干净']
==========================================================================================
这的确定不下来
```

```
['这', '的', '确定', '不', '下来']
==========
乒乓球拍卖完了
['乒乓球', '拍卖', '完了']
==========
香港中文大学将来合肥一中进行招生宣传今年在皖招8人万家热线安徽第一门户
['香港', '中文', '大学', '将来', '合肥', '一中', '进行', '招生', '宣传', '今年', '在', '皖', '招', '8', '人', '万', '家', '热线', '安徽', '第一', '门户']
==========
在伦敦奥运会上将可能有一位沙特阿拉伯的女子
['在', '伦敦', '奥运会', '上将', '可能', '有', '一', '位', '沙特阿拉伯', '的', '女子']
==========
美军中将竟公然说
['美军', '中将', '竟', '公然', '说']
==========
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']
==========
将信息技术应用于教学实践
['将', '信息', '技术', '应', '用于', '教学', '实践']
==========
天真的你
['天', '真的', '你']
==========
我们中出了一个叛徒
['我们', '中', '出', '了', '一个', '叛徒']
```

双向最大匹配算法

```
==========
这个仅仅是一个小测试
['这个', '仅仅', '是', '一个', '小', '测试']
==========
这仅仅是一个小测试
['这', '仅仅', '是', '一个', '小', '测试']
==========
李小福是创新办主任也是云计算方面的专家
['李', '小', '福', '是', '创新', '办', '主任', '也', '是', '云', '计算', '方', '面的', '专家']
```

```
================================================================================
========================
实现祖国的完全统一是海内外全体中国人的共同心愿
['实现', '祖国', '的', '完全', '统一', '是', '海内外', '全体', '中', '国人', '的',
'共同', '心愿']
================================================================================
========================
南京市长江大桥
['南京市', '长江', '大桥']
================================================================================
========================
中文分词在中文信息处理中是最最基础的，无论机器翻译亦或信息检索还是其他相关应用，如果涉及中
文，都离不开中文分词，因此中文分词具有极高的地位。
['中文', '分', '词', '在', '中文', '信息', '处理', '中', '是', '最最', '基础',
'的', '，', '无论', '机器', '翻译', '亦', '或', '信息', '检索', '还是', '其他', '相
关', '应用', '，', '如果', '涉及', '中文', '，', '都', '离不开', '中文', '分',
'词', '，', '因此', '中文', '分', '词', '具有', '极', '高', '的', '地位', '。']
================================================================================
========================
蔡英文和特朗普通话
['蔡', '英文', '和', '特', '朗', '普通话']
================================================================================
========================
研究生命的起源
['研究', '生命', '的', '起源']
================================================================================
========================
他从马上下来
['他', '从', '马上', '下来']
================================================================================
========================
老人家身体不错
['老人家', '身体', '不错']
================================================================================
========================
老人家中很干净
['老人', '家中', '很', '干净']
================================================================================
========================
这的确定不下来
['这', '的', '确定', '不', '下来']
================================================================================
========================
乒乓球拍卖完了
['乒乓球', '拍卖', '完了']
================================================================================
========================
香港中文大学将来合肥一中进行招生宣传今年在皖招8人万家热线安徽第一门户
['香港', '中文', '大学', '将来', '合肥', '一中', '进行', '招生', '宣传', '今年',
'在', '皖', '招', '8', '人', '万', '家', '热线', '安徽', '第一', '门户']
================================================================================
========================
在伦敦奥运会上将可能有一位沙特阿拉伯的女子
['在', '伦敦', '奥运会', '上将', '可能', '有', '一', '位', '沙特阿拉伯', '的', '女
子']
```

```
==========
美军中将竟公然说
['美军', '中将', '竟', '公然', '说']
==========
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']
==========
将信息技术应用于教学实践
['将', '信息', '技术', '应', '用于', '教学', '实践']
==========
天真的你
['天', '真的', '你']
==========
我们中出了一个叛徒
['我们', '中', '出', '了', '一个', '叛徒']
```

LSTM + CRF 算法

```
==========
这个仅仅是一个小测试
这个/仅仅/是/一个/小/测试
==========
这仅仅是一个小测试
这/仅仅/是/一个/小/测试
==========
李小福是创新办主任也是云计算方面的专家
李/小/福是/创新/办/主任/也/是/云计/算方面/的/专家
==========
实现祖国的完全统一是海内外全体中国人的共同心愿
实现/祖国/的/完全/统一/是/海内外/全体/中国/人/的/共同/心愿
==========
南京市长江大桥
南京市/长江/大桥
==========
中文分词在中文信息处理中是最最基础的，无论机器翻译亦或信息检索还是其他相关应用，如果涉及中文，都离不开中文分词，因此中文分词具 有极高的地位。
中文分词/在/中文信息/处理/中/是/最/最/基础/的/，/无论机器/翻译/亦/或/信息/检索/还是/其他/相关/应用/，/如果/涉及/中文/，/都/离/ 不/开中/文分词/，/因此/中文分词/具有/极高/的/地位/。
```

```
==========
蔡英文和特朗普通话
蔡/英文/和/特朗/普通话
==========
研究生命的起源
研究/生命/的/起源
==========
他从马上下来
他/从/马上/下来
==========
老人家身体不错
老/人家/身体/不错
==========
老人家中很干净
老/人家/中/很/干净
==========
这的确定不下来
这/的/确定/不/下来
==========
乒乓球拍卖完了
乒乓球/拍卖/完/了
==========
在伦敦奥运会上将可能有一位沙特阿拉伯的女子
在/伦敦/奥运会/上将/可能/有/一/位/沙特/阿拉伯/的/女子
==========
美军中将竟公然说
美军/中将竟/公然/说
==========
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
==========
将信息技术应用于教学实践
将/信息/技术/应用/于/教学/实践
==========
天真的你
天真/的/你
==========
我们中出了一个叛徒
我们/中/出/了/一个/叛徒
```

结果分析

基于匹配的分词方法在一些句子上的分词结果相比 LSTM + CRF 的方法会更合理，分析原因可能是基于词典的分词方法划分的依据是已经存在的词，而基于神经网络的无词典方法则有可能在分词中出现自己造词的情况，把本该连接的词语分开，而本该分开的词语却没有分开。但是神经网络的强大之处在于它能够在无词典的情况下得到与有词典相近的性能。

另外，基于词典和不基于词典的方法在人名上表现得都不好，基于神经网络的方法在机构名称上识别的也不够准确。不过神经网络可以通过网络结构的改变、训练数据的增加而提升性能。但是基于词典的方法对于人名等的识别就是几乎不可能，比如“蔡英文”，如果语料库没有这个名字，那么就会将“蔡”和“英文”分开，那比如随便一个人的名字，不叫“英文”叫“汉文”，那这个词可能就不会出现在词典中，则基于匹配的方法就会将”汉“和”文“也分开了。

子词压缩

代码位于 `max_matching` 目录下的 `bpe.py` 文件中。

代码运行命令：

```
cd max_matching
python bpe.py
```

由于语料库巨大，只选取了前 100 个句子，并利用前向最大匹配算法实现子词压缩 BPE 算法，部分结果如下：

```
['李鹏', '江泽民', '改革开放', '温家宝',
'中国人民', '播音员主持人', '新华社记者',
'现代化建设', '解放思想实事求是', '扶贫工作',
'先进典型', '全国各族', '有中国', '中国特色',
'经济建设', '归侨侨眷', '世界各国', '特色社会主义',
'伟大旗帜', '十五大精神', '基本路线', '新年音乐会',
'贫困地区', '国家主席', '广播电台', '全国代表大会',
'高举邓小平理论', '贯彻落实', '团结一致',
'劳动模范代表', '中共中央政治局', ...]
```

由结果可以看出，子词压缩中提取出了许多人名和比较常用的组织机构名称。这些在分词时实际上是很容易分错的。

子词压缩也算是一种自然语言处理分词技术，将文本切分成更小的子词单元，而不是传统的单词或字符。子词压缩技术可以提高分词的精度和召回率，尤其对于那些难以用传统分词方法准确切分的文本。

对于中文分词，由于中文语言的复杂性，传统的基于规则或统计的分词方法可能无法完全准确地切分出所有的命名实体。子词压缩技术可以提供更加精细的分词粒度，将文本切分成更小的子词单元，从而有可能提高命名实体的识别精度。

例如，对于一个包含多个音译外来词或新词的句子，传统的分词方法可能无法准确地切分出这些词。子词压缩技术可以将这些音译外来词或新词切分成更小的子词单元，从而更好地捕捉其语义信息。

因此，子词压缩可以作为一种辅助技术，提高中文分词中命名实体的识别精度。但是，子词压缩并不能完全替代传统的中文分词方法，而是可以与其结合使用，以达到更好的分词效果。