机器学习实验报告

1 实验基本信息

GitHub 网址：https://github.com/lancopku/pkuseg-python

项目名称：pkuseg:一个多领域中文分词工具包

论文题目：PKUSEG: A Toolkit for Multi-Domain Chinese Word Segmentation

作者：Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, Xu Sun

发表时间：2019 年 6 月

2 实验目的

2.1 研究问题

分词，即文本分割（Word segmentation），是将书面文本划分为有意义的单元（例如单词，句子或主题）的过程。文本分割既是人类在阅读文本时的心理过程，同时也是计算机实现文本阅读的过程。本实验研究的问题为汉语分词，即将一个完整的汉语文本分割为若干有意义的单元。

2.2 研究意义

中文分词（Chinese Word Segmentation, CWS）是中文自然语言处理的基础步骤。由于单词定义了中文的基本语义单位，因此分词的质量直接影响到下游如机器翻译等任务的质量和性能。近年来，中文分词取得了长足的发展，表现最佳的系统主要基于条件随机字段（CRF）。但是，尽管取得了令人鼓舞的结果，但这些方法仍然严重依赖于特征工程(Feature Engineering)。为了解决这个问题，本项目探索了使用神经网络以使机器自动学习更好的分词方式。

3 基本原理

pkuseg-python 主要基于经典的 CRF 模型，并辅以 ADF 训练方法和精调的特征，以实现更快的训练速度、更高的测试效果和更好的泛化能力：

1. 在 CRF 模型中，特征选取对分词结果和分词性能有着不小的影响，获得一套效果好、泛化能力较强、分词速度适中的特征往往需要耗费大量时间。本项目的代码中包含了这样一套精调的特征，在领域内的训练和测试表明，pkuseg 使用的特征可以有效提升不同语料的测试集上的效果。

2. ADF 训练方法则可以加快训练速度和收敛效果，为 DIY 用户和希望自己训练模型的用户提供较好的训练体验。

4 实验步骤

1. 使用 pip install 安装工具包，可以通过 PyPI 安装(自带模型文件)，在控制台输入：

```
pip3 install pkuseg
```

2. 更新到最新版本，输入：

```
pip3 install -U pkuseg
```

3.进行分词测试（以下测试环境均为 Windows 10 + Visual Studio Code 1.45.0 + Python 3.7）

A. 使用默认配置进行分词

```
1. import pkuseg                          #调用 pkuseg
2. seg = pkuseg.pkuseg()                  #以默认配置加载模型
3. text = seg.cut('我爱北京交大')          #进行分词
4. print(text)
```

运行结果：

```
['我', '爱', '北京', '交大']
```

默认模型分词适用于用户无法确定分词领域，或文本内容含多个分词领域时的情况。

B. 细领域分词

我们首先使用默认配置对一段含有医学分词领域的文本进行分词，看看会有什么效果：

```
1. import pkuseg
2.
3. seg = pkuseg.pkuseg()
4. text = seg.cut('连花清瘟胶囊可用于新冠病毒性肺炎轻型、普通型引起的发热、咳嗽、乏力
   ')     #进行分词
5. print(text)
```

运行结果为：

```
['连花', '清瘟', '胶囊', '可', '用于', '新冠', '病毒性', '肺炎', '轻型', '、
', '普通型', '引起', '的', '发热', '、', '咳嗽', '、', '乏力']
```

我们可以看到，分级结果仅从语法角度将各个单元分割开，但并没有将诸如“连花清瘟胶囊”之类的医学术语名词作为一个整体单元进行分割。

下面使用细分领域模型对同一段文本进行分词。在使用细领域模型时，需要在`seg = pkuseg.pkuseg()`的括号内填入对应的模型名称。pkuseg 除默认配置外，额外提供了四种细分领域模型，分别为`"news"`，新闻领域模型；`"web"`，网络领域模型；`"medicine"`，医药领域模型；`"tourism"`，旅游领域模型。运行代码时，如果是第一次使用该细领域模型，程序会进行自动下载。在这里我们选择`"medicine"`，医药领域模型。

```
import pkuseg

seg = pkuseg.pkuseg(model_name='medicine')  #程序会自动下载所对应的细领域模型
text = seg.cut('连花清瘟胶囊可用于新冠病毒性肺炎轻型、普通型引起的发热、咳嗽、乏力') # 进行分词
print(text)
```

运行结果为：

```
['连花清瘟胶囊', '可', '用于', '新冠病毒性', '肺炎', '轻型', '、', '普通型', '引起', '的', '发热', '、', '咳嗽', '、', '乏力']
```

可以看到，使用医学细领域模型后，“连花清瘟胶囊”、“新冠病毒性”等医学专业术语没有被分割，更好地保留了整体意义。

C. 分词并进行词性标注

在对具有复杂语法结构的文本进行分词时，或对错误分词结果进行分析时，往往需要分词结果的词性进行分析。pkuseg 提供了多达 36 种词性的显示，具体请见 tags。下面我们选取一些具有复杂语法结构的文本进行分词。

（1）明明明明明白白白喜欢他，但他就是不说。

运行结果：

```
['明明', '明明', '明白', '白白', '喜欢', '他', '，', '但', '他', '就是', '不说', '。']
```

运行结果正确。

（2）今天下雨，我骑车差点摔倒，好在我一把把把把住了。

运行结果：

```
['今天', '下雨', '，', '，', '我', '骑车', '差点', '摔倒', '，', '，', '好在', '我', '一把把', '把', '把', '住', '了', '。']
```

后半句话的分词结果应为［'我', '一把', '把','把', '把', '住', '了', '。'］，而程序给出的分词结果是['我', '一把把', '把', '把', '住', '了', '。']。我们让程序显示分词词性来分析一下。

```
1. import pkuseg
2.
3. seg = pkuseg.pkuseg(postag = True) #显示分词词性
4. text = seg.cut('今天下雨，我骑车差点摔倒，好在我一把把把把住了。')
5. print(text)
```

运行结果：

```
[('今天', 't'), ('下雨', 'v'), ('，', 'w'), ('我', 'r'), ('骑车', 'v'), ('差点', 'n'), ('摔倒', 'v'), ('，', 'w'), ('好在', 'd'), ('我', 'r'), ('一把把', 'm'), ('把', 'q'), ('把', 'v'), ('住', 'v'), ('了', 'y'), ('。', 'w')]
```

可以看到，机器错误地将“一把把”当作了数词，后面的“把”当成了量词。“数词+量词”当然是符合语法结构的，但机器机械地配合语法结构却导致了对文本意义的理解错误。

4 实验结论

本项目的 pkuseg 工具包提供了一个多领域的预训练分词模型。当然，这个工具包还有很多需要解决的问题。除了在上文中提到的错误示例之外，在下面进行的数次实验中，曾多次出现分词不准确或分词词性显示错误的结果，碍于报告篇幅所限，这些错误的结果和结果分析未能一一展示。

但是，瑕不掩瑜，pkuseg 工具包内含的多领域的模型可以更有针对性的分词，较现有的通用模型在多领域中有优势。pkuseg 工具包为我们提供了一个已经训练好的、可以较为简单使用、可以直接获得的模型，具有良好的“开箱即用”的特性，免去了我们自己重新训练的周折。根据项目成员介绍，他们也将持续对代码质量、错误处理、兼容性、鲁棒性等诸多问题进行改进，并将在近期推出实现上更为高效、运行速度更快的版本，让我们拭目以待吧。