实验题目

不同分词方法的实施与对比

实验内容

利用给定的中英文文本序列（见 Chinese.txt 和 English.txt），分别利用以下给定的中英文分词工具进行分词并对不同分词工具产生的结果进行简要对比分析。

遇到和解决的问题

问题一

- 问题：在尝试使用 `StanfordCoreNLP` 进行中文分词时，发生了报错现象。报错信息为编码不正确，使用的为 `4.4.0` 版本。
- 解决：怀疑是 `Python` 版本与 `StanfordCoreNLP` 版本不匹配引发的错误。`github` 仓库中提供的版本对应表 `Python` 版本最低到 `3.7`，而我使用的是 `3.6` 版本。之后更改为 `3.7.0` 版本，但分词结果全为空。最后更改到 `3.9.1` 版本（尝试了 `github` 仓库中提供样例使用的版本），得到正确的结果

Py Version	CoreNLP Version
v3.7.0.1 v3.7.0.2	CoreNLP 3.7.0
v3.8.0.1	CoreNLP 3.8.0
v3.9.1.1	CoreNLP 3.9.1

问题二

- 问题：在尝试使用 `Spacy` 进行英文分词时，加载模型使用 `spacy.load("en")`，被告知，该方法已经被抛弃，建议使用 `spacy.load("en_core_web_sm")`。之后，使用 `spacy.load("en_core_web_sm")`，被告知找不到模型 `en_core_web_sm`
- 解决：`python -m spacy download en_core_web_sm` 大概率是不会成功的，我也确实没有通过这种方式成功安装。直接从 `github` 上下载，然后使用命令 `pip install en_core_web_sm-2.3.0.tar.gz` 进行安装，即可成功安装

实验步骤

中文分词

jieba

Paddle Mode

该模式的突出优点为，不会强行分词，一个词的长度可以比较长，比如出现了 `湖北神丹健康食品有限公司`，`3月15日晚间` 等符合人类阅读习惯的长短语。专有词的分析也比较到位，分出了 `神丹牌`， `莲田牌` 等其他模式不能分出的词。分词效果较好。略有不足的是，其他模式都能识别的 `央视`,在这里拆分成了两个字。运行速度略慢于其他模式。

Full Mode

该模式运行结果符合其特点：把句子中所有的可以成词的词语都扫描出来。比如将 `湖北省` 分成了 `湖北` 和 `湖北省`，将 `消费者` 分成 `消费` 和 `消费者` 等。将分词结果拼接起来，容易改变原句子的意思。

Default Mode

该模式相较于 `Full Mode` 来说，该模式容易将不同的形容词结合成一个词，而不容易将形容词与名词结合在一起。比如将 `土鸡蛋` 分成 `土`， `鸡蛋`。将 `好土商标` 分成了 `好土`， `商标`。词性区分比较明显。但是不能很好地处理专有名词，比如将 `新京报` 分成 `新京`，`报`。

Search Mode

其分词结果与精确模式相差不大，在精确模式的基础上，对长词进行了再拆分。比如将`有限公司`分为了`有限`， `公司`和`有限公司`。再分词确实有利于搜索时触发关键词。

- 代码

```
jieba.enable_paddle()  # 启动paddle模式。 0.40版之后开始支持，早期版本不支持
    seg_list = jieba.cut(data, use_paddle=True)  # 使用paddle模式
    print("Paddle Mode: " + '/'.join(list(seg_list)))

    seg_list = jieba.cut(data, cut_all=True)
    print("Full Mode: " + "/ ".join(seg_list))  # 全模式

    seg_list = jieba.cut(data, cut_all=False)
    print("Default Mode: " + "/ ".join(seg_list))  # 精确模式

    seg_list = jieba.cut_for_search(data)  # 搜索引擎模式
    print("Search Mode: " + "/ ".join(seg_list))
```

- 结果

Paddle Mode: 央/视/315/晚会/曝光/湖北省/知名/的/神丹牌、莲田牌“土鸡蛋”/实为/普通/鸡蛋/冒充/，/同时/在/商标/上/玩/猫腻，/分别/注册/“鲜土”、/注册/“/好土/”/商标/，/让/消费者/误/以为/是/“土/鸡蛋/”。/3月15日晚间/，/新京报/记者/就/此事/致电/湖北神丹健康食品有限公司/方面/，/其/工作/人员/表示/不知/情，/需要/了解/清楚/情况/，/截至/发稿/暂未/取得/最新/回应/。/新京报/记者/还/查询/发现/，/湖北神丹健康食品有限公司/为/农业/产业化/国家/重点/龙头/企业/、/高新技术/企业/，/此前/曾/因/涉嫌/虚假/宣传/“/中国/最大/的/蛋品/企业/”/而/被/罚/6万元/。
Full Mode: •/ 央视/ 315/ 晚会/ 曝光/ 湖北/ 湖北省/ 知名/ 的/ 神丹/ 牌/ 、/ 莲/ 田/ 牌/ “/ 土鸡/ 鸡蛋/ ”/ 实为/ 普通/ 鸡蛋/ 冒充/ ，/ 同时/ 在/ 商标/ 标上/ 玩/ 猫腻/ ，/ 分别/ 注册/ “/ 鲜/ 土/ ”、/ 注册/ “/ 好/ 土/ ”/ 商标/ ，/ 让/ 消费/ 消费者/ 误以为/ 以为/ 是/ “/ 土鸡/ 鸡蛋/ ”。/ 3/ 月/ 15/ 日/ 晚间/ ，/ 新/ 京报/ 记者/ 就此/ 此事/ 致电/ 湖北/ 神丹/ 健康/ 食品/ 有限/ 有限公司/ 公司/ 方面/ ，/ 其/ 工作/ 工作人员/ 作人/ 人员/ 表示/ 不知/ 不知情/ 知情/ ，/ 需要/ 了解/ 清楚/ 情况/ ，/ 截至/ 发稿/ 暂/ 未取/ 取得/ 最新/ 回应/ 。/ 新/ 京报/ 记者/ 还/ 查询/ 发现/ ，/ 湖北/ 神丹/ 健康/ 食品/ 有限/ 有限公司/ 公司/ 为/ 农业/ 农业产业/ 产业/ 产业化/ 国家/ 重点/ 龙头/ 龙头企业/ 企业/ 、/ 高新/ 高新技术/ 技术/ 企业/ ，/ 此前/ 曾/ 因涉嫌/ 涉嫌/ 虚假/ 宣传/ “/ 中国/ 最大/ 的/ 蛋品/ 企业/ ”/ 而/ 被/ 罚/ 6/ 万元/ 。
Default Mode: •/ 央视/ 315/ 晚会/ 曝光/ 湖北省/ 知名/ 的/ 神丹/ 牌/ 、/ 莲田牌/ “/ 土/ 鸡蛋/ ”/ 实为/ 普通/ 鸡蛋/ 冒充/ ，/ 同时/ 在/ 商标/ 上/ 玩/ 猫腻/ ，/ 分别/ 注册/ “/ 鲜土/ ”/ 、/ 注册/ “/ 好土/ ”/ 商标/ ，/ 让/ 消费者/ 误以为/ 是/ “/ 土/ 鸡蛋/ ”/ 。/ 3/ 月/ 15/ 日/ 晚间/ ，/ 新/ 京报/ 记者/ 就/ 此事/ 致电/ 湖北/ 神丹/ 健康/ 食品/ 有限公司/ 方面/ ，/ 其/ 工作人员/ 表示/ 不知情/ ，/ 需要/ 了解/ 清楚/ 情况/ ，/ 截至/ 发稿/ 暂未/ 取得/ 最新/ 回应/ 。/ 新/ 京报/ 记者/ 还/ 查询/ 发现/ ，/ 湖北/ 神丹/ 健康/ 食品/ 有限公司/ 为/ 农业/ 产业化/ 国家/ 重点/ 龙头企业/ 、/ 高新技术/ 企业/ ，/ 此前/ 曾/ 因涉嫌/ 虚假/ 宣传/ “/ 中国/ 最大/ 的/ 蛋品/ 企业/ ”/ 而/ 被/ 罚/ 6/ 万元/ 。
Search Mode: •/ 央视/ 315/ 晚会/ 曝光/ 湖北/ 湖北省/ 知名/ 的/ 神丹/ 牌/ 、/ 莲田牌/ “/ 土/ 鸡蛋/ ”/ 实为/ 普通/ 鸡蛋/ 冒充/ ，/ 同时/ 在/ 商标/ 上/ 玩/ 猫腻/ ，/ 分别/ 注册/ “/ 鲜土/ ”/ 、/ 注册/ “/ 好土/ ”/ 商标/ ，/ 让/ 消费/ 消费者/ 以为/ 误以为/ 是/ “/ 土/ 鸡蛋/ ”/ 。/ 3/ 月/ 15/ 日/ 晚间/ ，/ 新/ 京报/ 记者/ 就/ 此事/ 致电/ 湖北/ 神丹/ 健康/ 食品/ 有限/ 公司/ 有限公司/ 方面/ ，/ 其/ 工作/ 作人/ 人员/ 工作人员/ 表示/ 不知/ 知情/ 不知情/ ，/ 需要/ 了解/ 清楚/ 情况/ ，/ 截至/ 发稿/ 暂未/ 取得/ 最新/ 回应/ 。/ 新/ 京报/ 记者/ 还/ 查询/ 发现/ ，/ 湖北/ 神丹/ 健康/ 食品/ 有限/ 公司/ 有限公司/ 为/ 农业/ 产业/ 产业化/ 国家/ 重点/ 龙头/ 企业/ 龙头企业/ 、/ 高新/ 技术/ 高新技术/ 企业/ ，/ 此前/ 曾/ 涉嫌/ 因涉嫌/ 虚假/ 宣传/ “/ 中国/ 最大/ 的/ 蛋品/ 企业/ ”/ 而/ 被/ 罚/ 6/ 万元/ 。

自定义词典

```
央视
神丹牌
莲田牌
土鸡蛋
新京报
湖北神丹健康食品有限公司
龙头企业
315晚会
```

- 新增代码

```
jieba.load_userdict("../exp2/Userdict.txt")
```

各方法针对自定义词典中的词的分词有了比较大的提升，但仍保留了各自特点。比如搜索引擎模式，将`湖北神丹健康食品有限公司`分成了。但是仍有分不出来的情况，比如后两种方法没能分词`土鸡蛋`，`Paddle Mode`仍然没能分词`央视`等。

```
湖北/ 神丹/ 健康/ 食品/ 有限/ 公司/ 湖北神丹健康食品有限公司
```

- 结果

Paddle Mode: 央/视/315/晚会/曝光/湖北省/知名/的/神丹牌、莲田牌“土鸡蛋”/实为/普通/鸡蛋/冒充/，/同时/在/商标/上/玩/猫腻，/分别/注册/“鲜土”、/注册/“/好土/”/商标/，/让/消费者/误/以为/是/“土/鸡蛋/”。/3月15日晚间/，/新京报/记者/就/此事/致电/湖北神丹健康食品有限公司/方面/，/其/工作/人员/表示/不知/情，/需要/了解/清楚/情况/，/截至/发稿/暂未/取得/最新/回应/。/新京报/记者/还/查询/发现/，/湖北神丹健康食品有限公司/为/农业/产业化/国家/重点/龙头/企业/、/高新技术/企业/，/此前/曾/因/涉嫌/虚假/宣传/“/中国/最大/的/蛋品/企业/”/而/被/罚/6万元/。

Full Mode: •/ 央视/ 315晚会/ 晚会/ 曝光/ 湖北/ 湖北省/ 知名/ 的/ 神丹/ 神丹牌/ 、/ 莲田牌/ “/ 土鸡/ 土鸡蛋/ 鸡蛋/ ”/ 实为/ 普通/ 鸡蛋/ 冒充/ ，/ 同时/ 在/ 商标/ 标上/ 玩/ 猫腻/ ，/ 分别/ 注册/ “/ 鲜/ 土/ ”、/ 注册/ “/ 好/ 土/ ”/ 商标/ ，/ 让/ 消费/ 消费者/ 误以为/ 以为/ 是/ “/ 土鸡/ 土鸡蛋/ 鸡蛋/ ”。/ 3/ 月/ 15/ 日/ 晚间/ ，/ 新京报/ 京报/ 记者/ 就/ 此/ 此事/ 致电/ 湖北/ 湖北神丹健康食品有限公司/ 神丹/ 健康/ 食品/ 有限/ 有限公司/ 公司/ 方面/ ，/ 其/ 工作/ 工作人员/ 作人/ 人员/ 表示/ 不知/ 不知情/ 知情/ ，/ 需要/ 了解/ 清楚/ 情况/ ，/ 截至/ 发稿/ 暂/ 未取/ 取得/ 最新/ 回应/ 。/ 新京报/ 京报/ 记者/ 还/ 查询/ 发现/ ，/ 湖北/ 湖北神丹健康食品有限公司/ 神丹/ 健康/ 食品/ 有限/ 有限公司/ 公司/ 为/ 农业/ 农业产业/ 产业/ 产业化/ 国家/ 重点/ 龙头/ 龙头企业/ 企业/ 、/ 高新/ 高新技术/ 技术/ 企业/ ，/ 此前/ 曾/ 因涉嫌/ 涉嫌/ 虚假/ 宣传/ “/ 中国/ 最大/ 的/ 蛋品/ 企业/ ”/ 而/ 被/ 罚/ 6/ 万元/ 。

Default Mode: •/ 央视/ 315晚会/ 曝光/ 湖北省/ 知名/ 的/ 神丹牌/ 、/ 莲田牌/ “/ 土鸡蛋/ ”/ 实为/ 普通/ 鸡蛋/ 冒充/ ，/ 同时/ 在/ 商标/ 上/ 玩/ 猫腻/ ，/ 分别/ 注册/ “/ 鲜土/ ”/ 、/ 注册/ “/ 好土/ ”/ 商标/ ，/ 让/ 消费者/ 误以为/ 是/ “/ 土鸡蛋/ ”/ 。/ 3/ 月/ 15/ 日/ 晚间/ ，/ 新京报/ 记者/ 就/ 此事/ 致电/ 湖北神丹健康食品有限公司/ 方面/ ，/ 其/ 工作人员/ 表示/ 不知情/ ，/ 需要/ 了解/ 清楚/ 情况/ ，/ 截至/ 发稿/ 暂未/ 取得/ 最新/ 回应/ 。/ 新京报/ 记者/ 还/ 查询/ 发现/ ，/ 湖北神丹健康食品有限公司/ 为/ 农业/ 产业化/ 国家/ 重点/ 龙头企业/ 、/ 高新技术/ 企业/ ，/ 此前/ 曾/ 因涉嫌/ 虚假/ 宣传/ “/ 中国/ 最大/ 的/ 蛋品/ 企业/ ”/ 而/ 被/ 罚/ 6/ 万元/ 。

Search Mode: •/ 央视/ 晚会/ 315晚会/ 曝光/ 湖北/ 湖北省/ 知名/ 的/ 神丹/ 神丹牌/ 、/ 莲田牌/ “/ 土鸡/ 鸡蛋/ 土鸡蛋/ ”/ 实为/ 普通/ 鸡蛋/ 冒充/ ，/ 同时/ 在/ 商标/ 上/ 玩/ 猫腻/ ，/ 分别/ 注册/ “/ 鲜土/ ”/ 、/ 注册/ “/ 好土/ ”/ 商标/ ，/ 让/ 消费/ 消费者/ 以为/ 误以为/ 是/ “/ 土鸡/ 鸡蛋/ 土鸡蛋/ ”/ 。/ 3/ 月/ 15/ 日/ 晚间/ ，/ 京报/ 新京报/ 记者/ 就/ 此事/ 致电/ 湖北/ 神丹/ 健康/ 食品/ 有限/ 公司/ 湖北神丹健康食品有限公司/ 方面/ ，/ 其/ 工作/ 作人/ 人员/ 工作人员/ 表示/ 不知/ 知情/ 不知情/ ，/ 需要/ 了解/ 清楚/ 情况/ ，/ 截至/ 发稿/ 暂未/ 取得/ 最新/ 回应/ 。/ 京报/ 新京报/ 记者/ 还/ 查询/ 发现/ ，/ 湖北/ 神丹/ 健康/ 食品/ 有限/ 公司/ 湖北神丹健康食品有限公司/ 为/ 农业/ 产业/ 产业化/ 国家/ 重点/ 龙头/ 企业/ 龙头企业/ 、/ 高新/ 技术/ 高新技术/ 企业/ ，/ 此前/ 曾/ 涉嫌/ 因涉嫌/ 虚假/ 宣传/ “/ 中国/ 最大/ 的/ 蛋品/ 企业/ ”/ 而/ 被/ 罚/ 6/ 万元/ 。

SnowNLP

该方法针对专有词的提取不是很理想，容易将一个词拆分成不同的部分。但是提供了很多有意思的功能，比如拼音，词性标注等

- 代码

```
def snow_nlp(data):
    s = SnowNLP(data)
    print(s.words)
    print(s.pinyin)
```

- 结果

```
['\ufeff', '央视', '315', '晚会', '曝光', '湖北省', '知名', '的', '神丹', '牌', '、', '莲', '田', '牌', '“', '土', '鸡蛋', '”', '实', '为', '普通', '鸡蛋', '冒充', '，', '同时', '在', '商标', '上', '玩猫', '腻', '，', '分别', '注册', '“', '鲜', '土', '”、', '注册', '“', '好', '土', '”', '商标', '，', '让', '消费者', '误', '以为', '是', '“', '土', '鸡蛋', '”。3', '月', '15', '日', '晚间', '，', '新京', '报', '记者', '就', '此事', '致电', '湖北', '神', '丹', '健康', '食品', '有限公司', '方面', '，', '其', '工作', '人员', '表示', '不', '知情', '，', '需要', '了解', '清楚', '情况', '，', '截至', '发稿', '暂', '未', '取得', '最新', '回应', '。', '新京', '报', '记者', '还', '查询', '发现', '，', '湖北', '神', '丹', '健康', '食品', '有限公司', '为', '农业', '产业化', '国家', '重点', '龙头', '企业', '、', '高新技术', '企业', '，', '此前', '曾', '因', '涉嫌', '虚假', '宣传', '“', '中国', '最', '大', '的', '蛋品', '企业', '”', '而', '被', '罚', '6', '万', '元', '。']

['\ufeff', '央', 'shi', '315', 'wan', 'hui', 'bao', 'guang', 'hu', 'bei', 'xing', 'zhi', 'ming', 'de', 'shen', 'dan', 'pai', '、', 'lian', 'tian', 'pai', '“', 'tu', 'ji', 'dan', '”', 'shi', 'wei', 'pu', 'tong', 'ji', 'dan', 'mao', 'chong', '，', 'tong', 'shi', 'zai', 'shang', 'biao', 'shang', 'wan', 'mao', 'ni', '，', 'fen', 'bie', 'zhu', 'ce', '“', 'xian', 'tu', '”、', 'zhu', 'ce', '“', 'hao', 'tu', '”', 'shang', 'biao', '，', 'rang', 'xiao', 'fei', 'zhe', 'wu', 'yi', 'wei', 'shi', '“', 'tu', 'ji', 'dan', '”。3', 'yue', '15', 'ri', 'wan', 'jian', '，', 'xin', 'jing', 'bao', 'ji', 'zhe', 'jiu', 'ci', 'shi', 'zhi', 'dian', 'hu', 'bei', 'shen', 'dan', 'jian', 'kang', 'shi', 'pin', 'you', 'xian', 'gong', 'si', 'fang', 'mian', '，', 'qi', 'gong', 'zuo', 'ren', 'yuan', 'biao', 'shi', 'bu', 'zhi', 'qing', '，', 'xu', 'yao', 'liao', 'jie', 'qing', 'chu', 'qing', 'kuang', '，', 'jie', 'zhi', 'fa', 'gao', 'zan', 'wei', 'qu', 'de', 'zui', 'xin', 'hui', 'ying', '。', 'xin', 'jing', 'bao', 'ji', 'zhe', 'huan', 'cha', 'xun', 'fa', 'xian', '，', 'hu', 'bei', 'shen', 'dan', 'jian', 'kang', 'shi', 'pin', 'you', 'xian', 'gong', 'si', 'wei', 'nong', 'ye', 'chan', 'ye', 'hua', 'guo', 'jia', 'zhong', 'dian', 'long', 'tou', 'qi', 'ye', '、', 'gao', 'xin', 'ji', 'shu', 'qi', 'ye', '，', 'ci', 'qian', 'zeng', 'yin', 'she', 'xian', 'xu', 'jia', 'xuan', 'chuan', '“', 'zhong', 'guo', 'zui', 'da', 'de', 'dan', 'jing', 'qi', 'ye', '”', 'er', 'bei', 'fa', '6', 'wan', '元', '。']
```

• THULAC

该模型对专有名词的识别有不错的能力，可以识别出 `神丹牌_nz`， `莲田牌_nz` 等。但是该能力不稳定，可能是受上下文影响。比如前面不能识别 `土鸡蛋`，将其分成 `土_a` `鸡蛋_n`，后面却可以识别出 `土鸡蛋_n`。前面可以识别出 `新京报_nz`，后面却将其识别成了 `新_a` `京报_n`。但总体识别结果仅次于 `Paddle Mode`，优于前面出现的其他模

型。

- 代码

```
def thulac_nlp(data):
    thu1 = thulac.thulac()  # 默认模式
    text = thu1.cut(data, text=True)  # 进行一句话分词
    print(text)
```

- 结果

```
•_c 央视_v 315_m 晚会_n 曝光_v 湖北省_ns 知名_a 的_u 神丹牌_nz 、_w 莲田牌_nz “_w 土_a 鸡蛋_n ”_w 实_a 为_v 普通_a 鸡蛋_n 冒充_v ，_w 同时_c 在_p 商标_n 上_f 玩_v 猫腻_n ，_w 分别_d 注册_v “_w 鲜土_n ”_w 、_w 注册_v “_w 好_a 土_n ”_w 商标_n ，_w 让_v 消费者_n 误_d 以为_v 是_v “_w 土鸡蛋_n ”_w 。_w 3月_t 15日_t 晚间_t ，_w 新京报_nz 记者_n 就_p 此事_r 致电_v 湖北_ns 神丹_nz 健康_a 食品_n 有限公司_n 方面_n ，_w 其_r 工作_v 人员_n 表示_v 不_d 知情_v ，_w 需要_v 了_u 解_v 清楚_a 情况_n ，_w 截至_v 发稿_v 暂_d 未_d 取得_v 最新_a 回应_v 。_w 新_a 京报_n 记者_n 还_d 查询_v 发现_v ，_w 湖北_ns 神丹_nz 健康_a 食品_n 有限公司_n 为_p 农业_n 产业化_v 国_m 家_q 重点_n 龙头_n 企业_n 、_w 高新技术_n 企业_n ，_w 此前_t 曾_d 因_p 涉嫌_v 虚假_a 宣传_v “_w 中国_ns 最_d 大_a 的_u 蛋品_n 企业_n ”_w 而_c 被_p 罚_v 6万_m 元_q 。_w
```

NLPIR

该模型倾向于利用词性进行分词，容易将专有词，按不同词性划分成好几个词。常见的术语如`有限公司`， `高新技术`等分词效果较好。对不常见的`土鸡蛋`则会将其分成`土`和`鸡蛋`

- 代码

```
def pynlpir_nlp(data):
    pynlpir.open()
    ans = pynlpir.segment(data)
    print(ans)
```

- 结果

[('央', 'verb'), ('视', 'verb'), ('315', 'numeral'), ('晚会', 'noun'), ('曝光', 'verb'), ('湖北省', 'noun'), ('知名', 'adjective'), ('的', 'particle'), ('神', 'noun'), ('丹', 'distinguishing word'), ('牌', 'noun'), ('、', 'punctuation mark'), ('莲', 'noun'), ('田', 'noun'), ('牌', 'noun'), ('“', 'punctuation mark'), ('土', 'noun'), ('鸡蛋', 'noun'), ('”', 'punctuation mark'), ('实', 'adjective'), ('为', 'verb'), ('普通', 'adjective'), ('鸡蛋', 'noun'), ('冒充', 'verb'), ('，', 'punctuation mark'), ('同时', 'conjunction'), ('在', 'preposition'), ('商标', 'noun'), ('上', 'noun of locality'), ('玩', 'verb'), ('猫腻', 'noun'), ('，', 'punctuation mark'), ('分别', 'adverb'), ('注册', 'verb'), ('“', 'punctuation mark'), ('鲜', 'adjective'), ('土', 'noun'), ('”', 'punctuation mark'), ('、', 'punctuation mark'), ('注册', 'verb'), ('“', 'punctuation mark'), ('好', 'adjective'), ('土', 'noun'), ('”', 'punctuation mark'), ('商标', 'noun'), ('，', 'punctuation mark'), ('让', 'verb'), ('消费者', 'noun'), ('误', 'adverb'), ('以为', 'verb'), ('是', 'verb'), ('“', 'punctuation mark'), ('土', 'noun'), ('鸡蛋', 'noun'), ('”', 'punctuation mark'), ('。', 'punctuation mark'), ('3月', 'time word'), ('15日', 'time word'), ('晚间', 'time word'), ('，', 'punctuation mark'), ('新京报', 'multiword expression'), ('记者', 'noun'), ('就', 'adverb'), ('此事', 'pronoun'), ('致电', 'verb'), ('湖北', 'noun'), ('神', 'noun'), ('丹', 'distinguishing word'), ('健康', 'adjective'), ('食品', 'noun'), ('有限公司', 'noun'), ('方面', 'noun'), ('，', 'punctuation mark'), ('其', 'pronoun'), ('工作', 'verb'), ('人员', 'noun'), ('表示', 'verb'), ('不', 'adverb'), ('知', 'verb'), ('情', 'noun'), ('，', 'punctuation mark'), ('需要', 'verb'), ('了解', 'verb'), ('清楚', 'adjective'), ('情况', 'noun'), ('，', 'punctuation mark'), ('截至', 'verb'), ('发稿', 'verb'), ('暂', 'adverb'), ('未', 'adverb'), ('取得', 'verb'), ('最新', 'adjective'), ('回应', 'verb'), ('。', 'punctuation mark'), ('新京报', 'multiword expression'), ('记者', 'noun'), ('还', 'adverb'), ('查询', 'verb'), ('发现', 'verb'), ('，', 'punctuation mark'), ('湖北', 'noun'), ('神', 'noun'), ('丹', 'distinguishing word'), ('健康', 'adjective'), ('食品', 'noun'), ('有限公司', 'noun'), ('为', 'preposition'), ('农业', 'noun'), ('产业化', 'verb'), ('国家', 'noun'), ('重点', 'noun'), ('龙头', 'noun'), ('企业', 'noun'), ('、', 'punctuation mark'), ('高新技术', 'noun'), ('企业', 'noun'), ('，', 'punctuation mark'), ('此前', 'time word'), ('曾', 'adverb'), ('因', 'preposition'), ('涉嫌', 'verb'), ('虚假', 'adjective'), ('宣传', 'verb'), ('“', 'punctuation mark'), ('中国', 'noun'), ('最', 'adverb'), ('大', 'adjective'), ('的', 'particle'), ('蛋品', 'noun'), ('企业', 'noun'), ('”', 'punctuation mark'), ('而', 'conjunction'), ('被', 'preposition'), ('罚', 'verb'), ('6万', 'numeral'), ('元', 'classifier'), ('。', 'punctuation mark')]

StanfordCoreNLP

该模型也是主要依据词性而不是语义来进行分词，不能在专有词的分词上取得比较好的结果

- 代码

```
def stanford_nlp(data):
    # _*_coding:utf-8_*_
    with StanfordCoreNLP(r'D:\stanford-corenlp-full-2018-02-27',
lang='zh') as nlp:
        print(nlp.word_tokenize(data))
```

- 结果

```
['\ufeff', '央视', '315', '晚会', '曝光', '湖北省', '知名', '的', '神
丹', '牌', '、', '莲', '田', '牌', '“', '土', '鸡蛋', '”', '实为',
'普通', '鸡蛋', '冒充', '，', '同时', '在', '商标', '上', '玩', '猫
腻', '，', '分别', '注册', '“', '鲜土', '”', '、', '注册', '“', '好',
'土', '”', '商标', '，', '让', '消费者', '误以为', '是', '“', '土',
'鸡蛋', '”', '。', '3月', '15日', '晚间', '，', '新京报', '记者', '就
此事', '致电', '湖北', '神丹', '健康', '食品', '有限', '公司', '方面',
'，', '其', '工作', '人员', '表示', '不知情', '，', '需要', '了解',
'清楚', '情况', '，', '截至', '发稿', '暂', '未', '取得', '最新', '回
应', '。', '新京报', '记者', '还', '查询', '发现', '，', '湖北', '神
丹', '健康', '食品', '有限', '公司', '为', '农业', '产业化', '国家',
'重点', '龙头', '企业', '、', '高', '新', '技术', '企业', '，', '此
前', '曾', '因', '涉嫌', '虚假', '宣传', '“', '中国', '最', '大',
'的', '蛋品', '企业', '”', '而', '被', '罚', '6万', '元', '。']
```

英文分词

NLTK

英文分词与中文分词有很大的区别，这里的分词不像上面的中文分词那样，这里不用考虑短语，专有词等。需要做的就是将一个句子中的所有单词都拆分出来。分词效果还是比较不错的，但是没有主动忽略标点。

- 代码

```
def nltk_nlp(data):
    ans = nltk.word_tokenize(data)
    print(ans)
```

- 结果

```
['Trump', 'was', 'born', 'and', 'raised', 'in', 'the', 'New',
'York', 'City', 'borough', 'of', 'Queens', 'and', 'received',
'an', 'economics', 'degree', 'from', 'the', 'Wharton', 'School',
'.', 'He', 'was', 'appointed', 'president', 'of', 'his', 'family',
"'s", 'real', 'estate', 'business', 'in', '1971', ',', 'renamed',
'it', 'The', 'Trump', 'Organization', ',', 'and', 'expanded',
'it', 'from', 'Queens', 'and', 'Brooklyn', 'into', 'Manhattan',
'.', 'The', 'company', 'built', 'or', 'renovated', 'skyscrapers',
',', 'hotels', ',', 'casinos', ',', 'and', 'golf', 'courses', '.',
'Trump', 'later', 'started', 'various', 'side', 'ventures', ',',
'including', 'licensing', 'his', 'name', 'for', 'real', 'estate',
'and', 'consumer', 'products', '.', 'He', 'managed', 'the',
'company', 'until', 'his', '2017', 'inauguration', '.', 'He', 'co-
authored', 'several', 'books', ',', 'including', 'The', 'Art',
'of', 'the', 'Deal', '.', 'He', 'owned', 'the', 'Miss',
'Universe', 'and', 'Miss', 'USA', 'beauty', 'pageants', 'from',
'1996', 'to', '2015', ',', 'and', 'he', 'produced', 'and',
'hosted', 'The', 'Apprentice', ',', 'a', 'reality', 'television',
'show', ',', 'from', '2003', 'to', '2015', '.', 'Forbes',
'estimates', 'his', 'net', 'worth', 'to', 'be', '$', '3.1',
'billion', '.']
```

Spacy

相较于 `NLTK` 划分得更细，但是这样可能会将一些复合词分成两半。比如将 `co-authored` 分成 `co`，`-`，`authored`，这样是不太好的。

- 代码

```
def spacy_nlp(data):
    nlp = spacy.load("en_core_web_sm")
    doc = nlp(data)
    ans = [token.text for token in doc]
    print(ans)
```

- 结果

```
['Trump', 'was', 'born', 'and', 'raised', 'in', 'the', 'New',
'York', 'City', 'borough', 'of', 'Queens', 'and', 'received',
'an', 'economics', 'degree', 'from', 'the', 'Wharton', 'School',
'.', 'He', 'was', 'appointed', 'president', 'of', 'his', 'family',
"'s", 'real', 'estate', 'business', 'in', '1971', ',', 'renamed',
'it', 'The', 'Trump', 'Organization', ',', 'and', 'expanded',
'it', 'from', 'Queens', 'and', 'Brooklyn', 'into', 'Manhattan',
'.', 'The', 'company', 'built', 'or', 'renovated', 'skyscrapers',
',', 'hotels', ',', 'casinos', ',', 'and', 'golf', 'courses', '.',
'Trump', 'later', 'started', 'various', 'side', 'ventures', ',',
'including', 'licensing', 'his', 'name', 'for', 'real', 'estate',
'and', 'consumer', 'products', '.', 'He', 'managed', 'the',
'company', 'until', 'his', '2017', 'inauguration', '.', 'He',
'co', '-', 'authored', 'several', 'books', ',', 'including',
'The', 'Art', 'of', 'the', 'Deal', '.', 'He', 'owned', 'the',
'Miss', 'Universe', 'and', 'Miss', 'USA', 'beauty', 'pageants',
'from', '1996', 'to', '2015', ',', 'and', 'he', 'produced', 'and',
'hosted', 'The', 'Apprentice', ',', 'a', 'reality', 'television',
'show', ',', 'from', '2003', 'to', '2015', '.', 'Forbes',
'estimates', 'his', 'net', 'worth', 'to', 'be', '$', '3.1',
'billion', '.']
```

StanfordCoreNLP

可以看出，该方法分词的结果明显优于以上两种英文分词模型。可以将`family's`识别出来，而不是像上面一样将其拆分成两个部分。可以正常识别`co-authored`，并且能够与将`$`与数字作为一个整体进行识别。这样是符合人类阅读习惯的，和人工分词基本一致。但运行速度明显慢于其它几种方法。

- 代码

```
def stanford_nlp(data):
    # _*_coding:utf-8_*_
    with StanfordCoreNLP(r'D:\stanford-corenlp-full-2018-02-27',
lang='zh') as nlp:
        print(nlp.word_tokenize(data))
```

- 结果

```
['Trump', 'was', 'born', 'and', 'raised', 'in', 'the', 'New',
'York', 'City', 'borough', 'of', 'Queens', 'and', 'received',
'an', 'economics', 'degree', 'from', 'the', 'Wharton', 'School.',
'He', 'was', 'appointed', 'president', 'of', 'his', "family's",
'real', 'estate', 'business', 'in', '1971', ',', 'renamed', 'it',
'The', 'Trump', 'Organization', ',', 'and', 'expanded', 'it',
'from', 'Queens', 'and', 'Brooklyn', 'into', 'Manhattan', '.',
'The', 'company', 'built', 'or', 'renovated', 'skyscrapers', ',',
'hotels', ',', 'casinos', ',', 'and', 'golf', 'courses.', 'Trump',
'later', 'started', 'various', 'side', 'ventures', ',',
'including', 'licensing', 'his', 'name', 'for', 'real', 'estate',
'and', 'consumer', 'products.', 'He', 'managed', 'the', 'company',
'until', 'his', '2017', 'inauguration.', 'He', 'co-authored',
'several', 'books', ',', 'including', 'The', 'Art', 'of', 'the',
'Deal.', 'He', 'owned', 'the', 'Miss', 'Universe', 'and', 'Miss',
'USA', 'beauty', 'pageants', 'from', '1996', 'to', '2015', ',',
'and', 'he', 'produced', 'and', 'hosted', 'The', 'Apprentice',
',', 'a', 'reality', 'television', 'show', ',', 'from', '2003',
'to', '2015.', 'Forbes', 'estimates', 'his', 'net', 'worth', 'to',
'be', '$3.1', 'billion', '.']
```

实验结果

本次实验的结果在实验过程中展示更好，因为需要针对不同方法得到的结果进行分析。这里就不再展示实验结果。

实验总结

中文分词

中文分词中，`jieba`中的`paddle`方法分词效果是最好的。该方法可以在一定程度上识别名称和专有名词，不会刻意地将一个比较长的词分成不同的部分。它相交于其它方法最显著的不同就是出现了`湖北神丹健康食品有限公司`这一名称，这在其他模型中是没有出现过的。但是其速度在`jieba`中略慢。

其次，效果第二好的就是`THULAC`。该模型对专有名词的识别有不错的能力，可以识别出`神丹牌_nz`，`莲田牌_nz`等。但是该能力不稳定，可能是受上下文影响。比如前面不能识别`土鸡蛋`，将其分成`土_a 鸡蛋_n`，后面却可以识别出`土鸡蛋_n`。前面可以识别出`新京报_nz`，后面却将其识别成了`新_a 京报_n`。但总体识别结果仅次于

`Paddle Mode`。

上面评价一个模型的好坏都是通过，其分词后的结果是否符合人类阅读习惯来划分的。但是针对不同的用途，可以使用不同的模型。像`jieba`中的搜索引擎模式，倾向于将词的彻底拆分。不仅保留拆分的词，而且对拆分后的词进行再拆分，且保留结果。这有利用用户输入关键字，就能搜索得到想要的内容。

其他方法各有优劣，但是`StanfordCoreNLP`有很明显的缺点，使用复杂，运行速度慢。

英文分词

从实验结果看，分词效果为 `StanfordCoreNLP` > `NLTK` > `Spacy`。
`StanfordCoreNLP`分词效果与人工分词相比差异不大，准确分词出`family's`，`co-authored`，`$3.1`等复合型结构。`NLTK`能够分词出`co-authored`，`Spacy`会将`co-authored`分成`co`，`-`，`authored`三部分，效果不佳。

总的来说，上述各种方法都能起到分词的作用，且都具有自己的特点。英文分词相对中文分词来说要简单一些，英文分词主要任务是区分`word`，而中文分词会设计到很多`phrase`。