大作业1-报告

词性标注与分词

赖泽强，刘文卓，钱泽，谭超

展示内容

1. 中文分词：
    1. 最大匹配法
    2. Uni-Gram模型
    3. 隐式马尔可夫模型（2-Gram)
2. 词性标注
    1. 隐式马尔可夫模型（2-Gram）

中文分词

1. 最大匹配法
2. Uni-Gram模型
3. 隐式马尔可夫模型（2-Gram)

最大匹配法

- 正向、逆向、双向

- 双向：根据正向和逆向的切分结果以及中文分词一般原则，选出更好的那个。

- 中文分词的一般原则：切分出词的数量越少越好，切分出单字的数量越少越好

效果

	Precision	Recall	F1	字/s
前向	0.843	0.907	0.874	32000+
后向	0.845	0.909	0.876	10000+
双向	0845	0.909	0.876	7600+

测试集：Bakeoff 2005/PKU（分词均为这个）

问题1:日期与数字

1. 大量的日期和数字没能正确分词

2. 语料库中没有

这次 实际上 只 核实 了 6 0 多 人 ， 另 有 2 0 0 多 人 没有 核实 。 据 乡政府 的 说法 ， 是 因 村民 阻挠 无法 进行 。 但 在 没有 对 全部 签名 进行 核实 的 情况 下 ， 乡政府 却 上报 有关 部门 ， 声称 因 6 0 人中 有人 不 是 出于 个人 意愿 ， 罢 免 书 无效 。 不过 ， 即使 对 这 6 0 人 的 调查 ， 乡政府 的 同志 也 承认 ， 如果 换 了 人 核实 ， 可能 结果 会 有 不同 。

令 人 疑惑 的 是 ， 直到 1 1 月 3 0 日 还 坚持 罢 免 书 无效 的 乡政府 ， 突然 来 了 个 1 8 0 度 的 大 转弯 ， 不再 提 核实 问题 ， 仓促 决定 于 1 2 月 5 日 召开 罢 免 大 会 。

2 0 0 0 年 1 1 月 5 日 ， 湖北省 京山 县 三 阳 镇 党委 书记 、 原 镇长 董 烈 宏 下乡 检查 秋收 工作 ， 不知不觉 地 来到 了 小 阜 村 四 组 村民 夏 世 清 老汉 的 房前 。

解决方案

- 日期和数字出现规律较为单一
- 人工定义规则
  - 所有数字单独分做一个词
  - 若数字末尾有“年”，“月”，“日”，和其合并成一个词。
- 缺点：不能覆盖所有情况，以文字出现的日期和数字情况多样，难以用规则描述。如，上千，一两等等

效果

改进后

● 10月 9日 至 11日， 中国 共产党 第十五 届 中央委员 会 第五 次 全体 会议 在 北京 举行 。
全会 指出， 我们 已经 胜利 实现 了 现代化 建设 的 前 两 步 战略 目标， 经济 和 社会 全面 发展， 人民 生活 总体 上 达到 了 小康 水平 。 全会 审议 并 通过 《 中共中央 关于 制定 国民经济 和 社会 发展 第十 个 五年计划 的 建议 》 。

● 11月 8日， 厦门 特大 走私案 首 批 案件 一审 公开 宣判， 杨 前线 、 庄 如 顺 、 蓝 甫 、 叶 季 谌 等 14 人 被 依法 判处 死刑 。 在 此 之前， 3月 8日 和 9月 14日， 经 最高 人民法院 核准， 对 腐败 分子 胡 长清 、 成 克 杰 分别 执行 死刑 。 党风 廉政 建设 和 反 腐败 斗争 加大 了 力度， 取得 了 新 的 明显 成效 。

● 11月 9日， 经过 200 多 位 专家 学者 历时 5年 的 努力， “ 夏商周 断代 工程 ” 正式 公布 《 夏商周 年表 》， 把 我国 的 历史 纪年 由 西周 晚期 的 共和 元年， 即 公元前 841年 向前 延伸 了 1200多年 。

评测对比

改进前

	Precision	Recall	F1	字/s
前向	0.843	0.907	0.874	32000+
后向	0.845	0.909	0.876	10000+
双向	0845	0.909	0.876	7600+

改进后

	Precision	Recall	F1	字/s
前向	0.891	0.922	0.907	10000+
后向	0.893	0.925	0.909	9000+
双向	0.894	0.925	0.909	5000+

代码

```
def __is_date_or_number(self, string):
    '''判断一个字符串是否为数字或者日期
    '''
    length = len(string)
    if(length < 2):
        return 0
    #print(length)

    if(string.isdigit()):
        return 1
    elif((string[length-1]=='年' or string[length-1]=='月' or
          string[length-1]=='日'or string[length-1]=='时'
          or string[length-1]=='分') and string[0:length-2].isdigit()):
        return 1
    elif(length == 2 and(string[length-1]=='年'
            or string[length-1]=='月' or string[length-1]=='日'
            or string[length-1]=='时' or string[length-1]=='分') and string[0].isdigit()):
        return 1
    else:
        return 0
```

问题2:数据集

- 人民日报标注集，若用 gb2312 进行解码会报类似下面的错误，使用 gbk 或 gb18030 可正确解码。

  UnicodeDecodeError: 'gb2312' codec can't decode byte 0xe9 in position 7524: illegal multibyte sequence

- 山西分词语料库的分隔符可能为一个空格或多个空格，因此切分的时候要使用任意空白符切分

  line.strip().split(' ') # *错误*　　line.strip().split() #*正确*

中文分词

1. 最大匹配法
2. Uni-Gram模型
3. 隐式马尔可夫模型（2-Gram)

Uni-Gram模型

- 最大匹配法
  - 只考虑了词语是否在词典中
  - 没有考虑整个句子是否正确

Unigram: 人类 的 生活 会 变 得 怎样 ？ ☑

前向： 人类 的 生活会 变 得 怎样 ？ ✖

后向： 人类 的 生活会 变 得 怎样 ？ ✖

各部队党委普遍运用民主**生活会**、个人述职、群众评议等形式，认真开展批评与自我批评 。

- 采用Uni-Gram模型，计算句子出现的概率
  - $\mathrm{P}(Sent) = P(W_1) \times P(W_2) \times \cdots \times P(W_n)$
  - $P(W) = \frac{Occur_w}{Total\_words}$
- 通过动态规划求解最优分词方案

效果

	Precision	Recall	F1	字/s
普通	0.844	0.922	0.881	8400+
识别日期/数字	0.892	0.937	0.914	3600+

分析

- Uni-Gram的本质是尽可能的选择出现概率大的词。
- 其没有考虑上下文。
- 以生活会为例，Uni-Gram很难分出生活会这个词，而分成生活和会。
- 考虑上下文，即用2-Gram或更高，还可以改进。

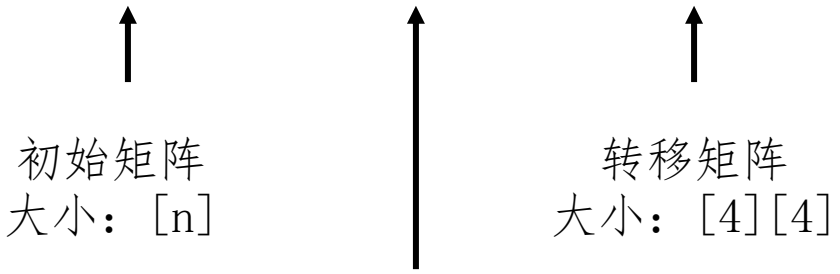
中文分词

1. 最大匹配法
2. Uni-Gram模型
3. 隐式马尔可夫模型（2-Gram)

隐式马尔可夫模型

- 基于字标注的模型
- 标记：B（词的开头），M（词的中部），
  E（词的结尾），S（单字）
- 使用$P(T|\mathrm{W})$作为评测标准。 $P(T|\mathrm{W})$：给定字串出现标记串的概率
- $P(T|\mathrm{W}) = P(t_1|t_0)\times P(w_1|t_1)\times P(t_2|t_1)\times P(w_2|t_2)\times\cdots\times$
  $P(t_n|t_{n-1})\times P(w_n|t_n)$

- $P(T|\mathrm{W}) = P(t_1|t_0) \times P(w_1|t_1) \times P(t_2|t_1) \times P(w_2|t_2)$



- 共有4种标记
- 设字典里有n个字

编程实现

- 第一步：转换数据集

- 第二步：构造字与标记的映射字典（映射到整数）。

unk:	0	B:	0
我:	1	M:	1
你:	2	E:	2
…		S:	3

- 第三步：编码，即用字典将数据集中的字符串全部映射为整数。

- 第四步：训练模型（构造三个矩阵）
  - 方法：遍历编码后的训练集，计数
  - 注意矩阵的平滑问题
- 第五步[可选]：保存模型（三个矩阵）
  - 保存到文件里,下次用就不用训练了
- 第六步：预测
  - 使用维特比算法求解最可能的标记序列

```
if __name__ == '__main__':
corpus_path = 'datasets/199801.txt'
corpus = utilities.load_renmin(corpus_path)
idxed_corpus, (obsv2idx, idx2obsv), (hide2idx, idx2hide) = index_corpus(corpus)
dicts = (obsv2idx, idx2obsv, hide2idx, idx2hide)
save_dicts(dicts, "hmm_para")

builder = HmmMatBuilder(idxed_corpus, len(obsv2idx.keys()),len(hide2idx.keys()))
builder.build()
builder.save("hmm_para")

obsv2idx, idx2obsv, hide2idx, idx2hide = load_dicts("hmm_para")
builder = HmmMatBuilder()
builder.load("hmm_para")

hmm = Hmm()
hmm.setup(builder.sp_mat, builder.tp_mat, builder.ep_mat, builder.num_obsv, builder.num_hide)

seq = ['19980101-01-001-002','中共中央','总书记', '、', '国家', '主席', '江', '泽民']
idxed_seq = [obsv2idx[word] for word in seq]

idxed_pos = hmm.find_hidden_state(idxed_seq)
pos = [idx2hide[idx] for idx in idxed_pos]
print(idxed_seq)
print(" ".join(seq))
print(idxed_pos)
print(" ".join(pos))
```

效果

	Precision	Recall	F1	字/s
隐式马尔可夫	0.777	0.792	0.785	20000+

- Bad
- 原因1: 未登录字还没做处理，有些句子没法分词。
- 原因2: 没有充分利用字典的信息

- 值得注意的是：马尔可夫模型可以较好的识别出人名。

问题1:稀疏矩阵的平滑

- 测试的时候，有些组合可能训练集中没有出现。
  - 如，发射矩阵[或][B] = 0，这时候P(T|W)就会直接变为0。
  - 这显然是我们不想要的，因此我们要认为的给矩阵中的所有0赋一个值，这就叫矩阵平滑。

$$P(T|\mathrm{W}) = P(t_1|t_0)\times P(w_1|t_1)\times P(t_2|t_1)\times P(w_2|t_2)$$

解决方法

- 方法:
  - 简单的+1平滑
  - Good Turing

问题2:标记解析

- ['B', 'E', 'B', 'E', 'S', 'B', 'E', 'B', 'E', 'B', 'E', 'B', 'E', 'B', 'E', 'B', 'E', 'S', 'S', 'B', 'E', 'B', 'M', 'M', 'E', 'B', 'E', 'B', 'E', 'B', 'M', 'E', 'B', 'E', 'S', 'S', 'B', 'E', 'B', 'E', '‘B’]
- **解析后：**专访/老瓦/：/可能/参加/北京/奥运/中国/领先/不/是/悲剧/2007/年06/月18/日/14/：/52/大/江网
- **实际应该是：**专访老瓦：可能参加北京奥运中国领先不是悲剧2007年06月18日14：52大江网.
- 少了个句号。因此需要特别注意解析方法

分析

- 问题在于，在解析的时候，我们组采用的方法是在遇到结束标记E的时候，把词语加进去。
- 如果一个B没有对应的E，则无法加入。

```
# 由tag还原回词
words = []
lo, hi = 0, 0
for i in range(len(tags)):
    if tags[i] == 'B':
        lo = i
    elif tags[i] == 'E':
        hi = i+1
        words.append(sentence[lo:hi])
    elif tags[i] == 'S':
        words.append(sentence[i:i+1])
```

解决方案

- 由于转移矩阵 BB, BS, MS, MB, SM, SE, EE, EM 都一定为0。（这些是不可能出现的转移）
- 可能出现的只有BE, BM, ME, MM, SS, SB, EB, ES
- 而这些只有BM, MM, SB, EB可能造成解析错误。
- 并且出错的时候，标记序列一定以上面四个中的某个结尾

- 因此，只需单独处理这些情况即可。

```python
# 由tag还原回词
words = []
lo, hi = 0, 0
for i in range(len(tags)):
    if tags[i] == 'B':
        lo = i
    elif tags[i] == 'E':
        hi = i+1
        words.append(sentence[lo:hi])
    elif tags[i] == 'S':
        words.append(sentence[i:i+1])

if tags[-1] == 'B':
    words.append(sentence[-1])  # 处理 SB,EB
elif tags[-1] == 'M':
    words.append(sentence[lo:-1])

assert len(sentence) == len("".join(words)), "还原失败,长度不一致"
```

未来的工作 & 改进点

- 尝试使用2-Gram或更高
  - $\mathrm{P}(Sent) = P(W_1|\mathrm{W}_0)\times P(W_2|W_1)\times\cdots\times P(W_n|W_{n-1})$
- 把大句子切成小句子

词性标注

1. 隐式马尔可夫模型（2-Gram)

隐式马尔可夫模型

- 基本与分词相同
- 标记集变为 -- {v, n, adj, p, …}
- 使用分好词的句子作为输入
  - 也可以使用字作为输入，那样的话就是分词与词性标注同时进行的联合模型。标记集变为 {bv,mv,ev,sv,…}

效果

	Precision	Recall	F1	字/s
隐式马尔可夫	～0.88	～0.88	～0.88	slow

- 测试集：人民日报语料库

- 测试时，用马尔可夫模型求解速度较慢。
- 该值为近似值。

其他

1. 双向LSTM（分词）

双向LSTM

- 基于字标注的分词模型。
- 基于统计的分词模型。
- F1值大概在0.88左右（PKU数据集）

谢谢大家