

中文分词系统

王艺丹
1190201303

摘要

本次实验针对中文分词选取不同的分词方法与模型，对中文分词进行初步的了解与实践。对于词典构建，实现了基于 list 的双 Trie 树进行优化；对于正反向最大匹配分词，结合已有的数据结构 Trie 树实现了算法优化，分词效率显著提高；实现了计算分词性能评价的常用指标，如准确率和召回率与 F 值；对于 N 元语法模型，实现了 2 元语法模型，并基于已有模型结合隐马尔可夫模型实现未登录词的识别；最后进行整体性能优化。本文最后基于各部分内容进行总结分析，论述结果与后续可能的优化。

1 绪论

在信息处理领域中，词是最小的能独立运用的语言单位。而中文信息处理技术在自然语言处理领域中起步较晚，且中文不同于拉丁语体系，不存在天然的分隔符，不能完全套用目前成熟的英文信息处理技术。因此，在中文信息处理过程中，需要先对成段的中文词序列进行分割，得到语言语义学上有意义的词。所以，中文分词是中文信息处理的基础，在中文信息处理领域中有着很关键的地位。可见，对中文自动分词技术的研究具有非常重要的意义，可促进中文信息处理技术的快速发展。中文分词作为中文自然语言处理领域的重要基础部分，随着近些年的研究与发展，目前较为成熟的分词方法可分为以下三种：

a) 基于规则的方法；b) 基于匹配的方法；c) 基于统计模型的方法。

2 相关工作

基于规则的方法利用构词原理结合标注的词性等信息，构建基于句法语义规则的分析系统，配合语法信息字典并补充大量消除歧义的信息。该方法的优点是具有针对性和暂时较高的准确率，但由于句法构造的领域相关性，适应性较差，词典与歧义消解处理难维护。**本次实验结合正则匹配，实现了日期序列的识别切分。**

基于匹配的方法又称机械分词方法，它是按照一定策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功（识别出一个词），该方法对于词典构建要求较高，效率较低。目前存在正向最大匹配分词、反向最大匹配分词、双向最大匹配分词等基本机械分词方法。**本次实验实现了正反向最大匹配分词。**

基于统计模型的方法主要为基于字标注的机器学习模型方法，即字在字串的标注问题，该方法能平等地看待词典词和未登录词的识别。主要方法又最大概率方法和最大熵方法等。随着大规模语料库的建立，统计机器学习方法的研究和发展，基于统计的中文分词方法渐渐成为了主流方法。主要统计模型有：N 元文法模型（N-gram），隐马尔可夫模型（Hidden Markov Model，HMM），最大熵模型（ME），条件随机场模型等。**本次实验实现了二元语法模型。**

针对各个传统分词方法以及实验要求，本次实验具体主要完成工作如下：

- 基于 list 列表的 Trie 树词典数据结构优化构建

- 基础正反向最大匹配分词，并结合已有字典数据结构进行优化
- 分词评价性能指标的计算
- 二元语法模型的实现

3 实验成果引导

针对各部分，均有单独的文件夹 $part_i$ ， $part_i$ 中包含 $data$ 、 $result$ 文件夹与 $.py$ 源程序，分别存放下发的训练集与要求生成的相应文件。 $.py$ 源程序中有 $main$ 方法，其他详细运行过程操作均已在代码中以注释形式给出。

4 实验过程

下文所提及的所有字典，均应用正则表达式匹配判断日期序号与量词，后续不做重复阐述。

4.1 词典的构建

分词标准：为了在封闭测试集中获得较高的分词结果，将词典文件除去日期序号的全部词语加入词典。由于专有名词在分词时的结果为更细分的词组，故不将专有名词作为完整的单个词语加入字典。

词典文件格式：词典文件dic.txt为utf-8编码的文本文档，每一行为一个单个的词。按拼音首字母进行排序。

对词典的分析：

- ✧ 后续处理未登录词数据时注意到GBK无法对一些字符进行编码，故选择utf-8编码，支持格式更为全面丰富
- ✧ 为使其在封闭测试集效果最优，选择将词语全部加入字典中
- ✧ 词典有序，按照拼音首字母排序

4.2 正反向最大匹配分词实现

该部分要求以最少的代码实现，即不追求代码算法优化，具体实验过程如下：

4.2.1 正向最大匹配分词

具体算法流程如下：

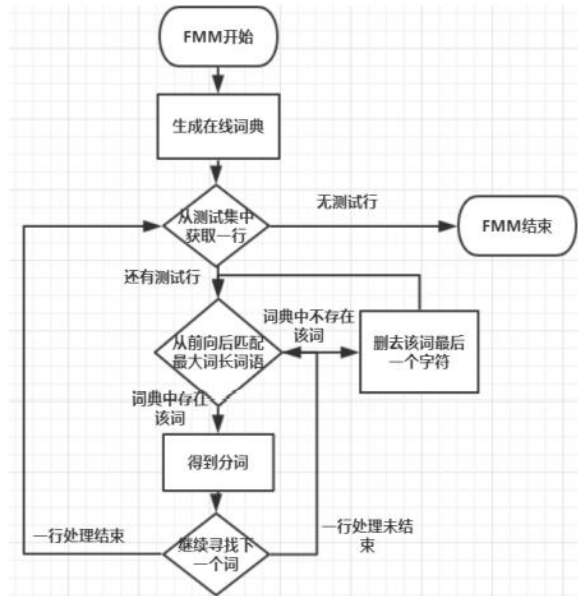


图 1. 正向最大匹配流程图

4.2.2 反向最大匹配分词

具体算法流程如下：

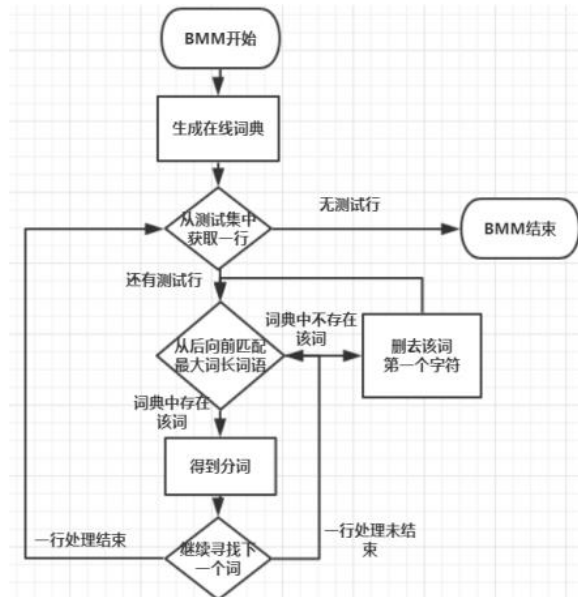


图 2. 反向最大匹配流程图

4.2.3 实验收获心得

由于该部分不允许使用内置字典dict数据结构，而遍历查找一个词语是否在词典中的时间复杂度为 $O(num_{words})$ ，可知对于 m 个句子逐个进行最大前缀匹配，逐句循环，时间复杂度为 $O(m * num_{words} * MAX_LEN^2)$ 。无法得到最终结果。该部分成功由优化后的词典结构与正反向匹配算法获得。

仅仅是 23031 句话与 55309 个词，就已经很难得到结果，易知效率在实际工程中的重要性与必要性。

4.3 正反向最大匹配分词效果分析

中文分词性能的评价指标主要有精确率、召回率、F 值等。其中精确率表示预测结果中，预测为正样本的样本中，正确预测为正样本的概率；召回率表示在原始样本的正样本中，最后被正确预测为正样本的概率。F 值则为综合评价精确率与召回率的综合指标。

4.3.1 指标计算函数实现

对于精确率：

$$Precision = \frac{TP}{NP + TP}$$

对于召回率：

$$Recall = \frac{TP}{FN + TP}$$

对于 F 值：

$$F = \frac{2PR}{P + R}$$

该部分对代码及实验工具不做要求，具体代码思想不做过多阐述。主要思想为将分词结果转换为形如 [(0,1), (1,4), (4,6), (6,10)] 的区间列表，通过集合的交并运算计算得到 TP, NP, FN，最终得到 P, R, F。

4.3.2 正反向匹配分词结果差异分析

由于汉语自身的特点，导致反向最大匹配效果往往优于正向最大匹配。这是因为，在汉语中，较长的词语往往出现在句子的后部，例如汉语中限定语在前，中心语在后的偏正结构（如“中国人民”）。因此后向匹配往往可以获得比从前向后更好的分词效果（中国/ 人民）；而从前向匹配则通过尽可能得到最长前缀词语的方式，导致结果中心语在前（中国人/ 民）。

因为汉语的这种偏正结构，导致反向最大匹配算法误差相对较小。再例如字符串“硕士研究生”，通过正向最大匹配，我们得到（硕士/ 研究生/ 产），这显然不是恰当的。而通过最大反向匹配，我们得到（硕士/ 研究/ 生产）。

由于汉语中偏正结构较多，后向匹配可以适当提高精确度。统计结果表明，单纯使

用正向最大匹配的错误率约为 1/169，单纯使用逆向最大匹配的错误率约为 1/245。

4.4 基于机械匹配的分词系统的速度优化

该部分针对数据结构与算法两方面进行优化。分词效率时间显著提升。

4.4.1 字典数据结构优化

易知常规的字典树通过字典数据结构使得查找时间复杂度为 $O(1)$ ，插入复杂度为 $O(len)$ ，而 list 列表查找为遍历查找，时间复杂度为 $O(n)$ ，所以可以结合哈希散列函数，使得列表的插入与查找时间复杂度优化至 $O(1)$ ，进而构建 Trie 树。

实现 TrieNode 与 Trie，根节点默认不存储任何字符，思想与常规方法的字典树相同，故不做赘述。选择 djb 哈希函数与线性探测实现词语的插入与查找。

4.4.2 正反向最大匹配算法优化

正向最大匹配：由于字典树本身具有最大前缀性质，故无需通过 MAX_LEN 进行迭代匹配，只需要按照句中汉字的顺序，从字典树根节点出发进行搜索，则该路径上为词语末尾的最深节点位置，即为最大匹配分词的一个词语结果，循环从分词结果的下一个汉字进行上述匹配过程即可。

反向最大匹配：在读取离线 txt 字典时，额外维护一个逆序字典树（将每一个词语逆序添加在字典中）；在反向最大匹配时，将句子逆序并结合逆序字典树，流程与 FMM 相同。最后将结果逆序输出即可。

总结：易知对一个长度为 len 的句子进行分词的时间复杂度为 $O(len)$ ，对于 m 句话的分词复杂度为 $O(len * m)$ ，效果显著提高。

4.4.3 优化结果分析

优化前后分词时间对比如下：

TimeCost.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

优化前FMM时间为：155663.2546s

优化后FMM时间为：19.818308s

优化前BMM时间为：146168.0281s

优化后BMM时间为：13.162235s

图 3. 优化结果对比

4.5 基于统计语言模型的分词系统实现

二元语法即指每个单词的概率仅取决于该单词之前的单词。本次实验针对该部分，对训练集进行处理，构建二元语法字典、词频字典、前缀字典、对句子建立有向无环图并应用 Viterbi 算法，实现了二元语法模型。

4.5.1 所需字典的构建

1) 离线二元语法字典构建

二元语法字典每行格式形如：词 前词 词频，则针对分词语料，分别在每一句句首与句尾加入 'BOS/' 与 '/EOS'，从第二个词开始循环遍历，利用 *pre* 记录前词，利用 *bi_dic[word][pre]* 记录词频，最终逐行写入 *bi_dic.txt* 构成离线二元语法词典。

部分展示如下：



```

EOS 住 1
EOS 鞭炮 1
EOS 响 2
EOS 白昼 1
EOS 手心 1
EOS 懈怠 1
EOS 喜悦 1
EOS 节日 1
EOS 讲话 8
EOS 元旦 2

```

图 4. *bi_dic.txt* 字典结果格式可视化

2) 读入离线字典

对于离线字典逐行遍历：

```

freq_dic[word] += int(freq)
bi_dic[word][pre] = int(freq)

```

即可获得二元语法字典与词频字典。
freq_dic 可视化格式结果如下：



```

年轻 76
年轻人 35
年轻力壮 1
年轻化 2
年轻有为 1
年过花甲 1
年迈 2
年近花甲 1
年逾古稀 1
年金 1


```

图 5. *freq_dic* 字典结果格式可视化

3) 前缀字典构建

前缀字典不重复的记录某一单词的所有可能前缀，用于后续有向无环图的建立。对于待写入前缀字典的词语，设置下标 *idx* 从词尾向前遍历，无重复的将该词词首至下标处的前缀加入前缀字典。并记录词频，如果统计字典中有该前缀则词频为该词对应词频，否则等于 0，则：
 $prefix_dic[w] = 0$ 即表示统计字典无该词。

prefix_dic 可视化结果格式如下：



```

prefix_dic.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
年轻人 35
年轻力壮 1
年轻力 0
年轻化 2
年轻有为 1
年轻有 0
年过花甲 1
年过花 0
年过 0
年迈 2

```

图 6. *freq_dic* 字典结果格式可视化

4.5.2 相关算法思路流程概述

1) DAG 词图构建

通过字与位置的映射，查找前缀字典与对应词频，表示一个句子所有可能的切分结果。

从当前位置 *k* 出发，通过下标 *i* 遍历句子，分别查找当前位置到 *i* 的词语在 *prefix_dic* 中的词频，不为 0 则将 *i* 记录在 *DAG[k]* 中，表示一种可能的分词结果。以下图为例

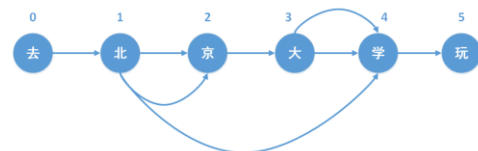


图 7. DAG 词图解析

对应输出结果为：

```
{0:[0], 1:[1,2,4], 2:[2], 3:[3,4], 4:[4], 5:[5]}
```

2) 负对数概率计算

防止概率下溢，将概率转为负对数概率求和。由于二元语法模型当前词仅与前词有关，为防止数据稀疏的情况，选择拉普拉斯平滑进行处理，有：

$$P(\text{word}|\text{pre}) = \frac{\text{freq}(\text{word}|\text{pre}) + 1}{\text{freq}(\text{pre}) + \text{words_num}}$$

分子前项即为二元语法字典中对应的：

bi_dic[word].get(pre, 0)

分母前项即为频率词典中对应的:

`freq_dic.get(pre, 0)`

利用减法计算负对数概率并返回即可。

3) 有向无环带权图构建

通过已有的DAG词图与概率计算公式,按拓扑序生成有向无环带权图。为了避免重复词语或字对结果的影响,仍然建立字与位置的映射,用元组 (i, k) 表示边,对应的图 $graph[i] = \{(i, k): \log P\}$,表示从当前位置可能的切分结果与对应的概率。

处理句首时,令前词 $pre = 'BOS'$;句中正常循环处理;句尾令 pre 为句尾词语,当前词语设为 $'EOS'$,即可。

4) Viterbi 算法得到最小负对数概率路径

首先正向按拓扑序遍历有向无环带权图 $graph$,遍历 $graph[i]$,对于每一个 $key[1]$,在前驱节点中,选择前驱节点对应的权值与自身权值加和最小的前驱节点,维护一个最优前驱节点与累计的权。

再逆向从 $'EOS'$ 遍历其维护的前驱节点,并加入到列表中,最后将列表逆序输出,即为最小负对数概率路径。返回最小负概率对数以及最优分词路径。

5) 分词

首先根据句子及词典生成 DAG 词图,再根据 DAG 生成带权有向图 gra ,再通过 Viterbi 算法获得最优路径与位置的映射 $path$ 。

则 $path$ 中每个元组(词首下标,词尾下标),映射到句子中即为一个分词结果,遍历 $path$ 得到分词后的各个词语,存于列表 div_result 中并返回。

6) 分词结果生成

对于待分词结果文件逐行处理,将`'/' .join(div_result)`写入结果文件即可。

4.5.3 不同分词方法的性能分析对比

1) 基于规则的分词方法

本次实验仅针对日期序号、数词量词、等识别使用了规则匹配的方法。该方法对待分词结果的规则性要求较高,而常规的汉语句子规则性较为复杂,如果单一使用规则匹配的方法,对规则的分析制定要求与耗费较高,实现较为困难。但由于规则的给出,基于规则的分词方法对于

分词内容本身的含义理解较强,故对于歧义与未登录词的处理表现良好。

2) 基于匹配的分词方法

本次实验的正反向最大匹配分词即为基于匹配的分词方法,显然,正反向最大匹配分词较于其他分词模型实现较为容易,但对训练集要求较高。基于匹配的分词只是将词语与已有字典进行比对,对词语内容本身的理解无关,若已有字典与待分词所含词语相关性较差,效果将很不理想。无法较好的处理歧义与未登录词。

3) 基于统计模型的分词方法

本次实验的二元语法模型即为基于统计模型的分词方法。可知,基于统计模型的分词方法是对可能的分词结果的极大概率估计,模型种类多且较为复杂,实现较为困难。由于与前词状态的相关性以及大规模的概率统计,基于统计模型的分词方法往往表现较好,对于歧义的处理与未登录词识别多数情况可以正确做出判断。但一般需要结合规则匹配实现。

5 总结

本次实验首先使用了简单的数据结构实现前向与后向的机械匹配算法,通过实现 Trie 树并优化算法,减少了单次最大词匹配的时间,并使用了更好的 Hash 函数来减少碰撞,大大减少了算法的时间与空间复杂度。

对于分词任务,合理的评估指标是衡量一个算法优劣的关键。有了合理的衡量指标,我们就能进行更加合理的探索,并挖掘中文分词相关的更多启发式信息。

统计语言模型能够利用大量语料的优势,通过直接对复杂的中文分词的概率分布进行建模,摆脱了机械、理解分词难以对复杂的中文进行建模的局限性。但是为了更好地发挥各种方法的长处,我们不能盲目认定某一种方法是解决问题的“银弹”,而应该取长补短,比如利用理解分词的启发式知识辅助统计语言模型进行分词等。

6 未来工作

在词典的构造,以及 MM 的匹配策略上融入更多的中文语言语义知识,进一步提高

基于统计语言模型方法的歧义消除能力以及分词精度。

使用更高级的语言模型，如近年来兴起的深度学习构建语言模型，尝试 LSTM，Bert 或他们的预训练神经网络，并采用条件随机场对分词任务进行建模。使用预训练这一深度学习范式，能够达到接近 SOTA 的效果。

参考文献

- 杨文珍, et al. 2021. 基于逆向最大匹配分词算法的汉盲翻译系统. 浙江大学虚拟现实实验室. <https://kns.cnki.net/KXReader/Detail?invoice=ICzdH2z4uNz7RGK%2FjQCMqQn9PCcHWM3McCZrnZfB15%2BHMj7%2FaEQ7aU5fBQveZN7dQlOWISUBLiGS0k8DnSKHMuHgTDYi0%2BfViglL0xm7%2FV3JPQUWyLKH8smZpLSYuVMHKUxLnvtnlfqKlk6gD6A%2BlprVMCIupY%2FnlZ9K0AUJAAw%3D&DBC CODE=CJFD&FileName=JYRJ202110016&TABLEName=cjfdauto&nonce=CB818F6C326B4FF48E9FF376F49A2A43&uid=&TIMESTAMP=1638975165212>
- 老顽童. 2017. 结巴分词 3—基于汉字成词能力的 HMM 模型识别未登录词. <https://www.cnblogs.com/zhibzz2007>
- 宗成庆. 2008. 统计自然语言处理