汉语分词系统

石翔宇
1190200523
xyu.shi@hit.edu.cn

摘要

本次实验目的是构建一个汉语分词系统。在课堂学习汉语自动分词原理和方法基础上，全面掌握汉语分词的若干关键技术，包括从词典的建立、分词算法的实现、性能优化和评价等环节。本次实验所要用到的知识和技能如下：语料库相关知识，正反向最大匹配分词算法，N 元语言模型相关知识，分词性能评价常用指标，基本编程能力（文件处理、数据统计等），相关的（查找）算法及数据结构实现能力。

1 介绍

中文分词指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。我们知道，在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是句段能通过明显的分界符来简单划界，而词是没有一个形式上的分界符的。虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂得多、困难得多。

由于英文的语言使用习惯，通过空格我们很容易拆分出单词；而中文字词接线模糊往往不容易区别哪些是“字”，哪些是“词”。这也是为什么我们想把中文的词语进行切分的原因。

当前研究所面临的问题和困难主要体现在三个方面：分词的规范、歧义词的切分和未登录词识别。

分词的规范 中文因其自身语言特性的局限，字 (词) 的界限往往很模糊，关于字 (词) 的抽象定义和词边界的划定尚没有一个公认的、权威的标准。曾经有专家对母语是汉语者调查结果显示，对汉语文本中“词”的认同率仅有百分之七十左右。正是由于这种不同的主观分词差异，给汉语分词造成了极大的困难。尽管在 1992 年国家颁布了《信息处理用现代词汉语分词规范》，但是这种规范很容易受主观因素影响，在处理现实问题时也不免相形见绌。

歧义词切分 中文中的歧义词是很普遍的，歧义词即同一个词有多种切分方式，该如何去处理这种问题。普遍认为中文歧义词有三种类型：

1. 交集型切分歧义，汉语词如 AJB 类型，满足 AJ 和 JB 分别成词。如“大学生”一种切分方式“大学/生”，另一种切分方式“大/学生”。你很难去判定那种切分正确，即使是人工切分也只能依据上下文，类似的有“结合成”、“美国会”等等。

2. 组合型切分歧义，汉语词如 AB，满足 A，B，AB 分别成词。如“郭靖有武功高超的才能”中的“才能”，一种切分“郭靖/有/武功/高超/的/才能”，另一种切分“中国/什么/时候/才/能/达到/发达/国家/水平”显示是不同的切分方式。

3. 混合型切分歧义，汉语词包含如上两种共存情况。如“郭靖说这把剑太重了”，其中“太重了”是交集型字段，“太重”是组合型字段。

未登录词识别 未登录词又称新词。这类词通常指两个方面，其一是词库中没有收录的词，

其二是训练语料没有出现过的词。未登录词主要体现在以下几种：

1. 新出现的网络用词：如“屌丝”、“蓝牙”、“蓝瘦香菇”、“房姐”、“奥特”、“累觉不爱”等。

2. 研究领域名称：特定领域和新出现领域的专有名词。如“苏丹红”、“禽流感”、“埃博拉”、“三聚氰胺”等。

3. 其他专有名词：诸如城市名、公司企业、职称名、电影、书籍、专业术语、缩写词等。如“成都”、“阿里巴巴”、“毛主席”、“三少爷的剑”、“NLP”、“川大”等。

综述所述，处理汉语词边界、歧义词切分和未登录词切分问题比较复杂，其中未登录词的影响大大超过了歧义词的影响，所以如何处理未登录词是关键问题。

本文将在课堂学习汉语自动分词原理和方法基础上，实现汉语分词的若干关键技术，包括词典的构建、正反向最大匹配分词、效果分析、基于统计语言模型的分词系统等环节。

2 相关工作

早在 80 年代就有中文分词的研究工作，曾有人提出“正向最大匹配法”、“逆向最大匹配法”、“双向扫描匹配法”、“逐词遍历法”等方法，共计多达 16 种之多。由于这些分词方法多是基于规则和词表的方法，随着统计方法的发展，不少学者提出很多关于统计模型的中文分词方法。关于规则的中文自动方法主要包括基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法。

基于字符串匹配的分词方法 基本思想是基于词典匹配，将待分词的中文文本根据一定规则切分和调整，然后跟词典中的词语进行匹配，匹配成功则按照词典的词分词，匹配失败通过调整或者重新选择，如此反复循环即可。代表方法有基于正向最大匹配和基于逆向最大匹配及双向匹配法。

基于理解的分词方法 基本思想是通过专家系统或者机器学习神经网络方法模拟人的理解能力。前者是通过专家对分词规则的逻辑推理并总结形成特征规则，不断迭代完善规则，其受到资源消耗大和算法复杂度高的制约。后者通过机器模拟人类理解的方式，虽可以取得不错的效果，但是依旧受训练时间长和过拟合等因素困扰。

基于统计的分词方法 统计的中文分词方法包括：

1. 基于隐马尔可夫模型的中文分词方法。基本思想是通过文本作为观测序列去确定隐藏序列的过程。该方法采用 Viterbi 算法对新词识别效果不错，但具有生成式模型的缺点，需要计算联合概率，因此随着文本增大存在计算量大问题。

2. 基于最大熵模型的中文分词方法。基本思想是学习概率模型时，在可能的概率分布模型中，认为熵最大的进行切分。该法可以避免生成模型的不足，但是存在偏移量问题。

3. 基于条件随机场模型的中文分词方法。基本思想主要来源最大熵马尔可夫模型，主要关注的字跟上下文标记位置有关，进而通过解码找到词边界。因此需要大量训练语料，而训练和解码又非常耗时。

综上所述，关于词典和规则的方法其分词速度较快，但是在不同领域取得效果差异很大，还存在构造费时费力、算法复杂度高、移植性差等缺点。基于统计的中文分词，虽然其相较于规则的方法取得不错的效果，但也依然存在模型训练时间长、分词速度慢等问题。针对这些问题，本文提出基于隐马尔可夫统计模型和自定义词典结合的方法，其在分词速度、歧义分析、新词发现和准确率方面都取得不错效果。

3 技术细节

我们将在本节分别介绍词典的构建、正反向最大匹配分词、基于统计语言模型的分词系统的具体实现细节。

3.1 词典的构建

词典是分词任务的基础，没有词典的分词任务就如无源之水。我们将训练集中的词性和所有辅助标注的符号去除掉，得到单个词。

需要注意的是，我们并不是将训练集的所有词都加到词典中。首先，训练集中每行开头的信息不加入词典。其次，训练集中重复的词不加入词典。这样不仅可以减少词典的噪声，同时也提高了词典的时空性能。

不同的分词方法对词典的要求不同。有的方法只要求词典中包含词即可，我们将这样的词典称为零元词典。而有的方法需要词典统计出各个词的出现频率，称之为一元词典。有的方法甚至需要统计出各个词及其前一个词共同出现时的频率，被称之为二元词典。

形式化地，我们定义词典的容量为 N，单个词为 w_i, $i = 1, 2, \ldots, N$，词 w_i 的出现频率为 $P(w_i)$，词 w_i 及其前一个词 w_{i-1} 共同出现的频率为 $P(w_i|w_{i-1})$。则零元词典的每个条目为

$$w_i, \quad i = 1, 2, \ldots, N \tag{1}$$

一元词典的每个条目为

$$w_i \ P(w_i), \quad i = 1, 2, \ldots, N \tag{2}$$

二元词典的每个条目为

$$w_i \ P(w_i|w_{i-1}), \quad i = 1, 2, \ldots, N \tag{3}$$

3.2 正反向最大匹配分词

最大匹配分词基于一种非常朴素的思想，对于每个分割出来的词组，我们总是希望这个词组的长度是最大的。其主要原理都是首先切分出单字串，然后和词典进行比对，如果当前切分是一个词那么就记录下来，否则通过增加或者减少一个单字，继续比较，一直还剩下一个单字则终止，如果该单字串无法切分，则作为未登录处理。

顾名思义，正向最大匹配分词就是从待分单句的前面开始，按照词典中的词进行切分，直到切分完成。而反向最大匹配分词则是从待分序列的末尾开始，按照词典中的词进行切分，直到切分完成。

该算法的朴素时间复杂度为 $O(L^2)$，但经过优化可以降低到 $O(kL)$，其中 k 为词典中词的最大长度。

3.3 正反向最大匹配分词中词典的结构

TODO

3.4 基于统计语言模型的分词系统

基于统计语言模型的分词的基本思想是把每个词看做是由字组成的，如果相连的字在不同文本中出现的次数越多，就证明这段相连的字很有可能就是一个词。基于统计语言模型的分词一般有如下两步操作：

1. 建立统计语言模型（n-gram）。
2. 对句子进行单词划分，然后对划分结果做概率计算，获取概率最大的分词方式。这里就用到了统计学习算法，如隐马尔科夫模型（HMM），条件随机场（CRF）等。

3.4.1 统计语言模型

统计语言模型在信息检索，机器翻译，语音识别中承担着重要的任务。这种模型结构简单、直接，但同时也因为数据缺乏而必须采取平滑算法。这里主要介绍 n 元语言模型（n-gram）。

每一个字节片段称为 gram，对所有 gram 的出现频度进行统计，并且按照事先设定好的阈值进行过滤，形成关键 gram 列表，也就是这个文本的向量特征空间，列表中的每一种 gram 就是一个特征向量维度。

该模型基于这样一种假设，第 k 个词的出现只与前面 $k-1$ 个词相关，而与其它任何词都不相关，整句的概率就是各个词出现概率的乘积。这些概率可以通过直接从语料中统计

k 个词同时出现的次数得到。常用的是一元的 Uni-Gram 和二元的 Bi-Gram。

如果我们有一个由 M 个词组成的序列，我们希望算得概率 $P(w_1, w_2, \ldots, w_M)$，根据链式规则，可得

$$P(w_1, w_2, \ldots, w_M) = P(w_1) * P(w_2 \mid w_1) \ldots P(w_M \mid w_1, \ldots, w_{M-1}) \tag{4}$$

这个概率显然并不好算，不妨利用马尔科夫链的假设，即当前这个词仅仅跟前面几个有限的词相关，因此也就不必追溯到最开始的那个词，这样便可以大幅缩减上述算式的长度。即

$$P(w_1, w_2, \ldots, w_M) = \prod_{i=1}^{M} P(w_i \mid w_{i-n+1}, \ldots, w_{i-1}) \tag{5}$$

上式的 n 指的就是的 n 元语言模型中的 n。

由此我们可以得到一元模型和二元模型的定义：当 $n = 1$ 时，一元模型即为：

$$P(w_1, w_2, \ldots, w_M) = \prod_{i=1}^{M} P(w_i) \tag{6}$$

当 $n = 2$ 时，二元模型即为：

$$P(w_1, w_2, \ldots, w_M) = \prod_{i=1}^{M} P(w_i \mid w_{i-1}) \tag{7}$$

接下来，在给定的训练词典中，我们可以利用贝叶斯定理，将上述的条件概率值都统计计算出来。

对于一元模型，

$$P(w_i) = \frac{C(w_i)}{N} \tag{8}$$

对于二元模型，

$$P(w_i \mid w_{i-1}) = \frac{C(w_i \mid w_{i-1})}{C(w_{i-1})} \tag{9}$$

这也对应了上面我们提到的一元词典和二元词典。

3.4.2 基于隐马尔科夫模型的分词

隐马尔可夫模型（HMM）是将分词作为字在句子中的序列标注任务来实现的。其基本思路是：每个字在构造一个特定词语时都占据着一个特定的位置即词位，一般采用四结构词位：B（词首），M（词中），E（词尾）和 S（单独成词）。比如：“中文/分词/是/文本处理/不可或缺/的/一步/!”的标注后的形式为“中/B 文/E 分/B 词/E 是/S 文/B 本/M 处/M 理/E 不/B 可/M 或/M 缺/E 的/S 一/B 步/E ！/S”。

致谢

这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。这里是致谢。

参考文献

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2331–2336.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1–11.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 199–209.

Daniel Fried, Mitchell Stern, and Dan Klein. 2017. Improving neural parsing by disentangling model combination and reranking effects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 161–166.

Juneki Hong and Liang Huang. 2018. Linear-time constituency parsing with rnns and dynamic programming. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 477–483.

Nikita Kitaev and Dan Klein. 2018a. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2675–2685.

Nikita Kitaev and Dan Klein. 2018b. Multilingual constituency parsing with self-attention and pre-training. *CoRR*, abs/1812.11760.

Jiangming Liu and Yue Zhang. 2017a. In-order transition-based constituent parsing. *Transactions of the Association for Computational Linguistics*, 5:413–424.

Jiangming Liu and Yue Zhang. 2017b. Shift-reduce constituent parsing with neural lookahead features. *Transactions of the Association for Computational Linguistics*, 5:45–58.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron C. Courville, and Yoshua Bengio. 2018. Straight to the tree: Constituency parsing with neural syntactic distance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1171–1180.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017a. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 818–827.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017b. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1695–1700.

Zhiyang Teng and Yue Zhang. 2018. Two local models for neural constituent parsing. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 119–132.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2773–2781.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 434–443.