基于 HMM 的中文词性标注系统

技术报告

第 1 版

姓名：何润康
学号：2018E8018661096
分组：单独完成

总页数	11	正文	9	附　录	无	日期：　2019 年　7 月　21 日
编制：何润康			审核：何润康			修改：何润康

目录

1. 引言

1.1 项目目标

本项目是为了实现一个基于 HMM（隐马尔科夫模型）的词性标注系统。词性标注（Part-of-Speech tagging 或 POS tagging)，又称词类标注或者简称标注，是指为分词结果中的每个单词标注一个正确的词性的程序，也即确定每个词是名词、动词、形容词或其他词性的过程。HMM 是一种基于统计的概率模型，常用来解决序列标注问题。在本项目中，使用 HMM 模型，结合人民日报 1998 年 1 月语料库，构建词性标注系统，最后得出词性标注的结果，并进行分析和总结，完成对自然语言处理课程的初步实践。

1.2 国内外相关工作

词性标注是自然语言处理的基础任务之一，它可以提高信息检索的效果和效率，因此在信息检索领域有着非常重要的作用。关于中文词性标注的方法，比较典型的有五类，分别是基于规则的方法、基于统计的方法、规则和统计相结合的方法、基于转换的学习方法（Brill 标注器） 和近年来流行的深度学习方法。其中基于规则的方法需要语言学家对特定的语言进行特征规则提取，而且对每一种语言都进行规则的提取，是一件很难且繁琐的事情[1]；基于统计的方法则通过建立统计模型，基于语料库对特定的语言问题进行建模，常见的模型有 HMM（隐马尔科夫模型）[2]、CRF（条件随机场）和基于最大熵的方法；规则和统计相结合的方法[3]，则结合了规则的消歧能力和统计方法对上下文的建模能力，显示出更好的性能；基于转换的学习方法，如 Brill 标注器[4]，通过自动发现并修正上次标记的错误，来不断地总结出有用的规则，从而改进标记的性能；另外，近年来研究人员也尝试用深度学习方法进行词性标注[5, 6, 7]。

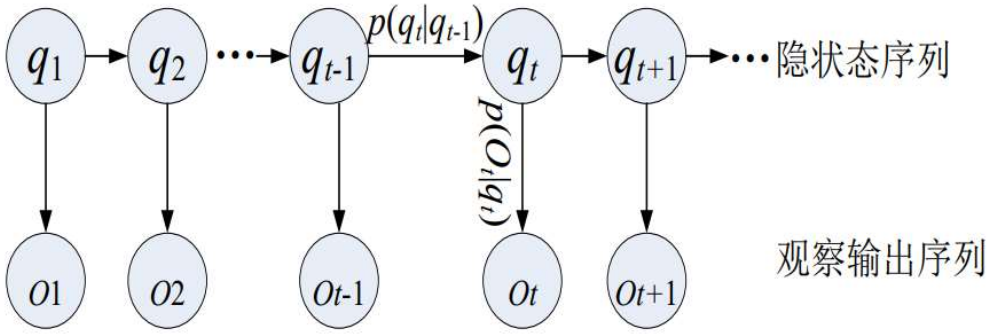
本项目通过实现经典的 HMM，基于人民日报 1998 年 1 月语料库，构建词性标注系统，并分析实验结果，最后提出改进的思路。

2. 处理过程

2.1 核心思想

词性标注问题可以看成是序列标注问题，即给定一个词的序列（也就是句子），找出最可能的词性序列（标签是词性）。HMM 是解决序列标注问题的经典模型，而 Viterbi 算法是 HMM 的解码算法，即找到概率最大的隐含状态序列。

HMM 是用来描述隐含未知参数的统计模型，图解如下：



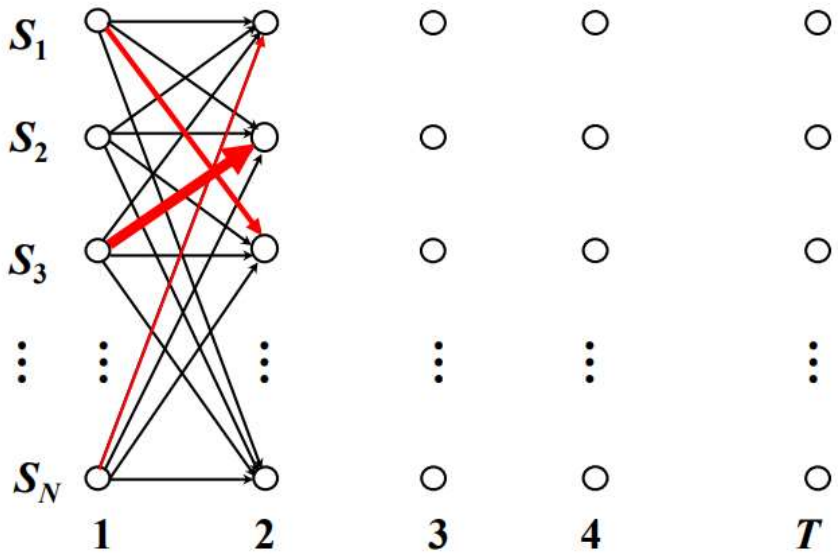
模型的参数由三部分组成，分别是初始状态概率、状态转移概率和发射概率，分别从训练语料库中统计获得。

Viterbi 算法是一种通用的解码算法，是基于动态规划的求序列最短路径的方法。序列最短路径的求解，满足动态规划的两个要素，即重叠子问题和最优子结构。

重叠子问题是指，问题的求解需要计算很多重复的子问题，如果不对这些子问题的结果进行存储，就会导致重复求解，导致复杂度变为指数级，比如从时刻 T 往前看，求解到达状态 S1 的最短路径，则要计算之前从时刻 1 到 T-1 最短路径这个子问题，同时 S2 的子问题与之是重复的。依次往前推，就可以发现子问题具有大量的重叠；最优子结构为原问题的解（即从时刻 1 到 T 的最短路径），可以由子问题的解合成，从自上而下分析，时刻 T 的最短路径取决于时刻 T-1 的最短路径，再加上时刻 T-1 到 T 的状态转移概率等，即 T 的最短路径必然包含 T-1 的最短路径。证明采用 cut-and-paste 方法，即如果不是最短，则把不是最短的替换为最短的。

具体实现上，采用自下而上的方法，先把时刻 1 的最短路径求解后，再依次构造后续时刻的解，最后构造出原问题的解。

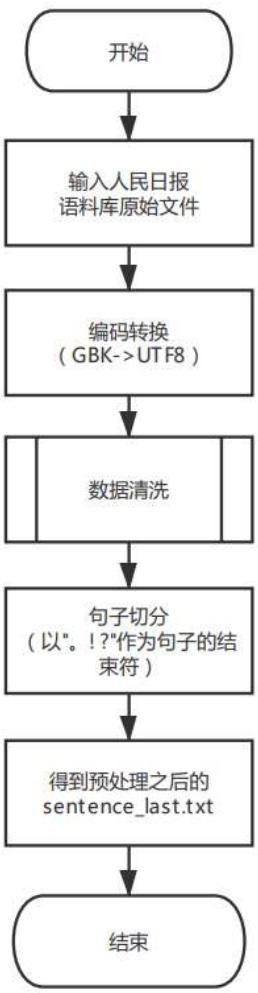
图解如下：

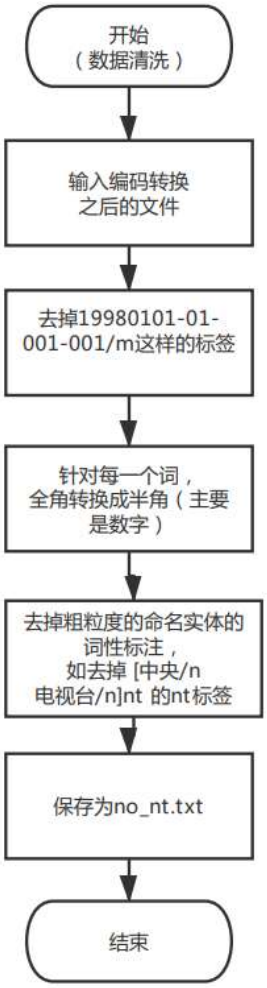


2.2 主要模块流程

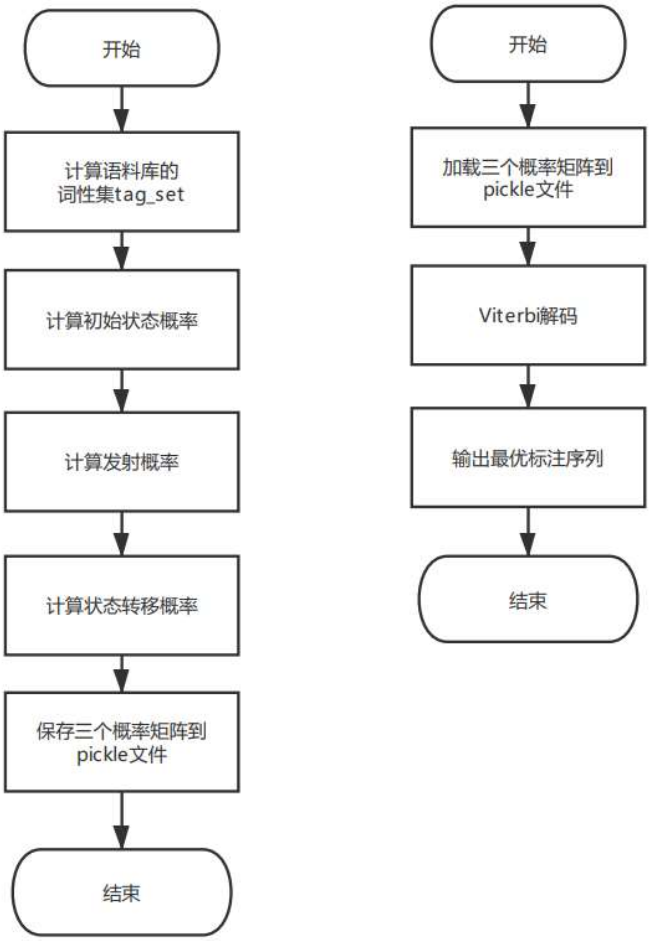
本项目包括三个主要模块，分别是预处理模块、HMM 和 Viterbi 模块、训练和评测模块。

预处理模块是针对人民日报 1998 年 1 月语料库的原始文件，经过编码转换、数据清洗和句子切分之后，得到可供后续 HMM 参数计算的语料文件。其流程图如下：

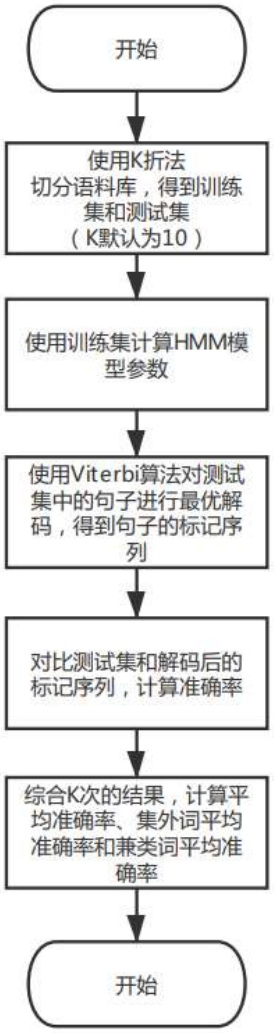




HMM 和 Viterbi 模块是核心模块，HMM 模块用于计算模型的三个主要的参数矩阵，分别是初始状态概率、发射概率和状态转移概率。需要注意的是，因为三个矩阵均存在数据稀疏的问题，所以需要进行平滑处理，本次采用了三种平滑，即简单平滑、加一平滑和 Good-Turing 平滑，简单平滑即统一给零概率值赋予一个很小的值。Viterbi 算法用于解码出一条最短路径，也即找出最优的标记序列。其流程图如下：



训练和评测模块是用来整合整个训练和测试流程的模块，因为单次训练得出准确率具有随机性，为了降低准确率的随机性，更加合理地得出模型的准确率，采用 K 折法对数据集进行 K 等份的拆分，不重复地每次取其中一份做测试集，其他 K-1 份作为训练集，运行 K 次，之后对标记的准确率取平均值。其流程图如下：



2.3 实验结果及分析

2.3.1 准确率

本项目使用 Sklearn 的 Kfold 函数（K=10），对数据集进行交叉验证，最后对结果取平均值，得到模型在人民日报语料库上的平均准确率。另外，因为集外词和兼类词的识别和标记是分词和词性标注任务中非常关键的问题，所以又对集外词和兼类词的占比和准确率进行了统计。模型的总体平均准确率、集外词平均准确率和兼类词平均准确率计算公式为：

$$总体准确率 = \frac{正确标记的个数}{测试集中词语总数}$$

$$集外词准确率 = \frac{正确标记的个数}{测试集中集外词总数}$$

$$兼类词准确率 = \frac{正确标记的个数}{测试集中兼类词总数}$$

结果见下表：

	总体	集外词	兼类词
平均准确率	90.42%	37.05%	87.22%
占比	100%	2.51%	38.57%

可知，总体准确率初步达到了要求，因为本项目使用的 HMM 是最基本的模型，数据平滑使用的是简单平滑[1]，所以达到 90%以上说明 HMM 作为经典模型的性能是比较好的。另外，集外词（未登陆词）的准确率很低，只有 37.05%，说明 HMM 对集外词的识别效果不够好，也可能是训练语料中命名实体的语料太少，无法对命名实体进行更高效的建模。集外词主要是命名实体，包括人名、地名和机构名等，虽然集外词占比只有 2.51%，但如果提高集外词的识别率到 90%，则可以使总体准确率提高 1.5%。兼类词的准确率尚可，说明 HMM 这种基于上下文的统计方法，有效地利用上下文信息确定了兼类词在具体语境下的词性。但兼类词的标记仍有提高的空间，如果提高到总体准确率，则可以使总体准确率提高 1.2%。

针对集外词，尤其是命名实体的识别和标记，可以对人民日报语料库进行再标记，如采用 IOB 标注法，然后训练出命名实体的 HMM 模型，也可以采用专门的命名实体识别语料库，训练好之后先对测试集进行初步标记，识别出其中的命名实体，之后再结合基于 HMM 的词性标注模型，进行后续的标记。

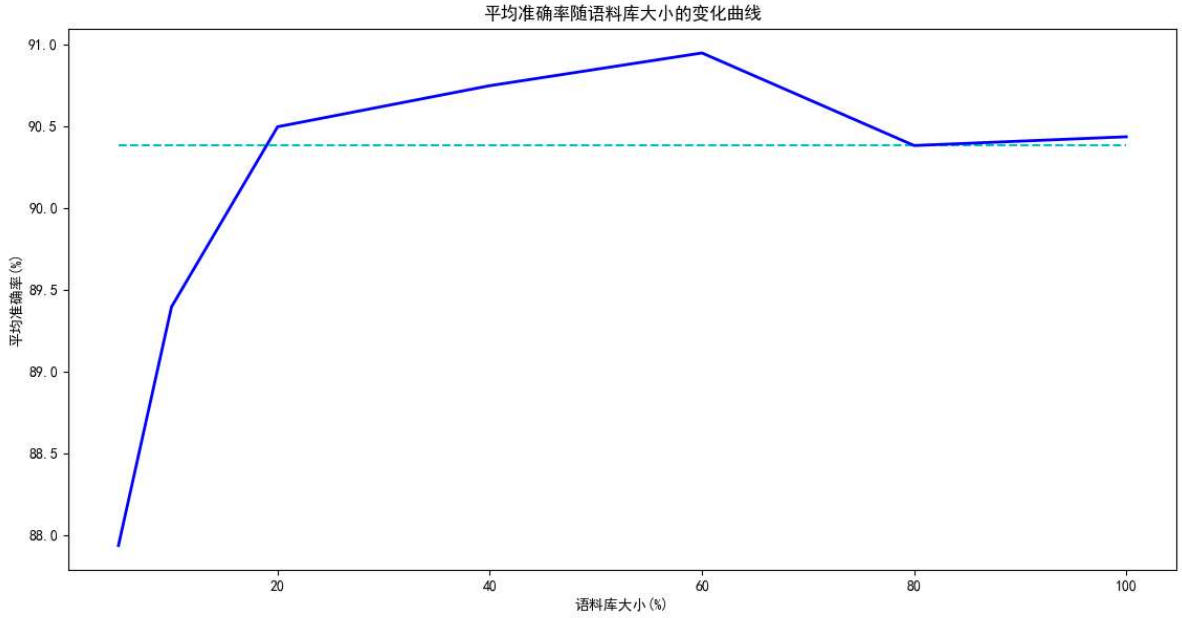
针对兼类词的标记，为了提高标记的准确率，考虑结合规则的方法，按兼类词搭配关系和上下文语境建造词类消歧规则。另外，考虑到 HMM 只使用三个矩阵建模，无法利用复杂的特征，所以考虑替换 HMM 为 CRF 和最大熵等可以建模任意复杂特征的模型。HMM 的另一个不足是，由于状态的转移是一阶的，相当于 Bigram，没有利用更多的上下文信息，所以可以考虑使用二阶的 HMM。

[1] 本项目尝试过其他的平滑方法，但测试的结果远不如使用简单平滑的好，使用加一平滑，准确率降低 10%，使用 Good-Turing，准确率降低 30%，但不知道是不是平滑代码写错了。暂且使用简单平滑的结果作为最终结果。

另外 HMM 模型只是利用了上文的信息，并没有全面利用上下文的信息，类似单向的 LSTM，目前我只能想到基于下文的 HMM 模型，如果要结合这两种算法，可以对一个待标记的句子正向解码一遍，再反向解码一遍，取最短路径概率最大的结果作为最终结果，或者对比正向和反向的解码结果，把标记一致的标签首先放在结果集里，对不一致的标签，则调用其他算法，比如 Bigram，统计哪种标记的概率更大一些。

2.3.2 语料库大小与模型性能的关系

项目对比了语料库大小与模型性能的关系，实验方法是依次利用 5%、10%、20%、40%、60%、80%和 100%大小的语料库作为数据集，并进行 K 折交叉验证（K=10），得出模型的平均准确率随着语料库大小的变化关系图，如下：



由上图可知，在训练语料很小的情况下，增加用于训练的语料的大小，将明显增加模型的准确率，但随着语料库逐渐增大，准确率趋于稳定（在 90.4%左右）。在 60%语料库大小的情况下，平均准确率最大，这可能是测试方法导致的，因为测试方法是每次取全体语料的前百分之多少，而不是在全体语料中再做一次 K 折（随机选取百分之多少）后取平均。

当前语料库使用的是一月的所有语料，实际上有全年的语料，但没有找到，所有没办法得知使用全年的数据后，模型性能的情况。另外，上图的结果不意味着继续增加语料库大小，对模型的性能提升影响不大。但不管怎样，增加语料库都会帮助模型把集外词变成集内词，并增加兼类词上下文的信息。这些对提升性能是有帮助的。

2.3.3 训练和标记速度

训练速度是指训练集中所有词的个数，除以计算模型矩阵所消耗的时间，也即把训练的时间均摊给每一个词，得到训练速度，单位是个/秒。标记速度则是把测试集中所有词的个数，除以使用 Viterbi 算法解码所消耗的时间，得出模型在实验机器上的标记速度，单位是个/秒。结果见下表[2]：

训练速度（个/秒）	标记速度（个/秒）	相差倍数
175512.31	475.05	369.46

从表中可知，训练速度远远快于标记速度，这是因为训练时只用单纯地统计转移概率和发射概率，均摊到每个词的时间复杂度是线性的，而测试时，Viterbi 算法的复杂度为N^2T，其中 N 是词性标签总数，为 44，T 是句子的长度，经统计，T 的平均值为 25.88，则测试的时间复杂度均摊到每个词也是线性的，只不过有一个很大的系数N^2，导致训练和测试的速度差异很大。

另外，以平均句长 25.88 算，一秒可以标记接近 20 个句子，基本满足日常使用的性能要求。如果想要实现更快的标注，可以并行化，因为每个句子的解码过程都是完全独立的。

3. 总结与展望

本项目通过动手实践，构建了一个基于 HMM 的中文词性标注系统，在人民日报 1998 年 1 月的新闻语料库上获得了 90.42%的准确率。在这个过程中，一步步摸索，调试，增加功能，终于完成了项目。另外，项目因为是做的经典的 HMM，后续有很多可以优化的工作，但由于没有时间在本项目中完成。这些工作在结果分析这一节中已经给出。

总体来说，本项目有两个地方需要改进。其一是需要进行开放测试。如果一个系统不经过在实际的评测中，而只是在严格标注的新闻语料库中评测，则无法得知系统在其他应用下的性能，这势必会影响系统的应用场景。本项目也曾考虑过做开放测试，但因为开源的语料库太少，而且标注集很不一样，所以无法做封闭测试。其二是需要对平滑算法进行重新探索，本次使用的简单平滑，严格意义上并不是真正的平滑算法，改进的方向是继续调试 Good-Turing，另外，这方面的平滑方法也被一些研究人员所研究，比如基于插值的方法。

[2] 测试环境说明：Intel Core i5-4200，Windows 10，Python 3.6

4. 参考文献

[1] 王丽杰, 车万翔, 刘挺. 基于 SVMTool 的中文词性标注[J]. 中文信息学报, 2009, 23(4): 16-22.
[2] 胡春静, 韩兆强. 基于隐马尔可夫模型 (HMM) 的词性标注的应用研究[J]. 计算机工程与应用, 2002, 6: 62-64.
[3] 张民, 李生. 统计与规则并举的汉语词性自动标注算法[J]. 软件学报, 1998, 9(2): 134-138.
[4] Brill E. A simple rule-based part of speech tagger[C]//Proceedings of the third conference on Applied natural language processing. Association for Computational Linguistics, 1992: 152-155.
[5] Zheng X, Chen H, Xu T. Deep learning for Chinese word segmentation and POS tagging[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 647-657.
[6] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.
[7] Santos C D, Zadrozny B. Learning character-level representations for part-of-speech tagging[C]//Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014: 1818-1826.

5. 致谢

本项目使用的开源工具有：numpy、sklearn.KFold，使用的语料库为北京大学 1998 年 1 月人民日报语料库，在此表示感谢。