链滴

BosonNLP分词技术解密

作者：BosonNLP
原文链接：https://ld246.com/article/1445499320074
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

在九月初BosonNLP全面开放了分词和词性标注引擎以后，很多尤其是从事数据处理和自然语言研究朋友在试用后很好奇，玻森如何能够做到目前的高准确率？希望这篇文章能够帮助大家理解玻森分词后的实现原理。

众所周知，中文并不像英文那样词与词之间用空格隔开，因此，在一般情况下，中文分词与词性标注往是中文自然语言处理的第一步。一个好的分词系统是有效进行中文相关数据分析和产品开发的重要证。

玻森采用的结构化预测模型是传统线性条件随机场（Linear-chain CRF）的一个变种。在过去及几年分词研究中，虽然以字符为单位进行编码，从而预测分词与词性标注的文献占到了主流。这类模型虽实现较容易，但比较难捕捉到高阶预测变量之间的关系。比如传统进行词性标注问题上使用Tri-gram征能够得到较高准确率的结果，但一阶甚至高阶的字符CRF都难以建立这样的关联。所以玻森在字符码以外加入了词语的信息，使这种高阶作用同样能被捕捉。

分词与词性标注中，新词识别与组合切分歧义是两个核心挑战。玻森在这方面做了不少的优化，包括特殊字符的处理，对比较有规律的构词方式的特征捕捉等。例如，近些年比较流行采用半监督的方式通过使用在大规模无标注数据上的统计数据来改善有监督学习中的标注结果，也在我们的分词实现上所应用。比如通过使用accressory variety作为特征，能够比较有效发现不同领域的新词，提升泛化力。

我们都知道上下文信息是解决组合切分歧义的重要手段。而作为一个面向实际商用环境的算法，除了准确率上的要求之外，还需要注意模型算法的时间复杂度需要足够高效。例如，相比于普通的Linear-hain CRF，Skip-chain CRF因为加入了更多的上下文信息，能够在准确率上达到更好的效果，但因为它在训练和解码过程，不论是精确算法还是近似算法，都难以达到我们对速度的要求，所以并没有在们最终实现中采用。一个比较有趣的分词改进是我们捕捉了中文中常见的固定搭配词对信息。譬如，如“得出某个结论”、“回答某个提问”等。如果前面出现“得出”，后面出现“结论”，那么“出”和“结论”作为一个词语出现的可能性就会很大，与这种相冲突的分词方案的可能性就会很小。类固定搭配也可以被建模，用于解决部分分词错误的问题。

怎样确定两个词是否是固定的搭配呢？我们通过计算两个词间的归一化逐点互信息(NPMI)来确定两个的搭配关系。逐点互信息（PMI），经常用在自然语言处理中，用于衡量两个事件的紧密程度。归一逐点互信息（NPMI）是逐点互信息的归一化形式，将逐点互信息的值归一化到-1到1之间。如果两个在一定距离范围内共同出现，则认为这两个词共现。筛选出NPMI高的两个词作为固定搭配，然后将组固定搭配作为一个组合特征添加到分词程序中。如“回答”和“问题”是一组固定的搭配，如果在注“回答”的时候，就会找后面一段距离范围内是否有“问题”，如果存在那么该特征被激活。

$$\mathrm{npmi}(x;y) = \frac{\mathrm{pmi}(x;y)}{-\log\left[p(x,y)\right]}$$

归一化逐点互信息(npmi)的计算公式

$$\mathrm{pmi}(x;y) \equiv \log\frac{p(x,y)}{p(x)p(y)} = \log\frac{p(x|y)}{p(x)} = \log\frac{p(y|x)}{p(y)}.$$

逐点互信息(pmi)的计算公式

可以看出，如果我们提取固定搭配不限制距离，会使后面偶然出现某个词的概率增大，降低该统计的定性。在具体实现中，我们限定了成为固定搭配的词对在原文中的距离必须小于一个常数。具体来看可以采用倒排索引，通过词找到其所在的位置，进而判断其位置是否在可接受的区间。这个简单的实有个比较大的问题，即在特定构造的文本中，判断两个词是否为固定搭配有可能需要遍历位置数组，次查询就有O(n)的时间复杂度了，并且可以使用二分查找进一步降低复杂度为O(logn)。

其实这个词对检索问题有一个更高效的算法实现。我们采用滑动窗口的方法进行统计：在枚举词的同维护一张词表，保存在当前位置前后一段距离中出现的可能成词的字符序列；当枚举词的位置向后移时，窗口也随之移动。这样在遍历到“回答”的时候，就可以通过查表确定后面是否有“问题”了同样在遇到后面的“问题”也可以通过查表确定前面是否有“回答”。当枚举下一个词的时候，词

也相应地进行调整。采用哈希表的方式查询词表，这样计算一个固定搭配型时间复杂度就可以是O(1)。

滑动窗口词语计数

词语	词频	词语	词频
迅速	1	回答	1
提出	1	的	1
问题	1		

通过引入上述的上下文的信息，分词与词性标注的准确率有近1%的提升，而对算法的时间复杂度没改变。我们也在不断迭代升级以保证引擎能够越来越准确，改善其通用性和易用性。今后我们也会在BsonNLP微信账户更多享我们在自然语言处理方面的经验，欢迎关注！