新词发现

什么是新词？

在中文自然语言处理任务中，分词一般是第一步，分词结果的好坏，会对后续任务质量造成严重影响。而现有的分词方法中，无论是基于何种模型何种方法，拥有一个较高质量的词典，都能极大的优化分词结果。

目前常说的新词（New Words）或未登录词(Unlisted Words），并未明确加以区分，通常未登录词被定义未加入词典的有意义词。这些词一般包括缩略词、专有名词、派生词、复合词等等，它们一般随着时间的推移出现。

新词发现方法

本文介绍一种常用的无监督新词发现方法。它主要依据两个统计特征，凝聚度和信息熵。

什么样的特征才能定义为一个词？

我们首先可能会想到，切割后的文本片段出现的频数，出现频数大于某一阈值的片段，可以作为候选词。当然，只考虑频数是肯定不够的，比如在一段文本中，“的电影”出现了 389 次，“电影院”只出现了 175 次，然而我们却更倾向于把“电影院”当作一个词，因为直觉上看，“电影”和“院”凝固得更紧一些。 [1]

为了证明“电影院”一词的内部凝固程度确实很高，我们可以计算一下，如果“电影”和“院”真的是各自独立地在文本中随机出现，它俩正好拼到一起的概率会有多小。在整个 2400 万字的数据中，“电影”一共出现了 2774 次，出现的概率约为 0.000113 。“院”字则出现了 4797 次，出现的概率约为 0.0001969 。如果两者之间真的毫无关系，它们恰好拼在了一起的概率就应该是 0.000113 × 0.0001969 ，约为 2.223 × 10-8 次方。但事实上，“电影院”在语料中一共出现了 175 次，出现概率约为 7.183 × 10-6 次方，是预测值的 300 多倍。类似地，统计可得“的”字的出现概率约为 0.0166 ，因而“的”和“电影”随机组合到了一起的理论概率值为 0.0166 × 0.000113 ，约为 1.875 × 10-6 ，这与“的电影”出现的真实概率很接近——真实概率约为 1.6 × 10-5 次方，是预测值的 8.5 倍。计算结果表明，“电影院”更可能是一个有意义的搭配，而“的电影”则更像是“的”和“电影”这两个成分偶然拼到一起的。[1]

当然，作为一个无知识库的抽词程序，我们并不知道“电影院”是“电影”加“院”得来的，也并不知道“的电影”是“的”加上“电影”得来的。错误的切分方法会过高地估计该片段的凝合程度。如果我们把“电影院”看作是“电”加“影院”所得，由此得到的凝合程度会更高一些。因此，为了算

出一个文本片段的凝合程度，我们需要枚举它的凝合方式——这个文本片段是由哪两部分组合而来的。令 p(x) 为文本片段 x 在整个语料中出现的概率，那么我们定义“电影院”的凝合程度就是 p(电影院) 与 p(电) • p(影院) 比值和 p(电影院) 与 p(电影) • p(院) 的比值中的较小值，“的电影”的凝合程度则是 p(的电影) 分别除以 p(的) • p(电影) 和 p(的电) • p(影) 所得的商的较小值。[1]

凝聚度

有了上文的概念，我们可以把凝聚度用以下公式表示。

$$PMI(A,B) = \log \frac{P(AB)}{P(A)P(B)}$$

其中，AB是一个候选词，A、B是组成其的两部分，P(AB)是AB出现的频率，P(A)和P(B)是A、B出现的频率。

tips

PMI 方法的缺点是过高估计了低频且总是相邻出现的字串间的结合强度。例如,
“啰”和“嗦”、“蝙”和“蝠”等在语料库中低频且总是相邻出现, 这些字串的 PMI 值非常高, 包含这些低频字串的垃圾串的 PMI 值也非常高, 例如“很啰”和“嗦”、“的蝙”和“蝠”等。[2]

为了解决上述问题，实践中采用了指数函数来代替线性函数，即

$$PMI(A,B)^k = \log \frac{P(AB)^k}{P(A)P(B)}$$

与参考论文中略微不同，在我所用到的数据中，k=1.5时效果较好。

左右熵

只有凝聚度的概念，还不足以来确定一个词，还需要从外部进行考虑。如果一个候选文本片段为一个词，则它除了内部的紧密程度高以外，外部的变化程度应该很大。直观理解为，一个词应该能够使用与很多不同场景，与它左右衔接的部分应该有足够的多样性，否则，它可能并不是一个独立的词。

考虑“被子”和“辈子”这两个片段。我们可以说“买被子”、“盖被子”、“进被子”、“好被子”、“这被子”等等，在“被子”前面加各种字；但“辈子”的用法却非常固定，除了“一辈

子”、“这辈子”、“上辈子”、“下辈子”，基本上“辈子”前面不能加别的字了。“辈子”这个文本片段左边可以出现的字太有限，以至于直觉上我们可能会认为，“辈子”并不单独成词，真正成词的其实是“一辈子”、“这辈子”之类的整体。可见，文本片段的自由运用程度也是判断它是否成词的重要标准。如果一个文本片段能够算作一个词的话，它应该能够灵活地出现在各种不同的环境中，具有非常丰富的左邻字集合和右邻字集合。[1]

“信息熵”可以用来衡量蕴含信息的丰富程度，熵越高，则能传输越多的信息，熵越低，则意味着传输的信息越少。

首先给出信息熵的公式。

$$entropy = -\sum_{i}^{n} p_i \log p_i$$

如果某个结果的发生概率为 p ，当你知道它确实发生了，你得到的信息量就被定义为 - log(p)。p 越小，你得到的信息量就越大。即你认为越不可能发生的事情发生了，带给你的震惊程度就越大，而习以为长的事情发生了（p越大）,带给你的震惊程度就越小

我们用信息熵来衡量一个文本片段的左邻字集合和右邻字集合有多随机。考虑这么一句话“吃葡萄不吐葡萄皮不吃葡萄倒吐葡萄皮”，“葡萄”一词出现了四次，其中左邻字分别为 {吃, 吐, 吃, 吐} ，右邻字分别为 {不, 皮, 倒, 皮} 。根据公式，“葡萄”一词的左邻字的信息熵为 - (1/2) • log(1/2) - (1/2) • log(1/2) ≈ 0.693 ，它的右邻字的信息熵则为 - (1/2) • log(1/2) - (1/4) • log(1/4) - (1/4) • log(1/4) ≈ 1.04 。可见，在这个句子中，“葡萄”一词的右邻字更加丰富一些。[1]

“被子”一词一共出现了 956 次，“辈子”一词一共出现了 2330 次，两者的右邻字集合的信息熵分别为 3.87404 和 4.11644 ，数值上非常接近。但“被子”的左邻字用例非常丰富：用得最多的是“晒被子”，它一共出现了 162 次；其次是“的被子”，出现了 85 次；接下来分别是“条被子”、“在被子”、“床被子”，分别出现了 69 次、 64 次和 52 次；当然，还有“叠被子”、“盖被子”、“加被子”、“新被子”、“掀被子”、“收被子”、“薄被子”、“踢被子”、“抢被子”等 100 多种不同的用法构成的长尾……所有左邻字的信息熵为 3.67453 。但“辈子”的左邻字就很可怜了， 2330 个“辈子”中有 1276 个是“一辈子”，有 596 个“这辈子”，有 235 个“下辈子”，有 149 个“上辈子”，有 32 个“半辈子”，有 10 个“八辈子”，有 7 个“几辈子”，有 6 个“哪辈子”，以及“n 辈子”、“两辈子”等 13 种更罕见的用法。所有左邻字的信息熵仅为 1.25963 。因而，“辈子”能否成词，明显就有争议了。“下子”则是更典型的例子， 310 个“下子”的用例中有 294 个出自“一下子”， 5 个出自“两下子”， 5 个出自“这下子”，其余的都是只出现过一次的罕见用法。事实上，“下子”的左邻字信息熵仅为 0.294421 ，我们不应该把它看作一个能灵活运用的词。当然，一些文本片段的左邻字没啥问题，右邻字用例却非常贫乏，例如“交响”、“后遗”、“鹅卵”等，把它们看作单独的词似乎也不太合适。我们不妨就把一个文本片段的自由运用程度定义为它的左邻字信息熵和右邻字信息熵中的较小值。[1]

使用左右信息熵的最小值来作为候选词的左右熵，可能会遇到一个常见问题。

假如某个词经常作为句首或者句尾，此时它的单侧信息熵会很小，因此使用阈值过滤时可能会把它们漏掉。因此，在实践中，对于这种情况，采用添加随机数的方法，给两侧增加padding，可以避免漏掉此类词语。

具体实施

在实际应用中，凝聚度和左右熵缺一不可，两者需要同时作为评价指标。否则，只看凝固程度的话，程序会找出“巧克”、“俄罗”、“颜六色”、“柴可夫”等实际上是“半个词”的片段；只看自由程度的话，程序则会把“吃了一顿”、“看了一遍”、“睡了一晚”、“去了一趟”中的“了一”提取出来，因为它的左右邻字都太丰富了。[1]

此外，候选词串的长度d需要设置最大长度，长度的增加会极大的增加计算复杂度（一般长度为5的字符串已经可以包含99%的词）。

结合词频、候选串长度、凝聚度和信息熵，就可以实现一个简易的新词发现系统。考虑到很多结果是已有词汇，加上词典过滤，就可以发现“新词”了。

结果展示

对部分娱乐新闻抽取，得到的新词结果。

复仇者联盟 奚梦瑶 嫁入豪门 孤芳不自赏 左耳失聪 快乐大本营 怪奇物语 撒狗粮 易烊千玺 晓彤 极限挑战 林丽莹 楚云浩 橘子君 欧阳娜娜 欲望之城 沈梦辰 泡芙 玉瑾 王俊凯 琅琊榜 白敬亭 盛一伦 秋瓷炫 程晓玥 纪凌尘 羞羞的铁拳 翟天临

可以看到，娱乐类新闻里的新词，大多数还是人名。

对部分社会类新闻抽取，得到的新词结果。

红树林基金会 征求意见稿 混淆行为 轻度污染 网瘾少年 森博会 公共自行车 载货汽车 梅姨 橙色预警 奥德赛之旅 投稿邮箱 全力抢救 王者荣耀 炫酷 冲击吸收 恋爱结婚 铁路专用线