基于机器学习方法与搜索引擎验证的缩略语预测*

焦 妍，王厚峰

北京大学 计算语言学研究所，北京 100871

E-mail: jy8939@gmail.com; wanghf@pku.edu.cn

摘 要：在自然语言中广泛使用的缩略语是重要的新词来源之一，成为了自然语言处理的一大问题。本文研究了从完整形式预测缩略语形式的方法。首先，使用 CRF 模型对完整形式预测，形成一定量的缩略候选，再利用搜索引擎得到的结果信息对各候选依次评估，通过打分和重排序，选择最终缩略结果。实验结果表明，增加网络信息之后，预测的缩略语准确率(top-1)可以提高 5 个百分点。

关键词：缩略语；CRF 模型；搜索验证

Abbreviation Prediction Using Machine Learning Method and Search Engine Verification

Jiao Yan, Wang Houfeng

Institution of Computational Linguistics, Peking University, Beijing 100871

E-mail: jy8939@gmail.com; wanghf@pku.edu.cn

Abstract: Abbreviations are commonly used in natural language texts. They have become a huge resource of Unknown Words, which brings challenges in Natural Language Processing. This article proposes a strategy of predicting abbreviation from full form. For a full form, it firstly generates a number of possible candidates using CRF. Then each of the candidates is re-scored according to the results of Web Search Engine based on different statistic methods. The candidate with highest score is selected as the abbreviation. Experiments show the precision of result improves 5% compared with single CRF prediction.

Keywords: abbreviation; CRF model; Search Engine

1 引言

缩略语在自然语言中被大量应用，是未登录新词的一大“贡献者”，给自然语言处理带来了诸多困难。在汉语分词、词性标注、命名实体识别、机器翻译和信息检索等领域都受到了缩略语问题的干扰。大规模的完整形式与缩略语对照库是解决上述问题的重要资源。从完整形式出发推导缩略形式是构建对照库的途径之一。这一过程也称为缩略语预测。

目前已有不少针对英文的缩略语研究[1]。汉语缩略语处理近年来也开始受到重视，并取得了一定的成果。代表性的研究包括 Chang 的工作[2][3]。在汉语缩略语预测方面，也已有研究报道。孙栩等使用支持向量回归(SVR)的方法对不同的缩略语候选进行打分和重排[4]。孙栩还研究了在序列标注基础上引入隐变量的方法 DPLVM，它比 CRF 更具一般性[5]。而 Yang 使用 CRF 模型生成候选，并利用完整形式和缩略语的字符串长度关系建立模型，进行重排序[6]。计峰专门针对汉语机构名的缩略预测也使用序列标注方式[7]。此外，随着 Internet 的迅速发展，网上隐藏了大量的有用信息，可以充分利用网络资源进行缩略语处理。Jiang 研究了使用搜索引擎结合线索词的方法挖掘缩略语和完整形式的关系[8]。Liu 研究了使用 Web 资源获取汉语缩略语完整形式的方法[9]。谢丽星则利用查询日志和锚文字作为桥梁，挖掘汉语缩略语和完整形式匹配对 [10]。

缩略语的形成受多种因素的影响，很难找到完全统一的规律，单纯使用机器学习方法或者规则方法都难以覆盖缩略语生成的各种现象。Jiang, Liu 和谢丽星的研究表明，充分利用已有的资源，

*本文受国家自然科学基金资助（编号: 60973053, 91024009,90920011）和博士点基金（编号: 20090001110047）资助。

特别是网络资源，对缩略语分析也具有很好的辅助作用。

基于上述已有的研究，本文提出了将机器学习方法与网络信息相结合进行缩略语预测。首先通过序列标注模型 CRF 对完整形式进行标注，产生可能的缩略语候选。再进一步利用搜索引擎返回的结果，对候选进行重排序和验证，从而得到最终的缩略形式。

2 基于序列标注模型的缩略语候选生成

2.1 缩略模型

本文将完整形式生成缩略语的过程看作一个序列标注问题。

定义 2.1 (序列标注) 序列标注问题即：给定长度为 n 的输入序列 $X_1X_2..X_n \in X^n$，形成输出序列 $Y_1Y_2..Y_n \in Y^n$，其中 X_i 来自一个可数集合 X，Y_i 来自有穷集合 Y，且 Y_i 是对应 X_i 的标记。

基于标注模型，可以得到缩略语的生成过程：

定义 2.2 (缩略生成模型) 定义集合 X 为所有汉字，标注集为 Y={S;K}，其中 S 表示“略过”(skip)，K 表示“保留”(keep)。从一个完整形式字序列 $X_1X_2..X_n \in X_n$ 生成相应缩略语的过程如下：

（1）生成标记序列 $Y_1Y_2..Y_n \in Y_n$；

（2）设其中所有标记为 K 的指标从小到大排列为 $1 \leqslant i_1 < i_2 < .. < i_m \leqslant n$；

（3）$X_{i1},X_{i2} .. X_{im} \in X^m$ 即为相应缩略语字序列。

通过序列标注的方法对完整形式进行序列标注，再抽取标记为 K 的字顺次连接，便得到缩略形式。例如：“北京理工大学”，若对应的标注序列为“北/K 京/S 理/K 工/K 大/S 学/S”，则缩略形式即为“北理工”。

2.2 CRF 模型

条件随机场（CRF）是一种判别式概率模型，被广泛用于序列标注问题中。它利用无向图模型定义了一个给定输入序列{Xi}时标记序列{Yi}的条件分布，最常用的链式 CRF 结构如图 2.1。

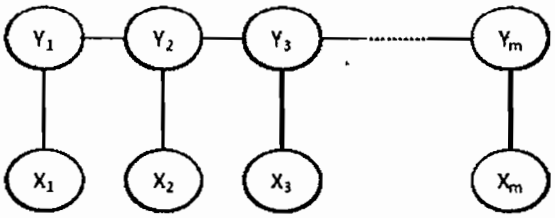

图 2.1 链式 CRF 结构

2.3 特征模板设计

在 CRF 模型中，本文使用的特征模板如下,简称为特征 1-6：

特征模板
1. X_i 的汉字以及拼音；
2. X_{i-1} 的汉字以及拼音；
3. (X_{i-j},X_{i-j+1}) 的汉字二元组和拼音二元组，其中 $j \in \{0;1;2\}$；
4. X_{i-j} 是否为数字，其中 $j \in \{0;1;2;3\}$；
5. $[[X_{i-j}=X_{i-j+1}]]$，其中 $j \in \{0;1;2\}$；
6. $[[X_{i-j}=X_{i-j+2}]]$，其中 $j \in \{0;1;2;3\}$。

3 基于搜索引擎返回结果的重排序

本文利用搜索引擎返回的结果，对机器学习得到的候选进行重排序。

我们的测试表明，使用上述特征模板经过 CRF 模型预测的缩略形式的候选，按条件概率的 Top-k 计算，有如下覆盖率（即前 k 个结果中包含正确结果的百分比）：

表 3.1 由 CRF 模型得到的 top-k 覆盖率

	Top-1	Top-3	Top-5	Top-10	Top-15	Top-20	Top-25
coverage	56.87%	74.91%	81.27%	90.35%	93.87%	94.67%	95.46%

表 3.1 显示，前 10 个候选的正确覆盖率为 90.35%。为了检验重排对 Top-1 的影响，同时又尽可能控制计算复杂性，本文只选择 CRF 的前 10 个候选进行打分和重排。

我们使用www.baidu.com 进行相关搜索，根据前 20 个搜索结果的信息对前 10 个缩略语候选评估和打分。用到的信息包括标题 title、摘要 snippet，URL 地址以及搜索引擎检索到的结果数量 resultNum。我们分别采用了以下几种统计方法进行打分，并最终与 CRF 模型的条件概率值一并综合实现重排序。

3.1 基于缩略语的搜索

将前 10 个缩略语候选中的每个候选在百度搜索引擎中搜索，取前 20 个返回结果进行分析和打分。包括如下两种打分。

（1）基于标题的打分

含有缩略语的文章，其完整形式很可能在标题中出现。针对每个缩略语 abbr,可以从 20 个返回结果中统计完整形式 full 在多少个标题中出现，以 titleFullCount(abbr)表示，同时统计被单独标红（强调显示）的缩略形式在多少个标题中出现，以 titleAbbrCount(abbr)表示。利用这两种信息，可以排除掉属于完整形式的一部分，但不能构成正常词汇的那些候选。比如“粮食交易会”的候选“粮食交会”并非正确缩略语，但搜索结果表明前 20 条中有 14 条标题包含“粮食交易会”，而每条标题都不包含“粮食交会”这个词。首先，引入公式(3.1)和(3.2)

$$\text{titleAbbrCount}(\text{abbr}) = \sum_{j=1}^{20} \text{ifOccur}(\text{title}_j[\text{abbr}], \text{abbr})\ ^{1} \tag{3.1}$$

$$\text{titleFullCount}(\text{abbr}) = \begin{cases} \sum_{j=1}^{20} \text{ifOccur}(\text{title}_j[\text{abbr}], \text{full}), & \text{titleAbbrCount}(\text{abbr}) \neq 0 \\ 0, & \text{titleAbbrCount}(\text{abbr}) = 0 \end{cases} \tag{3.2}$$

（2）基于摘要的打分

与标题的统计方法类似，统计完整形式在多少个摘要中出现，以 snippetFullCoun(abbr)表示，同时统计被单独标红（强调显示）的缩略形式在多少个摘要中出现，以 snippetAbbrCount(abbr)表示。计算方式与(3.1)和(3.2)相同，将 title 替换为 snippet 即可。

3.2 缩略语与完整形式的对比

由于缩略语和完整形式表达同样的语义，那么如果一篇文章中包含了缩略语和完整形式的大量信息，则这个网页很有可能以较高的排名同时出现在二者的搜索结果中。因此对完整形式单独进行搜索，再与上一步得到的每个候选的搜索结果进行对比。本文选择网页地址 URL 和网页标题 title 分别进行对比。

[1] ifOccur(a,b) 函数，即判断 b 是否在 a 中出现，若出现则返回 1 否则返回 0。

（1）网页标题对比

考虑到两个网页可能具有相同的内容但不同源，因此对完整形式和缩略形式的搜索结果的标题进行比较。为解决同一标题重复出现的问题，取完整形式的前 10 个网页标题作为词典索引，指向 title 第一次出现的排名，以及 title 在前 10 个中的计数 dicCount。考虑到搜索结果排名的重要性和检索计算量，这里只采取完整形式的前 10 个（而非 20 个）搜索结果。将上一步得到的缩略语搜索的前 20 个标题，一一在标题词典中查询并根据完整形式搜索结果的排序 rank 赋予比对结果一定的权值 1/rank。得到公式(3.3)。

$$\text{titleCompare (abbr)} = \sum_{j=1}^{20} \text{dicCount}(\text{title}_j[\text{abbr}]) / \text{rank}(\text{title}_j[\text{abbr}]) \tag{3.3}$$

（2）网页地址 URL 对比

考虑到一些相似网页隶属于同一网站的不同子集，因此对 URL 先进行过滤，只考虑网站地址。计算方式与网页标题类似，将(3.3)中的 title 替换为 URL 即可。

3.3 基于线索词的搜索

经过上一步的实验发现，很多缩略语形式得到的结果是帖子、博文等较为不规整的资源，不包含全称，也与搜索全称的结果相去甚远。比如，分别搜索“俄罗斯国际航空公司”和搜候选“俄航”的前 20 条结果，二者相似度为 0，且“俄航”的摘要结果中也没有出现完整形式。

于是，我们使用了带特殊线索词的搜索形式。如，在百度中搜索<完整形式> 简称 <缩略候选>，获取前 20 条搜索结果。针对“俄罗斯国际航空公司”和“俄航”，可以搜索“俄罗斯国际航空公司 简称 俄航”。在返回结果的摘要中即可以看到“俄罗斯国际航空公司（Aerolot，简称俄航）”这样的句式。考虑到不同的表达形式，包括中间添加标点符号或英文等，归纳为正则表达式后进行匹配，即可得到带线索词的得分策略：

$$\text{cue (abbr)} = \sum_{j=1}^{20} \text{regMatch}(\text{snippets}_j[\text{full,abbr}]) \tag{3.4}$$

3.4 共现现象

为了进一步分析网络资源中缩略语和完整形式间的关系，我们也考虑了二者的共现现象，并根据共现进行搜索，其形式为“<完整形式> <缩略形式>”，利用搜索得到的结果数量和摘要信息分别进行打分。

（1）结果数量

一般情况下，一对完整形式和缩略候选的搜索结果较多，说明二者的共现现象更为明显，也意味着二者之间的关系更为密切。但是也有例外的现象。比如一个错误的缩略形式可能是一个单字，或是作为完整形式一部分的一个常用词，那么搜索<完整形式> <缩略形式>就可能出现非常多的结果而误导打分。如搜索“影片来源”候选的结果数量：

表 3.2 “影片来源”和候选的单独搜索结果数量和共现搜索的结果数量

共现排名	候选	缩略结果	共现结果	得分	共现排名	候选	缩略结果	共现结果	得分
1	片	108	56800000	0	5	片源	9160000	1850000	1850000
2	影片	108	29500000	0	6	片来源	212000	856000	856000
3	影	108	11800000	0	7	片来	5270000	331000	331000
4	源	108	2730000	0	8	影片源	62900	1010	1010

表 3.2 中完整形式“影片来源”的候选“片”，“影片”，“影”，“源”，“片源”都是错误的候选，

但与“影片来源”组合却得到比正确形式<影片来源><片源>得到更多的搜索结果。因此进行判定，检验第一步对缩略语单独搜索获得的结果数量 abbrResultNum，如果等于 10^8（百度把超过 1 亿条的都算做 1 亿条），且大于单独搜索完整形式的结果数目 fullResultNum，则得分为 0。这样例子中错误的常用词和单字就被过滤了，可以得到正确的候选“片源”。计算公式见(3.5)。

$$\text{CoOccurNum(abbr)} = \begin{cases} 0, & \text{if resultNum(abbr)}=10^8 \text{ and resultNum(full)}<10^8 \\ \text{resultNum(full,abbr)}, & \text{else} \end{cases} \tag{3.5}$$

（2）摘要信息

我们也按公式(3.6)统计了在返回结果的摘要中是否同时出现完整形式和缩略形式。

$$\text{CoOccurCount (abbr)} = \sum_{j=1}^{20} \text{ifOccur}(\text{snippet}_j[\text{abbr}], \text{abbr}, \text{full}) \tag{3.6}$$

3.5 综合

由上面的四种方法可以得到 9 个估值。分别对每个统计值 count 进行归一化：

$$\text{Norm}(\text{abbr}_i) = \begin{cases} 0, & \sum_{i=1}^{10} \text{Count}(\text{abbr}_i) = 0 \\ \text{Count}(\text{abbr}_i) / \sum_{i=1}^{10} \text{Count}(\text{abbr}_i), & \sum_{i=1}^{10} \text{Count}(\text{abbr}_i) \neq 0 \end{cases} \tag{3.7}$$

因此每个值的范围都是[0,1]，经过参数为 1 的平滑处理，与 CRF 得到的概率值相乘得到最终的分值：

$$\text{Score} = \text{CRF} \times (1+\text{titleAbbr}) \times (1+\text{titleFull}) \times (1+\text{urlCompare}) \times \ldots \times (1+\text{CoOccurCount}) \tag{3.8}$$

在综合计算后，便可以根据值的大小按降序排列，排在最前面的结果即为最优结果。

4 实验与分析

本文使用了北京大学计算语言学研究所收集的 8350 对完整形式与缩略语对照表进行测试。将对照表随机按照 9:1 的比例分割成了训练集和测试集（训练集和测试集的条目完全不相交）。

对于结果，依照[4]采用了两种评测方法——完全匹配正确率以及 Top-k 最佳覆盖率。完全匹配率即为正确预测出缩略语的测试用例占所有测试用例的比例。Top-k 覆盖率评测即是当系统返回的 k 个最佳候选中如果其中任何一个和答案完全相同，就算正确答案，因此 Top-k 覆盖率即为成功覆盖正确答案的测试用例数占所有测试用例的比例。

4.1 第一步：CRF 序列标注

本文利用 CRF++工具[1]建立序列标注模型，对测试数据进行序列标注，得到 Top-k 覆盖率的结果如表 3.1 所示，并选择前 10 个结果作为重排的候选。

4.2 第二步：利用网络资源重排

按 CRF 条件概率排列的前 10 个候选中，分别使用各个统计值与 CRF 结合得到结果。最后进行全部特征的组合打分。结果如表 4.1 所示。

结果显示单独使用各方法都有不同程度的提高，其中使用方法 1（缩略语搜索）以及方法 4（共现法）中的摘要，都有较好的结果。而方法 1 中的标题，方法 2（对比法）中的 URL 以及方法 4 的搜索数量对结果的提升效果不明显。可能由于摘要提供的内容更加丰富，标题和 URL 具有一定的局限性。最终合并所有的影响因子的效果最好，比单纯用 CRF 在准确率上提高了约 5%。

[1] CRF++工具见：http://crfpp.sourceforge.net/

表 4.1　使用不同统计方法打分得到的完全匹配正确率和 Top-k 覆盖率

	序列标注	缩略语		对比法		线索词	共现		全部
方法	CRF	Title	Snippet	Url	Title	Cue	Snippet	ResultNum	combine
Top1	56.87%	56.98%	58.23%	56.98%	57.32%	57.78%	58.12%	56.98%	61.75%
Top2	67.42%	68.56%	69.69%	68.67%	68.56%	68.33%	69.01%	67.76%	72.76%
Top3	74.91%	76.05%	77.30%	75.37%	75.60%	75.71%	75.60%	75.14%	79.11%
Top5	81.27%	82.29%	83.43%	81.84%	81.95%	81.95%	82.86%	81.38%	85.70%

5　结论

本文所提出的方法结合了机器学习和网络信息验证两个过程。利用 CRF 模型得到第一步结果。然后利用搜索引擎返回的结果信息进行验证，对第一步的结果进行纠正。本文对缩略语搜索、对比法、线索词法、共现法四种方法进行统计打分，从而使重排序得到了较好的结果。

由于网络资源较复杂，因此统计模型的提升效果不是特别明显，进一步的工作可以涉及优化搜索结果的验证模型，探索新的统计方法，以及设计更加合理的重排序算法。另一方面，从完整形式向缩略语转换中，有很多因素共同起作用，选择更加合理的特征也是要深入分析和探讨的。

参 考 文 献

[1] Manuel Zahariev. ACRONYMS[D]. PHD thesis, Simon Fraser University, 2004.

[2] J.Chang and Y.Lai. A Preliminary Study on Probabilistic Models for Chinese Abbreviations[C]. In *Proceedings of the Third SIGHAN Workshop on Chinese Language Learning*, 2004, Barcelona, Spain.

[3] Jing-Shin Chang and Wei-Lun Teng, Mining Atomic Chinese Abbreviation Pairs with a Probabilistic Single Character Word Recovery Model. In the Proceedings of SIGHAN Workshop on Chinese Language Processing, 2006.

[4] Xu Sun, Hou-Feng Wang, Bo Wang. Predicting Chinese Abbreviations from Definitions: An Empirical Learning Approach Using Support Vector Regression[J]. Journalof Computer Science and Technology. Jul. 2008, 23(4): 602-611.

[5] Xu Sun, Naoaki Okazaki, Jun'ichi Tsujii. Robust Approach to Abbreviating Terms:A Discriminative Latent Variable Model with Global Information[C]. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Suntec, Singapore: Association for Computational Linguistics, 2009, 905-913.

[6] Dong Yang,Yi-Cheng Pan, Sadaoki Furui. Automatic Chinese Abbreviation Generation Using Conditional Random Field[C]. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. Boulder, Colorado: Association for Computational Linguistics, 2009, 273-276.

[7] 计峰，高沫，邱锡鹏，黄萱菁. 中文机构名简称的自动生成研究[C]. 见：孙茂松，陈群秀主编，《中国计算语言学研究前沿进展》，清华大学出版社，2009.

[8] Guang Jiang, Cao Gungen, Sui Yuefei, Han Lu, and Shi Wang. A General Approach to Extracting Full Names and Abbreviations for Chinese Entities from the Web. Intelligent Information Processing 2010: 271-280.

[9] Hui Liu, Yuquan Chen, Lei Liu. Automatic Expansion of Chinese Abbreviations by Web Mining[C]. In *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence*. LNAI 5855, 2009, Springer.

[10] 谢丽星，孙茂松，佟子健，王灿辉. 基于用户查询日志和锚文字的汉语缩略语识别[C]. 见：孙茂松，陈群秀主编，《中国计算语言学研究前沿进展》，清华大学出版社，2009.