联合无监督词聚类的递归神经网络语言模型①

刘 章, 陈小平

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

摘 要: 研究表明, 在递归神经网络语言模型的输入层加入词性标注信息, 可以显著提高模型的效果. 但使用词性标注需要手工标注的数据训练, 耗费大量的人力物力, 并且额外的标注器增加了模型的复杂性. 为了解决上述问题, 本文尝试将布朗词聚类的结果代替词性标注信息加入到递归神经网络语言模型输入层. 实验显示, 在 Penn Treebank 语料上, 加入布朗词类信息的递归神经网络语言模型相比原递归神经网络语言模型困惑度下降 8~9%.

关键词: 递归神经网络; 词性标注; 布朗词聚类; 语言模型

Using Word Clustering to Improve Recurrent Neural Network Language Model

LIU Zhang, CHEN Xiao-Ping

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Previous studies proved that, adding part of speech tag information to the input layer of neural language model, can improve the performance significantly. But part of speech tag need hand-annotated data to train the tag model, which consumes a lot and the extra tagger also makes the model more complicated. To solve the problem, this article propose adding the results of brown clustering, instead of part of speech tag information to the input layer of the recurrent network language model. In the Penn Treebank corpus, the relative improvement over the original recurrent neural network language model reaches 8%~9%.

Keywords: recurrent neural network language model; part of speech tag; brown clustering; language model

1 引言

语言模型是语音识别、机器翻译、光学字符识别等系统的关键组成部分. 这些系统在人类生产生活中发挥日益重要的作用, 因此提高语言模型的性能具有非常重要的意义. N-gram 是一种简单有效的语言模型, 在过去几十年实际应用中一直占有主导地位. 但 N-gram 有其自身的缺点, N-gram 是离散空间上的语言模型, 数据稀疏性导致 N-gram 很难获得较好的泛化能力, 并且 N-gram 的独立性假设使其无法利用长距离的信息. 为了弥补 N-gram 的不足, 神经网络语言模型, 包括前馈神经网络语言模型[1]和递归神经网络语言模型[2], 被相继提出. 神经网络语言模型通过连续的向量空间来提高模型的泛化能力, 其中递归神经网络语言模型, 可以通过递归的隐含层学习整个历史信息, 它因此被视作一种具有深度架构的语言模型.

神经网络语言模型输入层通常只使用词的本身作为输入. 然而实验表明, 添加词性标注信息到输入层, 可以有效的提高神经网络语言模型的效果. 这是由于词性标注带有语义上的信息, 可以使模型捕获更长距离词的依赖关系, 文献[5]和[6]中曾分别把词性标注信息加入前馈神经网络语言模型和递归神经网络语言模型的输入层, 并分别观察到模型效果的显著提升. 然而使用词性标注信息有明显的缺点, 标注模型的训练依赖手工标注的数据, 手工进行标注数据的代价太大, 这增加了将模型用于新语言或领域语言的困难. 另外词性标注需要额外的标注器(一般基于隐马尔科夫模型或向量随机场模型), 在递归神经网络语言模型训练或测试的过程中实时标注, 这也增加了模型的复杂性.

① 收稿时间:2013-09-12;收到修改稿时间:2013-11-11

布朗词聚类方法是一种用来聚类相似单词的无监督凝聚式层次聚类方法, 无需手工标注的数据进行训练, 并且聚类后的结果包含丰富的语义信息. 很多自然语言处理领域的任务都使用其提高效果, 例如[3]中曾利用布朗词聚类提高命名实体识别的准确率. 将布朗聚类信息融合到递归神经网络中十分容易, 只需构造词典时, 将每个词对应的类信息加入到词结构中即可, 计算量不会有明显增加.

在本文中, 我们把布朗词聚类后的结果信息代替词性标注信息, 加入到递归神经网络语言模型的输入层, 在 Penn TreeBank 数据集上, 取得相较原递归神经网络语言模型困惑度下降 8~9%的效果.

2 背景知识介绍

2.1 语言模型与 N-gram

语言模型的目的, 是给语言中一个字符序列赋一个值来表示该字符序列出现的可能性. 给定字符序列

$$W = w_1 w_2 ... w_m = W_1^m \tag{1}$$

语言模型赋给 W 的概率可以通过公式(2)来计算.

$$P(W) = P(W_1^m) = \prod_{i=1}^{m} P(w_i \mid w_1^{i-1}) \tag{2}$$

N-gram 假设要预测的词只依赖前 n-1 个词, 即 $P(w_i \mid w_1^{i-1}) = P(w_i \mid w_{i-n+1}^{i-1})$. 通常情况下 n 被限制到 3 或 4, 即使这样, 假如词汇量大小为 50K, 一个 4-gram 也需要估计 $50K^2 bigrams$, $50K^3 trigrams$, $50K^4 4-grams$. 由于数据的稀疏性, 很多参数在训练阶段会被忽略. 因此, 很多低频或未出现的 n-grams 得不到很好的估计, 当词汇量增加的时候, 这种现象尤为严重.

2.2 布朗词聚类方法

布朗词聚类法是一种无监督凝聚式层次聚类算法, 它的输入是单词序列构成的文本, 输出是单词的分类或一个叶子节点表示单词的二叉树, 其中每个子树可以视作一个词类.

布朗词聚类法是一种不断迭代聚合的方法, 聚类刚开始的时候, 单词集合中的每个单词都代表一个簇, 我们将每两个簇都试着合并一次, 然后观察划分度量在每个合并上的结果, 找到结果最好的那次合并, 并依照这次合并形成下一次迭代所需的簇集. 这里划分度量是关键部分, 下面介绍划分度量和详细的聚类算法.

2.2.1 划分度量

用 V 表示单词集合, $n(x)$ 表示单词或词类 x 在语料中出现的次数, $n(x,y)$ 表示以单词或词类 x 开头并以单词或词类 y 结尾的组合在语料中出现的次数. 划分 $C :\to \{1,2,...k\}$ 表示单词集合的一个 k 分类.

定义:

$$Qulatily(C) = \sum_{c,c'} \frac{n(c,c')}{n} \log \frac{n(c,c')n}{n(c)n(c')} + \sum_{w'} \frac{n(w')}{n} \log \frac{n(w')}{n} \tag{3}$$

表示划分 C 的质量.

这里的 $Quality(C)$ 实际上是划分 C 确定的 classed based bigram 在输入文本 $w_1 w_2 \ldots w_n$ 上的概率值.

我们将方程(2)用以下各项表示:

$$P(c,c') = \frac{n(c,c')}{n} \tag{4}$$

$$P(w) = \frac{n(w)}{n} \tag{5}$$

$$P(c) = \frac{n(c)}{n} \tag{6}$$

$$I(C) = \sum_{c,c'} P(c,c') \log \frac{P(c,c')}{P(c)P(c')} \tag{7}$$

$$H = -\sum_{w} P(w) \log P(w) \tag{8}$$

可得:

$$Quality(C) = P(c,c') \log \frac{P(c,c')}{P(c)P(c')} + P(w) \log P(w)$$

$$= I(c) - H \tag{9}$$

这里 $I(c)$ 是相邻簇的互信息, H 是单词分布的熵. 对于确定的输入文本 H 不依赖划分 C, 因此最大化 $Quality(C)$, 实际上是最大化 C 确定的簇间的互信息 $I(c)$.

2.2.2 聚类算法

聚类的过程可以描述为, 每次聚合时选择最大化 $Quality(C)$ 的一对簇进行合并, 算法描述如图 1.

布朗词聚类的结果属于硬聚类结果(每个单词只对应到一个相应的类中). 词法或语义相似的单词, 会因为它们在文本中相似的位置, 被聚为同一类, 这使得布朗聚类后的结果包含丰富的语义信息.

初始化：将 V 中每个单词对应到一个簇中

以下步骤运行 $|V|-k$ 次：

1.对当前划分中的每对簇 $(c_i,c_j), i\neq j$，计算它们暂时合并后所对应的 $Quality(C_{temp}^{ij})$。

2.选取 (c_i,c_j) 使其对应的 $Quality(C_{temp}^{ij})$ 最大，合并 (c_i,c_j) 形成新的划分 C。

图 1 聚类算法描述

3 联合布朗词聚类信息的递归神经网络语言模型

3.1 模型框架

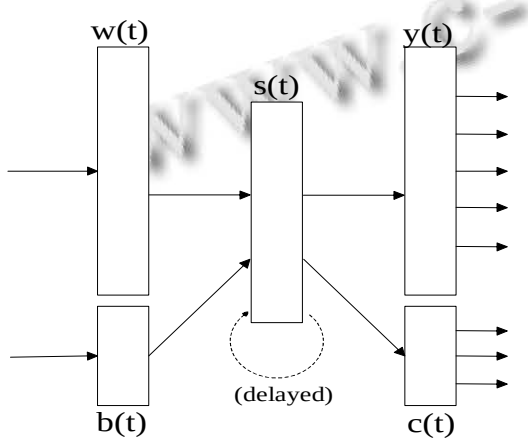

图 2 联合布朗聚类信息的递归神经网络语言模型结构

图 2 中式本文中所使用的递归神经网络语言模型结构. 如图所示, 它包含一个输入层, 一个隐含层和一个输出层.

输入层包含三个信息, $w(t)$、$b(t)$ 和 $s(t-1)$, $w(t)$ 表示词汇输入, $b(t)$ 表示 $w(t)$ 所对应的布朗类别信息. $w(t)$ 和 $b(t)$ 使用的都是“1 of n”编码, 例如,假设单词 $w(k)=tiger$ 在词汇表 V 中的位置是 k, 则用第 k 位置 1 其它位置 0 的一个 $|V|$ 维向量来表示 $w(k)=tiger$. $s(t-1)$ 表示模型上一次的状态信息, 即上一次的隐含层的拷贝.

因此输入为: $x(t)=[w(t),b(t),s(t-1)]$.

隐含层的神经元使用的是 sigmoid 激活函数.

$$s_j(t)=f(\sum_i x_i(t)u_{ji}) \tag{10}$$

其中 u_{ji} 是输入层和隐含层之间对应神经元连接的权重.

这里 $f(z)$ 是 sigmoid 激活函数

$$f(z)=\frac{1}{1+e^{-z}} \tag{11}$$

输出层分为两部分, 一部分为类别神经元, 另一部分为单词神经元. 给定当前单词 w_{i-1}, w_{i-1} 的布朗词类 b_{i-1}, 模型上次的隐含层状态 s_{i-1}, 计算下个单词 w_i 出现的概率公式可以分解为:

$$p(w_i|w_{i-1},b_{i-1},s_{i-1})=p(c_i|s_i)p_{c_i}(w_i|s_i) \tag{12}$$

计算输出时分为两步, 首先计算类别神经元:

$$c_l(t)=g(\sum_j s_j w_{jl}), l\in[0,|C|] \tag{13}$$

$|C|$ 为输出层类大小. 其中 $g(z)$ 是归一化函数.

$$g(z_m)=\frac{e^{z_m}}{\sum_k e^{z_k}} \tag{14}$$

然后计算单词神经元:

$$y_o(t)=g(\sum_j s_j w_{jo}), o\in W(c(w_i)) \tag{15}$$

其中 $W(c(w_i))$ 表示 w_i 所属的类的单词集合.

3.2 布朗词聚类信息的加入

在训练递归神经网络语言模型之前, 我们先用训练集对词表中的单词进行布朗聚类, 这样我们重新得到一个词表, 这里词表中每个单词都有一个布朗类别信息. 当对递归神经网络语言模型训练和计算时, 对每个输入的单词, 查询词表中的类别信息并一起加入输入层进行计算.

这里举例说明布朗词类信息能明显提高递归神经网络语言模型的效果的原因: 假设训练集中出现足够多关于 Monday, Tuesday, …, Friday 的句子, 那么布朗词聚类会把它们聚为一个类. 再假设测试集需要预测 it’s Friday 的概率, 但训练集中没出现 it’s Friday, 却有 it’s Monday 出现. 这时布朗类信息具有较强的指向性, 可以帮助正确的预测下个单词, 从而帮助模型取得更低的困惑度.

3.3 输出层分类

递归神经网络语言模型一个显著的缺点, 是它的高计算复杂性. 未采用输出层分类的递归神经网络语言模型计算复杂度为 $O=(2+H)\times H\times\tau+H\times V$. 其中 H 是

隐含层的大小, V 是词汇表大小, τ 是后传步数. 一般来说 $H \ll V$, 所以计算的瓶颈在于输出层和隐含层之间. 为了减少计算复杂度将输出层分为类部分和词部分, 这样计算复杂度可以减至:

$$O = (2+H)\times H\times\tau + H\times C \quad (16)$$

其中 H 是隐含层神经元数量, C 是输出层类的数量.

基于词频分类的基本思想是把单词根据词频均匀的分给各类, 换种说法是把单词按照它们的 unigram 概率分类. 实验中我们采用基于词频的分类方法.

```
vocab：单词表（已经按照词频从大到小排序）
vocab[i].cn：第 i 个单词的出现频率
vocab[i].classid：第 i 个单词的类标号
class_size：类大小
    初始化：df=0,a=0,b=0;
    for (i=0;i<|V|;i++)   b+=vocab[i].cn;
    for(i=0;i<|V|;i++){
    df+=vocab[i].cn/b;
    if(df>1) df=1;
    if(df>(a+1)/nclass){
    vocab[i].classid=a;
    if(a<class_size-1) a++;
    }
    else{
    vocab[i].classid=a;
    }
```

图 3 词频分类方法流程代码

3.4 训练

训练递归神经网络语言模型, 需要学习的参数有输入层和隐含层之间的权重矩阵 U, 以及隐含层与输出层之间的权重 W. 训练采用截断式时间后传算法(BPTT), BPTT 可以看作标准的后传算法在递归神经网络上的扩展. 使用截断式时间后传算法, 网络在学习时可以记住指定数目历史隐含层所包含的信息. 文献[4]中实验显示, 4~5 后传步数即可充分利用历史信息, 因此实验中我们采用的后传步数为 5.

训练算法总流程如下:

1. 布朗词聚类, 用训练集对词表中的单词进行聚类, 获得每个单词的类别。
2. 训练递归神经网络得到参数 U 和 W, 输入层传入的是单词和其对应的类别信息, 使用 BPTT 训练。

图 4 训练算法总流程

4 实验

4.1 实验准备

4.1.1 实验环境

实验的系统环境为 linux 3.2.0, 编译环境为 gcc 4.6.4. 硬件配置信息: CPU: 4 Intel(R) Core(TM) i5-2300 CPU @ 2.80GHz; 内存大小: 4GB.

4.1.2 语料选用与处理

实验采用的数据集是被语言模型领域研究者广泛使用的 Penn Treebank 数据集. Penn Treebank 数据集一共包含 24 个部分. 实验中采用的数据预处理方法如下: 将数据集的 0-20 部分用来作为训练集(930k 个单词), 21-22 作为验证集(74k 个单词), 23-24 作为测试集(82k 个单词). 词汇量限制为 10001 个, 未登录词统一用 <unk>表示.

4.1.3 实验评价标准

在众多语言模型评测标准中, 困惑度是最方便也是最为语言模型领域广泛采用的评价标准, 本文中采用困惑度(PPL)来评价模型的效果. 给定语言 L 的样本 $W = w_1 w_2 \ldots w_n$, 其困惑度定义为:

$$PPL(W) = \sqrt[K]{\prod_{i=1}^{K}\frac{1}{P(w_i \mid w_1 \ldots w_{i-1})}} = 2^{-\frac{1}{K}\sum_{i=1}^{K}\log_2 P(w_i \mid w_1 \ldots w_{i-1})} \quad (17)$$

模型赋给 W 的 PPL 值越高, 说明预测 W 出现的概率越低, 说明模型不能够较好的拟合该样本.

4.1.4 自由参数的处理

递归神经网络语言模型(RNNLM)与联合布朗词聚类的递归神经网络语言模型(bRNNLM)共同需要处理的自由参数有: 输出层类数、后传步数, 和隐含层的大小. 一般来说, 选取的输出层类数的大小, 应尽量使模型在效果没有明显损失的情况下, 能够充分降低计算所需时间. 在参照以前研究者的实验配置后, 本文在实验中把输出层类数设置为 300, 后传步数 bptt

设为 5(参见 3.4). 隐含层的增大会明显增大训练所需要的时间, 因此实验中的隐含层大小设在 0 至 500 之间.

4.2 实验及分析

实验比较了递归神经网络语言模型(RNNLM)与联合布朗词聚类的递归神经网络语言模型(bRNNLM), 在隐含层大小不同时的表现, 以及相同参数设置下的收敛速度. 另外, 实验评测了不同输入层的布朗词类数大小对效果的影响.

4.2.1 隐含层大小对结果的影响

这里 RNNLM 和 bRNNLM 的输出层类数设为 300, 后传步数 bptt 设为 5, bRNNLM 输入层类数大小设为 300. 从表 1 中可以看出相同条件下 bRNNLM 困惑度一直低于 RNNLM. 当隐含层大小为 400 时, RNNLM 模型对应的 PPL 为 140.14, 而 bRNNLM 对应的 PPL 只有 128.42, 当隐含层数目逐渐增大时, PPL 下降的百分比趋于 8~9%之间. 可见 bRNNLM 相对 RNNLM 有明显的下降, 但下降的幅度略低于加入词性标注信息的表现(PPL 下降 10%左右), 这可能是因为词性标注可以对多义词有不同的标注, 而布朗聚类是硬分类, 每个单词只对应到一个类中.

表 1 不同隐含层下 RNNLM 和 bRNNLM 的表现

	RNNLM	bRNNLM	PPL 下降百分比
50	165.35	152.37	7.9%
100	149.04	139.29	6.5%
200	144.96	132.33	8.7%
300	141.09	129.34	8.3%
400	140.14	128.42	8.4%

4.2.2 收敛速度比较

图 5 是隐含层大小为 300 时, RNNLM 和 bRNNLM 的收敛情况对比, 图的横轴是迭代周期, 纵轴是模型在验证集上的熵. 从图中可以看出, 在相同的迭代周期, bRNNLM 在验证集上的熵值始终要低于 RNNLM, 而且 bRNNLM 只需要 11 个迭代周期即可收敛, RNNLM 却需要 14 个迭代周期才最终收敛.

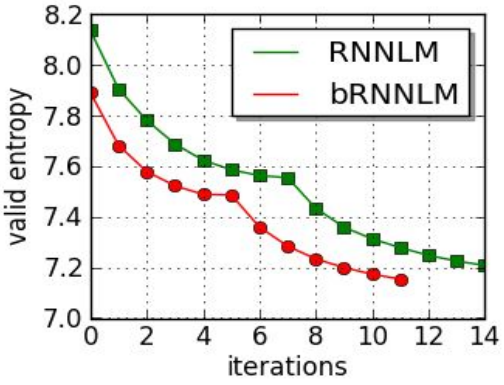

图 5 收敛比较

4.2.3 布朗类数大小对结果的影响

相比词性标注只有固定的词性数目, 布朗词聚类可以指定类数的值. 为了验证词类数目大小对效果的影响, 本文在[0-500]区间内计算了不同大小词类数对应的 PPL, 这里 bRNNLM 的隐含层大小设为 200, 后传步数设为 5, 输出层类数设为 300. 从图 6 中可以看到, PPL 从没加入任何类信息的 145 到词类数目增加为 10 时, 已经急剧下降至 133 左右. 词类数继续增加时, PPL 已经没有明显的变化, 联合图 7, 词类数目在区间[10-500]时, PPL 的值一直稳定在 130-134, 这说明输入层过多的布朗词类数对提高模型的效果没有帮助.

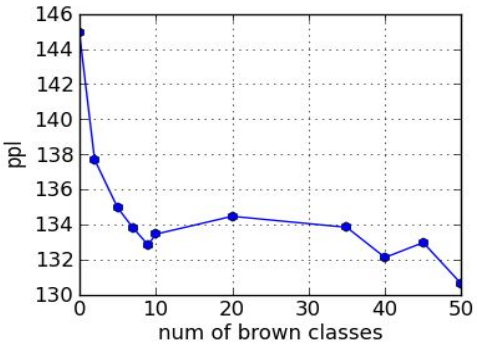

图 6 词类数目[0-50]对 PPL 的影响

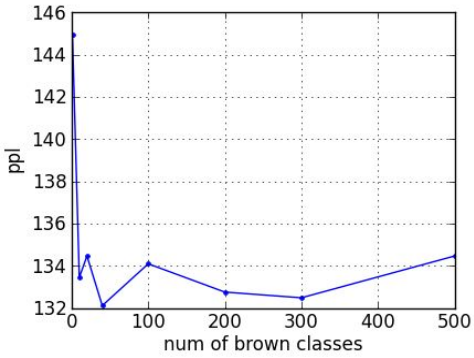

图 7 词类数目[0-500]对 PPL 的影响

5 结论与展望

目前，递归神经网络语言模型是语言模型领域最具潜力的语言模型框架，它在很多场合的表现都超出了传统的语言模型. 输入层加入词性标注信息的递归神经网络语言模型有更好的效果，但也有明显的不足. 针对词性标注需要手工数据进行训练的缺点，我们提出了使用布朗词聚类后的结果代替词性标注信息，加入输入层来提高递归神经网络语言模型效果的方法，这就节省了大量的人力物力. 我们在Penn TreeBank数据集上进行了实验，结果显示困惑度相比原递归神经网络语言模型下降了8~9%.

我们在实验中还发现，过多的布朗词类数对提高递归神经网络语言模型的效果，并没有明显的帮助，少量的类数即可充分的降低困惑度. 另外，布朗聚类是一种硬聚类方法，即每个单词只分配到一个对应的类中，这样即使是多义词也只有一个类别特征. 针对这个问题，下一步考虑改进布朗聚类方法，使其可以进行软分类(即一个词可以对应到多个类中)，然后测试效果.

参考文献

1 Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model. The Journal of Machine Learning Research, 2003, 3: 1137−1155.

2 Mikolov T, Karafiát M, Burget L, Cernocky JH, Khudanpur S. Recurrent neural network based language model. Proc. of Interspeech. 2010. 1045−1048.

3 Kombrink S, Mikolov T, Karafiát M, Burget L. Recurrent neural network based language modeling in meeting recognition. Proc. of Interspeech. 2011. 2877−2880.

4 Mikolov T, Kombrink S, Burget L, Cernocky JH, Khudanpur S. Extensions of recurrent neural network language model. Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE. 2011. 5528−5531.

5 Emami A, Jelinek F. A neural syntactic language model. Machine Learning, 2005, 60(1-3):195−227.

6 Yamamoto YWXLH, Matsuda S, Kashioka CHH. Factored language model based on recurrent neural network. Proc. of COLING. 2012. 2835−2850.

7 Si Y, Guo Y, Liu Y, Pan J, Yan Y. Impact of word classing on recurrent neural network language model. Intelligent Systems (GCIS), 2012 Third Global Congress on. IEEE. 2012. 100−103.

8 Miller S, Guinness J, Zamanian A. Name tagging with word clusters and discriminative training. Proc. of HLT-NAACL, 2004, 4: 337−342.

9 Turian J, Ratinov L, Bengio Y. Word representations: A simple and general method for semi-supervised learning. Proc. of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics. 2010. 384−394.

10 Brown PF, Desouza PV, Mercer RL, Pietra VJD, Lai JC. Class-based n-gram models of natural language. Computational Linguistics, 1992, 18(4): 467−479.

11 Lau R, Rosenfeld R, Roukos S. Trigger-based language models: A maximum entropy approach. Acoustics, Speech, and Signal Processing, 1993. ICASSP-93, 1993 IEEE International Conference on. IEEE. 1993, 2. 45−48.