中图分类号：TP391
论 文 编 号：10006SY1506330

北京航空航天大學

硕士学位论文

基于神经网络的高性能语言建模技术研究

作 者 姓 名　姜楠
学 科 专 业　计算机应用技术
指 导 教 师　荣文戈　副教授
培 养 院 系　计算机学院

Nerual Network Based Highly Efficient Language Modeling

A Dissertation Submitted for the Degree of Master

Candidate: Jiang Nan

Supervisor: Associate Prof. Rong Wenge

School of Computer Science & Engineering

Beihang University, Beijing, China

中图分类号：**TP391**

论 文 编 号：**10006SY1506330**

硕 士 学 位 论 文

基于神经网络的高性能语言建模技术研究

作 者 姓 名 姜楠　　申请学位级别 工学硕士

指导教师姓名 荣文戈　　职　　称 副教授

学 科 专 业 计算机应用技术　　研 究 方 向 自然语言处理

学 习 时 间 自 2015 年 09 月 01 日　起　　至 2018 年 月 日止

论文提交日期 2018 年 月 日　　论文答辩日期 2018 年 月 日

学位授予单位 北京航空航天大学　　学位授予日期 2018 年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期：　　年　　月　　日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期：　　年　　月　　日

指导教师签名：_____ 日期：　　年　　月　　日

摘　要

随着深度学习技术的发展，出现了丰富多样的基于神经网络的自然语言模型，这些神经网络模型一般都是通过学习大规模语料库来调整自身的参数分布，以提高模型在相关任务上的性能和精度。然而，对于目前的一些较大规模应用而言，语言模型的训练过程还显得较为缓慢，效率需要进一步提高。导致计算缓慢的原因是这些模型在训练和测试的时候，通常需要预测整个词表的单词从而选择出最佳的候选单词，该步骤占用了绝大部分计算资源和时间，一般称为大词表问题。为了应对这个挑战，研究人员提出了各种不同类型的方案，如：单词拆分算法、基于抽样的近似算法和层次概率模型等，其中层次概率模型只关注局部概率，可以很大程度降低训练和测试过程中所占用的计算资源，因此获得了广泛的关注。

针对大词表问题，为了减少模型的计算时间并提高模型效率，本论文探讨了基于二叉树和基于类别的两种层次概率模型，并设计了相应的改进策略，其主要改进要点包括：1）提出了一种改进的层次模型的编码策略，引入基于树和基于类的并行度更高的损失函数；2）针对层次化模型对于聚类算法的敏感性问题，对基于 N-gram、句法信息和语义信息的不同聚类算法对各种层次概率模型产生的影响进行了分析；3）对多种不同的结构化预测算法进行了分析，同时分析了语言模型在排序和打分两个任务中的选择策略。

在实验验证环节，本文在语言建模任务上进行了评测，采用 WikiText-2, WikiText-103 和 One Billion Words 数据集作为实验的基准数据。在模型的评价指标方面，本文不仅考虑了传统的困惑度误差，还将文本比较中经常用到的编辑距离（单词错误率）作为一个重要的比较指标。除此之外，还比较了不同模型之间的训练和测试阶段内存占用量，以及相同数据量情况下模型的计算速度和模型的收敛速度等衡量指标。实验结果表明，与其他概率归一化方法相比，加速比提高，得到更高效的树聚类，与其他基于抽样的优化相比，性能相对较好。

关键词：层次概率，层次聚类，神经语言模型，递归神经网络，自然语言处理

Abstract

Recent decades has witnessed great progress and achievements, in the filed of natural language processing. Variants of neural networks based architecture have been proposed and successfully applied to the neural language models, neural acoustic models, neural translation model and etc. These neural models can leverage knowledge in texts by learning parameters from massive online corpora, and abundant cases are presented over various text tasks, like sentence classification and word vector learning. While they are extremely slow for the real-world challenge, as they try to predict candidates from a large vocabulary, in the process of training and inference. As an alternative to vocabulary truncation and sampling-based approximation methods, we explore the historical proposed tree-based and class-based hierarchical softmax methods.

In this research, aiming at reducing neural model's computational time as well as making them compatible with general purpose modern graphics processing units, we introduce a series of efficient and effective approaches and categorise our contributions as: a) Firstly, we reform their structural composition and introduce a compact tree-based loss function for the tree-based hierarchical softmax methods and class-based loss function for the corresponding class-based hierarchical softmax; b) Secondly, we discuss the impact of several ngram-based, syntactic and semantic clustering algorithms for the vanilla hierarchical softmax as these structural models are sensitive to the hierarchical clustering methods; c) Thirdly, we discuss possible inference algorithms for the hierarchical softmax variants, assuring the model can fetch presumable predictions under different circumstances.

Finally, Our experiments were carried out on language modelling tasks with standard benchmarks datasets, i.e., WikiText-2, WikiText-103 and One Billion Word datasets. Except for the traditional perplexity metric, we also extended our comparison over the word error rate and memory footprint and etc. Consist improvement with several intrinsic evaluation criterions: word error rate and perplexity, were also achieved over other conventional optimisation methods.

Key words: Hierarchical Softmax, Hierarchical Clustering, Neural Language Model, Recurrent Neural Network, Natural Language Processing.

目　　录

图　目

表　目

第一章　语言建模实验结果与分析

前面两章分别介绍了基于树状和类别的层次概率模型的具体建模过程。在本章中，为了比较所提出的这两种分层概率模型与已有的算法（Baselines）在计算效率和精确性上的差异，我们将在三个标准文本数据集上进行循环语言建模任务的实验研究和结果分析。本章将讨论分析不同并行层次概率算法的实验结果，还会与其他大词表问题的优化加速算法相互对比。接下来还将从效率、可扩展性、参数大小等方面，对这些已有的方法进行实验研究并作讨论分析。

1.1　实验设置

这一小节将介绍实验所需要用到的文本数据集，还有语言建模实验的两种主要的评测指标和实验模型的实际参数配置和训练过程。

1.1.1　实验数据集

本次实验所采用的数据集主要是依照两个目的选取的：1）首先为了便于实验复现和互相对比，需要在小数据集上反映出参数变化对模型的最后的效果产生的影响；2）同时还需要大数据集上，展示模型参数在最佳参数配置下，比较模型之间的最优结果。所以在本次实验中，我们选取了三个标准文本数据集：Wikitext-2，Wikitext-103 和 One Billion Word 数据集。如表 1 所示，表中列举了这三个数据集的相关统计指标，包括：单词数量、句子数量、词表大小和词表外单词的比例（Out-of-Vocabulary）。需要注意的是，词表大小不能改变，因为不同词表大小的模型之前理论上是没有互相比较的意义的，其次是由于接下来要计算的一个重要的评测指标就是和词表大小成负相关关系的。所以表 1 中展示的数据已经预先固定了，不会再做任何的修改，例如：不能对数据集做单词大小写变化，数词转换操作或者分词操作。

其中，对于 Wikitext-2 和 Wikitext-103 数据集，训练、验证和测试集都是预先划分固定的，并且其词汇表大小也已经被定义[1]。这两个数据集拥有相同的验证和测试集，而 Wikitext-103 的训练集则比 Wikitext-2 的训练集大得多。所以我们还可以间接测算出增

[1]https://metamind.io/research/the-wikitext-long-term-dependency-language-modeling-dataset/

表 **1 WikiText-2, WikiText-103** 和 **One Billion Words** 数据集统计指标

数据集	类型	文章数	句子数量	单词数量	词表大小	OOV （%）
Wikitext-2	训练集	600	36,718	2,088,628	33,278	2.6%
	验证集	60	3,760	217,646		
	测试集	60	4,358	245,569		
Wikitext-103	训练集	28,475	1,801,350	103,227,021	267,735	0.4%
	验证集	60	3,760	217,646		
	测试集	60	4,358	245,569		
One Billion Word	训练集	—	30,301,028	768,646,526	793,471	0.28%
	验证集	—	6,075	153,583		
	测试集	—	6,206	159,354		

大训练集对测试集上的提升效果表现。对于第三个 “One Billion Word” 数据集[2]，它是由之前机器翻译数据集里面的文本融合而成，文本数据也取自维基百科（Wikipedia）[3]。为了评价的标准性和实验的可互比性，官方还提供了一套数据预处理脚本，同时指定了该脚本所需要的 Perl 语言版本，可见要求极为严苛。因为对于语言模型来说，Perl 内置函数版本不同，处理出来的文本也略有不同，语言模型之间的结果相互比较就会变得没有意义的。通常来说，词表小的模型更占优势。当用官方提供的脚本处理完数据后，我们将 “./train/” 目录中的所有数据视为训练集，选择 “./holdout/” 目录下面的第一和第二个数据集作为相应的验证和测试集。这些数据集的详细统计指标在表 1 中进行了说明。

1.1.2　实验评价指标

在本次实验研究中，当各种实验模型在验证数据集上训练到收敛后，我们就需要评价不同模型的性能差异，针对语言模型的两个标准评估度量标准来揭示针对不同的优化方法的优劣：困惑度（Perplexity，PPL）和单词误差率（Word Error Rate，WER）。其中，PPL 是一个内在度量指标（Intrinsic Metric），代表了在不同语境中选择下一个候选单词时的困惑程度。语言模型困惑度较低，意味着在相同词表下拥有更好的可预测性。此外，在整个测试集上，PPL 数值是与平均负对数似然值（NLL）呈指数相关关系，这证明训

[2]http://www.statmt.org/lm-benchmark/

[3]英文维基百科主页：https://en.wikipedia.org/wiki/Main_Page/

练过程中模型直接优化了 PPL 评测指标，其数学定义如下所示：

$$\mathrm{PPL}(w_1, \cdots, w_T) = \sqrt[T]{\frac{1}{\prod_{t=1}^{T} p(w_t | w_{1:t-1})}} \tag{1.1}$$

此外，WER 表示单词的莱文斯坦距离（Levenshtein Distance），用于衡量参考句子（Reference，记作 r）和预测句子（Hypothesis，记作 h）之间的相似度。它是编辑距离的一种衍生类型，被定义为错误识别的单词（删除，插入，替换）占总单词的百分比[4]：

$$\mathrm{WER} = \frac{\text{插入的单词数} + \text{删除的单词数} + \text{替换的单词数}}{\text{全部单词数量}} \tag{1.2}$$

其数学形式的定义公式为：

$$d_{r,h}(i,j) = \begin{cases} \max(i,j) & \text{如果 } \min(i,j) = 0 \\ \min \begin{cases} d_{r,h}(i-1,j) + 1, \text{//插入的单词} \\ d_{r,h}(i,j-1) + 1, \text{//删除的单词} \\ d_{r,h}(i-1,j-1) + 1(r_i \neq h_j), \text{//替换的单词} \end{cases} & \text{否则} \end{cases} \tag{1.3}$$

其中 $1(r_i \neq h_j)$ 指代的是示性函数，当且仅当 $r_i = h_j$ 的时候取值为 0，否则该函数取值为 1。$d_{r,h}(i,j)$ 表示的是参考句子 r 的第一个 i 个字符与预测句子 r 的第一个 j 个字符之间的距离。同时莱文斯坦距离还具有度量的延展性关系（三角不等式），即：两个字符串的距离不大于分别与第三个字符串的距离之和：

$$d_{r,h} + d_{r,s} \geq d_{r,s} \tag{1.4}$$

其中 s 代表另外一个生成的句子，$d_{r,h}$ 表示的是句子 r 和句子 h 之间的编辑距离。

除了以上的定量的度量指标，训练时间效率，词汇可伸缩性和运行时内存消耗等定性度量指标也应该被视为衡量不同模型的重要衡量维度。因此，我们在实验中分别从 GPGPU 和 CPU 的理论和经验角度分析了不同优化算法在不同评估指标上的具体数值结果。最后，统计实验数值并分析在三个标准文本数据集上的最终结果。

1.1.3 模型训练和参数配置

接下来来介绍实验模型的参数设置和训练过程。在具体算法实现中，每个模型都是使用 Theano 框架实现，而且都运行在一个独立的 GPGPU 设备上，模型之间不互相干

[4]https://martin-thoma.com/word-error-rate-calculation

扰。GPGPU 设备具有 12GB 的显存（设备型号是 Nvidia K40m），保证能进行大矩阵乘法的运算，所以我们能测算出大词表问题的具体计算代价。然而我们发现，在实际试验中随着模型参数维数的增加，单个 GPGPU 显存资源被快速利用殆尽。

然后是针对数据集做必要的预处理。对于 wikitext-2 数据集，最大句子长度固定为 256；对于 wikitext-103 数据集，最大句子长度固定为 100；对于 OBW 数据集来说，最大句子长度固定为 50，因为第三个数据集的词表最大，需要占用更多的显存，所以句子长度有所缩减。对于长度超过阈值的那部分字符串将直接删除，由于测量的是语言模型的单词级损失（Word-Level Loss）而不是句子级的分数（Sentence-Level Loss），因此删除部分的那句子对模型训练的影响是很小的。

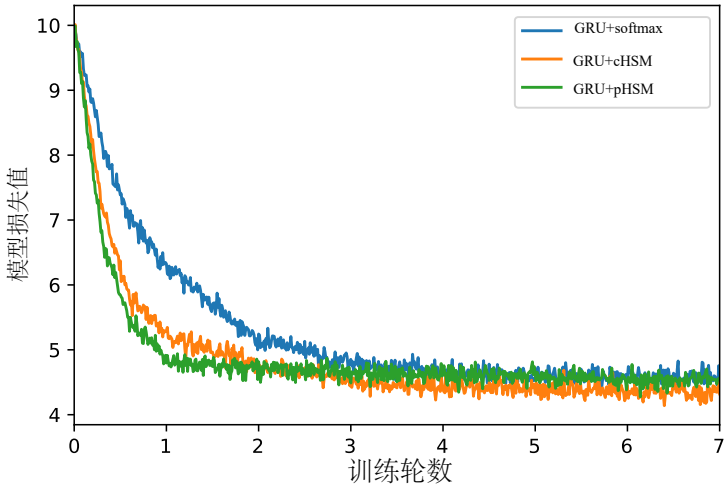

图 **1** 模型训练曲线图

另外在实验中，Adam 优化器（Optimizer）配合两种不同的学习率（Learning Rate）作为模型的优化函数，即设置 $\mathrm{lr} = 0.06$ 和 0.001。两种词表层级分解方法需要采用较大的学习率，而传统的 softmax 和采样近似方则需要采用较小的学习率。因为我们实验中发现，词表层次分解算法收敛很慢，需要配合更大的学习率。除此以外，每隔一定步数（$\mathrm{step} = 100$），学习率还需要逐渐缩小：$\mathrm{lr} \leftarrow \mathrm{lr} * 0.9$。

对于在 WikiText-2 数据集上运行的实验，我们在批处理大小（batch size）为 20 的 20 个时期内运行，直到我们观察到验证集上的最小 PPL。验证集上的损失数值约 4.8，这是最好的验证上的损失值。而对于 Wikitext-103 数据集，在训练集上优化参数需要大约 3-4 个轮数（Epochs），因为训练集大约比较小的一个大 50 倍。此外，训练集损失数值大约在 5.1 左右，因为在 wikitext-2 数据集上更高的原因是我们预测的词汇量要大得多，所以混淆程度（即训练损失）应该更高。虽然我们发现模型可以很容易地收敛到局部最

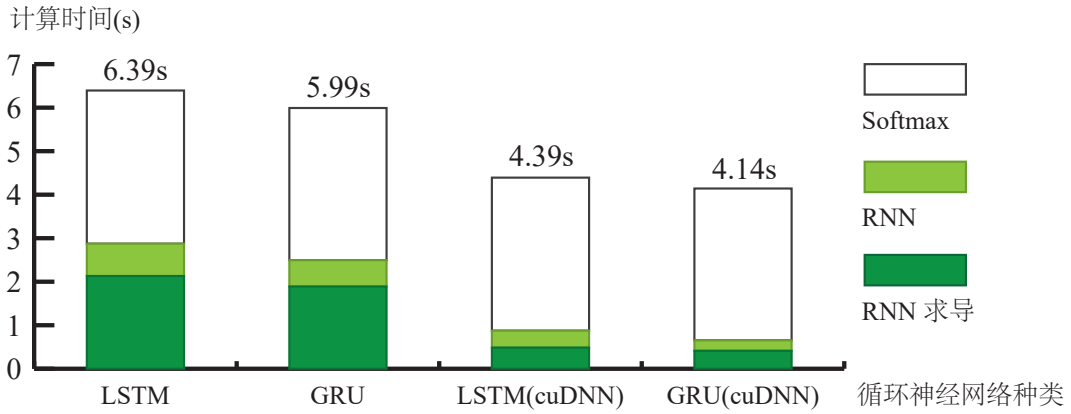

图 **2 wikitext-103** 数据集上测量语言模型三个部分的计算时间比较

小点，验证误差在这个局部最小值的范围内振荡，但是 Wikitext-2 和 Wikitext-103 的唯一区别是训练数据的大小，表示收集很多训练数据不一定能保证模型在相同的验证和测试集上学得更好，而不是更普遍的结果。

此外，我们在实验中发现在 OBW 数据集上运行实验相当具有挑战性，因为它需要更大的参数来拟合。所以我们应用 CUDA 实现的 RNN 模型 [54]，这将会把 RNN 部分所需要的计算时间降到最低，然而等待模型收敛到训练集最小值仍然需要 480 小时。

1.2 影响因素比较

这部分将讨论语言模型的大词表问题在具体实验中瓶颈和各种不同优化策略对对该问题的计算效率和性能的提升和分析。

1.2.1 词表层次化比较

首先在 wikitext-103 数据集，我们统计了语言模型中每个模块的计算时间消耗，如图 2 所示。我们计算运行不同的 RNN 模型（即 LSTM 模型和 GRU 模型）及其梯度（即，RNN Grad）所需的时间，以及 Wikitext-103 数据集的大词汇表上的常规 softmax，这词汇表大小与我们平时所采用的数据集的词表相当。我们分别用 Theano 框架和 CUDA 实现了 RNN 单元，一个采用 python 语言实现，第二个使用并行 C++ 语言实现。目前基于 CuDNN 实现的 RNN 模型计算时间最快，它里面的 RNN 的运行时间可以缩短到最短。我们将代码重复了 100 遍，统计了每个模块的总时间占用，然后分别计算其平均时间占用。这样做的目的是通过多次实验来保证实验数据准确性。

从图中可以看出，使用优化的 CUDA，softmax 模块的影响比 RNN 单元及其梯度更重要。Softmax 计算时间占用总计算时间随着词表的增大占比越来越大，并且已经超过

表 **2**　**Wikitext-103** 数据集上 **GPGPU** 和 **CPU** 的运行时内存和计算时间比较

算法	运行时内存占用	总计算时间 (ms)		前向计算时间 (ms)	
		CPU	GPGPU	CPU	GPGPU
Softmax	$\|\mathcal{H}\mathcal{V}\|$	510.4	262.1	352.2	62.9
cHSM	$2\|\mathcal{H}\|\sqrt{\|V\|}$	506.5	**40.6**	28.7	14.6
tHSM	$\|\mathcal{H}\|$	1,004.0	444.4	8.1	5.6
p-tHSM	$\|\mathcal{H}\|\log\|\mathcal{V}\|$	**383.5**	86.4	**7.0**	**1.4**

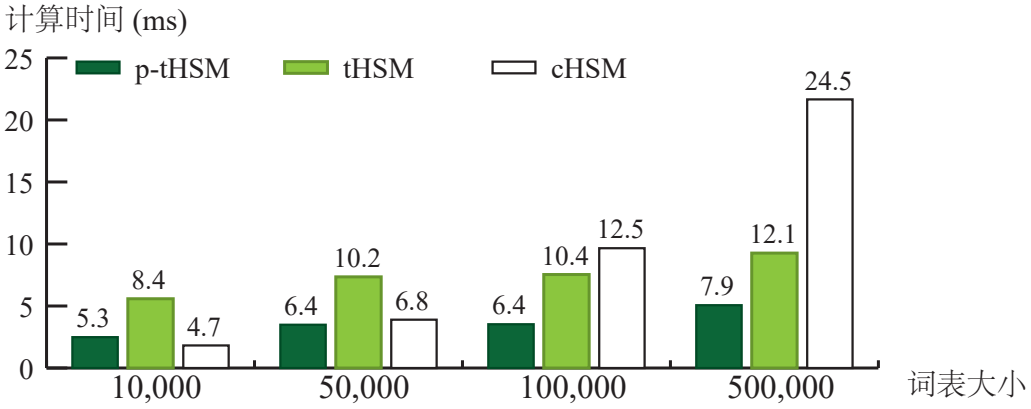

图 **3**　**cHSM, tHSM** 和 **p-tHSM** 算法随着词表大小的计算时间影响

了 50% 的计算时间，这就需要我们详细讨论 softmax 优化。

为了对用于训练相同批处理数据的经验时间复杂性和内存消耗进行基准测试，我们使用这些算法在表 2 中收集了 GPGPU 和 CPU 上的详细结果。我们尝试使用这些算法处理 WikiText-103 数据集，并计算处理一个批处理数据所需的平均时间。另外，输入句子的最大长度，隐藏层，输出词汇和批量大小分别设置为 $\{50, 256, 267735, 20\}$。此外，“总时间”过程表示前向传播和后向梯度优化的过程，“前进时间”过程表示从输入数据到计算模型成本所需的时间消耗。此外，我们计算了上述算法训练期间加载所需的内存。$|\mathcal{V}|$ 表示词汇大小，$|\mathcal{H}|$ 是隐藏层维度。在训练期间，tHSM 只消耗最小的存储器资源，而 p-tHSM 算法覆盖了更大的存储器集合，并且 p-tHSM 在考虑存储器消耗和速度时采用更大的存储器并且获得了更好的加速比。

为了验证与词汇大小相关的 cHSM，tHSM 和 p-tHSM 算法的可扩展性，结果展示在图 3 中。为了显示 p-tHSM 算法的影响，我们在这里不包含“Softmax”方法，因为与其他算法相比，它消耗了更多的计算时间，无法被放在这个图里面。很显然，cHSM 随着词汇大小的平方根（即，$O(|\mathcal{H}|\sqrt{|V|})$）进行缩放，而 p-tHSM 随着词汇大小的增加呈

现出稳定的表现。tHSM 算法随着此表大小的对数进行变化（即，$O(|\mathcal{H}|\log|\mathcal{V}|)$），我们的 p-tHSM 算法所需要的计算时间词表变大，与 tHSM 算法的差异越来越大。这一结果证明我们提出的计算模型更加优越。

在这些实验的基础上，我们可以得出结论：所提出的 p-tHSM 方法胜过历史记录 $O(|\mathcal{H}|\log|\mathcal{V}|)$），并且取得了令人满意的分层 softmax 方法的加速比。这种性能归因于基于 GPGPU 并行性的加速，也是由于 p-tHSM 方法的基本结构，能够将目标字树进行并行地计算。

1.2.2 搜索策略的影响

由于我们提出了三个关于推理阶段搜索策略的算法，其影响可以通过 WER 度量来观察。在这个结构化预测过程中，一个合适的搜索规则将帮助模型获得最小风险的最佳候选人，如表 4 所示。算法 **??**方法表示我们计算所有单词的得分，单词的概率用整个词汇全局归一化。

对于 p-tHSM 系列算法，我们比较了所提出的算法 **??**和传统的算法 **??**算法。算法 **??**比算法 **??**方法获得本地最佳结果花费的时间更少。由于基于树的模型在以前的决策中更容易失败，所以“全局”方法的 MER 分数要比算法 **??** 要好。

表 **3 Wikitext-2** 数据集上 **p-tHSM** 算法针对不同搜索算法的 **WER** 评测结果

	算法	计算时间（ms）	验证集（WER）	测试集（WER）
p-tHSM	算法 **??**	161	**76.67%**	**75.35%**
	算法 **??**	**30**	79.61%	79.32%

值得注意的是，对于 cHSM 方法算法 **??** 比算法 **??** 得到更好的 WER，尽管后一种方法获得了词汇表上的确切顶级候选。由于类结构中存在标签偏差问题，排名最高的词容易出现算法模型无法建模的小组。最后但并非最不重要的是，算法 **??** 和 **??**获得相同的单词排序，唯一的区别是算法 **??** 避免了冗余计算。所以他们达到了可比的 WER 得分，但算法 **??** 需要更少的时间。

1.2.3 循环网络模型的影响

从表 5 中可以看出，门控单元（即 LSTM 和 GRU 单元）的 RNN 模型在复杂度和字误码率方面比传统的 RNN Relu 和 RNN Tanh 模型表现得更好。因为门控功能可以避免梯度消失的问题。虽然它需要稍微多一些的时间进行计算。此外，LSTM 的计算公式

表 **4** **Wikitext-2** 数据集上 **cHSM** 算法针对不同搜索算法的 **WER** 评测结果

	算法	计算时间 (ms)	验证集（WER）	测试集（WER）
cHSM	算法 **??**	102	80.00%	80.02%
	算法 **??**	63	80.00%	80.02%
	算法 **??**	**44**	**77.09%**	**77.07%**

在 GPGPU 上比 GRU 单元更容易并行运行，因此 LSTM 比 GRU 模型需要更少的推理时间。

表 **5** **Wikitext-2** 数据集上不同循环网络针对 **PPL**、**WER** 和计算时间的影响

循环神经网络	计算时间（ms）	验证集（PPL / WER）	测试集（PPL / WER）
1×RNN Relu [55]	176.4	260.52 / 80.00%	238.75 / 80.02%
1×RNN Tanh [38]	176.2	250.57 / 79.61%	230.98 / 79.32%
1×LSTM [23]	**189.5**	180.98 / 77.16%	165.60 / 76.67%
1×GRU [56]	191.3	**179.59 / 77.09%**	**165.32 / 77.07%**
2×RNN Relu [55]	266.3	190.52 / 73.01%	198.75 / 73.02%
2×RNN Tanh [38]	266.3	189.57 / 72.62%	260.98 / 72.32%
2×LSTM [23]	**279.4**	164.98 / 71.17%	165.60 / 71.67%
2×GRU [56]	281.2	**158.59 / 70.08%**	**155.32 / 70.07%**

对于传统的 RNN 模型，通常通过时间反向传播（Back Propagation Through Time，BPTT）进行训练，梯度计算步骤中的努力花费在该训练批次中最长序列的长度上是线性的。因此，小批量生产可能效率低下，因为批次中的次序可能有不同的长度。这个问题的一个经典的解决方案是截断 BPTT 算法，因此 RNN 模型的梯度在截断长度上是恒定的，并且对于 GPGPU 设备上的小批量处理是高效的。而前向概率计算步骤仍然是相同的，唯一的影响是避免梯度反向传播步骤。从图中可以看出，较小的截断 BPTT 在时间效率上取得了较好的效果，较长的截断 BPTT 长度在 PPL 度量方面效果更好。

在训练过程中，如果在被截断的语句中存在某些依赖关系，则模型可以学习这种关系；如果这些依赖关系总是跨越不同的语句块，那么截断的语句已经打破了这些依赖，模型将无法学习这种关系。这是由于模型的参数导数无法回溯到最初的跨段的那个单词，并告诉前段中的某些内容对下一段是有用的。如果你在一个语句段中看到这种关系的话，模型能学习到这种关系是由于模型的参数能求导到前面对应的单词。与之所区别

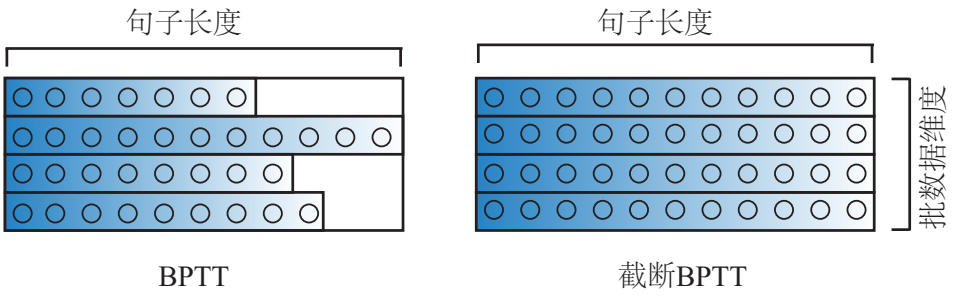

图 **4** **BPTT** 和截断 **BPTT** 算法示意图

的是，在测试时模型将能够在跨语句段中预测，因为模型的计算步骤是不断向前传播，不存在模型反向求导的计算过程。

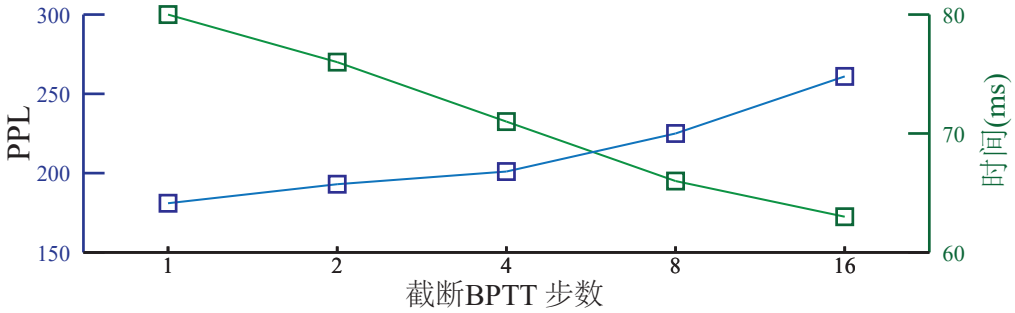

图 **5** **Wikitext-2** 数据集上 **BPTT** 和截断 **BPTT** 算法对 **RNN** 的影响

1.2.4 采样近似算法比较

基于抽样的算法的效率和准确性与样本大小密切相关，我们在下一次评估中测试了这个样本。我们测试了几个样本大小，以评估它们对停电和 NCE 近似值的影响，结果显示在表 6中。根据图 1 所示，中断算法比传统的 NCE 模型表现得相对较好，这与 Ji 等人的实验结果是一致的 [38]。

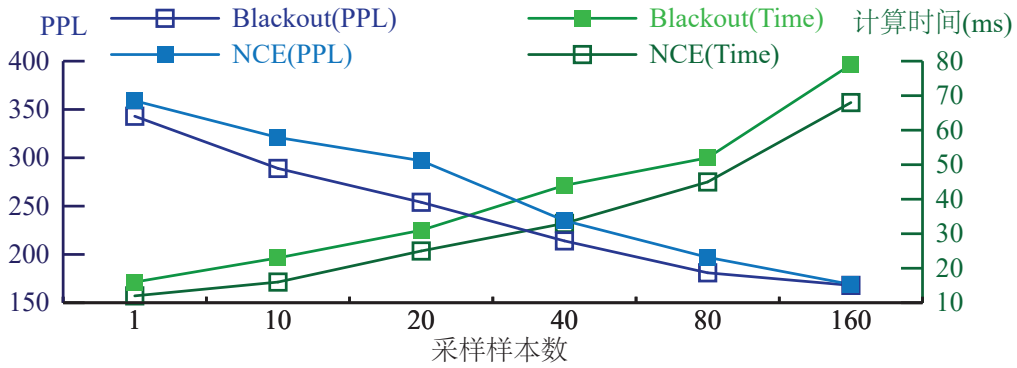

图 **6** **Wikitext-2** 上测试不同采样数量对 **NCE** 和 **Blackout** 算法的影响

我们发现一个样本大小和二元分类表现最差。训练过程涉及在真正的词概率阶段的

学习，以及在噪声概率估计阶段的学习。因此，这些算法只能在收集足够的噪声样本时才能准确地估计噪声概率。这些估计算法的性能随着噪声样本数量的增加而收敛到最优混淆度，但加速比 V/k 减小，其中 k 表示样本大小。因此，我们把这个术语作为一个超参数，应该根据验证集进行调整，所以在训练过程中，每个特定数据集的样本量都是固定的。尽管如此，采样方法在推理过程中是不能使用的，而采用原始的 softmax 方法。

1.2.5　单词聚类策略分析

Chen 等人发现这个单词的层次聚类算法对 cHSM 算法的性能很敏感 [30]。同样，为了获得较为稳定的 cHSM 算法和 p-tHSM 方法的性能，我们考虑了几个现有的聚类准则，如表 6 和表 7 所示。

表 **6** **Wikitext-2** 上评价不同聚类方法对 **p-tHSM** 算法的 **PPL** 影响

算法	建树时间	最大树深度	验证集（PPL）	测试集（PPL）
Uigram	3 分钟	12	218.42	216.05
Bigram	35 小时	21	186.23	189.58
Semantic	26 小时	18	**163.12**	**178.78**

一方面，我们将表 7中的 Wikitext-2 数据集中，应用前面所有提到的聚类方法，同时比较了不同的分支因子对算法效果的影响，并做了比较结果差异。从这张表中可以发现，随机洗牌方法对于其他算法的性能最差，因为它们没有提供有关先前分配的任何信息。此外，我们还观察到，该方法比其他人更难以接受可接受的训练损失，因此在训练集上花费更多的时间来优化参数。然后，发现在对类结构注入外部单字和双字的知识之后，模型确实在目标空间中学习了一个明确定义的单词分布。因此，它对随机和字母表系列取得了更好的结果。而且，用语法和语义算法进行分词的词汇比上述方法得到的结果要好得多，代价是计算分割方法。从实验结果可以看出，向具有先验知识的类结构注入会增强该方法的稳定性，通过平衡聚类时间和模型的准确性，可以调整分支因子，达到预期的结果。

另一方面，我们采用了单词，字母和词汇聚类方法来生成单词在树上的分布，详细的结果在表 7 中给出。与提供调整分支因子的自由度的 cHSM 方法不同，树聚类的实验是相当有限的。Uni-gram 聚类方法（即按照词频合并）在创建单词层次结构方面效率更高，而双字词和语义聚类花费更多时间来计算词汇表中单词之间的相似度矩阵。尽管如

表 7 **Wikitext-2** 数据集上不同聚类算法对 **cHSM** 算法的 **PPL** 和 **WER** 影响

聚类算法	均匀划分？	分支数	训练轮数	测试集（PPL / WER）	耗时（ms）
Random	是	10/3330	145	211.15 / 78.55	791
		20/1664	123	228.72 / 78.89	565
		40/832	103	234.36 / 79.21	321
		80/417	78	243.12 / 79.64	171
		160/208	57	253.38 / 80.08	92
		182/183	48	268.63 / 80.11	88
Alphabet	是	10/3330	141	199.01 / 78.07	773
		20/1664	120	211.34 / 78.23	551
		40/832	100	238.75 / 79.02	313
		80/417	90	241.75 / 79.34	174
		160/208	56	248.35 / 79.62	97
		182/183	45	258.57 / 80.02	87
Uni-gram	是	10/3330	134	211.51 / 77.41	788
		20/1664	122	220.01 / 77.71	549
		40/832	113	236.56 / 77.95	302
		80/417	91	241.12 / 78.25	170
		160/208	55	247.25 / 79.21	93
		182/183	42	253.35 / 79.92	86
Bi-gram	否	10/3672	150	208.11 / 77.32	801
		20/1923	121	217,34 / 77.64	621
		40/1123	102	228.87 / 78.14	588
		80/572	89	246.32 / 78.43	186
		160/340	76	252.33 / 79.51	97
Syntactic	否	10/3612	152	214.31 / 78.11	810
		20/1972	130	220.19 / 78.86	633
		40/996	101	232.33 / 79.33	543
		80/545	89	241.34 / 79.84	179
		160/235	70	262.34 / 80.14	134
Semantic	否	10/3570	133	208.77 / 77.41	819
		20/1873	114	218.31 / 77.78	641
		40/1092	91	225.38 / 78.35	521
		80/561	69	238.45 / 78.91	174
		160/244	44	256.75 / 79.41	103

此，考虑到树合并的规则，Bi-gram 方法考虑了二元共现统计和语义方法来评估特征空间中的欧氏距离。在困惑度量下，二元语义聚类方法比一元方法有更好的效果。最后，与 cHSM 方法相比，更深层次的树模型更适合于词汇聚类，适当的聚类可以提高树层次的效率。

1.3　模型总体评价

如表 8 所示，我们收集了上述三种标准语料库的验证和测试数据集的所有困惑和错误率结果。值得注意的是，我们采用了一层 GRU 单元作为所有这些算法的上下文表示，其维数设置为 256. 另外，对于 NCE 和 Blackout 近似，超参数 k 是对较小的 Wikitext-2 和 Wikitext-103 数据集设置为 $|\mathcal{V}|/20$，对于较大的 One Billion Words 数据集，设置为 $k = |\mathcal{V}|/200$。此外，对于 cHSM 方法，我们根据单词的单字分布来划分词汇。

考虑到 Wikitext-2 数据集上的结果，最初的 softmax 比其他算法获得了最好的分数，因为在 Blackout 和 NCE 近似中没有引入任何 cHSM 和 p-tHSM 算法的结构损失或基于抽样的变分损失。

表 **8** 所有模型在三个数据集上的 **PPL** 和 **WER** 的性能评测

数据集	算法	验证集（PPL/WER）	测试集（PPL/WER）
WikiText-2	GRU + Softmax	172.64 / 77.49%	162.09 / 77.07%
	GRU + NCE [55]	217.84 / 78.26%	199.54 / 78.02%
	GRU + Blackout [38]	221.15 / 77.72%	199.56 / 77.50%
	GRU + cHSM + Uni-gram [30]	253.18 / 78.25%	236.61 / 78.02%
	GRU + p-tHSM + Uni-gram [19]	218.42 / 78.15%	216.05 / 78.15%
	GRU + p-tHSM + Bi-gram [9]	186.23 / 78.15%	189.58 / 78.15%
WikiText-103	GRU + Softmax	130.38 / 72.15%	136.83 / 72.37%
	GRU + NCE [55]	164.78 / 73.22%	165.01 / 73.34%
	GRU + Blackout [38]	163.99 / 73.18%	162.76 / 74.22%
	GRU + cHSM + Uni-gram [30]	171.81 / 73.42%	166.74 / 73.18%
	GRU + p-tHSM + Uni-gram [19]	165.70 / 73.53%	166.11 / 72.44%
	GRU + p-tHSM + Bi-gram [9]	164.15 / 78.15%	163.55 / 77.85%
One Billion Words	GRU + Softmax	330.38 / 88.15%	330.83 / 88.37%
	GRU + NCE [55]	272.07 / 84.83%	276.11 / 84.34%
	GRU + Blackout [38]	268.67 / 84.23%	266.11 / 84.18%
	GRU + cHSM + Uni-gram [30]	225.36 / 80.32%	224.11 / 79.42%
	GRU + p-tHSM + Uni-gram [19]	231.44 / 87.53%	236.11 / 82.53%
	GRU + p-tHSM + Bi-gram [9]	221.55 / 81.15%	218.70 / 83.15%

对于第二个 Wikitext-103 数据集，Brown 聚类的 p-tHSM 方法不仅获得了比霍夫曼聚类方法更好的结果，而且表现也比其他方法好。另外，cHSM 模型能够获得与 p-tHSM

变体类似的结果，表明我们可以用其他合适的用于 cHSM 方法的聚类算法获得更好的结果。由于 Wikitext-103 和 Wikitext-2 数据集共享相同的测试集，因此发现最初的 softmax 通过更大的训练数据被收敛到更好的结果。此外，对于采样方法，它收敛于比 softmax 方法好得多的结果，同时提高了时间效率。

综上所述，在用字极性编码方案替代 tHSM 中的传统霍夫曼编码方案并且实现了基于紧密的树型模型 p-tHSM 之后，我们证明了这种新颖的编码方案允许在 GPGPU 上并行运行计算。这将原始 tHSM 的时间复杂度从 $O(|\mathcal{H}|\log|\mathcal{V}|)$ 减少到 $O(|\mathcal{H}||\mathcal{V}|)$ 并获得了最佳的加速比对于大量的词汇问题。此外，为了稳定 p-tHSM 模型的性能，我们测试了几种现有的层次聚类算法，发现基于树模型的词聚类与内部节点的二元分类密切相关。

1.4 本章小结

在本章中，首先定量分析了大词表问题，接下来比较了我们提出的模型的计算效率和传统计算法之间的差异。针对模型的初始化算法，我们讨论了三种层次聚类算法。最后，我们评估了几个单词层次聚类算法，以更有效的方式组织树中的单词。结果表明，与其他概率归一化方法相比，我们提出的层次概率算法具有很好的加速比，与其他基于抽样的优化算法相比，它性能相对更好。

参考文献

[1] 王建翔. 面向可读性评估的词向量技术研究及实现 [D]. 南京, 中国 : 南京大学, 2017.

[2] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786) : 504 – 507.

[3] Wang Z, Wang D. A Joint Training Framework for Robust Automatic Speech Recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(4) : 796 – 806.

[4] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[A]. Proceedings of 2nd International Conference on Learning Representations[C]. 2015.

[5] Bengio Y, Ducharme R, Vincent P. A Neural Probabilistic Language Model[A]. Proceedings of 14th Annual Conference on Neural Information Processing Systems[C]. 2000 : 932 – 938.

[6] Elman J L. Finding Structure in Time[J]. Cognitive Science, 1990, 14(2) : 179 – 211.

[7] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[A]. Proceedings of 11th Annual Conference of the International Speech Communication Association[C]. 2010 : 1045 – 1048.

[8] Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model[A]. Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing[C]. 2011 : 5528 – 5531.

[9] Brown P F, Pietra V J D, de Souza P V, et al. Class-Based n-gram Models of Natural Language[J]. Computational Linguistics, 1992, 18(4) : 467 – 479.

[10] Baroni M, Dinu G, Kruszewski G. Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors[A]. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics[C]. 2014 : 238 – 247.

[11] Bell R M, Koren Y. Lessons from the Netflix prize challenge[J]. SIGKDD Explorations, 2007, 9(2) : 75 – 79.

[12] Bengio Y, Courville A C, Vincent P. Representation Learning: A Review and New Perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798 – 1828.

[13] Bengio Y, Simard P Y, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157 – 166.

[14] 赵林, 杨保安, 谢志鸣. 一种新的基于结构的神经网络规则抽取方法 [J]. 计算机应用与软件, 2007, 24(6): 28 – 29.

[15] Mnih A, Hinton G E. Three new graphical models for statistical language modelling[A]. Proceedings of 24th International Conference on Machine Learning[C]. 2007: 641 – 648.

[16] Mnih A, Kavukcuoglu K. Learning word embeddings efficiently with noise-contrastive estimation[A]. Proceedings of 27th Annual Conference on Neural Information Processing Systems[C]. 2013: 2265 – 2273.

[17] Mnih A, Teh Y W. A fast and simple algorithm for training neural probabilistic language models[A]. Proceedings of the 29th International Conference on Machine Learning[C]. 2012.

[18] Mikolov T. Statistical language models based on neural networks[J]. Presentation at Google, Mountain View, 2nd April, 2012.

[19] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[A]. Proceedings of 27th Annual Conference on Neural Information Processing Systems[C]. 2013: 3111 – 3119.

[20] Chien J, Ku Y. Bayesian Recurrent Neural Network for Language Modeling[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(2): 361 – 374.

[21] Bengio Y, Ducharme R, Vincent P, et al. A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research, 2003, 3: 1137 – 1155.

[22] 贾玉祥, 王浩石, 昝红英, et al. 汉语语义选择限制知识的自动获取研究 [J]. 中文信息学报, 2014, 28(5): 66 – 73.

[23] Greff K, Srivastava R K, Koutník J, et al. LSTM: A Search Space Odyssey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2222 – 2232.

[24] Chung J, Kastner K, Dinh L, et al. A Recurrent Latent Variable Model for Sequential Data[A]. Proceedings of 29th Annual Conference on Neural Information Processing Systems[C]. 2015 : 2980 – 2988.

[25] Bradbury J, Merity S, Xiong C, et al. Quasi-Recurrent Neural Networks[A]. Proceedings of 4th International Conference on Learning Representations[C]. 2017.

[26] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8) : 1735 – 1780.

[27] 陈凯. 深度学习模型的高效训练算法研究 [D]. 安徽, 中国 : 中国科学技术大学, 2016.

[28] Pezeshki M. Sequence Modeling using Gated Recurrent Neural Networks[J]. CoRR, 2015, abs/1501.00299.

[29] Duda R O, Hart P E, Stork D G. Pattern Classification (2Nd Edition)[M]. Indianapolis, USA : Wiley-Interscience, 2000.

[30] Chen W, Grangier D, Auli M. Strategies for Training Large Vocabulary Neural Language Models[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics[C]. 2016.

[31] Schwenk H. Continuous space language models[J]. Computer Speech & Language, 2007, 21(3) : 492 – 518.

[32] Tucker R C F, Carey M J, Parris E S. Automatic language identification using sub-word models[A]. Proceedings of the 1994 IEEE International Conference on Acoustics, Speech and Signal Processing[C]. 1994 : 301 – 304.

[33] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics[C]. 2016.

[34] Gage P. A New Algorithm for Data Compression[J]. The C Users Journal, 1994, 12(2) : 23 – 38.

[35] Józefowicz R, Vinyals O, Schuster M, et al. Exploring the Limits of Language Modeling[J]. CoRR, 2016, abs/1602.02410.

[36] Kim Y, Jernite Y, Sontag D, et al. Character-Aware Neural Language Models[A]. Proceedings of the 30th AAAI Conference on Artificial Intelligence[C]. 2016 :

2741 – 2749.

[37] Bengio Y, Senecal J. Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model[J]. IEEE Transactions on Neural Networks, 2008, 19(4): 713 – 722.

[38] Ji S, Vishwanathan S V N, Satish N, et al. BlackOut: Speeding up Recurrent Neural Network Language Models With Very Large Vocabularies[A]. Proceedings of 3rd International Conference on Learning Representations[C]. 2016: 1 – 14.

[39] Zoph B, Vaswani A, May J, et al. Simple, Fast Noise-Contrastive Estimation for Large RNN Vocabularies[A]. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies[C]. 2016: 1217 – 1222.

[40] Gutmann M, Hyvärinen A. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics[J]. Journal of Machine Learning Research, 2012, 13: 307 – 361.

[41] 王龙, 杨俊安, 陈雷, et al. 基于循环神经网络的汉语语言模型并行优化算法 [J]. 应用科学学报, 2015, 33(3): 253 – 261.

[42] Goodman J. Classes for fast maximum entropy training[A]. Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing[C]. 2001: 561 – 564.

[43] Morin F, Bengio Y. Hierarchical Probabilistic Neural Network Language Model[A]. Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics[C]. 2005.

[44] Cline D, Razdan A, Wonka P. A Comparison of Tabular PDF Inversion Methods[J]. Computer Graphics Forum, 2009, 28(1): 154 – 160.

[45] Kronmal R A, Peterson A V. On the Alias Method for Generating Random Variables from a Discrete Distribution[J]. The American Statistician, 1979, 33(4): 214 – 218.

[46] Vaswani A, Zhao Y, Fossum V, et al. Decoding with Large-Scale Neural Language Models Improves Translation[A]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing[C]. 2013: 1387 – 1392.

[47] Minka T. Algorithms for maximum-likelihood logistic regression[A]. Virtual Communities and Social Capital. Social Science Computer Review[C]. 2001.

[48] Dugas C, Bengio Y, Bélisle F, et al. Incorporating Second-Order Functional Knowledge for Better Option Pricing[A]. Proceedings of 14th Annual Conference on Neural Information Processing Systems[C]. 2000 : 472 – 478.

[49] 牛雪婷. Huffman 算法的分析与应用 [J]. 中国科教创新导刊, 2009(28) : 87 – 87.

[50] Liang P. Semi-supervised learning for natural language[J]. Master's Thesis Mit, 2005.

[51] Aggarwal C C, Zhai C. Mining Text Data[M]. Berlin, Germany : Springer, 2012.

[52] Lafferty J D, McCallum A, Pereira F C N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data[A]. Proceedings of the 18th International Conference on Machine Learning[C]. 2001 : 282 – 289.

[53] Daks A, Clark A. Unsupervised Authorial Clustering Based on Syntactic Structure[A]. Proceedings of the ACL 2016 Student Research Workshop[C]. 2016 : 114 – 118.

[54] Appleyard J, Kociský T, Blunsom P. Optimizing Performance of Recurrent Neural Networks on GPUs[J]. CoRR, 2016, abs/1604.01946.

[55] Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models[A]. Proceedings of the 30th International Conference on Artificial Intelligence and Statistics[C]. 2010 : 297 – 304.

[56] Chung J, Gülçehre Ç, Cho K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. CoRR, 2014, abs/1412.3555.

[57] Fu R, Guo J, Qin B, et al. Learning Semantic Hierarchies via Word Embeddings[A]. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics[C]. 2014 : 1199 – 1209.

[58] Chen Y, Wang W Y, Rudnicky A I. Learning semantic hierarchy with distributed representations for unsupervised spoken language understanding[A]. Proceedings of 16th Annual Conference of the International Speech Communication Association[C]. 2015 : 1869 – 1873.

[59] Chen X, Wang Y, Liu X, et al. Efficient GPU-based training of recurrent neural network language models using spliced sentence bunch[A]. Proceedings of 15th Annual Confer-

ence of the International Speech Communication Association[C]. 2014 : 641 – 645.

[60] Mnih A, Hinton G E. A Scalable Hierarchical Distributed Language Model[A]. Proceedings of 22nd Annual Conference on Neural Information Processing Systems[C]. 2008 : 1081 – 1088.

[61] Al-Rfou R, Alain G, Almahairi A, et al. Theano: A Python framework for fast computation of mathematical expressions[J]. CoRR, 2016, abs/1605.02688.

[62] Marcus M P, Santorini B, Marcinkiewicz M A. Building a Large Annotated Corpus of English: The Penn Treebank[J]. Computational Linguistics, 1993, 19(2) : 313 – 330.

[63] Paul D B, Baker J M. The design for the wall street journal-based CSR corpus[A]. Proceedings of the 2nd International Conference on Spoken Language Processing[C]. 1992.

[64] Zhou M. Softplus Regressions and Convex Polytopes[J]. CoRR, 2016, abs/1608.06383.

[65] de Brébisson A, Vincent P. The Z-loss: a shift and scale invariant classification loss belonging to the Spherical Family[J]. CoRR, 2016, abs/1604.08859.

[66] Sundermeyer M, Ney H, Schlüter R. From Feedforward to Recurrent LSTM Neural Networks for Language Modeling[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(3) : 517 – 529.

[67] Grave E, Joulin A, Cissé M, et al. Efficient softmax approximation for GPUs[A]. Proceedings of the 34th International Conference on Machine Learning[C]. 2017 : 1302 – 1310.

[68] Hochreiter S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 1998, 6(2) : 107 – 116.

[69] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. CoRR, 2013, abs/1301.3781.

[70] Derczynski L, Chester S, Bøgh K S. Tune Your Brown Clustering, Please[A]. Recent Advances in Natural Language Processing[C]. 2015 : 110 – 117.

[71] Vincent P, de Brébisson A, Bouthillier X. Efficient Exact Gradient Update for training Deep Networks with Very Large Sparse Targets[A]. Proceedings of 29th Annual Confer-

ence on Neural Information Processing Systems[C]. 2015 : 1108 – 1116.

[72] Baltescu P, Blunsom P. Pragmatic Neural Language Modelling in Machine Translation[A]. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies[C]. 2015 : 820 – 829.

[73] Merity S, Xiong C, Bradbury J, et al. Pointer Sentinel Mixture Models[A]. Proceedings of 4th International Conference on Learning Representations[C]. 2017.

[74] Parada C, Dredze M, Sethy A, et al. Learning Sub-Word Units for Open Vocabulary Speech Recognition[A]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies[C]. 2011 : 712 – 721.

[75] Clark A. Combining Distributional and Morphological Information for Part of Speech Induction[A]. Proceedings of 10th Conference of the European Chapter of the Association for Computational Linguistics[C]. 2003 : 59 – 66.

[76] Martin S C, Liermann J, Ney H. Algorithms for bigram and trigram word clustering[J]. Speech Communication, 1998, 24(1) : 19 – 37.

[77] Chen X, Liu X, Wang Y, et al. Efficient Training and Evaluation of Recurrent Neural Network Language Models for Automatic Speech Recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(11) : 2146 – 2157.

[78] Kleinbaum D G, Klein M. Maximum likelihood techniques: An overview[J]. Logistic regression, 2010 : 103 – 127.

[79] Chelba C, Mikolov T, Schuster M, et al. One billion word benchmark for measuring progress in statistical language modeling[A]. Proceedings of 15th Annual Conference of the International Speech Communication Association[C]. 2014 : 2635 – 2639.

[80] Jean S, Cho K, Memisevic R, et al. On Using Very Large Target Vocabulary for Neural Machine Translation[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics[C]. 2015 : 1 – 10.

[81] Jordan M I. Serial order: A parallel distributed processing approach[J]. Advances in psychology, 1997, 121 : 471 – 495.

[82] Gatt A, Krahmer E. Survey of the State of the Art in Natural Language Generation: Core

tasks, applications and evaluation[J]. CoRR, 2017, abs/1703.09902.

[83] Li J, Ouazzane K, Kazemian H B, et al. Neural Network Approaches for Noisy Language Modeling[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(11): 1773 – 1784.

[84] Dehdari J, Tan L, van Genabith J. BIRA: Improved Predictive Exchange Word Clustering[A]. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies[C]. 2016: 1169 – 1174.

[85] Cohn T, Lapata M. An abstractive approach to sentence compression[J]. ACM Transactions on Intelligent Systems and Technology, 2013, 4(3): 41:1 – 41:35.

[86] Li Z, Tang J, Wang X, et al. Multimedia News Summarization in Search[J]. ACM Transactions on Intelligent Systems and Technology, 2016, 7(3): 33:1 – 33:20.

[87] Józefowicz R, Zaremba W, Sutskever I. An Empirical Exploration of Recurrent Network Architectures[A]. Proceedings of the 32nd International Conference on Machine Learning[C]. 2015: 2342 – 2350.

[88] Qu K, Chai X, Liu T, et al. Computer-Aided Diagnosis in Chest Radiography with Deep Multi-Instance Learning[A]. Proceedings of 24th International Conference on Neural Information Processing[C]. 2017: 723 – 731.