

北京航空航天大学中法工程师学院

硕士学位论文文献综述

论文题目：基于词表分解优化的循环神经网络
语言模型的研究

专 业：工业工程

研究方向：自然语言处理

作 者：是黎彬

学 号：ZY1624134

指导教师：荣文戈 副教授

北京航空航天大学中法工程师学院

2017 年 12 月 14 号

摘要

语言模型在自然语言处理领域起着非常重要的作用，该性能的好坏直接影响例如机器翻译和语音识别等很多任务的发展。随着循环神经网络的提出以及应用于语言模型上，计算机对人类语言的建模更加精确，但是也带来模型训练和运行速度都下降。尤其是当文本数据的数量急剧增长，单词的词表会变得巨大，这直接导致计算消耗十分大，包括计算复杂度太高所引起的运行耗时以及模型参数量太大所引起的内存占用过于庞大。为了解决这些问题，历史上人们开发了很多技术与模型，这些方法可以被分为三类：基于采样的近似算法，基于词表分解的算法，基于字母级别的编码模型。这些方法在一定程度上能缓解那些问题，但各自仍然存在一些缺陷。

本文首先概述语言模型的主要发展历史，详细阐述核心的建模方法——循环神经网络及其变种，解释大词表问题带来的重大挑战，列出了语言模型任务的国内外通用的标准数据集。然后上述所说的三大类前沿技术将会被总结分析优缺点，并重点讨论了基于词表分解的方法。最后，通过总结前人的方法，本文列出几个未来发展方向。

关键词: 大词表问题；词表分解；循环神经网络语言模型；自然语言处理

Abstract

Language model is a fundamental component in the field of neural language processing; its performance has direct influence on many tasks, e.g., machine translation and automatic speech recognition. With the propose of Recurrent Neural Network (RNN) and its inclusion in language model, language modeling by computer becomes more accuracy, yet producing the in-efficiency in the model' s training and operating. Especially when the text corpus growths explosively, vocabulary will become so huge that it causes both high computational complexity, which make it time-consuming to train as well as to infer, and excessive model size, which means the huge memory footprint. To tackle these problems, many techniques and models have been developed, mainly categorized as: sampling-based approximation, character level-based models, and vocabulary factorization-based methods.

In this paper we first review the development of the language model, containing the detail introduction of recurrent neural network and its variants, the explanation of the problem of over-large vocabulary, and some standard datasets. Afterwards, we explore those three kinds of advanced methods and analyze their merit and demerit. Finally, by summarizing the previous researches, we list some possible directions and the future works.

Keyword: Over-large Vocabulary; Vocabulary Factorization; RNN Language Model; Natural Language Processing

目录

摘要	1
1 引言	4
2 语言模型概述	5
2.1 N-gram 语言模型	5
2.2 前馈神经网络语言模型	6
2.3 循环神经网络语言模型	8
2.4 循环神经网络的变种	9
3 国内外研究现状及发展动态	11
3.1 大词表问题分析	11
3.2 基于采样的近似	11
3.3 基于字符级别的建模	11
3.4 基于词表分解的预测	11
4 总结与展望	11
参考文献	13

1 引言

近年来,互联网技术的快速发展,尤其是各大社交网络平台技术日趋完善,为人们营造了一个便于交流的世界。于是,在全球范围内互联网用户数量在不断增加,用户之间的交流、用户对互联网产品的评论以及各种信息的共享,这造成互联网上的数据爆炸式增长。其中,文本数据是用户信息传递的载体,如何从文本数据中自动挖掘有用的信息便成了一个各大互联网公司研究的热点。由此,对无标注的文本语言进行建模(Language Modeling, LM)就成了文本数据挖掘的一项非常重要的基础工作,很多文本挖掘应用场景,如信息检索 [1]、机器翻译 [2]、语音识别 [3],都需要以语言模型为基本模块。因此,此研究课题是具有很强的实用价值和研究意义的。

在深度学习的概念提出之前,统计语言模型(Statistical Language Model, SLM)是一种主要的对语言建模的方法,又称为 N 元语言模型(N-gram model [4])。自 2006 年 Hinton 提出深度学习(Deep Learning, DL)以来 [5],将深度学习应用于自然语言处理(Natural Language Processing, NLP)任务中已经成为自然语言处理界的趋势,尤其是在语言模型中的应用。研究者提出可以用神经网络(Neural Network, NN)对语言进行建模,包括前馈神经网络模型(Feed-Forward Neural Network, FFNN) [6] 和循环神经网络模型(Recurrent Neural Network, RNN) [7]。其中,RNN 由于其特殊的网络结构能够将当前词的历史信息存储起来,并作为当前词预测的依据。该结构克服了 N-gram 语言模型无法利用距离当前词较远的词的信息。此外,在 RNN 对一个词序列进行建模时,离散的词被映射到连续稠密的词嵌入(Word Embedding)空间,在这低维空间进行计算学习词序列的特征来学习词与词之间的依赖关系。因此,循环神经网络语言模型(Recurrent Neural Network Language Model, RNNLM)在模型困惑度(Perplexity, PPL)和词识别率上都取得了最好的结果 [8]。

尽管 RNNLM 已经取得了重大的成功,但该模型仍然存在着很多难点,其中一个最为显著的问题是词表过大(Over-large vocabulary)的问题 [9]。对此,由于语言模型旨在学习人类语言的模式,那么这意味着用于训练模型的人类语料越大,那么语言模型越能拟合语言模式并能因此模型生成的文本更像人说的语句一样。于是为了让模型更好的学习,这就不可避免的使得语料库数据量不断增长。词表过大的问题将会对 RNNLM 造成两个方面的巨大挑战:1)高计算复杂度(Computational complexity) [10]; 2)庞大的模型参数量(Model size) [11]。

为了强调这些挑战的严峻性，在此列举一个例子来对此进行分析。One Billion Words (OBW) 数据集是一个近年来公开的训练语言模型的语料数据集，它包含约 80 万个不同的词，是一个典型的能够测试模型能否胜任词表过大这个问题的数据集。如果我们考虑一个标准的 RNNLM，其中词嵌入向量的维度和 RNN 中的隐层的维度都设置为 1024，并且在计算机中用 32 位浮点数去储存这些数据，那么通过计算可知该模型的大小达到 7GB。如此巨大的模型对于当前 12GB 的图形处理元件 (Graphics Processing Units, GPUs) 是极其困难的，甚至当训练时候增大每一批次的数据量，该 GPUs 已经无法处理该模型了。此外，通过一些基础实验发现，该模型的训练非常耗时，甚至要花费 1 个月的时间才能训练好。由此可见词表过大对于 RNNLM 是一个巨大的且亟待解决的问题。

因此探讨研究语言模型的大词表问题，是目前理论应用到实际过程中必须要克服的问题，也是值得研究和探讨的问题。为了克服这个问题，历史上研究者们做了很多尝试并开发了一系列的方法。我们讨论了三个不同方向的方法：一类是基于采样技术的近似方法 (Sampling-based Approximation) [12, 13]；一类基于字符级别 (Char-level) 的建模加速方法 [14]；最后一类是基于词表分解 (Vocabulary Factorization) 的方法 [15, 8]。

2 语言模型概述

2.1 N-gram 语言模型

语言模型可以对一段文本的概率进行估计，对信息检索 [1]、机器翻译 [2]、语音识别 [3] 等任务有着重要的作用。形式化讲，统计语言模型的作用是为一个长度为 m 的字符串确定一个概率分布 $P(w_1; w_2; \dots; w_m)$ ，表示其存在的可能性，其中 w_1 到 w_m 依次表示这段文本中的各个词。一般在实际求解过程中，通常采用下式计算其概率值：

$$P(w_1; w_2; \dots; w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1; w_2) \cdots P(w_i|w_1; w_2; \dots; w_{i-1}) \\ \cdots P(w_m|w_1; w_2; \dots; w_{m-1}) \quad (1)$$

在实践中，如果文本的长度较长，公式 1 右部 $\cdots P(w_m|w_1; w_2; \dots; w_{m-1})$ 的估算会非常困难，因为出现 $w_1; w_2; \dots; w_{m-1}; w_m$ 的语段非常少，进而该模型的稀疏性特别严重。因此，研究者们提出使用一个简化模型： n 元模型 (n -gram

model)。在 n 元模型中估算条件概率时，距离大于等于 n 的上文词会被忽略，也就是对上述条件概率做了以下近似：

$$P(w_i|w_1; w_2; \cdots; w_{i-1}) \approx P(w_i|w_{i-(n-1)}; \cdots; w_{i-1}) \quad (2)$$

当 $n = 1$ 时又称一元模型 (unigram model)，公式2 右部会退化成 $P(w_i)$ ，此时，整个句子的概率为：

$$P(w_1; w_2; \cdots; w_m) = P(w_1)P(w_2) \cdots P(w_m) \quad (3)$$

从式中可以知道，一元语言模型中，文本的概率为其中各词概率的乘积。也就是说，模型假设了各个词之间都是相互独立的，文本中的词序信息完全丢失。因此，该模型虽然估算方便，但性能有限。

当 $n = 2$ 时又称二元模型 (bigram model)，将 n 代入公式2 中，右部为 $P(w_i|w_{i-1})$ 。常用的还有 $n = 3$ 时的三元模型 (trigram model)，使用 $P(w_i|w_{i-2}; w_{i-1})$ 作为近似。这些方法均可以保留一定的词序信息 [4]。

2.2 前馈神经网络语言模型

N-gram 语言模型的一个显著缺陷是：基于 2 式，新词和低频词难以得到有效的概率统计。基于此，人们发明了各种平滑算法，如 discount, back-off, interpolation 等 [16, 17]。这些方法在一定程度上改善了 n-gram 在低频词上的性能，但基于模型本身的缺陷，这一困难始终无法从根本上解决。

随着神经网络的兴起，人们开始尝试利用神经网络构造语言模型。与 n-gram 不同，神经网络对参数进行高度共享，因此对低频词具有天然的平滑能力。神经网络语言模型 (Neural Network Language Model, NNLM) 的最早由 Bengio 等人在 2001 年提出 [6]，近年来一些学者开始展开这方面的研究，并取得一系列成果，如 [18, 19, 20, 21]，但总体而言，对 NNLM 的研究还处在起步阶段。具体而言，NNLM 通过一个多层感知网络 (MultiLayer Perceptron, MLP) 来计算 2 式中概率。图 1 给出一个典型的 NNLM 语言模型。神经网络语言模型采用普通的三层前馈神经网络结构，其中第一层为输入层。Bengio 提出使用各词的词向量作为输入以解决数据稀疏问题，因此输入层为词 $w_{i-(n-1)}; \cdots; w_{i-1}$ 的词向量的顺序拼接：

$$x = [e(w_{i-(n-1)}); \cdots; e(w_{i-2}); e(w_{i-1})] \quad (4)$$

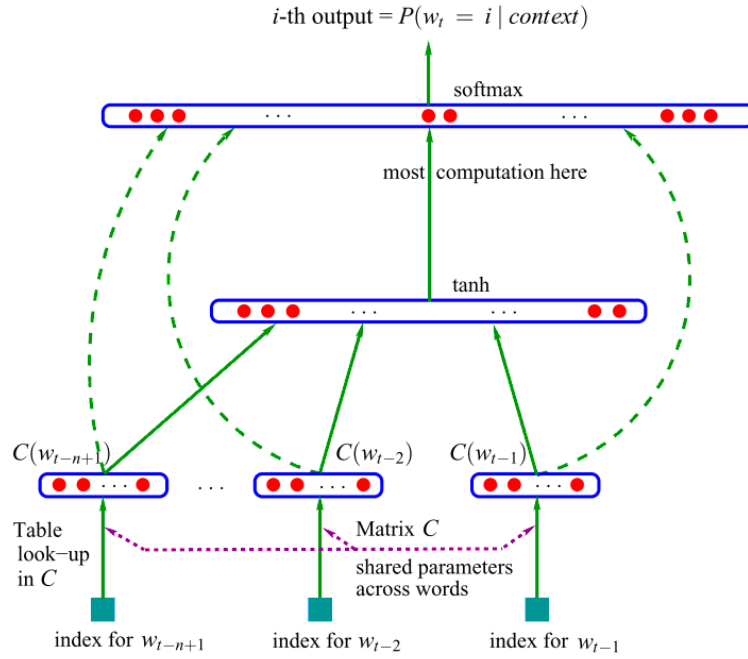


图 1: 前馈神经网络语言模型

当输入层完成对上文的表示 x 之后，模型将其送入剩下两层神经网络，依次得到隐藏层 h 和输出层 y ：

$$\begin{aligned} h &= \tanh(b(1) + Hx) \\ y &= b(2) + Wx + Uh \end{aligned} \quad (5)$$

其中 $H \in \mathbb{R}^{|h| \times (n-1)|e|}$ 为输入层到隐藏层的权重矩阵， $U \in \mathbb{R}^{|V| \times (n-1)|h|}$ 为隐藏层到输出层的权重矩阵， $|V|$ 表示词表的大小， $|e|$ 表示词向量的维度， $|g|$ 为隐藏层的维度。 $b(1), b(2)$ 均为模型中的偏置项。矩阵 $W \in \mathbb{R}^{|V| \times (n-1)|e|}$ 表示从输入层到输出层的直连边权重矩阵。由于 W 的存在，该模型可能会从非线性的神经网络退化成为线性分类器。Bengio 等人在文中指出，如果使用该直连边，可以减少一半的迭代次数；但如果没有直连边，可以生成性能更好的语言模型。因此在后续工作中，很少有使用输入层到输出层直连边的工作，下文也直接忽略这一项。如果不考虑 W 矩阵，整个模型计算量最大的操作，就是从隐藏层到输出层的矩阵运算 Uh ，后续的模型均有对这一操作的优化，这部分内容将在下一节介绍。

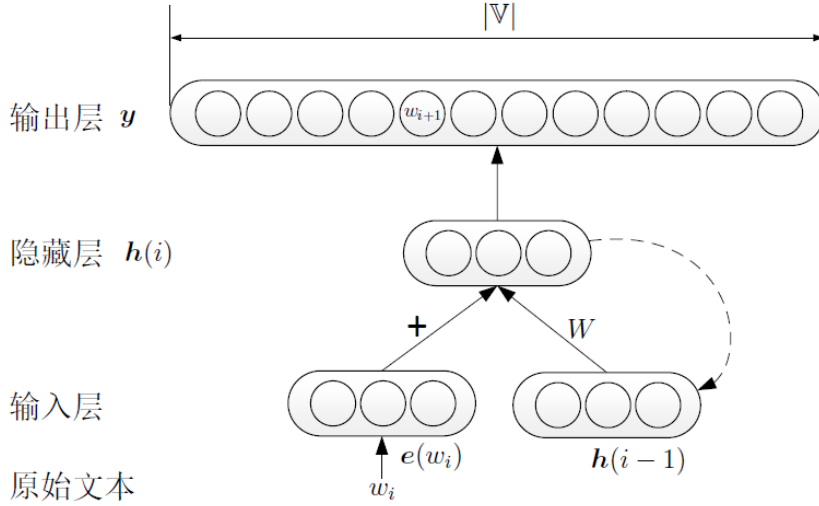


图 2: 循环神经网络语言模型 (RNNLM) 模型结构图

2.3 循环神经网络语言模型

Mikolov 等人提出的循环神经网络语言模型 (Recurrent Neural Network based Language Model, RNNLM) 则直接对 $P(w_i|w_1; w_2; \dots; w_{i-1})$ 进行建模, 而不使用公式 2 对其进行简化 [22, 7]。因此, RNNLM 可以利用所有的上文信息, 预测下一个词, 其模型结构如图 2 所示。

RNNLM 的核心在于其隐藏层的算法:

$$h(i) = \phi(e(w_i) + Wh(i-1)) \quad (6)$$

其中, ϕ 非线性激活函数。但与 NNLM 不同, RNNLM 并不采用 n 元近似, 而是使用迭代的方式直接对所有上文进行建模。在公式 6 中, $h(i)$ 表示文本中第 i 个词 w_i 所对应的隐藏层, 该隐藏层由当前词的词向量 $e(w_i)$ 以及上一个词对应的隐藏层 $h(i-1)$ 结合得到。

隐藏层的初始状态为 $h(0)$, 随着模型逐个读入语料中的词 $w_1; w_2; \dots$, 隐藏层不断地更新为 $h(1); h(2); \dots$ 。根据公式 6, 每一个隐藏层包含了当前词的信息以及上一个隐藏层的信息。通过这种迭代推进的方式, 每个隐藏层实际上包含了此前所有上文的信息, 相比 NNLM 只能采用上文 n 元短语作为近似, RNNLM 包含了更丰富的上文信息, 也有潜力达到更好的效果。RNNLM 的输出层计算方法与 NNLM 的输出层一致。

2.4 循环神经网络的变种

循环神经网络在对较短文本进行建模时表现良好，但当词序列过长时，便会出现梯度消失和梯度爆炸的问题，这使得词与词之间的长期依赖的特性和信息难以被攫取和记忆 [21]。针对这个问题，很多 RNN 的变种被开发出来以提升模型的建模性能，其中最主要的两种模型是长短记忆网络（Long Short-term Memory, LSTM）[23] 和门限记忆节点（Gated Recurrent Unit, GRU）[24]。LSTM 的计算公式定于如下 [25]：

- 输入门: 输入门：控制当前输入 x_t 和前一步输出 h_{t-1} 进入新的 cell 的信息量：

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i)$$

- 忘记门: 决定是否清楚或者保持单一部分的状态

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f)$$

- 变换输出和前一状态到最新状态

$$g_t = \phi(W^g x_t + U^g h_{t-1} + b^g)$$

- 输出门: 计算 cell 的输出

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o)$$

- cell 状态更新步骤: 计算下一个时间戳的状态使用经过门处理的前一状态和输入：

$$s_t = g_t \odot i_t + s_{t-1} \odot f_t$$

- 最终 LSTM 的输出: 使用一个对当前状态的 tanh 变换进行重变换：

$$h_t = s_t \odot \phi(o_t)$$

其中 \odot 代表对应元素相乘 (Element-wise Matrix Multiplication), $\phi(x)$, $\sigma(x)$ 的定义：

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \sigma(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

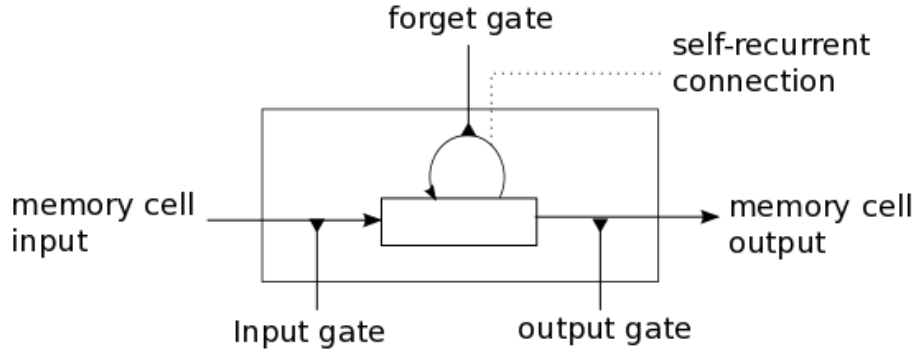


图 3: LSTM 模型

GRU 可以看成是 LSTM 的变种，GRU 把 LSTM 中的 forget gate 和 input gate 用 update gate 来替代。把 cell state 和隐状态 h_t 进行合并，在计算当前时刻新信息的方法和 LSTM 有所不同。下图是 GRU 更新 h_t 的过程 [26], 具体定义如下:

- 更新门 z_t : 定义保存多少以前的信息。

$$z_t = \sigma(W^z x_t + U^z h_{t-1})$$

- 重置门 r_t : 决定保留多少输入信息。

$$r_t = \sigma(W^r x_t + U^r h_{t-1})$$

- 节点内部更新值 \tilde{h}_t : 其次是计算候选隐藏层 (candidate hidden layer) \tilde{h}_t , 这个候选隐藏层和 LSTM 中的 \tilde{c}_t 是类似, 可以看成是当前时刻的新信息, 其中 r_t 用来控制需要保留多少之前的记忆, 如果 r_t 为 0, 那么 \tilde{h}_t 只包含当前词的信息:

$$\tilde{h}_t = \tanh(W^h x_t + U^h (h_{t-1} \odot r_t))$$

- 隐藏层输出值 h_t : 最后 z_t 控制需要从前一时刻的隐藏层 h_{t-1} 中遗忘多少信息, 需要加入多少当前时刻的隐藏层信息 \tilde{h}_t , 最后得到 h_t , 直接得到最后输出的隐藏层信息, 这里与 LSTM 的区别是 GRU 中没有 output gate:

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}$$

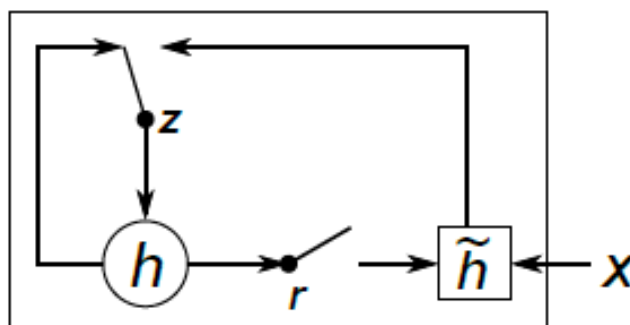


图 4: GRU 模型示意图

如果 reset gate 接近 0，那么之前的隐藏层信息就会丢弃，允许模型丢弃一些和未来无关的信息；update gate 控制当前时刻的隐藏层输出 h_t 需要保留多少之前的隐藏层信息，若 z_t 接近 1 相当于我们之前把之前的隐藏层信息拷贝到当前时刻，可以学习长距离依赖。一般来说那些具有短距离依赖的单元 reset gate 比较活跃（如果 r_t 为 1，而 z_t 为 0 那么相当于变成了一个标准的 RNN，能处理短距离依赖），具有长距离依赖的单元 update gate 比较活跃。

3 国内外研究现状及发展动态

3.1 大词表问题分析

本文已经在第1章节通过举例凸显出词表过大的问题，此小节简单解释词表过大

3.2 基于采样的近似

3.3 基于字符级别的建模

3.4 基于词表分解的预测

4 总结与展望

在机器学习领域有一个公认的观点是，模型选用的特征决定了机器学习算法所能达到的上界；而具体模型的选择，则决定了对上界的逼近程度。因此，特征表示在机器学习中是至关重要的一个步骤。在基于神经网络的语言模型中，尽管不同的模型有着不同的性能，但这些模型均基于分布假说，语义由其上下

文决定。这些模型所能表达出的语义，受到语料中各词上下文分布的约束。语言模型的构建是自然语言处理的基础工作，具有重要的理论意义和广阔的应用前景。本文对基于神经网络的语言模型建模技术中，最重要的两个问题，大词表问题和上下文信息的表示问题，进行深入分析，比较现有方法的优劣。

在基于神经网络的语言模型建模中，本文对现有的语言模型进行了系统的理论对比及实验分析。理论方面，本文阐述了各种现有模型的联系，从上下文的表示、上下文与目标词之间的关系两方面对模型进行了分类整理，并对其中一些重要的模型进行详细描述。在基于神经网络的文档表示技术中，本文分析了现有的文档表示技术：基于循环网络的语言模型和基于前馈神经网络的语言模型。进一步，我们还分析讨论了各种循环神经网络的变种，例如长短记忆网络和门限循环节点，并绘制模型结构图和具体数学公式。

并针对现有的大词表问题，提出了基于二叉树的多层概率分布函数来提高其效率。该方法克服了传统 softmax 的计算复杂度过高的问题，但是最初提出来的需要手工构造单词的类别关系。所以利用大数据文本信息，通过聚类算法能否有效提高模型的精确度是值得商榷的实验课题。因此，我们顺便讨论了目前已有的聚类策略，包括：按频率聚类，按照 2-gram 聚类的布朗聚类算法等等。

最近一两年，已经有人尝试跳出分布假说的框架，使用更广泛的信息，对语义进行建模。Weston 等人设计了一个模型，从知识库中获取语义表示，并用于提升关系抽取任务的性能 [27]。Wang 等人指出，利用知识库中的知识，可以进行词义消歧，进一步提升词向量的性能 [28]。未来的工作需要考虑，如何利用海量的多源异构的数据学习到的语言模型，从中挖掘出有用的信息，更好地对数据和知识进行表示。

参考文献

- [1] Rong Jin, Alex G. Hauptmann, and Cheng Xiang Zhai. Title language model for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 42–48, New York, NY, USA, 2002. ACM.
- [2] Paul Baltescu and Phil Blunsom. Pragmatic neural language modelling in machine translation. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 820–829, 2015.
- [3] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 338–342, 2014.
- [4] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.
- [5] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [6] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938, 2000.
- [7] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.

- [8] Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 5528–5531, 2011.
- [9] Xie Chen, Yongqiang Wang, Xunying Liu, Mark J. F. Gales, and Philip C. Woodland. Efficient gpu-based training of recurrent neural network language models using spliced sentence bunch. In *Proc. of INTERSPEECH*, pages 641–645, 2014.
- [10] Wenlin Chen, David Grangier, and Michael Auli. Strategies for training large vocabulary neural language models. In *Proc. of ACL*, 2016.
- [11] Xiang Li, Tao Qin, Jian Yang, Xiaolin Hu, and Tie-Yan Liu. Lightrnn: Memory and computation-efficient recurrent neural networks. In *Proc. of NIPS*, pages 4385–4393, 2016.
- [12] Yoshua Bengio and Jean-Sébastien Senecal. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Trans. Neural Networks*, 19(4):713–722, 2008.
- [13] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J MACH LEARN RES*, 13:307–361, 2012.
- [14] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. In *Proc. of AAAI*, pages 2741–2749, 2016.
- [15] Joshua Goodman. Classes for fast maximum entropy training. In *Proc. of ICASSP*, pages 561–564, 2001.
- [16] *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*. ISCA, 2009.

- [17] Guodong Zhou and Kimteng Lua. Interpolation of n-gram and mutual-information based trigger pair language models for mandarin speech recognition. *Computer Speech & Language*, 13(2):125–141, 1999.
- [18] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 238–247, 2014.
- [19] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SIGKDD Explorations*, 9(2):75–79, 2007.
- [20] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.
- [21] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.
- [22] Tomáš Mikolov. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April, 2012*.
- [23] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. LSTM neural networks for language modeling. In *Proc. of INTERSPEECH*, pages 194–197, 2012.
- [24] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2980–2988, 2015.
- [25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

- [26] Mohammad Pezeshki. Sequence modeling using gated recurrent neural networks. *CoRR*, abs/1501.00299, 2015.
- [27] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1366–1371. ACL, 2013.
- [28] Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2014.