南京大學

研 究 生 毕 业 论 文

（申 请 硕 士 学 位）

论　文　题　目　基于神经网络语言模型的文本向量表示研究

作　者　姓　名　牛力强

学科、专业方向　计算机技术

指　导　教　师　戴新宇 副教授

研　究　方　向　自然语言处理

2016 年 **3** 月 **9** 日

学　　号：**MF1333036**

论文答辩日期：**2016** 年 **6** 月 **1** 日

指 导 教 师：　　（签字）

A Research on Text Embeddings based on Neural Network Language Models

by

NIU Li-Qiang

Supervised by

Associate Professor DAI Xin-Yu

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER
in
Computer Technology

Department of Computer Science and Technology
Nanjing University

Mar 1, 2016

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于神经网络语言模型的文本向量表示研究

计算机技术 专业 2013 级硕士生姓名：牛力强

指导教师（姓名、职称）：戴新宇 副教授

摘　要

文本表示与建模是自然语言处理领域一项基础任务，传统文本表示方法是基于词袋模型，即将所有存在的词组成一个高维的 0/1 空间，若词出现，则对应维度置为 1，否则置为 0。词袋模型的好处在于简单高效，容易扩展，但同时也面临众多严重的问题，如维度灾难、数据稀疏表示、缺失语义表达能力等。而近年来随着深度学习技术在语音、图像、生物信息等领域取得重大的成果，研究者们也开始将深度学习技术应用到自然语言处理领域。特别地，自 2013 年谷歌研究员基于神经网络语言模型来学习分布式词向量表示起，越来越多基于深度学习技术来学习各层次文本语义向量表示的方法出现。

本文集中对基于神经网络语言模型的文本表示与建模问题进行了探索。相比传统方法以及前人基于神经网络的文本表示与建模方法，本文对其进行了多方面的扩展：

1. 提出主题语义向量学习模型 Topic2Vec：借助于潜在狄利克雷分布（Latent Dirichlet Allocation, LDA）挖掘到词的主题信息，将基于神经网络语言模型学习来学习词的语义向量表示模型 Word2Vec 扩展到主题语义向量表示。相比原始 LDA 中词以及主题文档等都以维度下标来表示而且词与主题之间的关系采用条件概率来衡量，Topic2Vec 将词及其主题映射至统一语义向量空间，词与主题之间关系采用余弦相似度来计算。
2. 提出一个能够联合学习词及其属性分布式表示的学习框架：将词向量表示学习方法扩展到联合学习词及其属性语义向量表示。相比单一模型的学习能力，联合学习框架使得词与其属性在学习过程中相互促进来得到更好的语义向量表示。
3. 提出融合词向量与 LDA 的模型：将词向量表示应用到 LDA 中来提升词主题发现。

针对扩展之后的模型方法，本文分别进行了多方面的实验评估，实验结果表明扩展之后的方法均优于原有的模型方法。

关键词： 自然语言处理；文本表示；深度学习；神经网络；文本建模

南京大学研究生毕业论文英文摘要首页用纸

THESIS: A Research on Text Embeddings based on Neural Network Language Models

SPECIALIZATION: Computer Technology

POSTGRADUATE: NIU Li-Qiang

MENTOR: Associate Professor DAI Xin-Yu

Abstract

Natural language

Traditional

Based on the

Text representation

keywords: Natural Language Processing, Text Representation, Deep Learning, Neural Networks, Text Modelling

前　言

在机器学习系统的设计过程中，数据的表示是一项基础工程，好的数据表示方法能够提升整个系统的性能。传统思路下研究者们重点关注如何设计出更好的模型系统来达到更好的结果，而数据的表示则大多采用人工设计的方法。但是，近年来随着互联网大数据时代的到来以及 2006 年深度学习技术的兴起，大数据及数据表示方法在机器学习系统中扮演的角色越来越重要。特别地，基于深度学习技术，采用多层神经网络学习数据的层次化表示迅速成为一大研究热点，而且已经取得不俗的成果。

在自然语言处理领域中，文本表示则是一项基础任务，好的文本表示方法将直接有益于后续各项自然语言处理任务。传统的文本表示方法是基于词袋模型，即将所有的词组成高维 0/1 特征空间，若词出现，则对应维度置为 1，否则置为 0。词袋模型的好处在于简单高效，但是面临众多严重的问题，如维度灾难、数据稀疏、缺失语义表达能力等。而最新的基于深度学习的文本表示方法通过多层神经网络的学习将文本中的多层结构（词、短语、句子、段落、文档等）映射至一个低维连续的空间，每一种类型的文本都对应一个低维连续值的向量。因此，新的深度学习文本表示方法完美的克服了原来词袋模型的弊端，而且在各类自然语言处理任务中取得了最好的结果。如基于神经网络的机器翻译、基于神经网络的文本分类、情感分析、复述识别等等。

基于近年来深度学习用于自然语言处理任务所取得的成果，本文集中对自然语言处理中的文本表示以及建模问题进行了探索。相比原有基于深度学习的文本表示与建模方法，本文对其进行了多方面的扩展：

1. 提出主题语义向量学习模型 Topic2Vec：借助于潜在狄利克雷分布（Latent Dirichlet Allocation, LDA）挖掘到词的主题信息，将基于神经网络语言模型学习来学习词的语义向量表示模型 Word2Vec 扩展到主题语义向量表示。相比原始 LDA 中词以及主题文档等都以维度下标来表示而且词与主题之间的关系采用条件概率来衡量，Topic2Vec 将词及其主题映射至统一语义向量空间，词与主题之间关系采用余弦相似度来计算。
2. 提出一个能够联合学习词及其属性分布式表示的学习框架：将词向量表示

学习方法扩展到联合学习词及其属性语义向量表示。相比单一模型的学习能力，联合学习框架使得词与其属性在学习过程中相互促进来得到更好的语义向量表示。

3. 提出融合词向量与 LDA 的模型：将词向量表示应用到 LDA 中来提升词主题发现。

针对扩展之后的模型方法，本文进行了多方面的实验验证评估，实验结果表明扩展之后的方法均优于原有的模型方法。

牛力强

2016 年夏于南京大学

目　　次

插图清单

附表清单

第一章　绪论

1.1　研究背景

自然语言同语音、图像并列为人工智能研究领域的三大重要元素。特别地，自 2006 年 Hinton 提出深度学习技术[3] 并且在机器学习以及人工智能领域取得重大突破。其中，语音识别、计算机视觉、图像处理等典型人工智能任务取得了重大进展与广泛应用。但是，不同于语音、图像等原始信号或者信息，自然语言是人类高度抽象之后的产物。因此，深度学习等机器学习技术在面对自然语言处理任务时依旧面临较大的挑战。

传统的自然语言处理任务包括了语言建模、机器翻译、文本分类、情感分析、句法分析、词性标注、命名实体识别、中文分词、复述识别、自动问答等等[4]。传统的自然语言处理方法大多是将有监督机器学习模型方法等根据对应的任务加以修改应用。例如，基于短语的统计机器翻译采用双语平行语料，依据源语言与目标语言词对齐方法来构建翻译模型，之后基于翻译模型来将源语言翻译为目标语言[5]；文本分类、情感分析、复述识别等都是将具体的任务转换为有监督机器学习中二分类或者多分类任务，不同的任务采取不同的特征提取方法，进而构建对应的分类器[4]；句法分析、中文分词等序列标注任务部分采用条件随机场等图模型方法，基于特征模板来提取特征进而构造有监督分类器来进行参数学习[6,7]。可以看到传统自然语言处理处理的最大局限在于采用有监督的机器学习方法，而现实中有监督方法所需的标记语料是相对有限且人工成本昂贵。因此，传统有监督机器学习方法在自然语言处理任务中的效果并不理想依旧面临很大的问题与挑战。

在此背景下，诸多研究者意识到深度学习的下一个突破口是自然语言处理方向。深度学习在语音以及图像等领域的成果得益于多层神经网络基于大量的未标注语料中的层次表示学习能力。如在图像中，神经网络通过层次化的学习可以学习到像素、边际、部分、图像的表示，之后通过这些不同层级的特征来构造分类器[8]。可以看到多层神经网络在处理图像时模拟人脑的处理过程，因此深度学习得以取得重大突破，语音识别也类似[9]。因此，研究者将自然语言

处理的突破寄望于深度学习。近年来，研究者也开始探索文本的层次化表示学习，并希望通过基于深度学习的文本表示方法来提升传统的自然语言处理任务。

1.2 研究目的与意义

对文本诸如词（Word）、主题（Topic）、句子（Sentence）、文档（Document）等进行建模在自然语言处理（Natural Language Processing, NLP）和信息检索（Information Retrieval, IR）领域是一项关键任务，其目的在于找到一类简短扼要的描述来表达文本的语义，使其能够促进大规模系统的高效处理并且有益于常见的任务，例如文本分类、聚类、摘要、相似性或者相关性估计。

在过去的几十年间，各种各样的模型和解决方法被提出来，例如词袋模型（Bag-of-Words, BOW)[10]、TF-IDF[11]、潜在语义分析（Latent Semantic Analysis, LSA）[12]、概率潜在语义分析（Probabilistic Latent Semantic Analysis, PLSA）[13]、潜在狄利克雷分布（Latent Dirichlet Allocation, LDA）[14] 等。词袋模型的好处在于简单高效，但是面临众多严重的问题，如维度灾难、数据稀疏、缺失语义表达能力等。

近年来，基于深度学习的文本表示方法通过多层神经网络的学习将文本中的多层结构（词、短语、句子、段落、文档等）映射至一个低维连续的空间，每一种类型的文本都对应一个低维连续值的向量，如词向量（Word Embedding）。因此，基于深度学习文本表示方法完美的克服了传统词袋模型的问题，而且在各类自然语言处理任务中取得了最好的结果[4]。如基于 Bengio 等人在 2003 年提出的神经网络语言模型[15]，Mikolov 等人在 2013 年提出了 Word2Vec 模型来学习词向量表示[1,16,17]；Le 和 Mikolov 在 2014 年提出学习句子和文档分布式向量表示的方法[18]；Stanford 大学 Socher 等人利用递归自编码器和动态 Pooling 技术来做复述识别取得 state-of-the-art 的结果[19]；Tang 等人在 2014 年利用开发了基于深度学习技术的系统用于 Twitter 情感分类[20]；还有各类基于递归神经网络和编码-解码器的统计机器翻译模型[21,22]。

由上可以看出，基于深度学习的文本表示与建模在克服传统方法缺点的同时，在各类自然语言处理任务中均取得较好甚至是最好的结果。但是，深度学习技术在 NLP 领域的应用远不止于此。因此，基于近年来深度学习用于自然语言处理任务所取得的成果，本文集中在自然语言处理中的文本表示以及建模问

题进行了深入的探索。相比原有的深度学习的文本表示与建模方法，本文对其进行了多方面的扩展：

1. 提出主题语义向量学习模型 Topic2Vec：借助于潜在狄利克雷分布（Latent Dirichlet Allocation, LDA）挖掘到词的主题信息，将基于神经网络语言模型学习来学习词的语义向量表示模型 Word2Vec 扩展到主题语义向量表示。相比原始 LDA 中词以及主题文档等都以维度下标来表示而且词与主题之间的关系采用条件概率来衡量，Topic2Vec 将词及其主题映射至统一语义向量空间，词与主题之间关系采用余弦相似度来计算。
2. 提出一个能够联合学习词及其属性分布式表示的学习框架：将词向量表示学习方法扩展到联合学习词及其属性语义向量表示。相比单一模型的学习能力，联合学习框架使得词与其属性在学习过程中相互促进来得到更好的语义向量表示。
3. 提出融合词向量与 LDA 的模型：将词向量表示应用到 LDA 中来提升词主题发现。

针对扩展之后的模型方法，本文进行了多方面的实验验证评估，实验结果表明扩展之后的方法均优于原有的模型方法。因此，尽管深度学习技术已经在文本表示与建模甚至是整个 NLP 领域做出不俗的成绩，但当我们面对具体的实际问题时候，仍需思考当前问题的不同之处以及现有方法的不足，对现有模型做出修改或者创新才能做到更好的效果。

1.3 论文结构

本文旨在对深度学习技术在自然语言处理领域中文本表示与建模这一问题的应用做进一步的探索，从（1）学习主题向量表示、（2）联合学习词及其属性分布式表示和（3）融合词向量表示与图模型三个大的方面来进行扩展，分析了现有方法的不足之处，提出了自己的模型方法，并且分别进行了理论分析与实验对比。

本文一共分为六章，后续章节安排如下：

- 第2章：语言模型与词向量表示。本章简要介绍本文的背景工作，分别介绍自然语言处理中的语言模型、Bengio 等人在 2003 年提出的神经网络语言模型[15] 以及 Mikolov 等人在 2013 年提出的基于神经网络语言模型学习分布式词向量表示的模型 Word2Vec[1,16,17]。

- 第3章：学习主题向量表示。本章主要工作是借助于潜在狄利克雷分布（Latent Dirichlet Allocation, LDA）挖掘到词的主题（Topic）信息，将 Word2Vec 扩展至学习主题 Topic 的分布式表示，并且提出了模型 Topic2Vec。
- 第4章：联合学习词和属性向量表示。本章主要工作是将 Word2Vec 扩展到一个能够联合学习词（Word）和属性（Attributes）分布式向量表示的统一框架，其中重点引入了三类词属性：词元（Lemma）、主题（Topic）、文档（Document），基于该框架实现：（1）学习主题 Topic 的分布式表示、（2）学习文档 Document 的分布式表示和（3）利用词元 Lemma 和主题 Topic 信息来提升词向量表示。该框架不仅可以学习到属性的分布式表示，而且可以利用属性知识来提高原来词向量表示。另外，该框架易于扩展，可以学习更多其他的词属性的分布式表示。
- 第5章：融合词向量与主题模型。
- 第6章：总结与展望。本章对前面的工作进行总结，并提出了进一步研究的展望。

第二章　语言模型与词向量表示

2.1　统计语言模型

统计语言建模（Statistical Language Modeling）的目的在于建立一个统计语言模型（Statistical Language Model, SLM），使其能够尽量准确的估计自然语言的分布。一个统计语言模型（SLM）是一个发生在字符串 S 上且能够反应字符串 S 作为一个句子出现频率的概率分布 P(s)。通过在统计语言模型中将各种各样的语言现象用简单的参数表达，SLMs 提供了一种容易的方法使得计算机能够处理复杂的自然语言。SLMs 最初始也是最重要的应用是语音识别，但是 SLMs 在其他各种各样的自然语言应用中扮演者重要的角色，诸如机器翻译、词性标注、智能输入系统和文本转换语音系统等。

不失一般性，我们先定义 $V^{\dagger}$ 为词汇表 V 中所有可能句子的集合，$V^{\dagger}$ 是一个无穷集，因为句子可以是任意无限长度。因此，我们给出如下语言模型的定义：

定义 2-1 (语言模型)　一个语言模型包含一个有限集 V 和一个函数 $p(x_1, x_2, ..., x_n)$ 因此：

1. 对于任意

$$\langle x_1, x_2, ..., x_n \rangle \in V^{\dagger}, p(x_1, x_2, ..., x_n) \geq 0 \tag{2-1}$$

2. 并且

$$\sum_{\langle x_1, ..., x_n \rangle \in V^{\dagger}} p(x_1, x_2, ..., x_n) = 1 \tag{2-2}$$

这里 $p(x_1, x_2, ..., x_n)$ 是一个在 $V^{\dagger}$ 中所有句子的概率分布。

常见的统计语言模型如下（这里以句子 $S: w_1, w_2, ..., w_n$ 为例说明）：

- **N-gram 语言模型**[23,24]
  - **Unigram 模型**假设当前词 w_i 只依赖于自己，因此我们按如下方式计算句

子 S 的概率：

$$P(w_1, w_2, ..., w_n) = \prod_i P(w_i|w_1, w_2, ..., w_{i-1}) \approx \prod_i P(w_i) \tag{2-3}$$

- **Bigram 模型**假设当前词 w_i 依赖于前一个词 w_{i-1}，因此我们按如下方式计算句子 S 的概率：

$$P(w_1, w_2, ..., w_n) = \prod_i P(w_i|w_1, w_2, ..., w_{i-1}) \approx \prod_i P(w_i|w_{i-1}) \tag{2-4}$$

- 按如上方法类推，我们可以扩展到 trigrams 模型，4-grams 模型以及 5-grams 模型。

其中，N-gram 模型参数可以通过极大似人估计（Maximum Likelihood Estimation, MLE）方法，如下所示：

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})} \tag{2-5}$$

- 扩展 N-gram 语言模型：基于类别的 N-gram 语言模型、语法的 trigrams 模型、序列 N-gram 模型等。
- 此外还有最大熵语言模型（Maximum Entropy Language Model）[25]、结构化语言模型（Structured Language Model）[26]、全句指数模型（Whole Sentence Exponetial Model）[27] 等。

2.2 神经网络语言模型

以上我们知道，统计语言模型的目标是学习一个关于自然语言中任意词所组成句子的联合概率函数。但是这个过程其实是异常艰难的，因为存在维度灾难（curse of dimensionality）的问题：在模型中被测试的一个词序列很有可能跟训练集中的所有句子不同。传统并且非常成功的基于 N-grams 的模型通过连接训练集中出现过的较短的覆盖词序列来提高语言模型的泛化能力。为了彻底克服维度灾难的问题，Bengio 等人在 2003 年提出了一个基于神经网络的语言模型（Neural Probabilistic Language Model, NPLM）[15]。该神经网络语言模型可以同时学习到（1）每个词对应的分布式表示和（2）以这些词分布式表示表达的词序列的概率函数。泛化能力可以保证是因为若某一从未出现过的词序列中的

词跟存在的句子中的词相似（词与词具有相似的分布式表示），则该词序列也会得到比较高的概率。

神经网络语言模型结构如下图2-1所示：

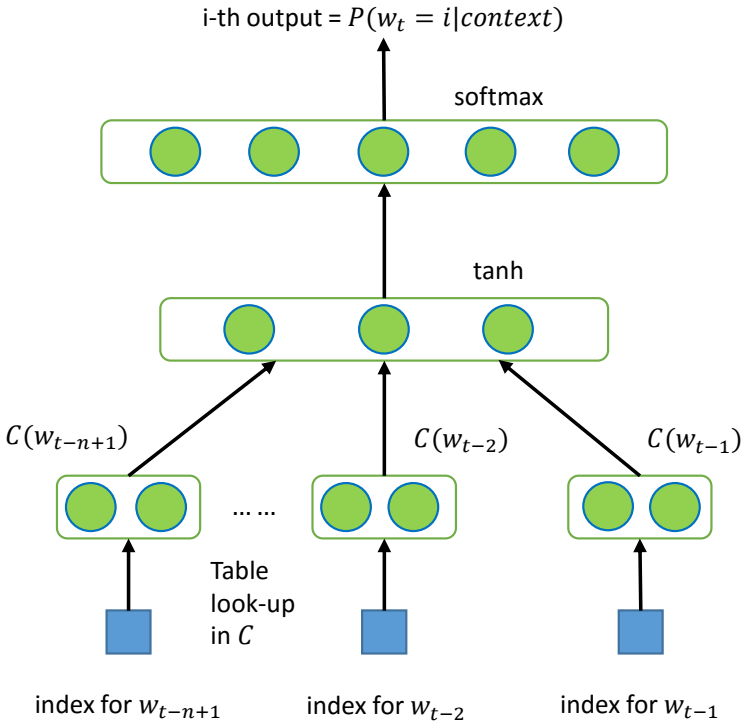

图 2-1: 神经网络语言模型结构图

这里主要介绍神经网络语言模型的训练过程如下：

- 训练数据：一个词序列 $w_1, ..., w_T$ 且任意 $w_t \in V$，这里 V 是一个大的有限词汇表。

  训练目标：学习一个好的模型 $f(w_t, ..., w_{t-n+1}) = \hat{p}(w_t|w_1^{t-1})$, 满足对任意 w_1^{t-1} 有 $\sum_{i=1}^{|V|} f(i, w_{t-1}, ..., w_{t-n+1}) = 1$ 且 $f > 0$。
- 将函数 $f(w_t, ..., w_{t-n+1}) = \hat{p}(w_t|w_1^{t-1})$ 分解为两部分：
  1. 映射矩阵 C 将 V 中任意元素 i 映射到实数向量 $C(i) \in \mathbb{R}^m$，代表词汇表里每一个词的分布式特征向量（distributed feature vector），实际中 C 是一个 $|V| \times m$ 的矩阵。
  2. 基于当前词的概率函数，用 C 来表示：函数 g 将上下文中词序列的特征向量映射为词汇表 V 中下一个词 w_t 出现的条件概率，g 的输出结果是一

个向量，其第 i 个元素估计概率 $\hat{p}(w_t|w_1^{t-1})$ 如图2-1所示。

$$f(i, w_i, ..., w_{t-n+1}) = g(i, C(w_{t-1}), ..., C(w_{t-n+1})) \tag{2-6}$$

- 函数 f 是两个映射的组合（C 和 g），其中 C 在上下文中被所有词所共享。映射 C 中参数时特征向量本身，表示为 $|V| \times m$，其中第 i 行 $C(i)$ 是词 i 的特征向量。函数 g 可以由一个前向或者递归神经网络或其他参数函数实现，参数为 ω。因此所有的参数集合是 $\theta = (C, \omega)$。

- 参数训练通过求解 θ 使得训练语料的惩罚 log 似然最大：

$$L = \frac{1}{T} \sum_t log(f(w_t, w_{t-1}, ..., w_{t-n+1}; \theta) + R(\theta) \tag{2-7}$$

在大多数情况下，神经网络包含一个隐藏层建立在词特征向量映射之上或者直接由词特征向量连接到输出。因此这里实际有两个隐藏层：共享词特征映射层 C 和普通的双曲正切隐藏层。更精确地，神经网络计算如下函数，选用 $softmax$ 输出层保证输出概率值总和为 1：

$$\hat{p}(w_t|w_{t-1}, ..., w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{2-8}$$

这里 y_i 是每一个输出词 i 的非规范化 log 概率，通过以下方式来计算，包含参数 b, W, U, d 和 H：

$$y = b + Wx + Utanh(d + Hx) \tag{2-9}$$

这里双曲正切 tanh 逐元的应用，W 可选置为 0（没有从输入词特征向量直接连接到输出层），x 是词特征层激活向量，通过矩阵 C 中所有输入词特征组合连接而成：

$$x = (C(w_{t-1}), C(w_{t-2}), ..., C(w_{t-n+1})) \tag{2-10}$$

令 h 作为隐藏单元的个数，m 作为每个词特征数。当词特征没有直接连接到输出层时，矩阵 W 置为 0。则模型的所有自由参数（free parameters）有输出偏置（biases）b（包含 $|V|$ 个元素），隐藏层偏置 d（包含 h 个元素），隐藏层到输出层权重 U（$|V| \times h$ 矩阵），词特征到输出层权重 W（$|V| \times (n-1)m$ 矩阵），隐藏层权重 H（$h \times (n-1)m$ 矩阵）和词特征 C

（$|V| \times m$ 矩阵）：

$$\theta = (b, d, W, U, H, C) \tag{2-11}$$

所有自由参数的总数为 $|V|(1 + nm + h) + h(1 + (n-1)m)$，主要影响因子是 $|V|(nm + h)$。

- 随机梯度下降用来训练该神经网络，通过如下遍历训练语料中第 t 个词的方式迭代更新：

$$\theta \leftarrow \theta + \epsilon \frac{\partial log\hat{p}(w_t|w_{t-1}, ..., w_{t-n+1})}{\partial \theta} \tag{2-12}$$

这里 ϵ 是学习率（learning rate）。

综上可以看到神经网络语言模型不仅可以学习到词序列即句子的概率值，而且还可以学习到每一个词所对应的特征向量。在文本表示特别是词的表示方法中我们重点关注词特征向量的计算，即矩阵 C。

2.3 传统词向量表示

2.4 分布式词向量表示

当前众多的自然语言处理（NLP）系统中，词被视作为原子单元，但是词与词之间的相似度却没有度量，是因为词被表示为词汇表中的下标索引（indices）。通常采用比较流行的 N-gram 模型用于统计语言模型，如今可以在所有有效数据中来训练 N-grams（万亿的词数级别[28]）。但是这类简单方法依赖于大量训练数据，缺失泛化能力。因此在很多任务中都存在缺陷，例如自动语音识别中的领域内数据是有限的；同样地，在机器翻译系统中，大多数语言所有的语料数据也仅仅包含十亿级别的词数或者更少。

随着近年来机器学习技术的进步，依赖大数据来训练更加复杂的模型已经成为可能并且通常效果优于简单模型。特别地，最成功的概念是利用词的分布式表示（distributed representations）[29]。例如，基于语言模型的神经网络模型相比 N-gram 模型表现更好[15,30]。

因此，Mikolov 等人在 2013 年基于神经网络语言模型提出 Word2Vec 模型来在大数据集上训练词的分布式向量表示[1,16,17]，包括 Continuous Bag-of-Words(CBOW) 和 Skip-gram 两种结构。下面分别简要介绍 CBOW 和 Skip-gram 模型：

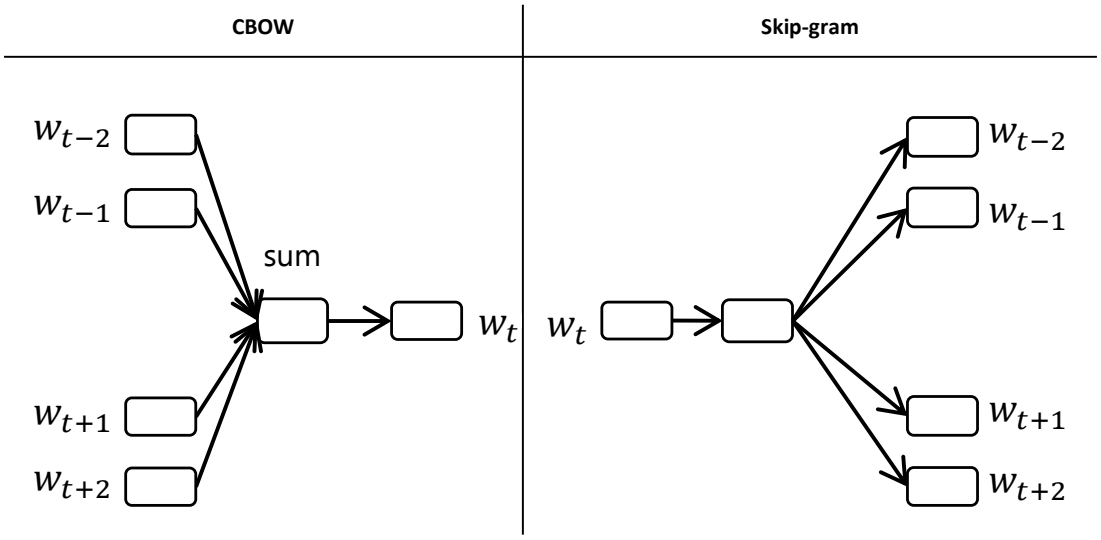

图 2-2: Word2Vec 结构图

- **CBOW 模型**

  假设给定词序列 $(w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2})$，其中 w_t 是当前词，其余词作为 w_t 的上下文（context）。如图2-2所示，CBOW 模型利用上下文所有词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 去预测当前词 w_t。训练时，给定一个词序列 $D = \{w_1, ..., w_M\}$，CBOW 模型的学习目标函数定义为最大化如下 log 似然：

$$L_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M} \log p(w_i|w_{cxt}) \tag{2-13}$$

  这里 w_{cxt} 表示当前词 w_i 的上下文。

- **Skip-gram 模型**

  如图2-2所示，Skip-gram模型利用当前词 w_t 去预测上下文所有词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$。训练时，给定一个词序列 $D = \{w_1, ..., w_M\}$，Skip-gram 模型的学习目标函数定义为最大化如下 log 似然：

$$L_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M} \sum_{-k\le c\le k, c\neq 0} \log p(w_{i+c}|w_i) \tag{2-14}$$

  这里 k 是上下文窗口大小。

另外，在公式2-13和2-14中，对任意变量 w_j 和 w_i，条件概率 $p(w_j|w_i)$ 通

过以下 $softmax$ 函数来计算：

$$p(w_j|w_i) = \frac{\exp(\mathbf{w_j} \cdot \mathbf{w_i})}{\sum_{w \in W} \exp(\mathbf{w} \cdot \mathbf{w_i})} \tag{2-15}$$

在实际 Word2Vec 训练中，考虑到 $softmax$ 函数中分母项数量级为 W，计算 $\nabla log p(w_j|w_i)$ 等比例于 W，而 W 通常是非常大的（$10^5 - 10^7$ 词项），因此采用一般的随机梯度下降算法计算代价太大，在实践中并不适用。因此 Mikolov 等人随机提出了加速 $softmax$ 的算法，包括层次 $softmax$（Hierarchical Softmax）和负采样（Negative Sampling）[16]。

- **层次 $softmax$**

  层次 $softmax$ 是一个计算高效的 $softmax$ 的近似方法，在神经网络语言模型中，最早由 Morin 和 Bengio 提出[31]。主要优势是替代原来神经网络中评估 W 个输出节点获得概率分布，层次 $softmax$ 只需评估 $log_2(W)$ 个节点。

  层次 $softmax$ 采用一棵以 W 个词作为叶子节点的二叉树，而且对于每一个节点，层次 $softmax$ 明确地表示其子孙节点的相对概率，这定义了一个赋予词以概率值的随机游走过程。

  更精确地，每一个词 w 可以由树的根节点通过合适的路径达到。设 $n(w, j)$ 为从根节点 root 到 w 路径中第 j 个节点，$L(w)$ 为这个路径的长度。因此 $n(w, 1) = root$ 且 $n(w, L(w)) = w$。另外，对任意的内部节点 n，设 $ch(n)$ 为一个 n 任意固定的孩子节点且令如果 x 是真，则 $[x]$ 为 1，否则为-1。因此层次 $softmax$ 按如下定义 $p(w_O|w_I)$：

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma([n(w, j+1) = ch(n(w, j))] \cdot v'^{T}_{n(w,j)} v_{w_I}) \tag{2-16}$$

  这里 $\sigma(x) = 1/(1+exp(-x))$，可以证明 $\sum_{w=1}^{W} p(w|w_I) = 1$。这使得 $log p(w_O|w_I)$ 和 $\nabla log(w_O|w_I)$ 的计算代价等比例于 $L(w_O)$，$L(w_O)$ 平均情况不会大于 $logW$)。并且，不同于标准 $softmax$ 中每个词 w 被赋予两个表示 v_w 和 v'_w，而层次 $softmax$ 中每个词 w 只有一个表示 v_w，并且二叉树中每一个内部节点 n 也有一个表示 v'_n。

  层次 $softmax$ 中树的结构对于性能有着重要的影响，Mnih 和 Hinton 基于对训练时间和结果模型的准确率的考虑探索了很多方法来构造树的结构[32]。

  在 Word2Vec 训练过程中，层次 $softmax$ 采用了霍夫曼树结构，对高频词赋

予短的编码来加速训练。因为在实践中观测到提前对词依据出现频率来进行聚簇在神经网络语言模型中是一种非常简单有效的加速技术[1,33]。

- **负采样**

层次 *softmax* 的一种替代方法是噪音对比估计（Noise Contrastive Estimation, NCE），最早由 Gutmann 和 Hycarinen 提出[34] 并且由 Mnih 和 Teh 用于语言建模中[35]。NCE 假设一个好的模型有能力通过逻辑斯蒂回归（logistic regression）从噪音中区分数据，这与 Collober 和 Weston 通过对噪音以上的数据进行排序来训练模型所用的 hinge 损失类似[4]。

NCE 可以被证明能够近似最大化 *softmax* 的 log 概率，Word2Vec 模型仅仅关心学习到高质量的词向量表示，因此可以简化 NCE 只需保证向量表示的质量即可，按如下方式定义负采样（Negative Sampling）：

$$log\sigma(v'^{T}_{w_O} v_{w_I}) + \sum_{i=1}^{k} E_{w_i} \sim P_n(w) \left[log\sigma(-v'^{T}_{w_i} v_{w_I} \right] \tag{2-17}$$

公式2-17可以用来替代 Word2Vec 中的每一个 $logP(w_O|w_I)$ 项。因此，任务变成利用逻辑斯蒂回归（logistic regression）从噪音分布 $P_n(w)$ 区分目标词 w_O，这里对于每一个数据样本都有 k 个负例样本。实验表明 k 的值范围在 5-20 之间对于小数据集有利，而对于大数据集，k 通常设置较小为 2-5 之间。负采样与 NCE 之间的主要区别在于 NCE 同时需要样本和噪音分布的数值概率，而负采样只需要样本。并且 NCE 目的在于最大化 *softmax* 的 log 概率，这个属性对于我们训练向量表示并不重要。

相比传统独热表示（one-hot representations）和词袋模型（Bag-of-Words），Word2Vec 学习词的分布式向量表示能够更好表征词的特征，泛化能力更强，因此在各项自然语言处理任务中取得最好的结果[1,16,17,36,37]。另外，基于 Google 开源 C 语言的 Word2Vec[①]，也有多个其他语言版本相继开源如 Java[②]和 Python[③]等。

受到词的分布式向量表示的启发，研究人员探索新的方法如 Pennington 等人提出基于全局上下文矩阵分解的 Glove 模型[38]、基于词的多义现象训练词的多个分布式向量表示[39,40]、融合 LDA 挖掘词的主题信息来学习词的向量表示解决一词多义的问题[2] 和更快速的学习词的分布式向量表示的方法[41,42]。

[①] C：https://code.google.com/archive/p/word2vec/
[②] Java：http://deeplearning4j.org/word2vec
[③] Python：https://radimrehurek.com/gensim/models/word2vec.html

也有基于词向量表示技术扩展至句子、文档[①]、词的情感、图结构、文本属性等[18,37,43–45]。

2.5 本章小结

本章着重介绍本文的背景工作，包括：（1）统计语言模型：简要介绍语言模型的原理与对于自然语言处理任务的意义，并列举了常用的 N-gram 模型以及其他语言模型；（2）神经网络语言模型：简要介绍神经网络语言模型的由来，重点说明神经网络语言模型的原理；（3）Word2Vec 词向量表示：简要介绍了 Google 基于神经网络语言模型提出 Word2Vec 模型来学习词的分布式表示，重点介绍 Word2Vec 的结构包括 CBOW 和 Skip-gram 以及训练过程。

本章内容是本文所有工作的基础，基于分布式假设（Distributed Hypotheses）和神经网络语言模型（Neural Network Language Models）来学习文本的分布式表示或向量表示（Distributed Representations or Embedding）的思想贯穿全文，并且训练过程也采用 Word2Vec 中的随机梯度下降（Stochastic Gradient Descent）以及负采样（Negative Sampling）技术。

[①] Doc2Vec：https://radimrehurek.com/gensim/models/doc2vec.html

第三章　学习主题的向量表示

3.1　潜在狄利克雷分布

潜在狄利克雷分布（Latent Dirichlet Allocation, LDA）[14] 是一个概率生成模型，LDA 假设每篇文档包含多个隐藏主题，而每一个主题则是建立在词汇表里所有词上的一个概率分布。简单来说，LDA 按如下方式来生成词序列：

- 对于文档 d 中第 n 个词 w_n：
  - 采样一个主题 $z_n \sim Multinomial(\theta_d)$
  - 采样一个词 $w_n \sim Multinomial(\phi_{z_n})$

通常在概率图模型中引入潜在隐藏变量，如 LDA 中引入隐藏主题（Topics），而采样极大似然估计法（Maximum Likelihood Estimate, MLE）和最大后验概率（Maximum a Posteriori, MAP）来直接推断模型参数会遇到无法直接求导或者计算代价太大的问题[14]。因此在实际中通常采用近似推断方法，包括拉普拉斯近似（Laplace approximation）[46]、变分近似（Variational Inference, VI）[47,48] 和马尔可夫链蒙特卡洛方法（Markov chain Monte Carlo methods, MCMC）[49] 等。通过 MCMC 中最简单的吉布斯采样（Gibbs sampling）[①]，依据当前全条件概率分布（full conditional distribution）对于每个词进行一定轮数主题采样或至收敛，可以推断学习到文档-主题概率矩阵 Θ 和主题-词概率矩阵 Φ[50]。依据已有的参数 Θ 和 Φ，可以对任意新来的句子进行同样的采样过程，收敛之后文档中每一个词都会被赋予一个主题。

3.2　学习主题向量表示

3.2.1　研究背景

依据前面的介绍，我们知道 LDA 可以挖掘文档中的主题结构信息，而且已经在自然语言处理（NLP）和机器学习（ML）等领域做出巨大的贡献[14]。

[①] http://gibbslda.sourceforge.net/

但是，LDA 中概率分布仅仅是语料中出现关系的统计结果，并且在实际中，概率分布表示（distrbutional representations）并不是特征表示最好的选择。近来，通过概念以及表示的学习，基于嵌入向量来表示词和文档的方法相继被提出，例如 Word2Vec[1] 和 Doc2Vec[18] 等，而且嵌入向量表示在许多任务中的结果比 LDA 的概率分布表示更好。

同时，由于词汇表通常在 $10^5 - 10^6$ 之间，因此 LDA 还面临严重的长尾现象（long-tail）：LDA 会赋予语料中的高频词以高概率而低概率的词则很难被选作为主题词。但是在实际中，低概率词有时候可以更好的表征主题。例如，LDA 会赋予词 *food* 高概率并且选为主题词而不是 *cheeseburger*，选用高概率词 *drug* 而不是 *aricept*，选用高概率词 *technology* 而不是 *smartphone*。从这些例子可以看出，LDA 基于语料的统计结果，会明显偏向于高频词（如 $food, drugs, technology$ etc.），而这些高频词词义通常比较泛，不够具体，不能够非常清晰具体地表征一个主题。相反，部分低频低概率词（如 $cheeseburger, aricept, smartphone$ etc.）词义更加具体，可以更好表征某一个主题。

最近，基于神经网络语言模型（NNLMs）学习的分布式表示将词和文档映射至一个低维的语义向量空间，并且在许多 NLP 和 ML 任务中实现了重要的结果[1,18]。特别地，Word2Vec 可以自动学习到词的概念以及词与词之间的语义和句法简单线性关系，例如词向量语义关系：$vec(\text{``}Berlin\text{''}) - vec(\text{``}Germany\text{''}) = vec(\text{``}Paris\text{''}) - vec(\text{``}France\text{''})$ 和词向量句法关系：$vec(\text{``}Write\text{''}) - vec(\text{``}Writing\text{''}) = vec(\text{``}Read\text{''}) - vec(\text{``}Reading\text{''})$[17]。Doc2Vec 在情感分析（Sentiment Analysis）任务中取得了最好的结果[18]。自然地，我们会想这个问题：如果将这些主题映射至语义空间将会发生什么？

3.2.2 Topic2Vec 模型

受到 Word2Vec 的启发，我们将主题和词整合到神经网络概率语言模型（NPLM）中。如图3-1所示，我们提出了 Topic2Vec 模型能在学习词向量表示的同时学习到主题的向量表示。Topic2Vec 同样分为 CBOW 和 Skip-gram 两种结构。例如，通过 LDA 的主题推断，给定一个词-主题序列 $(w_{t-2} : z_{t-2}, w_{t-1} : z_{t-1}, w_t : z_t, w_{t+1} : z_{t+1}, w_{t+2} : z_{t+2})$，其中每个词 w_i 都被 LDA 赋予一个主题 z_i。通过扩展 Word2Vec，在 Topic2Vec 中，CBOW 结构基于上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 来预测当前词 w_t 和主题 z_t，而 Skip-gram 结构给定当前词

w_t 和主题 z_t 来预测上下文中词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$。

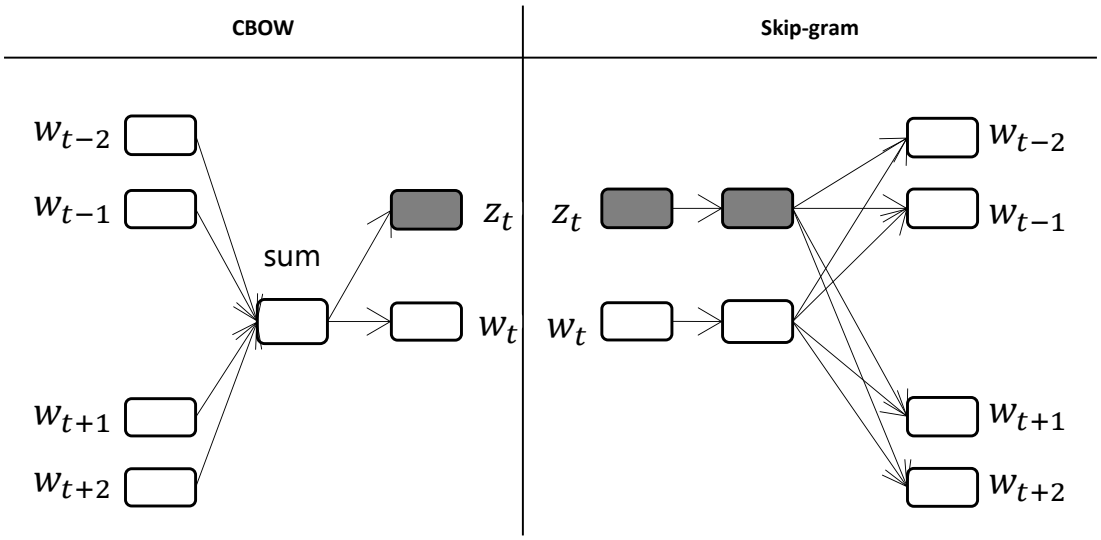

图 3-1: Topic2Vec 结构图

在 Topic2Vec 训练之前，需将原始语料数据通过 LDA 来给语料中每个词赋予一个主题。之后在训练过程中，给定一个文档的词-主题序列 $D = \{w_1 : z_1, ..., w_M : z_M\}$，其中 z_i 是词 w_i 被 LDA 所赋予的主题。训练学习目标通过最大如下 log 似然来定义，分别基于 CBOW 和 Skip-gram 模型:

$$L_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(z_i|w_{cxt})) \tag{3-1}$$

$$L_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le c\le k, c\neq 0}(\log p(w_{i+c}|w_i) + \log p(w_{i+c}|z_i)) \tag{3-2}$$

Topic2Vec 模型旨在学习词向量表示的同时能够学习到主题的向量表示。考虑到简单和高效的解决方法，我们沿用了 Word2Vec 中的优化策略。为了近似最大 $softmax$ 的概率，我们选用负采样（Negative Sampling）而没有用层次 $softmax$（Hierarchical Softmax）。随机梯度下降（Stochastic Gredient Descent, SGD）和后向传播（Back-Propagation, BP）算法用来优化我们的模型参数。同时，可以明显看出 Topic2Vec 模型的复杂度与数据规模呈线性关系，与 Word2Vec 一致。

	Topic_6		Topic_19		Topic_27		Topic_47	
LDA	word	prob.	word	prob.	word	prob.	word	prob.
	food	0.027	drug	0.031	medical	0.033	dog	0.011
	restaurant	0.008	drugs	0.019	hospital	0.024	garden	0.009
	eat	0.008	cancer	0.019	care	0.019	tree	0.009
	more	0.005	study	0.011	patients	0.018	dogs	0.009
	chicken	0.005	patients	0.011	doctors	0.016	plants	0.008
	cooking	0.005	treatment	0.009	health	0.013	trees	0.008
	eating	0.005	fda	0.009	doctor	0.009	animal	0.007
	one	0.005	heart	0.008	patient	0.009	plant	0.007
	good	0.005	risk	0.008	surgery	0.008	animals	0.006
	foods	0.005	more	0.007	center	0.008	zoo	0.006
Topic2Vec	word/topic	cos.	word/topic	cos.	word/topic	cos.	word/topic	cos.
	cheeseburgers	0.564	topic_62	0.618	topic_19	0.519	dogwood	0.498
	meatless	0.535	aricept	0.531	topic_62	0.478	dogwoods	0.494
	smoothies	0.534	topic_27	0.519	neonatal	0.466	topic_33	0.485
	topic_95	0.533	memantine	0.514	topic_13	0.457	bark	0.484
	meatloaf	0.530	enbrel	0.512	anesthesiologists	0.445	fescue	0.483
	tastier	0.530	gabapentin	0.511	anesthesia	0.439	aphids	0.478
	topic_52	0.527	colorectal	0.509	reconstructive	0.437	mulched	0.478
	cheeseburger	0.525	prilosec	0.507	comatose	0.437	azaleas	0.477
	concoctions	0.522	placebos	0.507	hysterectomy	0.433	shrub	0.475
	vegetarians	0.515	intravenously	0.504	ventilator	0.432	camellias	0.472
	Topic_53		Topic_67		Topic_79		Topic_93	
LDA	word	prob.	word	prob.	word	prob.	word	prob.
	government	0.022	www	0.028	computer	0.016	russia	0.028
	africa	0.015	com	0.023	technology	0.010	russian	0.027
	people	0.015	hotel	0.018	phone	0.009	putin	0.017
	african	0.011	travel	0.015	software	0.009	soviet	0.013
	country	0.009	trip	0.011	digital	0.008	moscow	0.012
	international	0.008	night	0.010	apple	0.008	president	0.010
	darfur	0.007	per	0.009	use	0.007	country	0.007
	sudan	0.007	day	0.008	system	0.006	former	0.007
	south	0.007	tour	0.008	microsoft	0.006	state	0.007
	human	0.007	cruise	0.007	up	0.006	union	0.006
Topic2Vec	word/topic	cos.	word/topic	cos.	word/topic	cos.	word/topic	cos.
	mozambique	0.428	fairmont	0.569	wirelessly	0.584	topic_88	0.469
	uganda	0.423	motorcoach	0.553	handhelds	0.573	boris	0.435
	ghana	0.419	stateroom	0.547	desktops	0.572	leonid	0.411
	addis	0.417	uniworld	0.540	pda	0.566	dmitry	0.404
	darfur	0.412	maarten	0.533	smartphone	0.566	vladimir	0.397
	burundi	0.408	tourcrafters	0.529	megabyte	0.562	mikhail	0.397
	lanka	0.407	wyndham	0.528	macbook	0.556	dmitri	0.396
	congo	0.406	cunard	0.527	handheld	0.549	alexei	0.394
	ababa	0.403	safaris	0.522	treo	0.549	eduard	0.392
	darfurians	0.402	trafalgar	0.518	modems	0.548	kasparov	0.391

图 3-2: 对比 LDA 和 Topic2Vec 模型列举出给定主题所包含的主题词

3.3 实验及分析

3.3.1 数据集

实验中，我们选用 English Gigaword Fifth Edition[①]作为训练数据来学习词和主题的向量表示。期间我们随机抽取了一定数量的文档来构建训练集如下描述：我们从包含 411032 文档的子目录 ltw_eng（Los Angeles Times）抽取了 100000 文档，其中每一个文档都包含超过 1000 个字符。另外，我们还去除出现次数少于 5 的英文单词和停用词（stop words）。最终，整个训练集包含大约 42000000 个英文单词，词汇表大小为 102644。

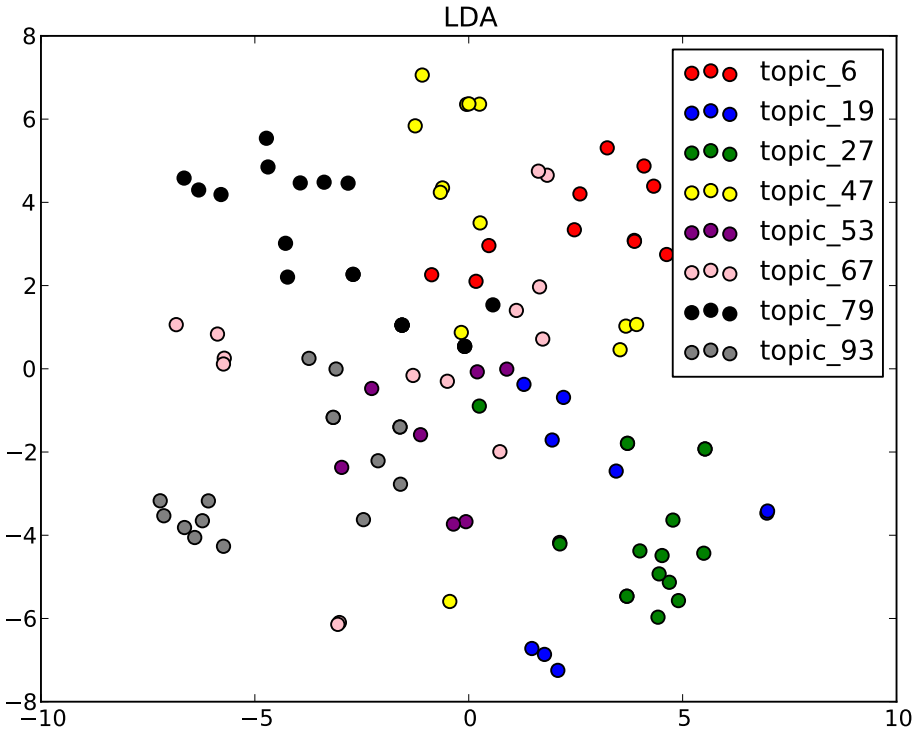

图 3-3: LDA 结果中每个主题所包含主题词基于 t-SNE 的在 2 维空间的映射

3.3.2 评价方法

在实验中，我们在 Skip-gram 结构下运行 Topic2Vec 模型来学习词和主题的分布式向量表示。接着我们将 Topic2Vec 和原有 LDA 进行了如下两个方面的对比：（1）我们基于选定的一些主题来选择出每个主题最相关的主题和词；（2）利用 t-SNE[51] 方法将最相关的主题和词映射至 2 维空间。在这两个过程中，我们通过如下方式来归纳主题所包含的最相关主题词：

- **LDA**：每一个主题是所有词的一个概率分布，因此我们依据词与主题之间的条件概率选择最相关的 $N = 10$ 个词作为主题词。
- **Topic2Vec**：主题和词被同等地映射至同一个低维的向量空间，因此我们可以通过余弦相似度（cosine similarity）来计算词与主题之间的相似度并且依据相似度来排序选择最相关的主题词。

[①] https://catalog.ldc.upenn.edu/LDC2011T07

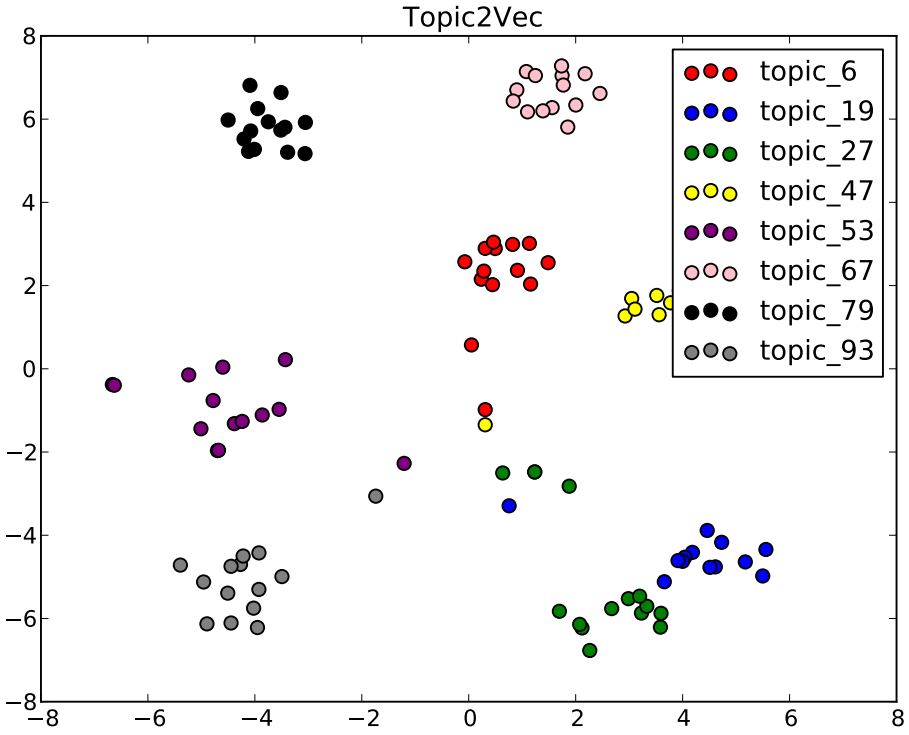

图 3-4: Topic2Vec 结果中每个主题所包含主题和词基于 t-SNE 的在 2 维空间的映射

3.3.3 实验结果分析

图 3-2展示出选定 8 个典型有意义的主题，LDA 和 Topic2Vec 分别给出了最相关的词和主题，我们通过如下细节的分析来理解它们两者之间的不同。如图 3-2所示，对于 Topic_19，LDA 返回的主题词如 *"drugs"*,‘‘*drugs"*,‘‘*cancer"* 和 *"patients"*，而 Topic2Vec 返回 *"aricept"*,‘‘*memantine"*,‘‘*enbrel"* 和 *"gabapentin"*，对于 Topic_27，LDA 返回的主题词如 *"medical"*,‘‘*hospital"*,‘‘*care"*,‘‘*patients"* 和 *"doctors"*，而 Topic2Vec 返回 *"neonatal"*,‘‘*anesthesiologists"*,‘‘*anesthesia"* 和 *"comatose"*。从 LDA 的结果来看，我们只知道 Topic_19 和 Topic_27 共享着相同关于 *patients* 或者 *medical* 的主题，但是我们无法更进一步获得这两个主题之间的不同。但是从 Topic2Vec 结果来看，我们可以轻易地发现 Topic_19 关注一个更加具体的关于药品（*"drugs"*）的主题（*"aricept"*,‘‘*memantine"*,‘‘*enbrel"* 和 *"gabapentin"*），而 Topic_27 则关注另一个更加具体的关于治疗状况（*"treatment"*）的主题（*"neonatal"*,‘‘*anesthesiologists"*,‘‘*anesthesia"* 和 *"comatose"*），Topic_19 和 Topic_27 本质意义是完全不同的。因此我们得出结论，Topic2Vec 在识别两

个相似主题时显得更有区别能力。

利用 t-SNE 降维方法，图3-3和图3-4分别展示出了 LDA 和 Topic2Vec 结果的每一个主题所包含最相关主题词在 2 维空间的映射。明显可以看出，Topic2Vec 的结果相同的主题词产生了更好的聚簇而在不同的主题之间产生更好的分离。相反地，LDA 并不能产生一个良好分离的映射，不同主题的主题词相互混合在一起。

综上，对于每一个主题而言，Topic2Vec 所选定的主题词相比 LDA 更加具有典型性和代表性。最终，Topic2Vec 可以更好的区分不同的主题。

3.4 本章小结

本章首先简要回顾了著名的主题模型潜在狄利克雷分布（LDA），分析了 LDA 方法中长尾以及概率分布表示存在的问题。之后借鉴 Word2Vec 基于神经网络语言模型学习词向量表示的思想，将主题和词同时整合到神经网络语言模型中并提出 Topic2Vec 模型，利用 Topic2Vec 模型，我们可以将隐藏主题与词映射至同一个语义向量空间。原则上，本章的目的在通过 Topic2Vec 模型学习新式的嵌入式主题向量表示。另外，通过实验结果的观察，Topic2Vec 面对相似的主题相比 LDA 展示出更强的区别能力。因此，我们可以得出结论：Topic2Vec 可以更好的建模主题与词之间的语义关系。

但是目前我们仅仅对于 Topic2Vec 和 LDA 进行了定性的评估与分析并且强调了它们二者本质的不同。在未来工作中，我们可以针对二者的不同进行更多细节的定量的分析，包括探索 Topic2Vec 对于传统自然语言处理任务的提升。

第四章　联合学习词及其属性的向量表示

4.1　研究背景

基于我们的认识，词可以表示为词汇表中的下标而文档可以用词袋模型（Bag-of-Words）或 N-gram 模型来表示。尽管这种表示策略简单高效，但同时也面临众多的不足，例如维度灾难（Curse of Dimensionality）、数据稀疏（Data Sparsity）和缺乏捕获词和文档语义信息（Semantic Information）的能力。

而近来，新式的分布式词向量（Dsitributed Word Representations）表示已经在诸多自然语言处理（NLP）任务中取得重大的成功，例如词性标注（POS-Tagging）、命名实体识别（Name Entity Recognition）和语言建模（Language Modeling）等[4,15,52,53]。另外，分布式表示方法已经被扩展到建模超越词级别的概念，例如短语、句子、文档[18]，实体与关系[54,55]、社交和引用网络等[44]。但是，大多数的模型只利用到局部上下文属性并且单独地学习特定任务相关的表示。因此，这些模型都缺乏可以融合多种属性并利用词及其属性联合学习的能力。

因此在本章，我们提出一个可以同时学习词及其属性的分布式表示的统一框架，其中词的属性可以是词的任意特性。自然地，词属性可以关联到句法关系（词性（POS-Tag）和词元（lemma））、文档结构关系（短语、主题和文档）或者其他信息（例如语言（language）、情感（sentiment）和人名（name of person））。如图4–1中例子所示，词 ″*scoring*″ 具有如下属性：″*football*″（主题）、’″*score*″（词元）和”David Villa opened the scoring. Lionel Messi notched a hat trick. There was also an own goal created by pressure from Messi.”（文档）。值得注意的是我们可以扩展我们的模型来学习更多词属性的分布式表示，例如 ″*David Villa opened the scoring*″（句子）、″*positive*″（情感）、″*English*″（语言）、″*NN*″（词性）和 ″*Lionel Messi*″（人名）等。

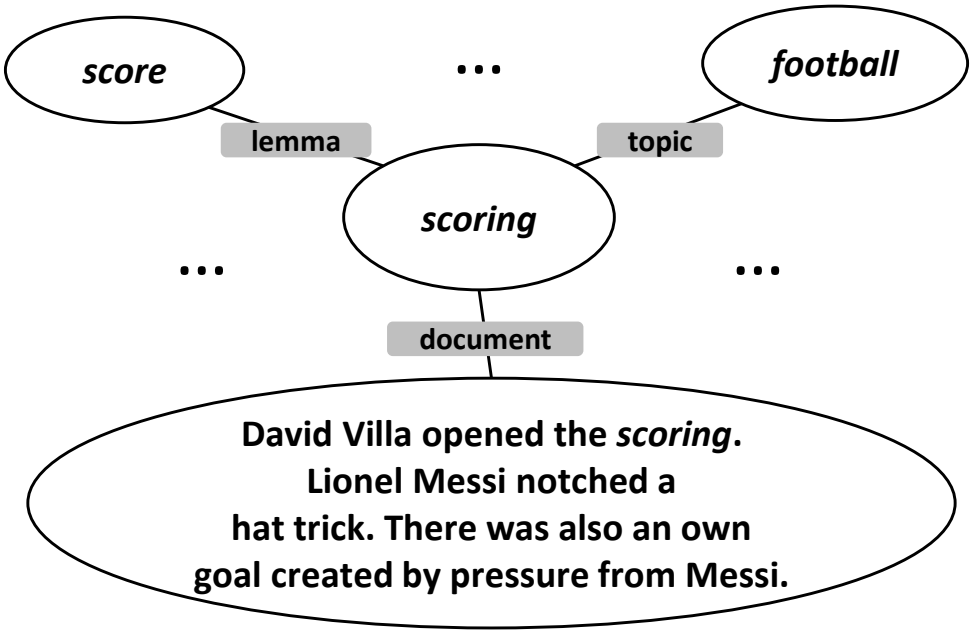

图 4-1: 词 "*scoring*" 及其节点属性图例

表 4-1: Word2Vec[1] 和模型 (TW, DW, LW and TLW) 中所用到词和属性对以及学习目标

模型	词和属性对	学习目标
Word2Vec	词	词向量表示
TW	词: 主题	主题向量表示与提升的词向量表示
DW	词: 文档	文档向量表示
LW	词: 词元	提升的词向量表示
TLW	词: 主题: 词元	提升的词向量表示

特别地，我们研究了三种词的属性包括主题、词元和文档。基于统一的学习框架，我们提出了四个具体的模型如表4-1所示：**TW** 整合主题属性来学习分布式主题向量表示，同时可以学习到提升的词向量表示；**DW** 旨在学习分布式文档向量表示；**LW** 整合词元属性来学习提升的词向量表示；**TLW** 则同时整合主题和词元属性来提升词向量表示。总结我们本章的工作如下：

- 我们提出了一个学习词和属性分布式表示的统一框架，见4.2.1。
- 基于统一的学习框架，我们提出的模型可以学习主题（见4.2.2）和文档（见4.2.3）的分布式表示。
- 我们提出的模型（TW, LW 和 TLW）可以利用额外的属性（主题和词元）来

提升词的向量表示（（见4.2.4））。

实验结果表明我们的模型不仅可以对于特定任务学习到属性表示，而且可以利用额外的属性知识来提升词的向量表示。

4.2 我们的模型

4.2.1 联合学习词和属性向量表示的统一框架

借鉴神经网络语言模型（NPLMs）和 Word2Vec 模型，我们提出了一个联合学习词和属性分布式向量表示的统一学习框架，如图4-2(c) 和 (d) 所示。例如给定一个词序列 $(w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2})$，其中 w_t 是当前词并且被赋予 k 个属性 $(a_{t,1}, ..., a_{t,k})$，CBOW 模型如图4-2(c) 所示，基于上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 来预测当前词 w_t 和 k 个属性 $(a_{t,1}, ..., a_{t,k})$，而 Skip-gram 模型给定当前词 w_t 和 k 个属性 $(a_{t,1}, ..., a_{t,k})$ 来预测上下文词 $(w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2})$。

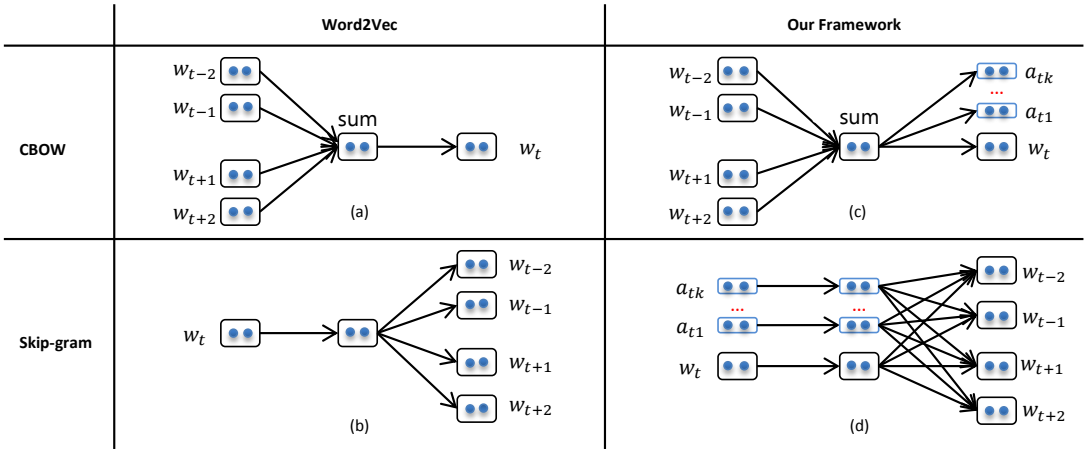

图 4-2: Word2Vec 和统一学习框架中的 CBOW 和 Skip-gram 模型结构对比图

基于我们提出的框架模型，可以明显地看出词和属性在学习过程中可以相互帮助提升来得到更好的向量表示。

特别地，在本章我们考虑了三种属性：主题、词元和文档并分别提出了相应的模型 TW、DW、LW 和 TLW。基于 Harris 及后来 Pantel 等人提出的词分布假设（distributional hypothesis）[56,57]，我们假设词属性也具有相似的分布假设。而且我们的模型也受到如下分布假设的驱动：

- **Hypothesis A**: 出现在相同上下文中的词具有相似的意义。（*"words that occur in the same contexts tend to have similar meanings" (Pantel, 2005)*）.
- **Hypothesis B**: 出现在相同上下文中的词所赋予的主题也是相似的。（*"topics assigned to words that occur in the same contexts tend to be similar"*）.
- **Hypothesis C**: 出现在相同上下文中的词所属的词元也是相似的。（*"lemmas of words that occur in the same contexts tend to be similar"*）.
- **Hypothesis D**: 出现在相同上下文中的词所属的文档也是相似的。（*"documents consisting of words that occur in the same contexts tend to be similar"*）.

4.2.2 TW 模型：学习主题向量表示

如表4-1所示，TW 考虑了词所赋予的主题属性旨在学习分布式主题向量表示。例如给定一个词-主题序列 $(w_{t-2}:z_{t-2}, w_{t-1}:z_{t-1}, w_t:z_t, w_{t+1}:z_{t+1}, w_{t+2}:z_{t+2})$，其中 w_t 是当前词并伴随着一个从 *GibbsLDA++*[①]学到的主题属性 z_t，CBOW 模型基于上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 来预测当前词 w_t 和主题 z_t，而 Skip-gram 模型给定当前词 w_t 和主题 z_t 来预测上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$。

当开始训练时，给定一个词-主题序列 $D = \{w_1:z_1, ..., w_M:z_M\}$，学习目标通过最大如下 log 似然来定义，分别基于 CBOW 和 Skip-gram 模型:

$$L_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(z_i|w_{cxt})) \tag{4-1}$$

$$L_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le c\le k, c\neq 0}(\log p(w_{i+c}|w_i) + \log p(w_{i+c}|z_i)) \tag{4-2}$$

值得注意的是，在上述公式4-1和公式4-2中，第一部分关于 w_i 基于 **Hypothesis A** 而第二部分关于 z_i 基于 **Hypothesis B**。不同于传统 LDA 中作为一个在词表上的概率分布的主题表示方法，TW 模型将词和主题属性映射至同一个语义空间。在相同的语义空间里，词和主题的相似度可以直接采用余弦函数来计算。

[①]http://gibbslda.sourceforge.net/

4.2.3 DW 模型：学习文档向量表示

如表4-1所示，DW 考虑了词所赋予的文档属性旨在学习分布式文档向量表示。例如给定一个词-文档序列 $(w_{t-2}:d_{t-2}, w_{t-1}:d_{t-1}, w_t:d_t, w_{t+1}:d_{t+1}, w_{t+2}:d_{t+2})$，其中 w_t 是当前词并伴随着一个包含这个词的文档属性 d_t，CBOW 模型基于上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 来预测当前词 w_t 和文档 d_t，而 Skip-gram 模型给定当前词 w_t 和文档 d_t 来预测上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$。

当开始训练时，给定一个词-文档序列 $D = \{w_1:d_1, ..., w_M:d_M\}$，学习目标通过最大如下 log 似然来定义，分别基于 CBOW 和 Skip-gram 模型：

$$L_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(d_i|w_{cxt})) \tag{4-3}$$

$$L_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le c\le k, c\neq 0}(\log p(w_{i+c}|w_i) + \log p(w_{i+c}|d_i)) \tag{4-4}$$

值得注意的是，在上述公式4-3和公式4-4中，第一部分关于 w_i 基于 **Hypothesis A** 而第二部分关于 d_i 基于 **Hypothesis D**。文档作为词的属性导致了包含在同一个文档中的所有词更难于区别，因此 DW 模型使得词的向量表示更差。因此在本文中，DW 仅仅关注与学习文档的分布式表示而不能够提升词的向量表示。

4.2.4 提升词向量表示的模型

- **TW** 如前面在4.2.2中描述，TW 模型可以联合学习词向量表示和分布式主题向量表示。同只利用到局部上下文的 Word2Vec 模型相比，TW 模型同时考虑了词和主题属性。自然地，我们期望运用额外的主题属性可以提升原来的词向量表示。
- **LW** 在形态学（Morphology）中，词元（Lemma）[①]是一组词的规范化（canonical）形式。在英语中，例如单词 *"go"*, *"goes"*, *"went"* 和 *"going"* 是同一个词素（Lexeme）的不同形式，以单词 *"go"* 作为词元，如图4-3所示。拥有相同词元的不同单词通常包含着相同的基本意义。

  如表4-1所示，LW 考虑了词所赋予的词元属性旨在提升词的向量表示。

[①] https://en.wikipedia.org/wiki/Lemma

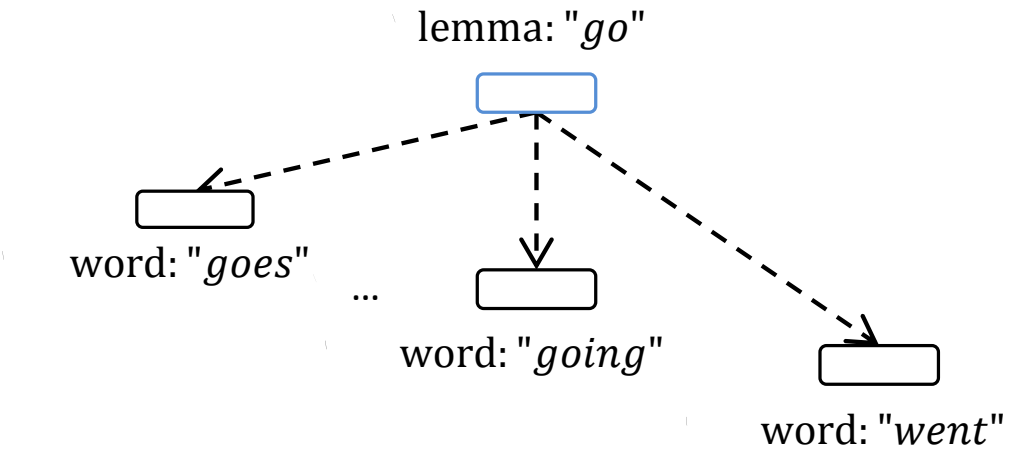

图 4-3: 形态学中一个词元及其变种词的例子

例如给定一个词-词元序列 $(w_{t-2}:l_{t-2}, w_{t-1}:l_{t-1}, w_t:l_t, w_{t+1}:l_{t+1}, w_{t+2}:l_{t+2})$，其中 w_t 是当前词并伴随着一个由 *WordNet Lemmatizer*[①]获取的词元属性 l_t，CBOW 模型基于上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 来预测当前词 w_t 和词元 l_t，而 Skip-gram 模型给定当前词 w_t 和词元 l_t 来预测上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$。

当开始训练时，给定一个词-词元序列 $D = \{w_1 : l_1, ..., w_M : l_M\}$，学习目标通过最大如下 log 似然来定义，分别基于 CBOW 和 Skip-gram 模型：

$$L_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(l_i|w_{cxt})) \tag{4-5}$$

$$L_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le c\le k, c\neq 0}(\log p(w_{i+c}|w_i) + \log p(w_{i+c}|l_i)) \tag{4-6}$$

值得注意的是，在上述公式4-5和公式4-6中，第一部分关于 w_i 基于 **Hypothesis A** 而第二部分关于 l_i 基于 **Hypothesis C**。

- **TLW** 如表4-1所示，TLW 同时考虑了词所赋予的主题和词元属性旨在提升词的向量表示。例如给定一个词-主题-词元序列 $(w_{t-2}:z_{t-2}:l_{t-2}, w_{t-1}:z_{t-1}:l_{t-1}, w_t:z_t:l_t, w_{t+1}:z_{t+1}:l_{t+1}, w_{t+2}:z_{t+2}:l_{t+2})$，其中 w_t 是当前词并伴随着一个主题属性 z_t 和一个词元属性 l_t，CBOW 模型基于上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ 来预测当前词 w_t、主题 z_t 和词元 l_t，而 Skip-gram 模型

[①]http://textanalysisonline.com/nltk-wordnet-lemmatizer

给定当前词 w_t、主题 z_t 和词元 l_t 来预测上下文词 $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$。

当开始训练时，给定一个词-词元序列 $D = \{w_1 : z_1 : l_1, ..., w_M : z_M : l_M\}$，学习目标通过最大如下 log 似然来定义，分别基于 CBOW 和 Skip-gram 模型:

$$L_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(z_i|w_{cxt}) + \log p(l_i|w_{cxt})) \tag{4-7}$$

$$L_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le c\le k, c\neq 0}(\log p(w_{i+c}|w_i) + \log p(w_{i+c}|z_i) + \log p(w_{i+c}|l_i)) \tag{4-8}$$

值得注意的是，在上述公式4-7和公式4-8中，第一部分关于 w_i 基于 **Hypothesis A**，第二部分关于 z_i 是基于 **Hypothesis B**，而第三部分关于 l_i 基于 **Hypothesis C**。

4.2.5 优化和学习过程

同样地，考虑到简单和高效的解决方法，我们沿用了 Word2Vec 中的优化策略。为了近似最大 $softmax$ 的 log 概率，我们选用负采样（Negative Sampling）而没有用层次 $softmax$（Hierarchical Softmax）[16]。随机梯度下降（Stochastic Gredient Descent, SGD）和后向传播（Back-Propagation, BP）算法用来优化我们的模型参数。

特别地，TW 模型集中来学习主题的分布式向量表示，而 DW 则学习文档的分布式向量表示。更甚者，TW、LW 和 TLW 模型可以提升词向量表示。在具体实现中，这些模型首先初始化词和主题等属性的向量表示，然后联合的学习各自的分布式表示。而对于 DW，我们仅仅对文档向量进行了随机初始化，对于词向量采用了预先从大规模和高质量的数据集中学习到的词向量来初始化，然后 DW 模型来学习文档的分布式表示，并且保持词向量表示不变。由我们模型的学习目标函数可以明显看出，模型的复杂度与数据集呈线性关系，与 Word2Vec 保持一致。

	Topic_6		Topic_19		Topic_27		Topic_79	
LDA	word	prob.	word	prob.	word	prob.	word	prob.
	food	0.027	drug	0.031	medical	0.033	computer	0.016
	restaurant	0.008	drugs	0.019	hospital	0.024	technology	0.010
	eat	0.008	cancer	0.019	care	0.019	phone	0.009
	more	0.005	study	0.011	patients	0.018	software	0.009
	chicken	0.005	patients	0.011	doctors	0.016	digital	0.008
	cooking	0.005	treatment	0.009	health	0.013	apple	0.008
	eating	0.005	fda	0.009	doctor	0.009	use	0.007
	one	0.005	heart	0.008	patient	0.009	system	0.006
	good	0.005	risk	0.008	surgery	0.008	microsoft	0.006
	foods	0.005	more	0.007	center	0.008	up	0.006
	dinner	0.004	use	0.007	treatment	0.007	music	0.006
	make	0.004	blood	0.007	hospitals	0.007	video	0.006
	fresh	0.004	women	0.006	heart	0.006	one	0.006
	chef	0.004	disease	0.006	dr	0.006	more	0.005
	made	0.004	percent	0.005	one	0.005	computers	0.005
TW	word/topic	cos.	word/topic	cos.	word/topic	cos.	word/topic	cos.
	cheeseburgers	0.564	topic_62	0.618	topic_19	0.519	wirelessly	0.584
	meatless	0.535	aricept	0.531	topic_62	0.478	handhelds	0.573
	smoothies	0.534	topic_27	0.519	neonatal	0.466	desktops	0.572
	topic_95	0.533	memantine	0.514	topic_13	0.457	pda	0.566
	meatloaf	0.530	enbrel	0.512	anesthesiologists	0.445	smartphone	0.566
	tastier	0.530	gabapentin	0.511	anesthesia	0.439	megabyte	0.562
	topic_52	0.527	colorectal	0.509	reconstructive	0.437	macbook	0.556
	cheeseburger	0.525	prilosec	0.507	comatose	0.437	handheld	0.549
	concoctions	0.522	placebos	0.507	hysterectomy	0.433	treo	0.549
	vegetarians	0.515	intravenously	0.504	ventilator	0.432	modems	0.548
	twinkies	0.514	adderall	0.502	checkup	0.429	camcorders	0.547
	veggie	0.513	inhibitor	0.502	pacemaker	0.428	toshiba	0.545
	panera	0.513	opioid	0.501	aneurysms	0.423	peripherals	0.545
	pepperoni	0.507	oncologists	0.501	respirator	0.423	android	0.544
	condiments	0.504	precancerous	0.501	caesarean	0.422	centrino	0.543

图 4–4: 对比 LDA 和 TW 模型列举出给定主题所包含的主题词

4.3 实验及分析

4.3.1 数据集

我们选用英语 Gigaword[①]作为训练数据来学习基础词向量表示。实际中，我们随机选择了一些文档并且按如下方式构造了两个不同大小的训练集：

- **DS-100k**：我们从包含了 411032 文档数的子目录 ltw_eng（Los Angeles Times）中选取了 100000 的文档，每个文档至少包含 1000 个字符。另外，我们去除掉出现次数少于 5 的词和停用词（stop words）。最终，DS-100k 数据集包含 42000000 的单词而整个词汇表的大小为 102644。
- **DS-500k**：我们还从包含了 1962178 文档数的子目录 nyt_eng（New York Times）中选取了 500000 的文档。在去除出现次数少于 5 的单词和停用词之后，DS-500k 最终大约包含了 2.1 亿的单词而整个词汇表的大小为 232481。

[①]https://catalog.ldc.upenn.edu/LDC2011T07

另外，我们在数据集 DS-100k 上运行了 *GibbsLDA* ++ 和 TW 模型来评估主题的向量表示，并且在 20*NewsGroup*[①] 上进行了文档向量表示的评估。

4.3.2 评估主题向量表示

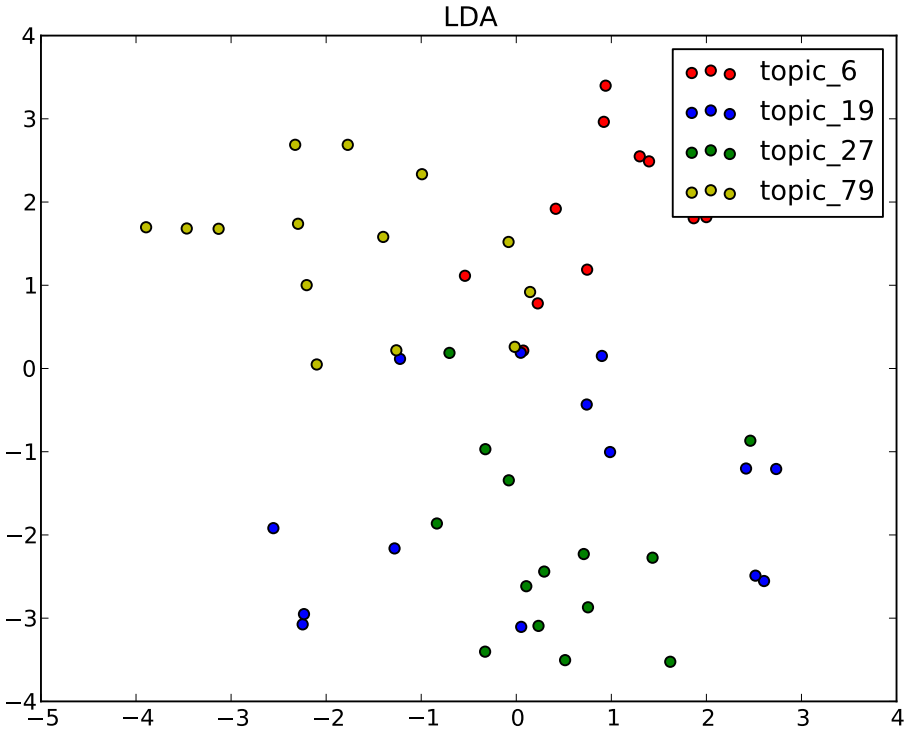

图 4–5: LDA 结果中每个主题所包含主题词基于 t-SNE 的在 2 维空间的映射

TW 模型能够在学习词向量表示的同时学习主题的向量表示。因此我们首先进行实验与原始 LDA[14] 模型来对比评估 TW 模型学习到的主题向量表示。我们按照如下方式来选取主题所包含的主题词：

- **LDA**：所有的主题都表示为词汇表上的概率分布，因此我们依据词与主题之间的条件概率选择最相关的 $N = 15$ 个词作为主题词。
- **TW**：所有的主题和词被同等地映射至同一个低维的向量空间，因此我们可以通过余弦相似度（cosine similarity）来计算词与主题之间的相似度并且依据相似度来排序选择最相关的主题词。

[①] http://qwone.com/ jason/20Newsgroups/

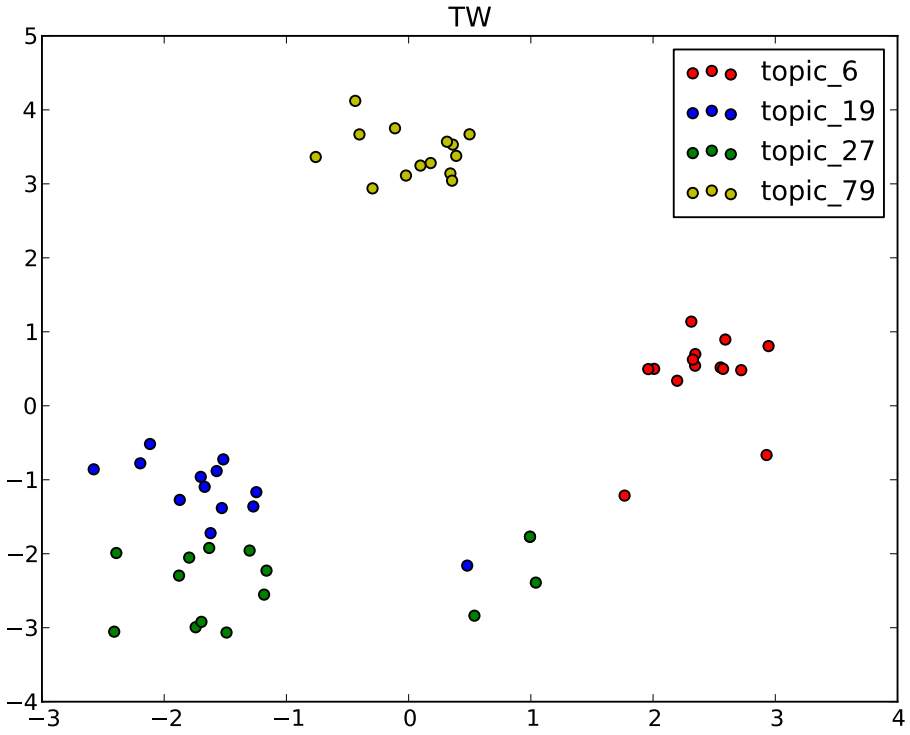

图 4-6: TW 结果中每个主题所包含主题词基于 t-SNE 的在 2 维空间的映射

图4-4展示出选定 4 个典型有意义的主题，LDA 和 TW 分别给出了最相关的词和主题，我们通过如下细节的分析来理解它们两者之间的不同。如图4-4所示，对于 Topic_19，LDA 返回的主题词如 *"drugs"*,‘‘*drugs"*,‘‘*cancer"* 和 *"patients"*，而 TW 返回 *"aricept"*,‘‘*memantine"*,‘‘*enbrel"* 和 *"gabapentin"*，对于 Topic_27，LDA 返回的主题词如 *"medical"*,‘‘*hospital"*,‘‘*care"*,‘‘*patients"* 和*"doctors"*，而TW返回*"neonatal"*,‘‘*anesthesiologists"*,‘‘*anesthesia"* 和*"comatose"*。从 LDA 的结果来看，我们只知道 Topic_19 和 Topic_27 共享着相同关于 *patients* 或者 *medical* 的主题，但是我们无法更进一步获得这两个主题之间的不同。但是从 TW 结果来看，我们可以轻易地发现 Topic_19 关注一个更加具体的关于药品（*"drugs"*）的主题（*"aricept"*,‘‘*memantine"*,‘‘*enbrel"* 和 *"gabapentin"*），而 Topic_27 则关注另一个更加具体的关于治疗状况（*"treatment"*）的主题（*"neonatal"*,‘‘*anesthesiologists"*,‘‘*anesthesia"* 和 *"comatose"*），Topic_19 和 Topic_27 本质意义是完全不同的。明显地可以看出 TW 在识别两个相似主题时显得更有区别能力。

利用 t-SNE[51] 降维方法，图 4-6和图 **??**分别展示出了 LDA 和 TW 结果的每一个主题所包含最相关主题词在 2 维空间的映射。明显可以看出，TW 的结果相同的主题词产生了更好的聚簇而在不同的主题之间产生更好的分离。相反地，LDA 并不能产生一个良好分离的映射，不同主题的主题词相互混合在一起。

综上，对于每一个主题而言，TW 所选定的主题词相比 LDA 更加具有典型性和代表性。最终，TW 可以更好的区分不同的主题。

4.3.3 评估文档向量表示

- **文本分类（Text Classification）** DW 集中学习文档的分布式向量表示，因此我们通过多分类（Multi-Class）的文本分类任务来评估文档向量表示。我们选用标准的 20*NewsGroup* 数据集，其中包含了从 20 个不同的新闻组收集到的大约 20000 的文档。考虑到 20*NewsGroup* 数据集对于训练词向量表示时数据不足的问题，我们首先基于大数据集 DS-500k 来训练得到词的向量表示。然后 DW 模型开始学习 20*NewsGroup* 中的文档的向量表示，并且保持词向量表示不变。

表 4-2: DW 模型与其他模型在 20*NewsGroup* 数据集上的实验对比。其他方法的结果见[2]。粗体表示所有结果中最好的结果。

模型		维度	准确率	精确度	回归率	F1 度量
LDA		80	72.2	70.8	70.7	70.0
PV-DM		400	72.4	72.1	71.5	71.5
PV-DBOW		400	75.4	74.9	74.3	74.3
DW	CBOW	300	74.4	73.9	73.5	73.4
		400	**75.8**	**75.4**	**74.9**	**74.8**
	Skip-gram	300	72.1	71.5	71.2	71.1
		400	72.9	72.4	72.1	72.2

对于每一个文档，DW 返回一个对应的向量作为这个文档的表示。紧接着我们利用采用了“一对剩余所有”（”one vs rest”）模式的 LIBLINEAR[①] 来做多类别的文本分类。为了评价我们模型的有效性，我们将 DW 模型与其他学习文档表示的模型进行了实验对比，包括 LDA 以及最近提出的段落向量表

[①] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

示模型（Paragraph Vector Model）[18]。其中，LDA 将每个文档表示为隐藏主题上的概率分布，而段落向量表示模型则将每个文档表示为一个低维的连续向量，包括分布式记忆模型（distributed memory model, PV-DM）和分布式词袋模型（distributed bag-of-words model, PV-DBOW）。如表4–2所示，DW 模型相比已有模型实现了具有竞争力的结果。值得注意的是，所有这些学习文档的低维连续值向量表示的方法的性能都不如传统的词袋模型（BOW）和 TWE-1 模型[2]，其中后两者方法均利用了语料数据的 TF-IDF 特征来帮助文本分类。

4.3.4 评估提升的词向量表示

最终我们通过如下的标准任务来评估提升的词向量表示：

- **词类比（Word Analogy）** 实验中采用两个数据集用于这个任务。其中谷歌数据集[16] 包含 10675 个句法问题（如 *young:yonger::large:larger*）和 8869 个语义问题（如 *Rome:Italy::Athens:Greece*）。微软数据集[17][①]包含了 8000 个句法问题（如 *good:better::rough:rougher*）。在每一个问题中，第四个单词是缺失的，而这个任务是正确地预测第四个单词。我们采用向量偏移（vector offset）方法[16] 来计算第四个单词的向量表示 $\mathbf{w_{fourth}} = \mathbf{w_{third}} + (\mathbf{w_{second}} - \mathbf{w_{first}})$，如果向量 $\mathbf{w_{fourth}}$ 与正确答案的向量具有最高的余弦相似度，则这个问题算作回答正确。

  我们通过将我们的模型与基准模型 Word2Vec[1] 和目前最好的模型 Glove[38][②]进行了结果对比。如表4–3所示，LW 和 TLW 模型相比 Word2Vec 在多数 Skip-gram 情况下表现更好，而 TW 不能。原有似乎是词元信息在词类比这个任务中相比主题信息可以获得更多的提升。更准确地，在 DS-100k 数据集上和 Skip-gram 情况下，TLW 在谷歌语义上提升了 +8.48% 而 LW 在微软句法上提升了 +5.57%。在更大的 DS-500k 数据集上和 Skip-gram 情况下，LW 在谷歌语义问题上提升了 +3.95% 而在微软句法问题上提升了 +2.72%。一般地，我们可以得出如下的结论：

  - 在词类比任务中，利用附加的词元信息可以得到更好的词向量表示。
  - 特别地在小数据集上，利用附加的词元信息可以使得词向量表示有重大的提升。但当数据集越来越大，额外的信息的作用会减弱。这个结果也与说法“更多的数据通常会打败更好的算法（More data usually beats

[①] http://research.microsoft.com/enus/projects/rnn/default.aspx
[②] http://nlp.stanford.edu/projects/glove/

better algorithms[60]）”保持一致。

表 4-3: 词类比任务上的准确率（%），值越高越好。我们将我们的模型（TW, LW 和 TLW）和基础模型 W2V（Word2Vec）以及目前最好的模型 Glove 进行对比。粗体数据表示每个数据集上的最好结果。时间是在一个 8GB 内存的单机上估计得来。

模型 (维度 =300)		数据集	Google			MSR	时间
			语义关系	句法关系	总体	句法关系	小时
CBOW	W2V	DS-100k	19.08	33.73	27.69	32.36	0.1
	TW	DS-100k	20.42	31.42	26.88	31.47	0.2
	LW	DS-100k	28.64	25.71	26.92	29.35	0.2
	TLW	DS-100k	28.15	27.32	27.67	30.21	0.2
Skip-gram	W2V	DS-100k	27.56	35.63	32.31	29.85	1.1
	TW	DS-100k	31.26	35.13	33.53	29.03	1.2
	LW	DS-100k	33.94	**37.13**(+1.50)	36.16	**35.42**(+5.57)	1.2
	TLW	DS-100k	**36.04**(+8.48)	36.60	**36.37**(+4.06)	34.65	1.3
Glove:iter=5		DS-100k	43.64	40.83	41.99	39.47	1.1
CBOW	W2V	DS-500k	30.57	50.57	41.74	44.97	2.1
	TW	DS-500k	28.12	49.60	40.12	43.93	2.2
	LW	DS-500k	41.80	46.11	44.21	42.43	2.2
	TLW	DS-500k	41.76	47.63	45.04	44.44	2.2
Skip-gram	W2V	DS-500k	41.77	50.63	46.89	43.38	6.8
	TW	DS-500k	41.46	49.46	45.93	41.39	7.4
	LW	DS-500k	**45.72**(+3.95)	**50.86**(+0.23)	**48.59**(+1.7)	**46.10**(+2.72)	7.2
	TLW	DS-500k	44.85	50.58	48.05	45.62	7.7
Glove:iter=5		DS-500k	51.32	49.12	50.09	46.36	6.3
Glove:iter=15		DS-500k	51.88	53.41	52.74	48.32	17.2

值得注意的是在表4-3中，Word2Vec 和我们的模型表现均差于 Glove。原因可能是由于 Glove 中利用全局的词-词共现次数矩阵（word-word co-occurrence counts）而在 Word2Vec 和我们的模型中采用的是局部上下文窗口（local context windows）。因此我们还需进一步的实验来比较这些相关的模型。

- **词相似度（Word Similarity）** 接着我们继续进行第二个词相似度的实验，

基于另一个 WordSim-353 数据集来验证我们模型的有效性，我们一致的将我们的模型和 Word2Vec 和 Glove 进行对比。这里词相似度任务中将人工评估的词与词之间的相似度与词向量计算的结果通过斯皮尔曼等级相关系数来衡量[①]，值越高说明相关性越高，表示词向量的结果与人的判断结果一致。如表4–4所示，我们的模型在词相似度任务中相比 Word2Vec 实现了重大的提升，并且比 Glove 模型更好。

表 4–4: WordSim-353 数据集上我们的模型（TW, LW 和 TLW）和 Word2Vec 以及 Glove 的进行斯皮尔曼等级相关系数（Spearman rank correlation coefficient）对比。粗体表示每个数据集上的最好结果。

模型 (维度 =300)		**语料**	$\rho \times 100$
Glove:iter=5		DS-100k	51.9
CBOW	Word2Vec	DS-100k	55.6
	TW	DS-100k	62.6
	LW	DS-100k	63.9
	TLW	DS-100k	65.0
Skip-gram	Word2Vec	DS-100k	61.5
	TW	DS-100k	63.7
	LW	DS-100k	**65.4**
	TLW	DS-100k	63.5
Glove:iter=5		DS-500k	50.8
Glove:iter=15		DS-500k	50.9
CBOW	Word2Vec	DS-500k	63.7
	TW	DS-500k	62.2
	LW	DS-500k	65.9
	TLW	DS-500k	**67.5**
Skip-gram	Word2Vec	DS-500k	65.8
	TW	DS-500k	63.7
	LW	DS-500k	64.6
	TLW	DS-500k	63.9

现在可以看出利用附加的主题和词元知识可以非常重大的提升原有的词向量表示，尤其是对于小数据集。因此我们可以想到在一些特定的领域内利

[①]https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

用已知的额外知识可以缓解数据不足的问题。

4.4 本章小结

在本章内容中，我们首先分析了现实中词通常伴随着很多特征或者属性一起出现，例如词的主题（topic）、词性（POS-Tag）、词元（lemma）、所属文档（document）、所在语言（language）或者人名（name of person）等等。而现实中大多数的模型往往是学习特定任务的向量表示并且缺乏融合多种信息来学习词向量表示的能力。因此我们重点提出了一个联合学习词和属性向量表示的统一框架。特别地，我们考虑了主题、词元和文档属性并且分别给出了四个具体的模型（TW, DW, LW 和 TLW）。从实验内容的观察和分析可以看出，基于提出的统一学习框架，我们的模型不仅可以学习到主题和文档的分布式向量表示，并在各自对应的任务中实现了明显和有竞争力的结果，而且还可以同时利用附加的知识信息来提升词的向量表示。

最后我们想强调的是我们所提出的框架具有灵活性和可扩展性，可以整合更多的属性。在未来的工作中，我们可以探索词的其他属性的向量表示学习，例如情感分析（sentiment analysis）中的情感（sentiment）、词性标注（POS-Tagging）中的词性（POS-Tags）和命名实体识别（name entity recognition, NER）中的人名（name of person）。

第五章　融合词向量与主题模型

5.1　融合词向量主题模型

5.2　实验及分析

5.3　本章小结

第六章　总结与展望

本文在第 **??**章中，

致　　谢

首先感谢我的父母

参考文献

[1] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[2] LIU Y, LIU Z, CHUA T-S, et al. Topical Word Embeddings.[C] // AAAI. 2015 : 2418 – 2424.

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

[4] COLLOBERT R, WESTON J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C] // Proceedings of the 25th international conference on Machine learning. 2008 : 160 – 167.

[5] CHIANG D. A hierarchical phrase-based model for statistical machine translation[C] // Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005 : 263 – 270.

[6] LAFFERTY J, MCCALLUM A, PEREIRA F C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[J], 2001.

[7] ZHAO H, HUANG C-N, LI M. An improved Chinese word segmentation system with conditional random field[C] // Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing : Vol 1082117. 2006.

[8] LEE H, GROSSE R, RANGANATH R, et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations[C] // Proceedings of the 26th Annual International Conference on Machine Learning. 2009 : 609 – 616.

[9] LEE H, PHAM P, LARGMAN Y, et al. Unsupervised feature learning for audio classification using convolutional deep belief networks[C] // Advances in neural information processing systems. 2009 : 1096 – 1104.

[10] ZHANG Y, JIN R, ZHOU Z-H. Understanding bag-of-words model: a statistical framework[J]. International Journal of Machine Learning and Cybernetics, 2010, 1(1-4): 43 – 52.

[11] SALTON G, MCGILL M J. Introduction to modern information retrieval[J], 1986.

[12] LANDAUER T K, FOLTZ P W, LAHAM D. An introduction to latent semantic analysis[J]. Discourse processes, 1998, 25(2-3): 259 – 284.

[13] HOFMANN T. Probabilistic latent semantic indexing[C] // Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999: 50 – 57.

[14] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993 – 1022.

[15] BENGIO Y, DUCHARME R, VINCENT P, et al. A Neural Probabilistic Language Model[C] // JOURNAL OF MACHINE LEARNING RESEARCH. 2003.

[16] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C] // Advances in neural information processing systems. 2013: 3111 – 3119.

[17] MIKOLOV T, YIH W-T, ZWEIG G. Linguistic Regularities in Continuous Space Word Representations.[C] // HLT-NAACL. 2013: 746 – 751.

[18] LE Q V, MIKOLOV T. Distributed representations of sentences and documents[J]. arXiv preprint arXiv:1405.4053, 2014.

[19] SOCHER R, HUANG E H, PENNIN J, et al. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection[C] // Advances in Neural Information Processing Systems. 2011: 801 – 809.

[20] TANG D, WEI F, QIN B, et al. Coooolll: A deep learning system for Twitter sentiment classification[C] // Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). 2014: 208 – 212.

[21] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[22] CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[23] NIESLER T, WOODLAND P. Variable-length category-based n-grams for language modelling[M]. [S.l.]: University of Cambridge, Department of Engineering, 1995.

[24] NIESLER T R, WOODLAND P C. A variable-length category-based n-gram language model[C] // Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on: Vol 1. 1996: 164 – 167.

[25] ROSENFELD R. A maximum entropy approach to adaptive statistical language modeling[J], 1996.

[26] CHELBA C. A structured language model[C] // Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics. 1997: 498 – 500.

[27] ROSENFELD R. A whole sentence maximum entropy language model[C] // Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on. 1997: 230 – 237.

[28] BRANTS T, POPAT A C, XU P, et al. Large language models in machine translation[C] // In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2007.

[29] HINTON G E. Learning distributed representations of concepts[C] // Proceedings of the eighth annual conference of the cognitive science society: Vol 1. 1986: 12.

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

[31] MORIN F, BENGIO Y. Hierarchical Probabilistic Neural Network Language Model.[C] // Aistats: Vol 5. 2005: 246 – 252.

[32] MNIH A, HINTON G E. A scalable hierarchical distributed language model[C] // Advances in neural information processing systems. 2009 : 1081 – 1088.

[33] MIKOLOV T, KOMBRINK S, BURGET L, et al. Extensions of recurrent neural network language model[C] // Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. 2011 : 5528 – 5531.

[34] GUTMANN M U, HYVÄRINEN A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics[J]. The Journal of Machine Learning Research, 2012, 13(1) : 307 – 361.

[35] MNIH A, TEH Y W. A fast and simple algorithm for training neural probabilistic language models[J]. arXiv preprint arXiv:1206.6426, 2012.

[36] dos SANTOS C N, GATTI M. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts.[C] // COLING. 2014 : 69 – 78.

[37] TANG D, WEI F, YANG N, et al. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification.[C] // ACL (1). 2014 : 1555 – 1565.

[38] PENNINGTON J, SOCHER R, MANNING C D. Glove: Global Vectors for Word Representation.[C] // EMNLP : Vol 14. 2014 : 1532 – 1543.

[39] QIU S, CUI Q, BIAN J, et al. Co-learning of Word Representations and Morpheme Representations.[C] // COLING. 2014 : 141 – 150.

[40] REISINGER J, MOONEY R J. Multi-prototype vector-space models of word meaning[C] // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2010 : 109 – 117.

[41] YOGATAMA D, FARUQUI M, DYER C, et al. Learning word representations with hierarchical sparse coding[J]. arXiv preprint arXiv:1406.2035, 2014.

[42] MNIH A, KAVUKCUOGLU K. Learning word embeddings efficiently with noise-contrastive estimation[C] // Advances in Neural Information Processing Systems. 2013 : 2265 – 2273.

[43] KALCHBRENNER N, GREFENSTETTE E, BLUNSOM P. A convolutional neural network for modelling sentences[J]. arXiv preprint arXiv:1404.2188, 2014.

[44] TANG J, QU M, WANG M, et al. Line: Large-scale information network embedding[C] // Proceedings of the 24th International Conference on World Wide Web. 2015 : 1067 – 1077.

[45] KIROS R, ZEMEL R, SALAKHUTDINOV R R. A multiplicative model for learning distributed text-based attribute representations[C] // Advances in Neural Information Processing Systems. 2014 : 2348 – 2356.

[46] WOLFINGER R. Laplace's approximation for nonlinear mixed models[J]. Biometrika, 1993, 80(4) : 791 – 795.

[47] JORDAN M I, GHAHRAMANI Z, JAAKKOLA T S, et al. An introduction to variational methods for graphical models[J]. Machine learning, 1999, 37(2) : 183 – 233.

[48] WAINWRIGHT M J, JORDAN M I. Graphical models, exponential families, and variational inference[J]. Foundations and Trends® in Machine Learning, 2008, 1(1-2) : 1 – 305.

[49] ANDRIEU C, DE FREITAS N, DOUCET A, et al. An introduction to MCMC for machine learning[J]. Machine learning, 2003, 50(1-2) : 5 – 43.

[50] GRIFFITHS T L, STEYVERS M. Finding scientific topics[J]. Proceedings of the National Academy of Sciences, 2004, 101(suppl 1) : 5228 – 5235.

[51] Van der MAATEN L, HINTON G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research, 2008, 9(2579-2605) : 85.

[52] TURIAN J, RATINOV L, BENGIO Y. Word representations: a simple and general method for semi-supervised learning[C] // Proceedings of the 48th annual meeting of the association for computational linguistics. 2010 : 384 – 394.

[53] HUANG Z, XU W, YU K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

[54] BORDES A, USUNIER N, GARCIA-DURAN A, et al. Translating embeddings for modeling multi-relational data[C] // Advances in Neural Information Processing Systems. 2013 : 2787 – 2795.

[55] SOCHER R, CHEN D, MANNING C D, et al. Reasoning with neural tensor networks for knowledge base completion[C] // Advances in Neural Information Processing Systems. 2013 : 926 – 934.

[56] HARRIS Z S. Distributional structure[J]. Word, 1954, 10(2-3) : 146 – 162.

[57] PANTEL P. Inducing ontological co-occurrence vectors[C] // Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005 : 125 – 132.

[58] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep Learning[H/OL]. 2016. http://goodfeli.github.io/dlbook/.

[59] DENG L, YU D. Deep learning: Methods and applications[J]. Foundations and Trends in Signal Processing, 2014, 7(3–4) : 197 – 387.

[60] RAJARAMAN A. More data usually beats better algorithms[J]. Datawocky Blog, 2008.

附录 A　图论基础知识

简历与科研成果

基本信息

牛力强，男，汉族，1991 年 10 月出生，出生于甘肃省甘谷县。

教育背景

2013 年 9 月 — 2016 年 6 月 南京大学计算机科学与技术系 硕士
2009 年 9 月 — 2013 年 6 月 南京大学计算机科学与技术系 本科

攻读硕士学位期间完成的学术成果

1. Liqiang Niu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. A Unified Framework for Jointly Learning Distributed Representations of Word and Attributes. In Proceedings of 7th Asian Conference on Machine Learning (ACML 2015) November 20-22, 2015, Hong Kong, JMLR: Workshop and Conference Proceedings 45:143–156, 2015
2. Liqiang Niu, Xinyu Dai, Jianbing Zhang, and Jiajun Chen. Topic2Vec: Learning Distributed Representations of Topics. In Proceedings of 2015 International Conference on Asian Language Processing (IALP 2015) pp. 193-196, 24-25 October, 2015, Suzhou, China

攻读硕士学位期间参与的科研课题

1. 国家自然科学基金面上项目 “无线传感器网络在知识获取过程中的若干安全问题研究”（课题年限 2010 年 1 月 — 2012 年 12 月），负责位置相关安全问题的研究。
2. 江苏省知识创新工程重要方向项目下属课题 “下一代移动通信安全机制研究”（课题年限 2010 年 1 月 — 2010 年 12 月），负责 LTE/SAE 认证相关的安全问题研究。

学位论文出版授权书

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：_____

_____年_____月_____日

论文题名	基于神经网络语言模型的文本向量表示研究				
研究生学号	MF1333036	所在院系	计算机科学与技术系	学位年度	2013
论文级别	☑ 硕士 ☐ 硕士专业学位 ☐ 博士 ☐ 博士专业学位 （请在方框内画勾）				
作者电话	15996276729		作者 Email	simpleniulq2013@gmail.com	
第一导师姓名	戴新宇 副教授		导师电话		

论文涉密情况：

☐ 不保密

☑ 保密，保密期：_____年_____月_____日 至 _____年_____月_____日

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。