中山大學
SUN YAT-SEN UNIVERSITY

本 科 生 毕 业 论 文

题　　目：面向自然语言理解的多向量表达学习算法

院　　系：软件学院

专　　业：软件工程（数字媒体技术）

学生姓名：张锐

学　　号：10389126

指导教师：林倞（教授）

二〇 一四 年　三　月

摘　要

词汇的表达在自然语言处理领域有着基础性的重要意义。最原始的词表空间向量表达存在诸多问题：向量的维度很高，但向量是离散表达，每个词的表达只有一维有信息；不同词汇的向量表达相互正交，词汇之间的联系难以表达。通过机器学习方法产生的较低维分布式词向量可以很好地克服这些不足，并且被证明能够提升许多自然语言处理任务的性能。其中的一大热门分支是无监督词向量。无监督词向量学习不需要经过标注的语料数据，可以充分发挥互联网大数据的优势。无监督词向量可以作为初始参数输入有监督学习算法，以提升有监督算法的学习效果。现有的无监督词向量学习算法多数使用单向量表达，即一个词汇仅用一个向量表达。然而在自然语言中，一个词汇往往有多个词义。使用单个向量表达词汇，会将词汇的多种词义混淆，对自然语言处理任务的效果造成不良影响。为消除算法和自然语言现象之间的这种不一致，本文提出一种新的无监督学习算法，通过上下文相关语言模型和词汇消歧义的联合学习，为词汇产生能够表达不同词义的多个向量。实验中，使用本算法学习产生的词向量预测人类对成对词汇（有上下文和无上下文）语义相似性的数值评分，比较预测结果与采集的人工标注结果的相似性。实验取得了良好结果，与单向量算法相比，本算法在预测准确度上有了较大提高。

关键词：　　词汇向量表达；无监督学习；自然语言处理；多义词

ABSTRACT

Word representation has a fundamental importance in the field of natural language processing. Primitive vocabulary-space represention is high-dimentional but discrete. It fails to capture the similarity and connection between words due to its orthogonality. Distributed vector-space representations produced by language models can cope with these shortcomings, and are proven successful in improving performance of various NLP tasks. Unsupervised vector-space representation is a major branch of vector-space representations. With no need for human labeled data, unsupvervised method can make use of the enormous amount of corpus on the Internet. Unsupervised vector-space representations helps improve performance of supervised methods as initial parameters or extra features. Most of exisiting vector-space models use single “prototype” vector to represent each word. In natural languages, however, homonymy and polysemy are very common phenomena; one word may have multiple senses. Inconsistence exists between single-prototype representation and multi-sense natural language words. To cope with this inconsistence, this thesis presents a novel method that produces multi-prototype word representations by co-learning context dependent language model and word sense disambiguation. Experiments are conducted of predicting human judgement of pairwise semantic word similarity between isolated words and words in context. Result shows the superiority of this method over existing single-prototype methods.

Keywords: Vector-space Word Representation, Unsupervised Learning, Natural Language Processing, Multi-sense words

目　录

第一章　概述/引言

1.1　词汇表达问题的背景和意义

对于人来说，自然语言中的词汇用文字（书面语）或音节（口头语）表示。使用计算机处理自然语言时，词汇必须用计算机能够处理的数学形式进行表达。词汇的表达方式对自然语言处理具有重大意义，直接影响自然语言处理算法的效果。

1.2　词汇表达问题的描述

词汇表达问题的本质，在于用数学的方式表示自然语言中词汇的性质。在自然语言中，词汇有很多属性，包括词性、时态、词义。词汇与词汇之间还存在许多联系，包括同义、反义、词性变化、语义变化等。自然语言的词汇还普遍存在一词多义现象，这进一步增加了词汇表达的难度。

解决问题的难点在于，如何发现词汇的属性，以及如何将这些属性以数学形式进行表达。对于词汇的属性，语言学的研究以及字典编纂的工作已经总结出了许多的规律。但是语言的应用领域众多，语言本身也在进行持续的发展进化，人难以穷尽所有的语言规律，人总结出的规律也不保证能适用于所有的语料。人工为每个词汇设计数学表达几乎是不可能的。因此，通过机器学习方法自动生成词汇的数学表达，是目前被学术界和工业界普遍采用的策略。

1.3　本文的工作

本文提出了一种新的算法，通过无监督机器学习生成词汇的多向量表达。其中向量的维度数、每个词汇的向量数量均可作为参数指定。本文的主要创新点有二：第一，将上下文相关语言模型和词汇消歧义两个任务结合起来，进行联合学习；第二，使用多个向量表示表示单个词汇。

1.4　论文结构简介

本文第二章将对词汇向量表达问题进行概括性的介绍，并对现有的词汇模型的优缺点进行分析。第三章将介绍本文提出的新的多向量词汇表达模型，包括其

理论基础和具体的概率模型。第四章将介绍第三章所提出的模型的训练算法，以及一些优化的技巧。第五章将介绍实验，包括实验所用测试数据集、参数设置、实例分析、量化对比。第六章将对本文进行总结，并提出对未来发展的展望。

第二章　词汇向量表达综述

2.1　词汇表达

词汇是自然语言的基本语义单位。对于人来说，词汇通过文字符号（书面）或者声音（口头）来表达。如何在计算机中表达词汇，是计算机自然语言处理中的一个基本问题。

2.1.1 离散词汇表达

离散表达是最为原始的一种词汇表达方法。假设词表有 N 个词，每个词使用一个 N 维的向量表示。向量的每维唯一对应词表中的一个词。每个词的向量表达中，只有对应该词的一维数值为 1，其他维数值均为 0。

离散词汇表达简单直观，但是具有严重的缺陷：表达向量维数高，但是非常稀疏，对存储空间是一种浪费；表达向量的信息量少，不足以充分表达词汇的语义信息；不同词汇的表达两两正交，难以使用线性代数运算衡量词汇之间的关系。

为克服离散词汇表达方法的缺点，分布式词汇表达方法被提出。

2.1.2 分布式词汇表达

分布式词汇表达（Distributed word representation），又称词向量表达，指使用低维、稠密的实数向量来表达词汇。词汇的语义信息通过向量各维的数值进行表达。随着互联网上文本数据的爆发性增长，以及深度学习技术的发展，分布式词汇表达广泛被应用在许多的文本挖掘任务中。许多研究者也提出了有效、高效的分布式词汇表达模型和训练算法。

自 Yoshua Bengio 的人工神经网络语言模型[1]发表以来，出现了许多基于人工神经网络模型的词向量表达学习算法。其中比较重要的工作有：Morin 和 Bengio 的层次人工神经网络模型[2]、Thomas Mikolov 等的 CBOW 和 Skip-gram 模型[3][4]、Ronan Collobert 等人的模型[5]。

词向量表达通常使用无监督机器学习算法产生，多数算法使用上下文相关的语言模型，以人工神经网络或其变种作为概率模型。无监督机器学习不需要人工对训练数据进行标注，可以充分利用互联网上海量的语料数据进行训练。训练所得的词向量可用于机器翻译、文本挖掘、文档分类等多种领域。

2.2 现有词汇向量表达模型的不足

现有的词向量表达模型及其训练算法依然存在不足，主要体现在：单向量难以完整表达语义信息；待训练参数过多，训练效率低，难以向大数据扩展。

2.2.1 单向量表达

大多数现有的词向量表达模型假设每个词汇只用一个向量表达。然而在自然语言中，一词多义的现象是非常普遍的，且一个词汇的多个词义之间的差距可能非常大。用同一个向量表达一个词汇的不同词义是不合理的，且会损害模型的表达效果。在这些模型中，单向量表达与词汇的多义性产生矛盾，限制了模型的表达能力，及其在自然语言处理任务中的表现。

近期的一些研究工作尝试通过为不同的词义训练不同的词向量来克服单向量表达的局限性。Reisinger 和 Mooney 提出了通过上下文聚类来训练多向量表达模型[6]。Huang 等人在这一工作的基础上，将人工神经网络模型和上下文聚类方法结合起来[7]。具体来说，这些方法将训练过程分为两步：第一步，基于“一个词汇只有一个词义”的假设，使用深度神经网络训练单向量词汇表达；第二步，基于上下文特征进行聚类，为多义词生成多个向量表达。这些多向量表达模型在许多自然语言处理任务中的表现与传统的单向量表达模型相比，有显著提升。

然而，现有的多向量表达模型在面对互联网带来的海量训练语料时，依然存在局限性。一方面，这些模型采用层数较深的网络，这导致整个模型中待训练的参数非常多，训练耗时较长。另一方面，这些模型的性能受聚类算法的影响很大，需要耗费时间实现和优化聚类算法。除此之外，基于聚类的模型缺乏在概率学上的理论解释，使得这些模型难以被应用在一些文本挖掘的任务，比如语言建模上。

2.3 相关工作

2.3.1 人工神经网络词汇表达模型

自 2003 年 Bengio 等人发表人工神经网络概率语言模型以来，出现了众多基于人工神经网络训练词向量表达的方法。2005 年，Morin 和 Bengio 发表了基于层次结构的 Softmax 模型[2]，使用树状层次结构代替原有的平铺结构，将算法的时间复杂度从O(n)降为O$(log(n))$，大大提高了算法的效率。2010 年，Mikolov 等人发表了 Word2Vec 项目，提出了 CBOW（Continuous Bag-of-word，连续词袋）模型，和 Skip-Gram 模型[3]。Word2vec 项目的两种模型对人工神经网络进行了进一步简化，并且吸收了 Morin 的树状层次结构，使得训练效率进一步提高。Mikolov 还提出了一种语义类比问题，用于评价词向量表达的质量。2013 年的 NIPS 会议上，Mikolov 等人发表了 Word2Vec 项目的改进版本，加入了负样本概率模型，再一次提高了模型的训练效率。

2.3.2 多向量词汇表达模型

上述词汇向量表达模型均基于“单向量假设”，即一个词汇只用一个向量进行表达。与前述方法不同，Huang 等人在 2012 年发表的人工神经网络模型使用了多个向量对一个词的不同词义进行表达，并将局部上下文特征和全文上下文特征结合起来，在估算词汇相似度的实验中取得了较好结果[3]。在 Huang 等人的方法中，多向量表达是通过对词的上下文特征进行聚类产生。上下文特征通过一个三层的人工神经网络基于“单向量假设”训练产生。每个聚类中心被用作对该词义的向量表达。Huang 等人的模型，及其所发表的实验结果，对本文有重要的启示作用。

值得注意的是，Reisinger 等人在 2010 年提出的“多原型词义向量空间表达”模型[]首先使用了多个向量对词的不同词义进行表达。该方法通过对对词的上下文特征进行聚类产生产生不同词义的不同向量表达。Huang 等人的方法实际上也基于 Reisinger 等人的这一方法。然而，Reisinger 的聚类方法使用的特征通过 TF*IDF（词频*文档频率的倒数）计数模型（又称“分布关系表达”[8]），而不是人工神经网络产生。因此，这里认为 Reisinger 等人的方法与本文关联度不高。

第三章　词汇多向量表达方法

3.1　词汇向量表达的理论基础

本文所介绍的词汇向量表达方法，以及大多数现有的词汇向量方法，都是基于概率语言模型的。概率语言模型本节将介绍词汇向量表达的理论基础。词汇向量表达的理论基础主要有两方面：如何表示词的特征，以及如何建立概率模型。

3.1.1 分布假设-用上下文作为词汇原始特征

如何从语料中提取词汇的原始特征，是词汇向量表达的前置基础问题。最原始而平凡的特征，就是词汇本身。为每一个词汇赋予一个唯一的标识，作为词汇的特征。计算机中，这种特征常用 1-of-v 的向量表示，即一个向量中有且仅有一维的值为 1，其他维的值全部为 0。这种表达方式有很大的局限性：稀疏向量信息量少，两两正交导致难以进行比较和计算。

1993 年，Pereira 等人提出了“分布假设”[9]：相似的词汇，应该出现在相似的上下文中。这一理论使得上下文特征成为了一种有效的词汇特征。词汇的上下文特征通常是不稀疏，也不正交的，这使得词汇特征之间的计算成为可能。词汇上下文的相似度和词汇语义的相似度正相关，使得词汇之间的相似度可以计算。而词汇相似度在许多自然语言处理任务中都有着重要意义。

3.1.2 概率语言模型

基于分布假设，词汇的上下文特征成为了有效的词汇特征。换个角度看，词汇的上下文特征即是词汇之间共同出现在同一上下文范围内的规律。自然地，这一规律适合使用概率模型进行描述。

求词汇的向量表达这一问题，可以转化为求一个概率模型，使得这一概率模型能最好地反映“词汇之间共同出现”这一事件的概率分布。

3.2　多向量词汇表达模型

本节将详细介绍本文提出的多向量词汇表达模型。由于新的模型基于 Mikolov 等人的 Skip-gram 模型[3]，本节将首先简要介绍 Skip-gram 模型，然后介

绍新的模型。

3.2.1 Skip-gram 模型

使用一个中心词的上下文范围内的其他词预测中心词的下一个词，这是建立概率语言模型的一种经典做法。与经典做法相反，Skip-gram 模型选择了用中心词预测上下文词的方式来建立概率语言模型。

设中心词为w_I，中心词上下文范围内的另一个词为w_O，条件概率$\mathrm{P}(w_O|w_I)$按以下方式建模：

$$\mathrm{P}(w_O|w_I) = \frac{\exp(V_{w_I}^T U_{w_O})}{\sum_{w \in W} \exp(V_{w_I}^T U_w)} \tag{3.1}$$

其中，W是包含所有词汇的词表集合。$U_w \in R^d$和$V_w \in R^d$分别表示词w的d维的“输入端”（人工神经网络输入端）和“输出端”（人工神经网络输出端）表达向量。需要注意的是，所有词输入端表达向量和输出端表达向量都是模型所需要学习的参数。

整个 Skip-gram 模型是一个三层的全连接人工神经网络。第一层（输入层）和第三层（输出层）的节点数为$|W|$（即词表大小），第二层（隐含层）的节点数为定义的表达向量维数d。U和V分别为第一第二层，第二第三层之间的连接权值矩阵。U和V也分别称作输入端表达向量矩阵和输出端表达向量矩阵。

综述中提到的其他人工神经网络往往采用四层或者更深的人工神经网络，且往往会使用非线性神经元激活函数。Skip-gram 模型使用三层网络，中间层使用线性激活函数，与其他模型相比，待训练的参数更少，训练效率有显著提高。

3.2.2 多向量 Skip-gram 模型

本文提出的多向量 Skip-gram 将多向量表达与 Skip-gram 模型进行了整合。与原本的 Skip-gram 模型类似，新模型的目标是对条件概率$\mathrm{P}(w_O|w_I)$进行建模。新模型同样使用三层人工神经网络，中间层使用线性激活函数。

新模型与 Skip-gram 模型的最主要区别在于：给定中心词w_I，其上下文范围内的w_O使用一个有限混合模型描述。混合模型中的每个子模型代表该词的一种词义*，以一个d维的向量进行表示。

具体来说，假设词w有N_w个词义，且以第h_w（$h_w \in \{1, 2 \ldots, N_w\}$）个词义出

现在语料中，那么条件概率$P(w_O|w_I)$展开可得：

$$P(w_O|w_I) = \sum_{i=1}^{N_{w_I}} P\left(w_O\middle|h_{w_i}=i,w_I\right)P\left(h_{w_i}=i\middle|w_I\right) \tag{3.2}$$

$$= \sum_{i=1}^{N_{w_I}} \frac{\exp\left(U_{w_O}^T V_{w_I,i}\right)}{\sum_{w\in W}\exp\left(U_w^T V_{w_I,i}\right)} P(h_{w_i}=i|w_I) \tag{3.3}$$

其中$V_{w_I,i} \in R^d$表示词w_I的第i个表达向量。这一公式所表达的意思是：条件概率$P(w_O|w_I)$是一个加权平均。参与加权平均的每一项分别是：给定词w_I以第i个词义存在，w_O出现在w_I上下文范围内的概率$P(w_O|h_{w_i}=i,w_I)$。权值则是词w_I以第i个词义存在的先验概率。

这一模型背后的思路十分直观：对于一个词的不同词义来说，它们的上下文词的分布通常是有区别的。例如，当 bank 一词表示“银行”时，其上下文词往往和金融领域有关；当 bank 表示“河岸”时，其上下文词往往和地理、环境有关。

公式(3.3)中的 Softmax 概率模型的计算时间复杂度较高，因为其分母$\sum_{w\in W}\exp\left(U_w^T V_{w_I,i}\right)$的时间复杂度为$O(|W|)$。$|W|$通常较大，根据训练语料不同，范围可能从数万到数百万。考虑到本模型中每个词有多个表达向量，$|W|$的数量级将会更大，模型的计算复杂度会更高。

3.2.3 使用 Softmax 树降低计算复杂度

为了克服 Softmax 回归的计算复杂度问题，学术界提出了一些改进的算法，例如 Softmax 树（Hierarchical Softmax Tree）[2]，负样本（Negative Sampling）[4]。本文算法采用 Softmax 树：词表中的所有词汇被组织成一颗树，树的每个叶节点代表一个词。一个二值向量$b^{(w_O)} \in \{-1,+1\}^{L_{w_O}}$表示从树的根节点至$w_O$所对应的叶节点的路径（$L_{w_O}$为该路径的长度）。

采用 Softmax 树之后，条件向量的公式变为：

$$P\left(w_O\middle|h_{w_i}=i,w_I\right) = \prod_{l=1}^{L_{w_O}} P\left(b_t^{(w_O)}\middle|w_I,h_{w_I}=i\right) = \prod_{l=1}^{L_{w_O}} \delta(b_t^{(w_O)} U_{w_O,t}^T V_{w_I,i}) \tag{3.4}$$

其中$\delta(x)$为 Sigmoid 函数，即$\delta(x) = 1/(1+\exp(-x))$。$U_{w_O,t}$表示 Softmax

树上路径中第t个节点对应的d维参数向量。将公式(3.4)带入公式(3.2)，替换掉原本的线性 Softmax 函数，就得到了计算复杂度更低的概率模型。每次计算条件概率的复杂度从$O(|W|)$降为$O(log(|W|))$。

本文采用 Huffman 编码树作为 Softmax 树的构造方法。高频词在 Huffman 编码树上的路径较短，低频词的路径较长。Huffman 编码使得语料能以更小的总长度进行表示，在本算法中，Huffman 树能够使得总的计算量更小。使用 Huffman 编码构造 Softmax 树这一方法的有效在 Mikolov 等人的工作[3]中得到了证实。

第四章　基于期待-最大化的训练算法

本章将对上一章所述多向量 Skip-gram 模型的训练算法进行介绍，然后与其他的算法进行对比，展示本算法更优秀的可伸缩性。

4.1　期待-最大化算法

4.1.1 符号定义

训练算法的目的，是获得每个拥有N_w个词义的词$w \in W$的N_w个表达向量。词w的表达向量集合表示为$V_w \in R^{d \times N_w}$。假设有M个词对作为训练样本：$\{(w_1, w), (w_2, w), \dots, (w_M, w)\}$，其中输入端词（中心词）均为同一个词$w$，需要通过模型预测的输出端词为$X = \{w_1, w_2, w_M\}$。形象地说，$X$是语料中词$w$的上下文窗口内的$M$个临近词。

为表述方便，这里改变上一章中的一些符号：h_m表示词w在词对(w_m, w)中的词义编号，$m \in \{1,2, \dots, M\}$。同时引进一些新的符号：词义的先验概率$P(h_{w_i} = i|w_I)$简化表达为π_i。定义$\gamma_{m,k}$（其中$m \in \{1,2, \dots, M\}$，$k \in \{1,2, \dots, N_w\}$）为隐含的二值变量，表示词在第$m$次出现时是否为第$k$个词义。

其他符号的定义与上文保持一致，为方便阅读，这里再次列出：$V_{w,i} \in R^d$表示词w的第i个词义的表达向量。$U_{w,t} \in R^d$表示 Softmax 树上从根节点到词w对应的叶节点的路径上的第t个节点的表达向量。

4.1.2 算法说明

模型中待学习的参数集合$\Theta = \{\pi_1, \dots, \pi_{N_w}; U; V_w\}$。隐含参数的集合为$\Gamma = \{\gamma_{m,k} | m \in (1,2, \dots, \mathrm{M}), k \in (1,2, \dots, N_w)\}$。结合公式(3.2)和(3.4)，$X$的对数概率表达为：

$$\log p(X, \Gamma|\Theta) = \sum_{m=1}^{M} \sum_{k=1}^{N_w} \gamma_{m,k} (\log \pi_k + \log P(w_m|h_m = k, w))$$

$$= \sum_{m=1}^{M} \sum_{k=1}^{N_w} \gamma_{m,k} (\log \pi_k + \sum_{I=1}^{L_{wm}} \log \xi(b_I^{(w_m)} U_{w_m,\mathrm{t}}^T V_{w,k})) \qquad (4.1)$$

由公式(4.1)可得，算法中的 E（期待）步和 M（最大化）步分别如下：

(1) E 步

E 步的目标是求隐含变量$\gamma_{m,k}$的期望值，用$\hat{\gamma}_{m,k}$表示。

$$\hat{\gamma}_{m,k} = \mathrm{p}\left(\gamma_{m,k} = 1 \middle| \mathrm{X}, \Theta\right) = \frac{\pi_k P(w_m|h_m=k,w)}{\sum_{i=1}^{N_m} \pi_i P(w_m|h_m=i,w)} \tag{4.2}$$

在第i次迭代中的模型参数用$\theta^{(i)}$b，Q 函数表示为：

$$Q\left(\theta, \theta^{(i)}\right) = \hat{\gamma}_{m,k}(\log \pi_k + \log P(w_m|h_m = k, w))$$

$$= \sum_{m=1}^{M} \hat{\gamma}_{m,k} \left(\log \pi_k + \sum_{I=1}^{L_{wm}} \log \xi(b_I^{(w_m)} U_{w_m,\mathrm{t}}^T V_{w,k})\right) \tag{4.3}$$

(2) M 步

π按照公式进行更新：

$$\pi_k = \frac{\sum_{m=1}^{M} \hat{\gamma}_{m,k}}{M}, k = 1,2,\ldots,N_w \tag{4.4}$$

求得隐含变量的期望值$\hat{\gamma}_{m,k}$，并更新π_k后，下一步是对模型参数$U_{w,t}$和$V_{w,k}$进行更新。注意到，这一优化问题是非凸的，且偏导数的零点$\frac{\partial \mathrm{Q}}{\partial U_{w_m,\mathrm{t}}} = 0$和$\frac{\partial \mathrm{Q}}{\partial V_{w,\mathrm{k}}} = 0$难以求得。因此，解优化问题的时候采用梯度下降法。Q 函数对表达向量U和V求梯度如下：

$$\frac{\partial \mathrm{Q}}{\partial U_{w_m,\mathrm{t}}} = \sum_{k=1}^{N_w} \hat{\gamma}_{m,k} b_I^{(w_m)} \left(1 - \xi\left(b_I^{(w_m)} U_{w_m,\mathrm{t}}^T V_{w,k}\right)\right) V_{w,k} \tag{4.5}$$

$$\frac{\partial \mathrm{Q}}{\partial V_{w,\mathrm{k}}} = \sum_{I=1}^{L_{wm}} \hat{\gamma}_{m,k} \sum_{k=1}^{N_w} b_I^{(w_m)} \left(1 - \xi\left(b_I^{(w_m)} U_{w_m,\mathrm{t}}^T V_{w,k}\right)\right) U_{w_m,\mathrm{t}} \tag{4.6}$$

交替迭代 E 步和 M 直至 Q 函数收敛，EM 算法完成。

4.1.3 对伸缩性的增强

为了增强算法的可伸缩性，以更好地适应大量训练数据，对上述 EM 算法进行了优化，提高了运算效率。

EM 算法的 E 步和 M 步中都需要对输入、输出向量进行内积，并计算 Sigmoid 函数值。然而，如果使用 Softmax 树，并且在 M 步中只进行一次梯度下降，那

么 M 步中可以直接使用在 E 步已经计算出来的向量内积和 Sigmoid 值，而不需要再次计算。使用这一策略，可以减少约一半的计算开销。

然而，这一优化策略也有不足。如果使用这一策略，M 步的优化效果将不能从二阶优化算法（如 L-BFGS，共轭梯度下降）中受益。因为这些二阶优化算法往往依赖于通过多步迭代达到收敛。

然而在实验中，不使用上述优化策略，并将 M 步中的优化算法更换为二阶优化算法（L-BFGS 和共轭梯度下降）并没有为实验结果带来显著提升。因此可以认为通过使用上述优化策略换取运算效率的提高是值得的。

4.2 与其他模型的对比

本节将说明多向量 Skip-gram 模型与其他多向量表达模型相比，在可伸缩性上的优势。这里选取 Eric Huang 等人于 2012 年发表的模型[7]（下文称作 EH 模型）作为对比对象。主要的对比指标是模型中需训练的参数数量。

定义$n_{embedding}$和n_{window}分别表示模型中总的表达向量数量，以及训练时上下文窗口的大小。n_{words}表示词表中词汇的数量，d表示表达向量的维度数。EH 模型需要计算两种上下文特征的神经网络输出，分别是局部上下文特征和全文上下文特征。定义这两种特征所对应的神经网络隐含层节点数分别为h_l和h_g。两模型的待训练参数数量对比见下表：

表格 1 两模型的待训练参数数量对比

模型	EH 模型	多向量 Skip-gram
参数数量	$dn_{words} + \mathrm{d}n_{embeddings} + (\mathrm{d}n_{window} + 1)h_l + (2d + 1)h_g$	$dn_{words} + \mathrm{d}n_{embeddings}$

观察表格可知，EH 模型比多向量 Skip-gram 模型多出了$(\mathrm{d}n_{window} + 1)h_l + (2d + 1)h_g$个待训练参数。这是因为 EH 模型的神经网络多了一层，且多考虑了一种特征（全文上下文特征）。在 Huang 等人的实验中，d、n_{window}、h_l和h_g分别被设置为 50，10，100，100，这使得两模型间的待训练参数数量进一步加大。

第五章　仿真/实验结果与分析

本章将介绍实验设置和实验结果。第一节将介绍实验使用的数据集和运行参数；第二节将进行定性的案例分析；第三节将展示在一个词汇相似度的公开测试数据集上的定量评价结果。

5.1　实验设置

5.1.1 训练数据集

为了与其他模型进行公平的对比，我们采用一个被广泛使用的公开数据集。这一数据集是 Wikipedia 在 2010 年四月的快照存档，由 Shaoul 发表[10]。数据集的长度为 9.9 亿词。本文主要的对比对象，EH 模型，在实验中也使用了这一数据集进行训练。

数据集在实验前经过了预处理，移除了词频低的词，最终保留了约 100 万个词频最高的词汇。和 Word2Vec 项目的实验[3]类似，纯数字的词汇被移除，100 个过于常见的词，如 how, for, we 等，也被移除。

5.1.2 训练程序参数

为了提高训练效率，算法中使用了 Softmax 树。与 Word2Vec 模型类似，构建 Softmax 树时采用了 Huffman 编码算法。词汇表达向量的维数和 Softmax 树节点的表达向量维数均设置为 50，与 EH 模型实验中的参数[7]保持一致。训练时的上下文窗口大小设置为 10，即中心词的前 5 个与后 5 个词。词义数量的设置上，将词频最高的约 7000 个词视为多义词，多数词设置为 10 个词义。

模型的学习速率上，参照 Word2Vec[3][4]项目，将初始学习速率设置为 0.025，且随训练进度线性减少至 0。实验结果显示，这一设置能够带来最好的表现。

EM 算法中的参数方面，EM 迭代的 batch-size 设置为 1，即每次只对一个训练样本进行 EM 迭代，原因如下：待训练的模型是高度非凸的，更小的 batch-size 导致更加频繁的参数更新，增加了接近全局最优值的概率。EM 迭代的次数设置为 1，即每个训练样本只进行一次 E 步和一次 M 步。实验表明，这一设置能够在词汇相似度测试中达到足够好的结果。增大 EM 迭代次数能够带来些微的性能提升，但是会成倍增加训练所需的时间。在以上配置下，本模型的训练速度大约

是 EH 模型的 3 倍。

5.2 实例分析

本节通过实例分析的方式，说明本算法在一定条件下能够有效区分词的不同词义。表 2 中列出了一些公认的多义词。对于每个词，列出有代表性的几个表达向量，这些向量的先验概率，以及与每个向量在向量空间中最相似的 3 个词。向量空间中的相似度，通过余弦相似度进行计算。

表格 2 典型的多义词实例

词	先验概率	最相似的词
apple_1	0.82	Strawberry, cherry, blueberry
apple_2	0.17	iphone, machintosh, Microsoft
bank_1	0.15	river, canal, waterway
bank_2	0.6	citibank, jpmorgan, bancorp
bank_3	0.25	stock, exchange, banking
cell_1	0.09	phones, cellphones, mobile
cell_2	0.81	protein, tissues, lysis
cell_3	0.01	locked, escape,handcuffed

通过观察表格，可以发现本模型生成的多向量表达有一些有趣的性质：

- 对于一个多义词，不同的表达向量表示其不同的词义。例如：apple 的第一个表达向量对应“一种水果”这一词义，其相似词为其他水果的名称；apple 的第二个表达向量对应信息科技领域的“苹果公司”这一词义，其相似词为苹果公司的产品名称，以及行业内其他企业的名称。
- 表达向量的先验概率能够在一定程度上反映不同词义在语料中的出现概率。例如：cell 一词最常见的词义为“细胞”，对应的表达向量有 0.81 的先验概率，明显高于其他两个词义“监狱单间”和“移动电话”。
- 通过将词义数设置为一个较大值（如 10），模型能够训练出对词义的更精细的分类。例如：观察表中的 bank 一词，其第二和第三个表达向量看上去较为相似，都表示财经领域的意思。然而，这两个向量表示的词义有细微的区别。第二个向量表示“银行”这一机构实体，其相似词为著名银行的名称；第三个向量更接近于表示“金融业务”，也就是在“bank”中进行的业务，因此其相似词为“股票”、“交易”等。本模型这种精细区分词义的能力能够增强向量的表达能力。

5.3 词汇相似度预测实验

5.3.1 测试数据集

本节将给出多向量 Skip-gram 模型与传统的单向量表达模型 Word2Vec[3][4]以及已有的多向量表达模型 EH 模型[7]的量化对比。

对比实验选用 Eric Huang 等人发表的 SCWS 词汇相似度测试数据集[7]。词汇相似度预测任务通过计算模型预测出的相似度与人工标注的相似度之间的 Spearman 排序相关度（Spearman's rank correlation）。也就是说，不直接比较相似度数值，而是比较两组相似度的排序序列。

在 SCWS 之前，已有一些词汇相似度预测数据集，如 Findelstein 等人发表的 WordSim353 数据集[11]，和 Rubenstein 等人发表的 RG 数据集[12]，但这些数据集并不适合用于多向量表达模型的量化评价。其原因有二：第一，这些数据集中没有足够多的多义词，无法体现多向量表达模型在多义词上的优势；第二，这些数据集中的词对没有给出上下文，使得多向量模型无法根据词的上下文推断词属于哪个词义。

为了解决这一问题，Eric Huang 等人构造并发表了带有上下文的词汇相似度测试数据集 SCWS。SCWS 包含 2003 对词，每个词各有一个句子作为上下文，由人工标注的每对词的相似度。在 SCWS 中，标注的相似度基于词汇在句子中的实际词义，而不是像之前的数据集那样，笼统地基于孤立的词汇。因此，在含有更多多义词，并且有上下文的数据集上得出的量化比较结果，更加具有说服力。

5.3.2 测试算法

本模型进行 SCWS 词汇相似度测试时的具体算法如下：

每个测试样本词对用$\{w_1, w_2\}$表示。定义中心词w的上下文窗口大小为$T+1$，则w共有T个临近词。w_1、w_2的上下文邻近词集合分别用$\text{Context}_1 = \{c_1^1, c_2^1, \dots, c_T^1\}$和$\text{Context}_2 = \{c_1^2, c_2^2, \dots, c_T^2\}$表示，其中$c_1^1$和$c_1^2$分别表示$w_1$和$w_2$的第 1 个邻近词。

根据贝叶斯定理，对于$i \in \{1,2,\dots,N_{w_1}\}$有：

$$\begin{aligned} \mathrm{P}\left(h_{w_1} = i | Context_1, w_1\right) &\propto P\left(Context_1 \middle| h_{w_1} = i, w_1\right) P\left(h_{w_1} = i | w_1\right) \\ &= \prod_{I=1}^{T} P(c_I^1 | h_{w_I} = i, w_I) P(h_{w_I} = i, w_I) \end{aligned} \tag{5.1}$$

其中$P(c_I^1|h_{w_I}=i,w_I)$可以由公式(4)计算得到，$P(h_{w_I}=i,w_I)$则是通过 EM 算法(公式(8))学习得到的词义先验概率。w_2同样适用于以上公式。这里假设在给定中心词的条件下，上下文邻近词之间相互独立。

定义w_1在上下文$Context_1$中最可能属于的词义为$\hat{h}_{w_1}$，即：

$$\hat{h}_{w_1}=\text{argmax}_{i\in\{1,2,\ldots,N_{w_1}\}}P(h_{w_1}=i|Context_1,w_1) \tag{5.2}$$

同理，定义w_2在上下文$Context_2$中最可能属于的词义为$\hat{h}_{w_2}$。

基于公式（），有两种计算词汇相似度的方法：MaxSim 和 WeightedSim。MaxSim 计算两个词在各自上下文中概率最高的一个表达向量的正弦相似度；WeightedSim 计算两个词所有表达向量两两的正弦相似度的加权平均，以计算出的表达向量在各自上下文中的概率为权。公式表达如下：

$$MaxSim(w_1,w_2)=Cosine(V_{w_1,\hat{h}_{w_1}},V_{w_2,\hat{h}_{w_2}}) \tag{5.3}$$

$$\text{WeightedSim}(w_1,w_2)=\sum_{i=1}^{N_{w_1}}\sum_{j=1}^{N_{w_2}}P(h_{w_1}=i|Context_1,w_1)P(h_{w_2}=j|Context_2,w_2)Cosine(V_{w_1,\hat{h}_{w_1}},V_{w_2,\hat{h}_{w_2}}) \tag{5.4}$$

其中$Cosine(x,y)$表示x，y两向量间的余弦相似度。$V_{w,i}\in R^d$表示词w的第i个表达向量。

5.3.3 实验结果

详细的量化对比结果见表 3。表中ρ表示 Spearman 排序相关度，数值越高，表示预测的相似度排序与人工标注结果越接近，模型表现越好。EH 模型的实验结果引用自 Eric Huang 的论文[7]。Word2Vec 模型的实验结果通过运行 Mikolov 等人公开的程序代码进行，训练中使用了 Softmax 树进行加速。由于 Word2Vec 模型为单向量表达模型，在进行 SCWS 测试时无法利用上下文信息。本模型使用了两种相似度算法进行测试，分别是上文所述的 MaxSim 和 WeightedSim 算法。所有模型的表达向量维数均设为 50。

从表 3 中可以看出，本模型的结果（65.4）优于单向量表达模型 Word2Vec（61.7），并且与目前多向量的顶尖水准 EH 模型（65.4）基本持平。实验结果表明，本模型在效果没有明显损失的情况下，实现了更加简化的模型和更高效的训练过程，在可伸缩性上具有优势。

通过对比本模型在两种测试算法下的不同结果可以发现，WeightedSim 算法下的结果（65.4）比 MaxSim 下（63.6）更好。WeightedSim 算法充分利用了词的所有表达向量，而 Maxsim 只利用了每个词各一个表达向量。这说明，充分利用多个表达向量能够更好地表达词汇的词义。

表 3 SCWS 数据集上 Spearman 排序相关度结果对比

模型	$\rho \times 100$
Word2Vec	61.7
EH 模型	65.7
本模型（MaxSim）	63.6
本模型（WeightedSim）	65.4

第六章　总结与展望

6.1　项目总结

本文提出了一种概率模型及训练算法，用于为自然语言词汇生成多向量表达。本模型主要基于 Word2Vec 中的 Skip-gram 模型。一方面，通过加入多向量表达，本模型实现了对多义词的较强表达能力；另一方面，通过优化网络结构和训练算法，本模型在不明显损失表达效果的前提下，实现了比已有的基于聚类算法的模型更高的训练效率，且不需进行聚类运算。

6.2　后续展望

本模型的主要优势是运算复杂度。在表达效果的定量评测中只能与业界前沿的算法持平。在未来，希望能将多向量模型与其他人工神经网络语言模型进行整合，进一步提升模型的表达效果。另外，希望能将本模型应用于真实的自然语言处理和文本挖掘任务中，以提高这些任务的效果。

致　谢

感谢微软亚洲研究院为我提供实习机会，以及进行研究、实验的条件。

感谢林倞老师，刘铁岩老师、高斌老师、边江老师对我的支持与指导。

感谢田飞师兄在数学模型和公式推导方面对我的帮助。

感谢戴涵俊同学在编程方面对我的帮助。

参考文献

[1]. Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language´ model. In *Journal of Machine Learning Research*, pages 1137 – 1155

[2]. Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246 – 252.

[3]. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. 1301-3781.

[4]. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems* 26, pages 3111 – 3119.

[5]. Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa.´ 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493 – 2537.

[6]. Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109 – 117. Association for Computational Linguistics.

[7]. Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873 – 882. Association for ComputationalLinguistics.

[8]. Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384 – 394. Association for Computational Linguistics.

[9]. Fernando C. N. Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In Proceedings of the 31st Annual Meeting of the Association for Computational Lninguistics (ACL-93), pages 183-190, Columbus, Ohio.

[10]. Westbury C Shaoul, C. 2010. The westbury lab wikipedia corpus.

[11]. Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan,

Gadi Wolfman, and Eytan Ruppin.2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

[12]. Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：　　张锐　　日期：　2014 年 5 月 6 日

毕业论文成绩评定记录

指导教师评语：

针对自然语言的词汇多义问题，本论文研究一种新的无监督学习算法，将上下文相关语言模型和词汇消歧义联合学习，从而生成词汇的多向量表达，并通过实验论证了方法的有效性。

论文对相关工作的分析讨论较为全面深入。所提出的方法体现了一定的学术创新贡献，也有具体的应用价值。写作方面整体结构完整，逻辑清晰，重点突出，图文格式规范，达到了本科毕业论文的要求。

成绩评定：

指导教师签名：　　年　　月　　日

答辩小组或专业负责人意见：

成绩评定：

签名（章）：　　年　　月　　日

院系负责人意见：

成绩评定：

签名（章）：　　年　　月　　日

附表一、毕业论文开题报告

论文（设计）题目：面向自然语言理解的多向量表达学习算法

（简述选题的目的、思路、方法、相关支持条件及进度安排等）

自然语言处理是当今计算机技术应用的热点领域。自然语言词汇在计算机内部的表达是计算机进行自然语言处理的基础，表达方式的选择直接影响语言模型的质量。

词汇表达的最基础方式是在词汇空间中表达，即：对于包含N个词的词表，每个词使用一个N维的向量表示，向量中的每一维唯一对应词表中的一个词。表示一个词汇的向量中，只有对应这个词汇的一维为 1，其他各维均为 0.

这一表达方式有众多不足。第一，词汇空间的维度数与词表大小相等，这一维度往往很大，导致机器学习的过程计算量较大。第二，词汇空间中不同词汇的表示向量两两正交，没有可计算性，也无法体现自然语言中词汇之间的联系。

为解决这一问题，词汇嵌入(word embedding)的概念被提出，即将词汇映射到一个维数较低且固定的空间进行表达。映射后的词汇表达不再是两两正交，且能够在一定程度上反映词汇之间的联系。

词汇嵌入的模型往往通过机器学习的方法求得，训练样本的数量与学习所得的模型的质量正相关。为了获得更好的模型，往往采用增加训练样本的方式。然而，与图像、音频等自然信号不同，语言是由人创造的抽象符号。语言中的许多规律，例如词汇间的语法关系（时态变化、词性变化）、语义关系（同义词、反义词），对于人来说是已知的。如果在词汇嵌入模型的学习过程中加入这些先验知识，可以预计学习所得的模型质量能够得到提高。

本文以 Google 的 Word2Vec 项目为基础进行实验，使用与之相同的数据集，并以 Word2Vec 作为实验效果的对照基准，期望提高 Word2Vec 在多个任务上的性能。

学生签名：张锐　　2013 年 11 月 11 日

指导教师意见：

1、同意开题（ √ ） 2、修改后开题（ ） 3、重新开题（ ）

指导教师签名：　　2013 年 11 月 11 日

附表二、毕业论文过程检查情况记录表

指导教师分阶段检查论文的进展情况（要求过程检查记录不少于 3 次）：

第 1 次检查

学生总结：

初步确定以“多义词”作为突破口，以 Word2Vec 项目的模型为基础进行研究。开始推导数学模型

指导教师意见：

切入点较有创意，应广泛调研相关文献，清楚了解领域现状。数学模型的推导要保证正确

第 2 次检查

学生总结：

确定以 EM 作为基本数学模型，相关公式的推导已经完成，正确性基本得到验证，开始进入实验阶段。

指导教师意见：

数学模型正确性没有问题，代码实现要保证正确性。

第 3 次检查

学生总结：

算法实现完成，代码调试完成。目前的量化实验结果离对比算法尚有一定距离，需要继续调节参数。

指导教师意见：

实验过程中注意记录整理数据，以便进行分析

第 4 次检查

学生总结：

指导教师意见：

学生签名： 张锐　　2014 年 5 月 6 日

指导教师签名： 林倞　　2014 年 5 月 6 日

总体完成情况	指导教师意见： 1、按计划完成，完成情况优（√） 2、按计划完成，完成情况良（ ） 3、基本按计划完成，完成情况合格（ ） 4、完成情况不合格（ ） 指导教师签名：　　2014 年 5 月 6 日

附表三、毕业论文答辩情况

答辩人		专 业	
论文（设计）题目			
答辩小组成员			

答辩记录：

记录人签名：　　年　　月　　日