目录

1 基本介绍

1.1 什么是词嵌入

词嵌入 (word embedding)，是一种稠密的词的分布表示方式，不同于传统的符号表示和词袋模型 (bag of words) 利用词-上下文共现矩阵 (word-context matrix) 的稀疏表示, 而是将词汇映射到一个连续的希尔伯特空间进行表示。基于语言学家 Zellig Haris 的假设 “在相近的上下文中出现的词汇具有相近的语义” [3] 进行训练，使得相近语义表现为希尔伯特空间中某种度量意义下的接近。

分布式的表示方式的最大优点是易于进行计算，尤其是非常适合作为神经网络的输入层，因而被广泛的应用到一系列自然语言处理的具体任务中，包括信息检索 [11], 问答系统 [17], 主题命名识别 [18] 和语义解析 [16] 等，并取得了非常好的效果。随着近几年来深度学习 (deep learning) 技术在自然语言处理领域得到广泛运用，对于词嵌入和更广泛的词表示学习成为这几年的研究热点。

1.2 之前的工作

之前有大量的关于词嵌入方面的工作, 绝大多数是基于向量的表示形式。从方法大致分为显示矩阵分解法、神经网络法和隐式矩阵分解法。

显示矩阵分解法 (Explicit Matrix Factorization Methods) 是一种传统的获得隐层空间的向量表示的方法. 其中经典的工作是隐语义分析法 (LSA, Latent Semantic Analysis)，通过对 term-doc 矩阵进行近似分解 (SVD) 的手段获得 document 的 topic 分布. 对于词分布表示而言，类似的工作是 Hypersapce Analogue to Language(HAL)[10], 通过分解 term-term 矩阵，即单词和在每个上下文单词出现的词数构成的矩阵，获得词的分布表示。由于这种方法存在着受高频词影响过大的的缺陷 (诸如 the, and, with 等停用词)，后续的研究表明通过分解 PPMI 矩阵 (positive pointwise mutual information) 是更好的替代方案 [7]。

另外一种方法是通过从上下文预测中心词的办法学习词的分布表示，通过浅层神

经网络学习预测任务来获得词所对应的向量。Bengio 在 2003 年提出用神经网络训练语言模型成功获得了高质量的词向量 [1]，同过神经网络学习如下条件概率：

$$x = (C(w_t - 1), C(w_t - 2), ..., C(w_t - n + 1))$$

$$y = b + Wx + Utanh(d + Hx)$$

$$\hat{P}(w_t|w_t - 1, ..., w_t - n + 1) = \frac{e^{y_{w_t}}}{\Sigma_i e^{y_i}}$$

由于输入层采取 concentration 的方法处理输入的窗口词表示，对窗口大小的可扩展性较差，并且正则化需要大量的计算，在面对大规模语料的时候效率并不高。后续的工作大多在提高这一框架的性能上展开。

2013 年 Thomas Mikolov 提出了一个更为简单迅速的训练词向量的框架 Word2Vec[12]，采用单层神经网络，通过计算 $w_i' w_j$ 来训练得到联合概率 $p(w_i, w_j)$。Word2Vec 包括 CBOW 和 skip-gram 两种计算方式 (Figure 1)，同时也利用了 Hierarchical Softmax 和 Negative Sampling 两种快速的正则化方法取代了 Bengio 的 Softmax 方法。至此快速获得高质量的词向量成为可能，也使得神经网络在自然语言处理任务中得到普及。随后提出的 Glove 模型 [14] 和 Swivel 模型 [15] 则在 Word2Vec 的基础上进一步地提升了词向量的训练速度和质量。

大量的实验表明，上述两类取得词向量的方法在语料足够大的情况下在得到词向量的质量上并没有绝对的优劣。2014 年 Omer Levy 等人更是证明了两类方法在数学上的统一性 [8], 即 Word2Vec 中的 SGNG(Skip-Gram with Negative Sampling) 等价于分解 word-context 的 PPMI 矩阵。随后在 O.Levy 于 TACL 发表的 [9] 一文中总结了目前位置常用的训练词嵌入的方法，将 Word2Vec 模型、Glove 模型划分为了隐式矩阵分解方法（Implicit Matrix Decomposition Methods），并通过大量的对照实验，认为相比模型中超参数的调整，上述两种矩阵分解模型的优劣很小，对于参数的控制要重于对模型架构和优化算法的研究。

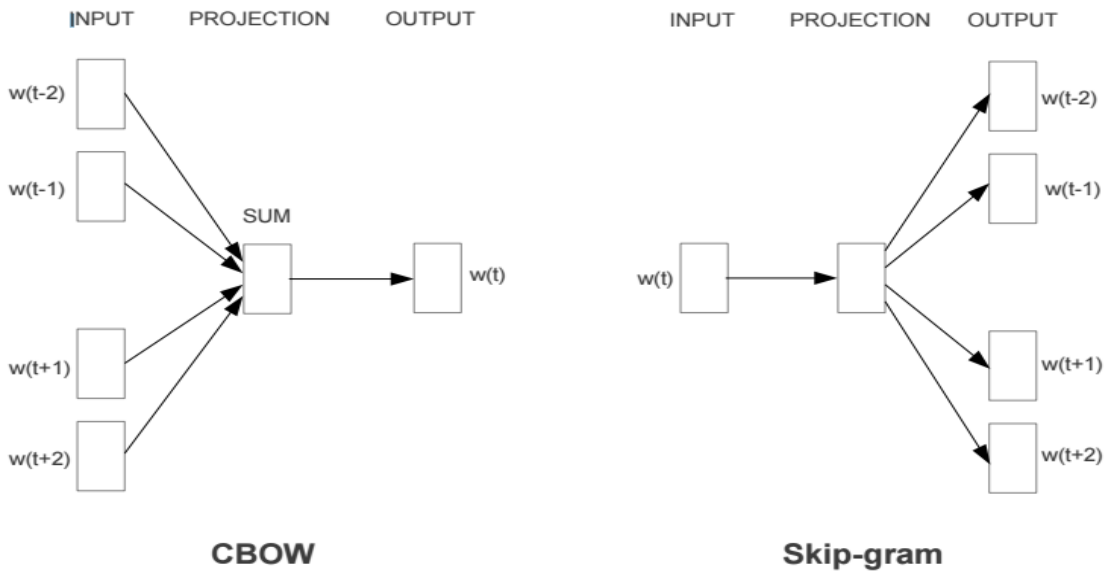

图 1: CBOW 模型和 skip-gram 模型的大致框架

此后的工作更多地专项词向量的细化工作，比如词向量可解释性的研究，在词向量的训练中引入语言学的一些数据集和监督学习的方法，非英语的词向量研究 (比如运用在汉语和日语上的词向量) 等。

1.3 本文的工作

从之前的工作中可以看出，词的向量表示已经得到了比较成熟的研究，并取得了广泛应用。然而单纯从表示方式上来说，词的向量表示也存在一些不足：1. 只用向量空间中的一个点表示词，会造成相当的语义损失，实际上各国语言的词汇中都普遍存在一词多义的现象，向量表示并不能刻画这种不确定性。2. 词向量的度量方式比较单一，不论是欧氏距离还是余弦距离，只能用来刻画语义的接近程度，对于语言学中其他语义关系，诸如上下位词 (hypernym-hyponym) 的刻画比较无力。因而最近关于词的分布式表示的研究，有许多致力于建立更具有语言学解释性，能够包含更多人工抽象的词汇信息 (fine-grained lexical resources) 的表示模型。

Luke Vilins 等人在 ICLR 2015 中发表的 Word Representation Via Gaussian Embedding[19] 文章中，提出了一种新的词的分布表示的形式—不同以往的词向量表示，而是试图将词表示成一种特定的分布 (以下简称词分布)。词分布通过通常含有一组参数

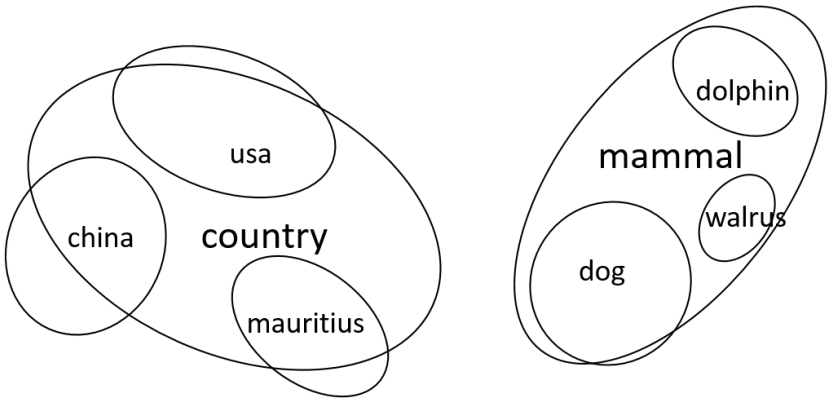

图 2: 词分布的示意图。这里选取的是选择高斯分布对词进行表示的情况。本图表现出了这种表示的两个特点：具有相近词义的分布的均值比较接近；词频较大和词义较宽泛的协方差较大。

表示分布的均值，这包含了词向量所涵盖的信息。此外，通过模型学习到的分布的其他参数，诸如方差等，提供了词的其他语义信息，诸如词义的不确定程度等，可以作为新的特征运用在具体的自然语言处理任务中。(图 2)

然而囿于篇幅，Vilins 等人的文章对于词分布的讨论并不是非常充分，除了他们在 future work 的环节提出的一些待探索的课题同时，也比较缺乏充分的实验探究。本文正是基于这些问题对词分布这一新颖的表示方式做更深入的讨论。本文的工作将分成三个部分。第一部分针对原文中对于“还有那些分布可以用于表示”尝试作出一些回答，并指出原文中作者设想的 student's t 分布并非是一个很好的选择。第二部分是对于词分布模型在词的相似度等实验上的表现，特别指出了一些超参数的选择会极大影响实验的结果。第三部分则针对原文作者提出的对于非对称关系学习的实验上的补充，并指出原文作者设想的学习这种关系的方法的效果不佳，最后给出自己的修改过的模型和学习策略。

2 理论分析

2.1 模型介绍

我们将语料经过 tokenize 得到的一个词典记为 $\mathcal{D}$，其中的每一个单词记为 w，自然有 $w \in \mathcal{D}$ 。通过设定窗口 (window) 的大小 (记为 i)，对于语料中位置为 n 的词 w_n，我们把 $w_n - i, ..., w_n - 1, w_n + 1, ..., w_n + i$ 称作上下文词，把所有上下文词的几何记为 $\mathcal{C}$，该集合中的每一个词记为 c，显然 $c \in \mathcal{C}$ 。一般情况下，我们有 $\mathcal{C} \subseteq \mathcal{D}$。对于下文

中关于单词和上下文词的表示，我们同样分别以 w 和 c 来表示以简化表达。

对于一个单词-上文词对样本 (w_i, c_j)，我们把 c_j 在语料中出现在 w_i 的窗口词对样本称为正样本 (positive)，反之则称为负样本 (negative)。考虑到各种词对样本出现的频率差异很大，单纯的二分类问题并不能很好的建立模型，为此我们引进能量函数 (energy function) 来给词对样本打分。

设输入为 x，输出为 y，我们定义能量函数为 $E_\theta(x, y)$，其中 θ 为能量函数的参数。能量函数只取非负值，其值越大，则意味着该样本 (x, y) 更 “正”，在训练词分布的模型中，即意味着词对样本在语料中出现的概率更大。通过对模型的目标函数进行优化，学习得到的参数 θ，即为 x 和 y 的分布表示。以 Word2Vec 的 SGNS 模型为例，其优化的能量函数为

$$E(w, c) = \sigma(w'c) \cdot \prod_k \sigma(w'c_N^k)$$

其中 $\sigma(x) = \frac{1}{1+e^(-x)}$ 为 sigmoid 函数，便可以看做以 sigmoid 函数作为其能量函数。为了很好的区分正样本和负样本，我们考虑引入排序损失函数 (ranking loss function)。设 $\mathcal{L}(w, c, c_n) = max(0, \varepsilon - E(w, c) + E(w, c_n))$，$\varepsilon$ 为正样本和负样本的一个边界 (margin)。当 $\varepsilon - E(w, c) + E(w, c_n) > 0$ 即 $E(w, c) - E(w, c_n) > m$ 时小损失函数，从而使得 $E(w, c)$ 和 $E(w, c_n)$ 的差值大于 ε。

相对于以往以向量为基础的词分布表示，本文讨论的模型希望将词表示为一个特定的分布，因而需要选取适合与计算分布间的能量函数。Vilins 等在 ICLR15 的文章中，选择了概率乘积核 (probability product kernel) 和 KL 散度 (Kullback-Leibler divergence) 进行建模 (以下分别简称为 IP 能量函数和 KL 能量函数)。IP 能量函数在所有连续分布构成的 $\mathcal{L}^2$ 空间中相当于内积，而 KL 能量函数则是信息论中常用的度量。下面以高斯分布为例进行说明。

对任何一个词 w，用一个高斯分布 $\mathcal{N}(\mu_w, \Sigma_w)$ 进行表示。对于 IP 能量函数，有：

$$\begin{aligned} E(w,c) &= D_K L(\mathcal{N}_w || \mathcal{N}_c) \\ &= \frac{1}{2}(tr(\Sigma_c^{-1}\Sigma_w) + (\mu_c - \mu_w)'\Sigma_c^{-1}(\mu_c - \mu_w)) - d + ln(\frac{det(\Sigma_c)}{det(\Sigma_w)}) \end{aligned}$$

对于 KL 能量函数，有：

$$\begin{aligned} E(w,c) &= D_K L(\mathcal{N}_w || \mathcal{N}_c) \\ &= \frac{1}{2}(tr(\Sigma_c^{-1}\Sigma_w) + (\mu_c - \mu_w)'\Sigma_c^{-1}(\mu_c - \mu_w)) - d + ln(\frac{det(\Sigma_c)}{det(\Sigma_w)}) \end{aligned}$$

其中 d 为设定的词分布的的维度。当选取的高斯分布具有一些特殊的条件，诸如协方差矩阵为对角矩阵时，上述两个能量函数还可以得到进一步简化。

2.2 分布的选取

从上面的推导可以看到，由于高斯分布良好的性质，两种能量函数都能够得到封闭解 (close form)。那么是否存在其他良好的分布函数，适合用作词的向量表示? Vilins 在其 ICLR15 的工作的 future work 一节曾经提到过可以尝试使用其他分布进行尝试，比如 Student’ s t，然而并没有给出相应的结论，为此我们将针对一些常见的分布进行讨论。特别要提出的是，分布的选择，以及能量函数的选择对于词的表示效果的影响是比较难以预测的，对于具体选择那种分布，本文更倾向与将易于计算摆在优先考虑的位置，因而具有解析形式的密度函数的一类分布是较优的选择。

首先我们讨论一般椭圆分布 (general elliptical distribution) 的情况。我们采取 [2] 中给出的定义，只考虑一类具有解析形式的密度函数的椭圆分布。设 $E(\mu, \Sigma, g)$ 为椭圆分布，且具有密度函数 $f(x)$ 如下：

$$f(x; \mu, \Sigma, g) = c_n |\Sigma|^{-\frac{1}{2}} g((x-\mu)'\Sigma^{-1}(x-\mu))$$

其中 $\mu \in \mathbb{R}^n$，Σ 是 n 维正定矩阵，c_n 是正则项，g 是一个在 $[0, \infty)$ 上勒贝格可测的非

负函数且满足:

$$\int_0^\infty t^{\frac{n}{2}-1}g(t)dt < \infty$$

我们先考虑两个分布的 IP 能量函数 $E(f_1, f_2)$:

$$E(f_1, f_2) = c_n^1 c_n^2 (|\Sigma_1||\Sigma_2|)^{-\frac{1}{2}} \int g_1 \cdot g_2(x)dx$$

$$g_1 \cdot g_2(x) = g_1((x-\mu_1)'\Sigma_1^{-1}(x-\mu_1))g_2((x-\mu_2)'\Sigma_2^{-1}(x-\mu_2))$$

关键在于积分 $\int g_1 \cdot g_2(x)dx$ 是否存在闭式解。首先将 $g(x)$ 延拓到 $(-\infty, \infty)$ 上的偶函数，则我们得到:

$$\begin{aligned}\int_{-\infty}^{\infty} g_1(x)g_2(x)dx &= 2\int_0^\infty g_1(x)g_2(x)dx \\ &= \int_{-\infty}^{\infty} g_1(x)g_2(-x)dx\end{aligned}$$

考虑函数的卷积:

$$g_1 \otimes g_2(x) = \int_{-\infty}^{\infty} g_1(t)g_2(x-t)dt$$

$$g_1 \otimes g_2(0) = \int_{-\infty}^{\infty} g_1(t)g_2(-t)dt$$

结合 (1)(2)，我们可以把求积分的解析解转化为求函数卷积的解析解。对于高斯分布而言，考虑到其在卷积下的不变性质 (两个高斯分布的卷积依然是高斯分布)，可以通过提取高斯核的技巧求解出解析解。然而对于一般的椭圆分布而言，g 的选择在可积性的约束下并不能很好的保证其卷积的情况。

以原文章中提到的 Student's t 分布为例，其 $g(t) = (1+\frac{x^2}{v})^{-\frac{v+1}{2}}$。关于 t 分布的卷积计算 [13] 给出了一个可用的结论：当 v_1 和 v_2 皆为奇数时存在解析解，否则不存在解析解。此外，选取 t 分布作为模型中对词的表示还会引起 IP 能量函数的导数难以求解，提高了模型训练的难度，因而高斯分布可以说是一个非常好的选择。

另外一系列常用的分布是指数族分布 (exponential family distribution)。指数族分布的密度函数表现为如下形式：

$$p_\theta(x) = exp(\mathcal{A}(x) + \theta'\mathcal{T}(x) - \mathcal{K}(\theta))$$

[5] 对于指数族分布的 IP 能量函数计算（即指数族分布的概率乘积核）进行了讨论。对于：

$$k(x, x') = \int_x p_\theta(x)^\gamma p_{\theta'}(x)^\gamma dx$$

当 $\gamma = 1$(probability product kernel) 时，只有 $\mathcal{A}$ 和 $\mathcal{T}$ 满足某些特殊的形式时才具有闭式解。

然而当 $\gamma = \frac{1}{2}$ 时，$k(x, x')$ 总能写成如下形式的闭式解：

$$\begin{aligned} k(x, x') &= \int_x p_\theta(x)^{\frac{1}{2}} p_{\theta'}(x)^{\frac{1}{2}} dx \\ &= \int_x exp(\mathcal{A}(x) + (\frac{\theta}{2} + \frac{\theta'}{2})^{\mathrm{T}}\mathcal{T}(x) - \frac{1}{2}\mathcal{K}(\theta) - \frac{1}{2}\mathcal{K}(\theta')) \\ &= exp(\mathcal{K}(\frac{\theta}{2} + \frac{\theta'}{2}) - \frac{1}{2}\mathcal{K}(\theta) - \frac{1}{2}\mathcal{K}(\theta') - \frac{1}{2}\mathcal{K}(\theta')) \end{aligned}$$

对于 $\gamma = \frac{1}{2}$ 时，$k(x, x')$ 被称作巴特查里亚距离（Bhattacharyya distance)。巴氏距离同样可以表示两个分布的相似程度，在合适的 $\mathcal{K}$ 的选择下我们依然能够的到易于计算的能量函数，至于其与 IP 能量函数孰优孰劣本文不做展开讨论。

[6] 对于指数族分布的 KL 散度的计算做出了讨论：

$$\begin{aligned} D(p \parallel p') &= E_p log\frac{p}{p_\prime} \\ &= E_p(\theta - \theta')^T\mathcal{T}(x) - \mathcal{A}(\theta) + \mathcal{A}(\theta') \\ &= (\theta - \theta')^T\mu - \mathcal{A}(\theta) + \mathcal{A}(\theta') \end{aligned}$$

其中 $\mu = E_p(\mathcal{T}(x))$。由此可见指数族分布在采取巴氏距离的情况下，能量函数有稳定

的解析解。对于指数族分布是否适合作为词分布表示，是值得进一步探究的地方。

为了训练的方便，以及专注于讨论模型在具体任务上的问题，本文沿袭前文的做法，统一采用高斯分布作为表示。

3 词表示学习

3.1 学习过程

根据前面一节的分析，我们采用高斯分布用作词分布的表示，同时为了减少参数的存储空间，我们选择协方差矩阵为对角矩阵的一类高斯分布，但当协方差矩阵具有别的一些良好性质，尤其是存在一些低阶的分解的时候也同样适用，这里就不在进一步讨论。

为了进行梯度下降方法，我们需要求解出能量函数关于参数的导数的表达形式，我们分别对 IP 能量函数和 KL 散度两种情况给出计算的结果。

对于 IP 能量函数而言：

$$E(w,c) = \mathcal{N}(0; \mu_w - \mu_c, \Sigma_w + \Sigma_c)$$

为了防止指数函数导致数值过大或过小，我们一般对它的自然对数进行计算，即：

$$log(E(w,c)) = -\frac{1}{2}det(\Sigma_w + \Sigma_c) - \frac{1}{2}(\mu_w - \mu_c)^T(\Sigma_w + \Sigma_c)^{-1}(\mu_w - \mu_c) - \frac{d}{2}log(2\pi)$$

因而进一步计算其导数：

$$\begin{aligned}
\frac{\partial logE(w,c)}{\partial \mu_w} &= \frac{\partial logE(P_w,P_c)}{\partial \mu_c} = -\Phi_{wc} \\
\frac{\partial logE(P_w,P_c)}{\partial \Sigma_w} &= \frac{\partial logE(P_w,P_c)}{\partial \Sigma_c} = \frac{1}{2}(\Phi_{wc}\Phi_{wc}^T - (\Sigma_w + \Sigma_c)^{-1})
\end{aligned}$$

其中 $\Phi_{wc} = (\Sigma_w + \Sigma_c)^{-1}(\mu_w - \mu_c)$。同样，对于 KL 散度而言：

$$E(w,c) = \mathcal{D}_{KL}(w||c)$$

其求导的结果如下：

$$\frac{\partial E(w,c)}{\partial \mu_w} = -\frac{\partial E(w,c)}{\partial \mu_c} = -\Phi'_{wc}$$
$$\frac{\partial E(w,c)}{\partial \Sigma_w} = \frac{1}{2}(\Sigma_w^{-1}\Sigma_c\Sigma_w^{-1} + \Phi_{wc})'\Phi'_{wc} - \Sigma_w^{-1}$$
$$\frac{\partial E(w,c)}{\partial \Sigma_c} = \frac{1}{2}(\Sigma_c^{-1} - \Sigma_w^{-1})$$

其中 $\Phi'_{wc} = \Sigma_i^{-1}(\mu_i - \mu_j)$。

此外还有一些训练中的细节需要注意：为了防止词分布的均值，协方差的数字过大和过小，需要对其作出一些正则化约束。比如对于协方差而言，可以设定两个超参数 c，$\mathcal{C}$，使得 Σ 满足 $cI < \Sigma < \mathcal{C}I$（$A > B$ 表示 $A - B$ 是正定矩阵）；对于 KL 散度这种非对称的能量函数，在进行 negative sampling 的时候需要分别对两个位置进行等。

相比前文的工作，本文特别要强调的是，关于协方差矩阵的范围 (这里相当于 Σ 的最大、最小特征值) 的选取，是非常影响结果的，并在后面的实验评价一节中会特地进行讨论。

3.2 实验设置

获得词表示最普遍的做法，是根据语义的上下文假设，本质上是用词对的共现频率作为特征的训练手段。在自然语言处理的工作中，高质标注数据集的稀缺一直是提高模型性能的一个瓶颈，而这种无监督的训练方法则可以充分利用互联网产生的大量文本信息，利用海量数据获得良好的训练效果。

适用于词分布的训练的数据集有很多,常用语料库,英文的有维基百科(Wikipedia)、华盛顿邮报（WSJ)、推特（Twitter）等，中文的则有人民日报、微博等。本文采取维基英文百科三月份的镜像进行训练[1]，训练采用了 seomoz 在 github 上公开的代码并加以修改。[2]

[1]可以从 https://dumps.wikimedia.org/处下载维基百科各时间点的镜像文件
[2]https://github.com/seomoz/word2gauss/

3.3 结果及分析

模型设置 为了控制实验参数的规模，本文的实验只保留语料中出现次数大于等于 500 的单词，并且去除停用词 (Stop Words)。基于便于计算的考虑，统一采用对角型协方差矩阵，表示的维度 d 选择为 50，详细的超参数见下表：

initial learning rate	0.1
margin[3]	0.1
Negative Sampling number	5
context window size	5
iteration	5

表 1: 词表示实验的超参数设置

两种度量得到的结果 我们分别通过 cosine 函数和 IP 能量函数作为衡量词相似度的两个指标，选取最相近的 20 个词，结果如下表：

Query Word(metric)	**Neighborhood Words**
math(IP)	theory, mathematics, physics, psychology, geometry
math(cosine)	algebra, arithmetic, calculus, mathematics, coursework
google(IP)	software, internet, youtube, microsoft, iso
google(cosine)	github, gmail, skype, facebook, yahoo
country(IP)	nation, countries, sweden, iceland, belgium
country(cosine)	barbadian, nation, ireland, sweden, kenya

表 2: 词分布表示在两种度量下得到的临近词的结果，其中 IP 表示用 IP 能量函数计算相似度，cosine 表示用余弦函数只计算均值的相似度。IP 能量函数得到的结果按照协方差矩阵的行列式由高到低排列。

从结果来看两种能量函数获取词相似度的能力相近，可见词分布包含了词向量模型对于词义建模的能力。此外，由于 IP 能量函数的结果按照协方差矩阵的行列式由大到小排序，可见具有较大行列式的词，如 internet，theory，nation，具有交广泛的含义，而

较小行列式的词，如 geometry，microsoft 等，则含义非常具体。

词义相似度评价 我们使用一些人工构造的数据集来进行词义相似度的评价。每个模型的得分是数据集中两个词的人工打分和模型的出的两个词的相似度的皮尔逊相关系数 (Pearson Correlation Coefficient) 的百分数。

其中，WordSim-353 包含 353 对单词和它们的相似度得分。这些得分是 0~9 的数字构成，共计 10 个人的打分并给出其平均分。Lexsim-999 与 WordSim-353 的构成方式类似，不过具有更多的词对，并且评测比前者要严格。

这里模型的在 sigma_min 和 simga_max 两个关键的超参数上分别取值为 0.5 和 1.0，其余的超参数与表 1的设置相同，得到结果如表 3:

Dataset	**SG \50d**	**GR \50d \cos**	**GR \50d \IP**
WordSim-353	59.89	58.01	42.87
SimLex-999	29.39	27.59	20.91

表 3: 词相似度测评的结果。SG 代表 Skip-Gram 模型，GR(Gaussian Representation) 代表本文中的模型。50d 表示维度为 50.cos 与 IP 分别指 cosine 函数和 IP 能量函数，从数据上看 IP 能量函数表现词相似度的能力相对较弱

超参数的选取 之前的工作中提到词分布表示模型对于一些超参数非常敏感，尤其是协方差的上下界的选取，然而并没有给出相应的实验和取值策略。本文针对这一问题给出了一组实验, 其结果见表 4

从结果中看，σ_{min} 取 0.5，σ_{max} 取 1.0 是唯一的同时在大小语料下皆表现出良好效果的选择，其余的实验并不能总结出一个比较好的调参策略。从理论上说，当 σ_{min} 和 σ_{max} 非常接近时，词分布表示退化成词向量表示，而 IP 与 cosine 两种度量的方式应当趋同，当然这样便失去了词分布表示的意义。我们认为这可能是这一表示模型的一个缺点。

σ_{min}	σ_{max}	**vocab**	**similarity(cosine)**	**similarity(IP)**
0.5	1.0	114186	**0.581***	**0.437***
0.6	1.0	114186	0.503	0.203
0.7	1.0	114186	0.618	0.161
0.5	1.0	64617	0.544	0.385
0.5	1.5	64617	0.203	0.159
0.3	1.2	64617	0.448	0.206

表 4: 关于 σ 的上下界选取对于词义相似度的影响，其中 $\mu_{max} = 4.0$，我们反复的实验一些组合，只有一组得到了较好的结果，可见这组参数的选取对于实验结果的影响是巨大的

4 上下位关系的学习

4.1 学习过程

词分布表示相比于其他向量表示模型的最大不同在于，它自然引进了 KL 散度这一非对称的度量，能够学习一些通过偏序关系形成的层次结构，比如上下位关系、语义的继承关系（entailment）等。ICLR15 的论文中，作者指出了该模型具有学习这种层次结构的能力，然而本文则想通过上下位词的学习试图说明原论文的模型学习的效果并不理想，并给出了改进方式和二者的对比实验。

首先解释一下什么是上下位词（hypernym-hyponym）关系。在语言学中，下位词（hyponym）相对于上位词（hypernym），是一种 is-a 的关系。例如“狗是一种哺乳动物”、“西瓜是一种水果”，“狗”、“西瓜”相对于“哺乳动物”、“水果”是下位词，反之则是上位词。(Figure 3) 上下位词关系的引入对诸如问答系统、自然语言推断（natural language inference）等任务的效果具有很好的提升。

在 ICLR15 的文章中，作者采用 KL 散度的能量函数，将具有上下位关系的词对作为正样本，并通过分别对上位词和下位词位置进行负采样得到的词对作为负样本，采用

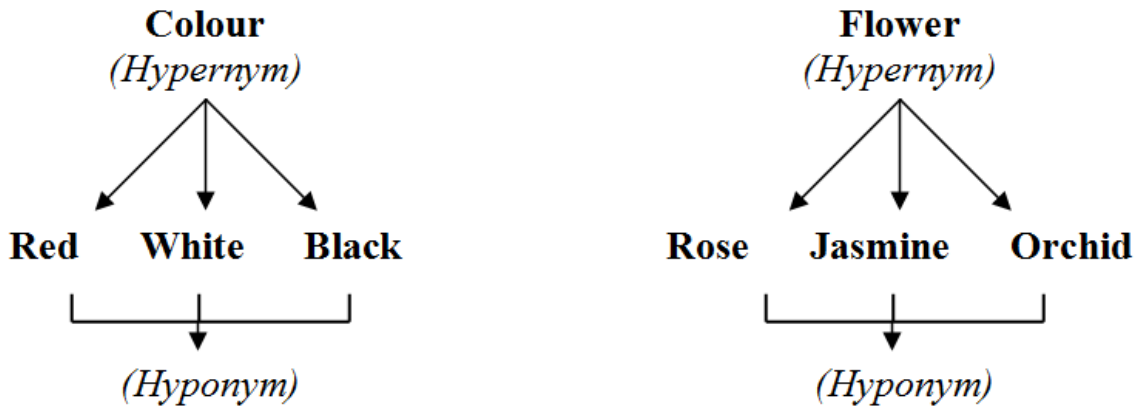

图 3: 上下位词的两个例子

和词表示学习同样的方式进行训练。其损失函数如下：

$$\begin{cases} Loss(v_{hyper}, v_{hypo}) & = max\{0, \varepsilon - D_{KL}(v_{hyper}||v_{hypo}) + D_{KL}(v_{hyper}||v_{neg})\} \\ Loss(v_{hyper}, v_{hypo}) & = max\{0, \varepsilon - D_{KL}(v_{hyper}||v_{hypo}) + D_{KL}(v_{neg}||v_{hypo})\} \end{cases}$$

从该损失函数可以看出，存在上下位关系的词对具有较大的 KL 散度值（本质上是增大上位词的协方差矩阵，减小下位词的协方差矩阵），并且与负样本存在一个边界，这个边界可以通过交叉验证（cross-validation）等手段获得。然而不得不指出的是，这种方式存在这如下的问题：

a) 负采样的方式不合适。[4] 曾提指出在对知识图谱三元组的两个本体进行负采样时，要使得进行采样的位置尽可能的减少会产生假负样本（false-negative）的情况。这同样适用于对于上下位词的采样。举一个例子：对于上下位词对（mammal.n.01，dog.n.01）对于下位词”dog.n.01” 进行采样时，由于其他的哺乳动物都是其下位词，有相当的概率得到的并非负样本。这种采样不正确的情况，会使得处于上位词位置的表示存在过拟合现象（即对任意的词都会判定为存在上下位关系）。从后面的实验部分可以看出，原模型的召回率（Recall）非常高。

b) 损失函数设计的并不合理。实际上，对于一个词对 (w_a, w_b) 而言存在着三种上下位情况（上下位关系记为 $\preceq$，下文沿用该记号）：$w_a \preceq w_b$、$w_b \preceq w_a$、w_a 和 w_b 无

关系。对于第一种情况的判断，由于 KL 散度的性质，只要存在 $D_{KL}(v_a||v_b) \preceq \varepsilon$ 即可，然而后两中情况的判断，还需要考虑 $D_{KL}(v_b||v_a)$ 与 ε 的关系。然而对于原文章中的损失函数而言，单纯的增大正样本和负采样样本之间的边际，有时会造成 $D_{KL}(v_hyper||v_neg)$ 或 $D_{KL}(v_neg||v_hypo)$，使得他们产生相反的上下位词的关系。从后面的实验中可以看出，这会造成模型的精准度（Precision）迅速下降。

为了解决这两个问题，本文同样提出了两种应对方法：

a) 只对上位词位置进行负采样。考虑到这里训练的负样本是不构成上下位词关系的词对，且上下位词呈现的是一种树状的层次结构，下位词的上位词是唯一确定的（相对词义而言，多义词会存在多个）。

b) 考虑新的损失函数：

$$\begin{aligned} Loss(v_{hyper}, v_{hypo}) = max\{0, \varepsilon - D_{KL}(v_{hyper}||v_{hypo}) + D_{KL}(v_{neg}||v_{hypo})\} \\ + min\{0, \varepsilon^{'} - D_{KL}(v_{neg}||v_{hypo}) + D_{KL}(v^{'}_{neg}||v_{hypo})\} \end{aligned}$$

在沿用原模型损失函数的基础上，我们新添加了一个对负样本的正则项和边际，来避免前面 b) 中出现的问题。这个损失函数同样反映了我们新的判别标准：对于词对 (w_a, w_b) 当 $D_{KL}(w_a||w_b) - D_{KL}(w_b||w_a) > \theta$ 时，认为 $w_a \preceq w_b$；当 $|D_{KL}(w_a||w_b) - D_{KL}(w_b||w_a) < \theta|$ 时认为 w_a 和 w_b 无上下位关系。这里的 θ 同样需要通过交叉验证来选取，不过一般情况下会大于训练时的 ε，这一点在后面的实验中会进行讨论。

4.2 实验设置

关于上下位词关系最常用的语料库是 WordNet[4]。WordNet 是一个英语词汇数据库（lexical database）。区别与常见的词典，WordNet 将所有的单词聚成一个同义词集

[4]George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

合（synsets）来表示一个共同的语义。为了方便的获得训练集和测试集，本文采用了NLTK[5]的 wordNet Interface, 获得了所有上下位词关系的传递闭包（transitive closures），共计 698587 个正样本词对，按一定比例获得训练集和测试集中的正样本，负样本则按照负采样的方法生成。我们这里不对分布的具体选取做出讨论，因此选取均值的维数为 50，协方差矩阵为对角型矩阵，迭代次数会在随后的结果中指明，负采样个数统一设定为 5。训练集和测试集中正样本的比例为 4:1，测试集正负样本比例为 1：1

4.3 结果及分析

我们将对模型中超参数的选取，本文改进的模型相比原模型的提升进行一些的对比实验。

边际 ε 、$\varepsilon^{'}$ 的选取对于结果的影响 我们设定最后判定结果的 $\theta = 5$，迭代次数为 3，分别考虑改变 ε 和 $\varepsilon^{'}$ 对于最后结果的情况:

ε,$\varepsilon^{'}$	**Precision**	**Recall**	**Accuracy**
5,0.3	0.81720	0.78560	0.80493
6,0.3	0.81710	0.79802	0.80967
7,0.3	0.81592	0.80801	0.81286
8,0.3	0.81410	0.81808	0.81564
8,0.1	0.81317	0.82050	0.81599
8,0.7	0.81617	0.81403	0.81534
8,1.3	0.81759	0.81034	0.81478

表 5: 两种 margin 的选择对于结果的影响

从实验的结果可以看出，ε 越大、$\varepsilon^{'}$ 越小，模型的 Accuracy 越高，但总的来说这两个参数的调整对模型的表现影响不大。

[5]Natural Language Toolkit, a python package to build natural language processing program

迭代次数对实验的影响　我们设定 $\theta = 3, \varepsilon = 5, \varepsilon^{'} = 0.3$，只改变训练集迭代的次数，得到如下结果：

iterations	**Precision**	**Recall**	**Accuracy**
3	0.71980	0.85908	0.76233
5	0.72693	0.86534	0.77015
10	0.73179	0.87426	0.77692
20	0.73441	0.88625	0.78287
50	0.73752	0.90859	0.79261

表 6: 迭代次数对结果的影响

从实验数据中可以看出，随着迭代次数的增加，模型效果得到显著的提升，从 3 次到 50 次迭代性能提高的近 3 个百分点。由于上下位词的训练集比较小，为了得到高质量的分布，相当的迭代次数是必要的。

两种模型的对照实验　考虑到不同超参数会影响模型的效果，我们特地选取了多种参数组合进行对照实验，其中本文修改过的模型的 ε 设定为 0.1，得到结果如下：

Implementation	**Original**			**Modified**		
(ε,θ,iterations)	Precision	Recall	Accuracy	Precision	Recall	Accuracy
5,3,10	0.63007	0.94993	0.69610	0.73033	0.87629	**0.77636***
5,3,50	0.64172	0.96508	0.71313	0.73722	0.91044	**0.79296***
7,5,10	0.70720	0.93409	0.77367	0.81485	0.82856	**0.82015***
7,5,50	0.70627	0.95981	0.78032	0.81912	0.88460	**0.84477***

表 7: 模型对比实验

可以看出，我们修改后模型的结果在相同的参数设置下要优于原模型。原模型的召回率远高于精准率，验证了之前对其对正样本过拟合的推断。

5 结论和展望

在作者原先工作的基础上，本文对词分布表示这一新的表示方式做了理论和实验上的探究。在理论分析一节中，我们论证了用于表示的分布不仅限于高斯分布一种情况，其他分布对于词表示是否具有更好的效果，这是一个值得探究的方向，比如本文中提出的指数族分布和巴氏距离，可以构建一个新的表示方法。

词分布表示最引人瞩目的地方在于其对于非对称关系的学习能力。不需要想向量表示模型那样自己构造非对称度量，KL 散度的自然引进不仅便于计算，具有很好的可解释性。实际上不光是对于上下位词这种非对称关系的学习，目前已经有一些工作将这一模型运用到句子的蕴含关系 (entailment)，知识图谱 (Knowledge Graph) 的本体-关系的学习等其他任务上去，并取得了很好的效果。

然而词分布模型仍然存在一些待改进的问题:

1. 对于词义的学习效果不算很好从词表示学习一节的实验中可以看出，相对于以前向量表示模型对于词义相似度的学习效果，词分布模型并不存在优势，在 IP 能量函数这一度量下甚至存在一定差距，这导致该模型在一些依赖词义相似度的任务中可能并没有太大优势。

2. 协方差学习到的信息不明确相比向量表示模型，词分布模型学习得到的协方差如何利用是一个问题。从实验结果可以看出协方差一定程度上反映除了词的多义性和抽象性，然而通过 IP 能量函数将协方差引入词义相似度的评测时，并没有取得优于向量表示的结果。如何理解和利用协方差学到的信息，并将其作为表示结合到目前的一些算法框架中仍然有待探究。最近利用协方差关于多义性的反映，用来通过混合高斯模型学习词的多义表示 (multi-prototype) 的任务是在这一方面很好的探究。

参考文献

[1] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[2] Eusebio Gómez, Miguel A Gómez-Villegas, and J Miguel Marín. A survey on continuous elliptical vector distributions. *Revista matemática complutense*, 16(1):345–361, 2003.

[3] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[4] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 623–632. ACM, 2015.

[5] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[6] Michael I Jordan. An introduction to probabilistic graphical models, 2003.

[7] Rémi Lebret and Ronan Collobert. Word emdeddings through hellinger pca. *arXiv preprint arXiv:1312.5542*, 2013.

[8] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014.

[9] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[10] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[11] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[13] S Nadarajah and DK Dey. Convolutions of the t distribution. *Computers & Mathematics with Applications*, 49(5):715–721, 2005.

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[15] Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*, 2016.

[16] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

[17] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM, 2003.

[18] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 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, 2010.

[19] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*, 2014.