基于非线性全局上下文的词嵌入

刘永彬^{①②}, 欧阳纯萍^{①*}, 钟东来^{②*}, 李涓子^②, 袁博志^②, 李奇^②

① 南华大学计算机学院, 衡阳 421001

② 清华大学计算机科学与技术系, 北京 100084

* 通信作者. E-mail: ouyangcp@126.com, tongleechung86@gmail.com

收稿日期: 2015-09-23; 接受日期: 2015-10-20; 网络出版日期: 2015-12-08

国家重点基础研究发展计划 (973 计划) (批准号: 2014CB340504)、国家自然科学基金重点基金 (批准号: 61533018)、国家自然科学基金青年基金 (批准号: 61402220)、国家自然科学基金中法合作项目 (批准号: 61261130588) 和中国博士后基金 (批准号: 2014M550733) 资助项目

摘要 针对当前词表示方法中的上下文的局限性, 文章提出了一个基于非线性全局上下文的词表示方法. 该方法主要分为两步骤, 首先利用维基百科的排歧页, 对文档中的当前词进行排歧处理, 以此来提高词表示的效果. 然后, 再针对传统词表示方法中的线性局部上下文问题, 利用依存和共指关系对语料进行分析, 得出基于非线性全局上下文的词表示向量. 文章选取英文维基百科数据集作为实验语料, 在定性分析方面, 该方法可以找到更接近当前词义的相关词, 因对多义词进行了排歧处理, 词表示结果上明显好于其他的方法. 在定量比较方面, 通过在 WordSim-353 数据集上对比实验表明, 该方法在 Spearman 相关系数上比其他方法高出 5%~10% 以上.

关键词 非线性全局上下文 共指消解 词义排歧 词向量 词嵌入 神经网络 深度学习 依存关系分析

1 引言

词表示作为自然语言处理领域中的一项基础研究工作, 随着深度学习等算法的提出再次受到了广泛的关注. 传统的词表示都是基于词袋模型 (bag-of-word model, BoW) [1], 该模型存在着高维、稀疏等问题, 需要利用各种技术对其进行降维和平滑. 同时词袋法还有一个缺点, 比如说丢失词语间的次序, 这就会导致不同的两句话使用了相同词语, 用词袋法表示出来时是完全一样 [2]. 随着基于深度学习的词嵌入 (Word Embedding) 理论的提出, 有效地解决了上述问题 [3]. 分布式表示 [4] 是基于深度学习的词表示方法的重要特性, 它可以使自然语言处理过程中的每个词被表示成稠密、低维和实值的向量. 但是, 当前的基于深度学习的词表示研究, 大部分都要求当前词的上下文被固定在一个窗口长度内. 这种固定窗口大小的方式有一个缺点, 就是很少考虑词与词之间的关系, 即它们仅考虑了词与词的“距离”因素, 却没考虑词语之间的语义依存关系.

例如, “Astronomers scientist discover star with telescopes” 这句话, 假设窗口选择为 2, 词 “discover” 的上下文是 “Australian”, “scientist”, “star”, “with” 这几个词, 很显然在窗外的 “telescopes” 这个词要比窗口内的某些词与 “discover” 的语义关系更紧密 [5]. 总的来说, 当前的基于深度学习的词表示方法对上下文的选取可以理解为一个局部特征的选择 (在一个窗口内), 欠缺了全局特征选择的综合考

引用格式: 刘永彬, 欧阳纯萍, 钟东来, 等. 基于非线性全局上下文的词嵌入. 中国科学: 信息科学, 2015, 45: 1588–1599, doi: 10.1360/N112015-00238

虑. 此外, 词表示的主流方法大部分忽视了一词多义的情况, 这也给词表示带来了很多噪音和不确定性的因素, 近两年这个问题引起了许多学者的关注.

因此, 本文提出一种基于共指消解和词义排歧的词表示方法. 本文的主要创新工作包括: (1) 利用共指关系, 扩展了 Levy 的局部非线性上下文为全局非线性上下文^[5]; (2) 利用维基百科中的实体排歧页和文章 (词条页) 中的实体内链信息对一词多义的问题进行了处理. 最后, 在英文维基百科的数据集上, 把我们的方法与 Word2Vec 等方法进行了多指标对比评测. 实验结果表明, 我们所提出的方法能够更有效更准确地对词进行表示, 进而为后续的自然语言处理应用研究奠定了良好的基础工作.

2 相关研究

词表示近年来在自然语言处理领域引起了广泛的关注, 已经有许多词表示学习的方法在多个领域得到了成功应用, 如机器翻译^[6,7]、情感分析 (sentiment analysis)^[8]、信息抽取^[9,10]、文档分类^[11]、知识库构建^[12~15]、自动问答系统^[16]等.

早期最常用的词向量表示方式是 one-hot 表示法, 即用一个词典大小的向量来表示一个词, 用 1 来表示该词在词典中的位置, 其余向量值均为 0. 这种表示方法不仅容易导致向量维数过高, 而且也不能很好地刻画词与词之间的相似性^[17]. 因此, 有学者提出了另一种分布式的词表示方式^[18], 通过训练将某篇文档中的每个词映射成一个固定长度的低维向量, 再将所有向量集中起来形成一个词向量空间. 而每一向量则为该空间中的一个点, 在这个空间上引入“距离”, 则可以根据词之间的距离来判断它们之间的 (词法、语义上的) 相似性, 这种词表示法也称为词嵌入.

随着深度神经网络的兴起, 有学者提出了基于深度神经网络算法生成词的 Word Embedding 方法. 其中最具代表性的是 Bengio 等提出的神经概率语言模型^[19], 采用三层神经网络模型, 利用前 $n-1$ 个单词, 来预测第 n 个单词. 尽管基于该模型的效果不错, 但是由于在输出层需要归一化处理使其训练和预测的时间较长, 影响实际的应用. 随后, Mikolov 等^[20]提出了更为简便的神经网络词表示模型 Word2Vec, 简化了 Bengio 的神经概率语言模型, 将中间层简化为输入层的向量累加求和运算, 输出层改为了层次的 softmax, 大大地提高了模型的执行效率.

但是, Bengio 和 Mikolov 等的方法都忽视了一词多义的情况. 因此, 很多学者在这方面提出了各种解决方法, 如 Huang 等提出的基于全局上下文和多义词的 Word Embedding 的方法^[21]. 该模型每个词都有一个 Local Context 计算的词向量表示, 还有一个 Global Context 计算的 Global semantic vector, 通过两个不同的上下文计算词向量以求解一个更为精确的表示, 同时利用局部上下文来进行 K-means 聚类, 凡是在不同类簇中的相同词都属于不同词义的词, 以此来解决一词多义的问题. 类似的方法解决多义词的问题还有 Reisinger 和 Mooney 提出的词表示的方法^[22]. 还有 Neelakantan 等针对一词多义的情况提出的基于非参估计的 Multiple Embeddings 的方法^[23]. 该模型利用非参模型对一词多义进行处理, 同时该模型可以进行在线学习 (online learning) 模式, 极大地提高了 Word Embedding 的效率. 清华大学孙茂松课题组的刘知远等, 利用 WordNet^[24] 语言知识库对一词多义进行排歧处理, 提出了基于词义表示和词义消歧联合推理的词表示模型^[25]. 还有一些有关神经网络词表示的研究工作也很突出^[26~29], 例如 Collobert 等提出 Senna 工具^[28,29], 该工具利用了神经网络对词进行向量化, 同时更为人们熟知的是它的自然语言处理的各种功能, 它可以进行词性标注 (part-of-speech tags), 模块划分 (chunking), 命名实体识别 (name entity recognition), 语义角色分析 (semantic role labeling) 和句法分析 (syntactic parsing) 等. 因此很多人认为它更像是一个自然语言处理的工具, 而不是一个基于神经网络的词表示学习算法. 还有 Levy 和 Goldberg 从理论上证明了 Word2Vec 其实近似等价于矩阵

分解 (matrix factorization) [30], 同时还提出了基于依存关系上下文的词表示方法 [5], 该方法利用了依存句法分析的结果对上下文的选择上进行优化.

在上下文分析方面, 上述方法大多都存在一个问题, 只利用了一个线性的局部上下文 [5], 而忽视了非线性的全局上下文. 而在解决多义词问题上, 上述方法大多采用无监督的对上下文进行聚类的方法, 该类方法事先不知道哪些词会有多义性, 因此需要大量的上下文聚类处理, 给执行的效率带来负担. 本文为了解决以上词表示中存在的问题, 提出了基于共指消解和词义排歧的词表示方法.

3 模型描述

3.1 问题描述

本研究问题可以形式化描述为: 用 $\mathcal{C}$ 和 $\mathcal{D}$ 分别表示语料集合和基于该语料的词典集合, w_{wsd} 表示排歧后的当前词, $\text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})$ 表示词 w_{wsd} 的基于依存和共指关系的非线性全局上下文, 也就是说 $\text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})$ 中的上下文都是和当前词 w_{wsd} 有着依存关系或者是通过共指关系建立的依存关系. 我们目标是通过上下文 $\text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})$ 和经过排歧后的 w_{wsd} 来学习出效果更精准的词向量, 词向量就是一个指定了固定长度的实值向量 $v(w_{\text{wsd}}) \in \mathbb{R}^m$, 其中 $v(w_{\text{wsd}})$ 就是当前词 w_{wsd} 的词向量, m 是这个向量的长度. 我们的目标函数设为

$$\prod_{w_{\text{wsd}} \in \mathcal{C}} p(w_{\text{wsd}} | \text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})), \quad (1)$$

实际应用中, 可以对式 (1) 取对数, 即目标函数为

$$\sum_{w_{\text{wsd}} \in \mathcal{C}} \log p(w_{\text{wsd}} | \text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})). \quad (2)$$

对于输出层我们采用 softmax 算法来完成. 于是我们有了以下结果:

$$p(w_{\text{wsd}} | \text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})) = \frac{e^{y_{w_t}}}{\sum_{i=1}^N e^{y_{w_i}}}, \quad (3)$$

其中, N 是词典 $\mathcal{D}$ 的长度, t 是词 w 在词典中的位置, 每一个 y_{w_i} 都是词 w_{wsd} 未正则化的概率, 它的计算如下:

$$y = b + Uh(w_{\text{wsd}} | \text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})) = \frac{e^{y_{w_t}}}{\sum_{i=1}^N e^{y_{w_i}}}, \quad (4)$$

其中, U (一个 $N \times h$ 的矩阵) 和 b 是 softmax 的参数, 它们都从训练算法中得来. 对于 softmax 算法, 本文采用和 Word2Vec 相同的层级 softmax 算法, 其中 Huffman 编码的使用及优化方法和 Word2Vec 的做法是一致的.

可以看出本文的目标函数和 Word2Vec 的目标函数基本上是一致的, 只是当前词 w_{wsd} 和上下文 $\text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})$ 的选择上不同. 本文的主要工作就是如何得到排歧后 w_{wsd} 和非线性的全局上下文 $\text{Context}_{\text{dep\&coref}}(w_{\text{wsd}})$. 整个模型分为两个步骤: 当前词的词义排歧和基于依存和共指关系的上下文选择. 下面首先对词义排歧处理进行介绍, 然后对基于依存和共指关系的上下文选择词表示方法进行描述.

图1 基于维基排歧页的多义词排歧

Figure 1 Word sense disambiguation based on wikipedia disambiguation pages

3.2 基于维基排歧页的词排歧

本文的方法利用当前最为流行的通用世界知识库维基百科的排歧页对语料中的一词多指问题进行处理, 相关的描述如下:

维基排歧页 在维基百科中, 当多个不同含义的实体有相同的名称时, 维基百科网站会提供一个排歧页面将不同含义相同名称的它们分开, 排歧页中还包含对这些不同实体的列表. 例如, 词“apple”的消歧页列举出“apple”相对应的所有不同实体, 其中包括著名的IT公司“Apple”、水果“apple”和披头士乐队的专辑“Apple”等. 对于每一个排歧页来说, 都相当于给了一个 mention 和对应的多个 entities 的映射关系^[31]. 本文会对维基网站中的每一个排歧页面构建一个 mention 和 entities 的映射 $\langle \text{mention}, En \rangle$, 其中 mention 就是各个不同实体的相同名称, En 是一个集合 $En = \{e_1, e_2, \dots, e_m\}$, 其中 e_i 表示同 mention 不同含义的实体, m 是这个排歧页中实体的个数. 本文维护一个所有的排歧页构建的 $\langle \text{mention}_i, En_i \rangle$ 映射集合.

维基文章内链 在维基百科中, 每篇文章 (article) 内部都会有很多带内链的实体出现, 这些实体也都有内链信息指向对应的页面. 本文假设文章内部的这些带有内链信息的实体是该文章的主题词表示. 所以, 每一个文章都构建了一个 $\langle \text{article}, \text{TopicEn} \rangle$ 映射, 其中 article 是这个文章标题, $\text{TopicEn} = \{Te_1, Te_2, \dots, Te_{tn}\}$ 是这篇文章内部所有带有内链信息的实体集合. 这样得到一个基于维基百科中所有文章的 $\langle \text{article}_i, \text{TopicEn}_i \rangle$ 映射集合.

文档主题词集合 对于需要处理的语料数据, 假设文档中发现的与维基 article 同名的词构成了该文档的主题词表. 对语料中的每一个文档 d_i 找出所有对应维基百科文章标题的词作为这篇文档的主题词集合. 这样每篇文档就会有一个主题词集合 $\langle w_{t1}, w_{t2}, \dots, w_{tn} \rangle$.

下面阐述如何利用上面构建的三个集合进行词义排歧, 模型框架如图1.

图1左边是对维基百科的排歧页进行处理后得到一个 mention 对应的 entities 集合 En , $En =$

$\{a_1, a_2, a_3, a_4\}$, 每个 a_i 是同 mention 不同含义的实体在维基中对应的 article, 每个 a_i 的主题词集合 TopicEn, $\text{TopicEn} = \{Te_1, Te_2, \dots, Te_{t_n}\}$. 图 1 右边是语料中的一个文档, 其中 w 是当前词, $\langle w_{t_1}, w_{t_2}, \dots, w_{t_i} \rangle$ 是该文档的主题词集合, 由和维基百科中的 article 标题检索匹配得到. 排歧的过程是对当前词和 $\langle \text{mention}_i, En_i \rangle$ 集合中的 mention 进行匹配, 如果当前词匹配到 mention, 表示该词是一个多义词, 接下来如图 1, 利用文档主题词集合 $\langle w_{t_1}, w_{t_2}, \dots, w_{t_i} \rangle$ 与 mention 对应的每个实体 a_i 的主题词集合 TopicEn_i 进行 overlap, 以 overlap 的个数作为实体 a_i 的 score, 最后取 score 最高的实体作为多义词的真正词义, 同时给当前词打上类别标签. 这样就得到一个排歧后的当前词 w_{wsd} . 具体如算法 1 所示.

算法 1 基于维基百科排歧页的多义词排歧

Input: Wikipedia disambiguation pages mapping collection:

$C_1 = \{\langle \text{name}_1, En_1 \rangle, \dots, \langle \text{name}_n, En_n \rangle\}, En = \{e_1, e_2, \dots, e_m\};$

Wikipedia pages mapping collection:

$C_2 = \{\langle \text{article}_1, \text{TopicEn}_1 \rangle, \dots, \langle \text{article}_{n_a}, \text{TopicEn}_{n_a} \rangle\}, \text{TopicEn} = \{e_1, e_2, \dots, e_{m_a}\};$

Training corpus: $\mathcal{D} = \{d_1, d_2, \dots, d_{n_d}\}$ topic words in $d_i: \langle w_{t_1}, w_{t_2}, \dots, w_{t_{d_n}} \rangle;$

Objective word: $w, w \in d_i;$

Output: WSD: $w_{\text{tag}};$

Initialize: $w_i;$

for each $d \in \mathcal{D}$ **do**

for each $w \in d_i$ **do**

if $w_i = \text{name}_j$ in C_∞ **then**

 Calculate overlap score for article: $\langle w_{t_1}, w_{t_2}, \dots, w_{t_{d_n}} \rangle$ and $\text{TopicEn} = \{e_1, e_2, \dots, e_{m_a}\};$

 Entity = $\max(\text{score}_1, \text{score}_2, \dots, \text{score}_{n_a});$

 Tag : $w_{\text{entity}};$

 Output : $w_{\text{entity}};$

else

 Break;

end if

end for

end for

3.3 基于依存和共指关系的上下文选择

对于基于依存和共指关系的上下文选择, 模型框架如图 2. 首先, 通过一个例子阐述共指关系在上下文选择中的作用, 如 “Astronomers discovers star with telescope. They work in observatories”, 传统的词表示方法利用的是线性词袋式固定长度的上下文来对当前词进行表示, 但是这样就存在着很多问题. 比如说, 在第一个例句中的, “discovers” 作为当前词, 如果选择窗口为 2, 那么其 Context={Astronomers, star, with}, 词 “telescope” 就没有被选入到这个窗口内, 但是很明显 “telescope” 要比 “with” 和 “discovers” 的关系更为紧密. 虽然 Levy 等提出了基于依存关系的上下文选择算法, 但是由于依存关系仅限在句内, 并不能跨句子进行, 所以它也仅仅是一个局部上下文. 例如, 例句中的 “Astronomers”, 利用 Levy 的方法就不能将例子的第二句中的 “observatories” 选入它的上下文中, 虽然它们之间关系比较紧密. 所以, 本文在 Levy 的方法基础上提出了基于共指关系的非线性全局上下文词表示方法. 共指关系广泛存在于人类的语言中, 通过观察维基百科数据, 接近 90% 的段落中存在着共指关系, 而且这些共指关系所描述的词语往往是整个段落甚至是这篇文章中的核心词语. 因此, 通

图 2 基于依存和共指关系的词表示模型

Figure 2 Word Embedding based on dependency and coreference parsing

图 3 一个跨句子上下文词表示的例子

Figure 3 Example of dependency context with coreference

过共指关系建立句子和句子之间的联系, 这样就在文档内通过共指关系构建了句与句的联系, 句内用依存关系建立词与词的联系, 如图 3 所示。

如图 2, 可以看出本文的方法选择的上下文不是线性词袋式的, 而是基于依存关系的局部上下文 $\text{Context}_{\text{dep}}(w)$ 和基于共指关系建立的全局上下文 $\text{Context}_{\text{coref}}(w)$ 。这样的上下文更能体现词的分布情况, 利用这样的非线性局部和全局上下文对词进行表示学习效果会更好。对于中间层和输出层, 本文采用的和 Word2Vec 一致的方法来进行推理和计算。表 1 是图 3 文本的上下文选择。

4 实验结果及分析

4.1 数据采集与预处理

本文采集 2014 年 5 月的英文维基百科¹⁾ 所有文章的摘要作为实验语料, 然后对其进行主题词抽取、依存和共指关系分析, 主题词抽取利用的是词检索技术来完成, 依存关系和共指关系分析采用的

1) <http://www.wikipedia.org/>.

表 1 基于依存和共指关系的上下文选择
Table 1 Coreference-based context extraction example

Word	Contexts
astronomers	discover, work
discover	astronomers, star, telescopes
star	discover
telescopes	discover
work	astronomers, observatories
observatories	work

是斯坦福大学的 NLP 工具²⁾ 来完成. 通过分析和整理, 实验语料有超过 7 百万的词 (word) 和 3 千万的上下文 (contexts).

1) **维基百科** 维基百科网站是一个依靠群体智能构建的世界知识库. 它是以文章 (article) 为基本单位, 文章内有摘要文本、属性文本 (包括信息框)、内链信息等内容. 因其规模巨大和开放性, 深受用户的喜爱, 尤其是对于从事语言计算的研究人员来说, 它就是一个带有标注信息的海量语料库. 因此, 本文同样选择英文维基数据作为语料进行词表示学习.

2) **维基排歧页** 人类的语言中经常会有一词多义、人重名等情况, 所以维基网站就给这些一词多指的问题建立了排歧页, 使用户感受到最佳的知识服务体验. 如 “Apple” 这个词, 在维基百科的排歧页里就有十多个不同的实体指向相对应的 “article”, 有水果类的 “apple”, 也有指向公司的 “Apple” 等, 这些排歧页给处理语料中的一词多义提供了非常好的先验知识. 因此, 本文给这些排歧页构建 K-Value 的映射集合. 其中 K 值是一词多指中的实体 mention, Value 是指同一 mention 对应的不同实体集合.

3) **维基百科内链** 维基百科数据最吸引语言计算学者的就是, 它的文章内部还有大量的内链信息, 即文本中出现其他的维基 article, 就会加上内部链接信息. 在本文的模型中假设这些内链信息是这个文章的主题词分布. 本文构建文章 title 和这些内链的映射表集合, 为排歧做准备.

4) **语料文档的主题词分布** 对于维基语料, 首先对其每篇文档进行主题词抽取. 文档中的所有和维基 article 标题相同的词构成这篇文档的主题词集合. 先对维基文章的所有 title 进行索引, 然后在此基础上查找语料中每篇文档的主题词, 这样就构建了每篇文档的主题词表.

5) **依存关系和共指关系标注** 对于语料中的依存关系和共指关系的标注问题, 本文采用了 Stanford NLP Tool 来进行处理. 语料是基于维基百科的常规语言语料, Stanford 的工具对标注常规语言的精度达到 90% 以上, 本文以这个工具标注实验语料作为依存关系和共指关系抽取的结果.

4.2 实验结果及分析

本文采用定性 (qualitative evaluation) 和定量 (quantitative evaluation) 两方面对实验结果进行评估. 在定性评估中对比的方法是 Word2Vec³⁾ 和 Levy 提出的 Word Embedding based on dependency parsing⁴⁾ (下文简写成 DEP) 方法, 在 Word2Vec 中选取滑动窗口的值分别是 2 和 5.

1) 定性分析 (qualitative evaluation) 第一个评价是从定性的方面进行, 通过手工随机挑选 4 个专有名词 (学校名、人名、地名、国家名) “hogwarts”, “turing”, “florida”, “china” 的最相似的前 5 个单词

2) <http://nlp.stanford.edu/index.shtml>.

3) code.google.com/p/Word2Vec/.

4) <http://irsr2.cs.biu.ac.il:9998/?word=h>.

表 2 目标词与其最相似的 5 个词

Table 2 Target words and their 5 most similar words, which are induced by different embeddings

Target	BoW2	BoW5	DEP	OurModel
hogwarts	evernight	dumbledore	sunnydale	gordonstoun
	sunnydale	hallows	collinwood	churchlands
	garderobe	half-blood	calarts	larkhill
	blandings	malfoy	greendale	esade
	collinwood	snape	millfield	oundle
turing	non-deterministic	non-deterministic	pauling	hawking
	finite-state	nondeterministic	hotelling	dedekind
	nondeterministic	deterministic	heyting	fitting
	bchi	finite-state	lessing	boltzmann
	primality	reducibility	hamming	feynman
florida	fla	jacksonville	texas	georgia
	alabama	tallahassee	louisiana	fla.
	gainesville	miami	georgia	carolina
	tallahassee	gainesville	california	virginia
	texas	sarasota	carolina	louisiana
china	korea	taiwan	korea	mongolia
	taiwan	fujian	iran	taiwan
	yunnan	guangdong	mongolia	thailand
	mongolia	yunnan	yunnan	korea
	japan	korea	cambodia	cambodia

(余弦相似性) 进行对比. 对比结果见表 2, BoW2 和 BoW5 分别表示滑动窗口为 2 和 5 的 Word2Vec 方法.

从表 2 中可以看出, 第一个词 hogwarts 是哈利波特中的魔法学校的名称, DEP 和本文提出的方法明显好于 Word2Vec 的结果, 本文提出的方法的结果都是学校的名称, 从语义相似度上来说更加接近目标词 hogwarts. 第二个词 turning 是科学家图灵的名字, 他是一位计算机领域和数学领域著名的科学家, 在 Word2Vec 的结果里出现最多的是一些数学或计算机领域中的概念, 而 DEP 和本文提出的方法中出现的都是人名, 不仅如此本文提出的结果全都是著名的科学家, 而 DEP 的结果中有一位不是科学家, lessing 是一位作家. 第三个词 florida 是美国的佛罗里达州名, 在 BoW2 和 BoW5 的结果中包括在美国的州和城市名字, 而 DEP 和本文提出方法的结果全都是美国的州名, 不过本文提出方法的结果里州都是地理位置靠近佛罗里达地域的, 而 DEP 的结果中加州并不靠近佛罗里达. 最后一个词 china, 是中华人民共和国的简称, 从结果上看, 对比的方法中总会有省份的名称出现, 但本文提出方法的结果要明显好于其他的结果. 因此, 从定性的对比结果来看, 本文提出的词表示模型对词的表示结果比其他的方法效果要好, 这表明本文提出的基于依存和共指关系的词表示产生的词向量质量更高.

本文还对比了一词多义的情况, 在表 3 显示了 3 个多义词的对比结果. 其中, 带上标 “*” 的是 Levy 提出的方法 DEP 的最相似的 5 个词结果, 带下标的是本文提出方法的结果. 可以非常明显地看出, Levy 的方法没有考虑一词多义的情况, 所以遇到多义词, 它的表示能力就会受限, 而本文提出的方

表 3 3 个多义词最相似的 5 个词的对比结果

Table 3 Most similar words of word embeddings learned by our model using the word sense disambiguation approach

Word or sense	Most similar words
apple*	apricot, acorn, blackberry, doritos, asus
apple_fruit	cherry, almond, pomegranate, pear, lemon
apple_inc.	acorn, netscape, oracle, mattel, microsoft
gates*	gate, doors, floodgates, doorways, sluices
gates_entrance	ramparts, entrances, barricades, escalators, doorways
gates_name	barrows , bunker, corwin, flagg, carver
star*	stars, heartthrob, dwarf, starlet, subgiant
star_cosmic	supergiant, deneb, tauri, pegasi, cephei
star_movie	starred, co-stars, starring, actresses, superstars

表 4 词类比结果

Table 4 Word total accuracy in percentages

Type	BoW_windows_5	Our model
capital common countries	44.86	59.61
capital world	21.37	34.73
currency	20.05	33.27
city in state	16.24	26.43
family	19.48	28.86
adjective to adverb	17.89	25.87
opposite	17.19	26.55
comparative	20.26	26.70
superlative	20.20	27.73
present participle	20.24	25.90
nationality adjective	21.35	28.71
past-tense	22.19	28.73
plural	22.96	29.33
plural-verbs	23.64	27.96

法利用了维基百科的排歧页来处理一词多义的情况, 所以实验结果明显提升了很多, 使得每个多义词都能够找到语义上最接近的它的词. 通过定性的分析结果显示, 本文提出的方法对词的表示效果更好, 质量更高.

2) 定量分析 (quantitative evaluation) 除了定性的词分析外, 本文还采用 Mikolov 在文献 [32] 中设计的一套词类比 (word-analogy) 的测试方法. 比如 “king” – “queen” = “man” – “woman”, 如果我们知道其中的三个词, 我们就能找到第四个词. 本文利用文献 [32] 中整理的数据集, 包括上万对的类比关系, 称之为 question-word task, 根据 question 的不同这些关系分成 14 个类型. 表 4 给出了词类比的的结果. 我们可以看到, 在这个任务中, 本文提出的方法大大地提高了整体精度. 提高精度的主要原因是大量的指代词被名词取代, 这丰富了 context 以至于提高了词表示的精度.

表 5 WordSim-353 数据上 Spearman 相关系数对比

Table 5 Spearman ρ correlation on WordSim-353, showing our models improvement over previous neural models for learning word embeddings. The results of the first three rows come from [21]

Model	Corpus	$\rho \times 100$
C&W*	Wiki_2010	49.8
C&W	Wiki_2010	55.3
Huang's Model	Wiki_2010	64.2
BoW_windows_2	Wiki_2014	63.3
BoW_windows_5	Wiki_2014	64.9
Levy's Model	Wiki_2014	65.3
Our Model	Wiki_2014	69.6

本文还在定量上对模型结果进行了评估. 为此, 使用 WordSim353 数据集^[33] 作为实验对比的标注数据集. 该数据集包含由人工对 353 个词对进行的相似性判定. 通过计算本文提出的方法在这个数据集词对上相似性得分和人工判断的相似性得分的 Spearman 相关系数来定量评估. 本文提出的方法与 Collobert 和 Weston 提出的方法 C&Ws^[34]、Huang 提出的方法^[21]、Word2Vec 和 Levy 提出的 DEP 方法^[5] 进行对比, 具体结果见表 5, 其中结果中的 C&W 和 Huang 的方法的结果直接来自于文献 [21]. 从表 5 可以看出本文提出的方法的结果是最好的, 实验结果显示在定量评估上本文提出的模型优于其他的方法.

通过定性和定量上对基于共指消解和词义排歧的词表示方法进行评估, 结果显示该方法在各个方面都明显好于同类方法. 由此, 可以证明该方法对当今现有的词表示方法在效果上有很大地提升.

5 结语

本文针对当前 Word Embedding 方法中存在的问题, 研究了利用依存和共指关系以及利用维基百科的排歧页对语料进行词义排歧处理对 Word Embedding 的效果提升问题, 提出了基于共指消解和词义排歧的 Word Embedding 的方法, 同时, 在英文维基百科语料上进行了实验, 结果显示本文提出的方法无论在定性还是定量的评估中都好于其他的方法, 且极大地改善了 Word Embedding 的效果.

参考文献

- Salton G, McGill M J. Introduction to Modern Information Retrieval. New York: McGraw-Hill, 1986
- Le Q V, Mikolov T. Distributed representations of sentences and documents. arXiv:1405.4053, 2014
- Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intel, 2013, 35: 1798–1828
- Hinton G E. Learning distributed representations of concepts. In: Proceedings of the 8th Annual Conference of the Cognitive Science Society. 1986, 1: 12
- Levy O, Goldberg Y. Dependencybased word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, 2014, 2: 302–308
- Devlin J, Zbib R, Huang Z, et al. Fast and robust neural network joint models for statistical machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, 2014, 1: 1370–1380
- Will Y Z, Richard S, Daniel M C, et al. Bilingual word embeddings for phrase-based machine translation. In: Proceedings of the Empirical Methods on Natural Language Processing, Seattle, 2013. 1393–1398

- 8 Tang D Y, Wei F R, Yang N, et al. Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, 2014, 1: 1555–1565
- 9 Zeng D J, Liu K, Lai S W, et al. Relation classification via convolutional deep neural network. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING-2014), Dublin, 2014. 2335–2344
- 10 Chen Y B, Xu L H, Liu K, et al. Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-2015), Beijing, 2015. 26–31
- 11 Lai S W, Xu L H, Liu K, et al. Recurrent convolutional neural networks for text classification. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15), Austin, 2015. 2267–2273
- 12 Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS), Las Vegas, 2013. 2787–2795
- 13 Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, Québec, 2014. 1112–1119
- 14 Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, 2015. 25–30
- 15 Ji G L, He S Z, Xu L H, et al. Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-2015), Beijing, 2015. 687–696
- 16 Zhou G Y, He T T, Zhao J, et al. Learning continuous word embedding with metadata for question retrieval in community question answering. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-2015), Beijing, 2015. 250–259
- 17 Turian J, Ratnoff L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2010. 384–394
- 18 Hinton G E, McClelland J L, Rumelhart D E. Distributed representations. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press, 1986. 77–109
- 19 Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. *J Mach Learn Res*, 2003, 3: 1137–1155
- 20 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS), Las Vegas, 2013. 3111–3119
- 21 Huang E H, Socher R, Manning C D, et al. Improving word representations via global context and multiple word prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2012, 1: 873–882
- 22 Reisinger J, Mooney R J. Multi-prototype vector-space models of word meaning. In: *Human Language Technologies: the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 2010. 109–117
- 23 Neelakantan A, Shankar J, Passos A, et al. Efficient non-parametric estimation of multiple embeddings per word in vector space. arXiv:1504.06654, 2015
- 24 George A M. Wordnet: a lexical database for english. *Commun ACM*, 1995, 38: 39–41
- 25 Chen X, Liu Z, Sun M. A unified model for word sense representation and disambiguation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, 2014. 1025–1035
- 26 Mnih A, Hinton G E. A scalable hierarchical distributed language model. In: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, 2009. 1081–1088
- 27 Turian J, Ratnoff L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2010. 384–394
- 28 Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, 2008. 160–167
- 29 Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch. *J Mach Learn Res*. 2011, 12: 2493–2537
- 30 Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization. In: Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS), Montréal, 2014. 2177–2185
- 31 Shen W, Wang J Y, Luo P, et al. LIEGE: link entities in web lists with knowledge base. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2012. 1424–1432

- 32 Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013
- 33 Finkelstein L, Gabrilovich E, Matias Y, et al. Placing search in context: the concept revisited. In: Proceedings of the 10th International Conference on World Wide Web, Hongkong, 2001. 406–414
- 34 Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, 2008. 160–167

Word embedding based on nonlinear global context

LIU YongBin^{1,2}, OUYANG ChunPing^{1*}, ZHONG DongLai^{2*}, LI JuanZi², YUAN BoZhi² & LI Qi²

1 School of Computer Science and Technology, University of South China, Hengyang 421001, China;

2 Knowledge Engineering Lab (Group), Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

*E-mail: ouyangcp@126.com, tongleechung86@gmail.com

Abstract With the aim of limiting the local context in word presentations, this paper presents a method for word representation that is based on nonlinear global context. First, the word senses can be disambiguated using the disambiguation page of Wikipedia, which can improve the effectiveness of the representation. Secondly, considering the linear local-context problem of the traditional word representation, we apply the dependency and coreference relation analysis to treat the global context of corpus, after which we obtain the word vector based on the nonlinear global context. Finally, we perform the experiments using the English Wikipedia. In the qualitative analysis, this method can find a related word that is closer to the current meaning of the word, and it is better than the Word2Vec method in the word-sense disambiguation. From the quantitative perspective, by comparing the experimental results in the WordSim-353 test data set, the experimental results show that the proposed method has a higher performance than the other methods in the Spearman correlation coefficient.

Keywords nonlinear global context, coreference resolution, word-sense disambiguation, word vector, word embedding, neural networks, deep learning, dependency parsing

LIU YongBin was born in 1978. He received his Ph.D. degree from the University of Science & Technology Beijing, China, in 2013. He is a vice-professor of computer science and a postgraduate supervisor. His current research interests include natural language processing, machine learning, knowledge engineering, and semantic web.

OUYANG ChunPing was born in 1979. She received her Ph.D. degree from the University of Science & Technology Beijing, China, in 2011. She is a vice-professor of computer science and a postgraduate supervisor. Her research interests include natural language processing and information retrieval.

ZHONG DongLai was born in 1986. He is a Ph.D. candidate in the Department of Computer Science and Technology, Tsinghua University. His research interests include knowledge-based construction and machine learning.

LI JuanZi was born in 1964. She is a Professor in the Department of Computer Science and Technology at the Tsinghua University. Her research interests are semantic web and semantic web services, text mining and knowledge discovery.