文章编号：1003-0077(2020)10-0027-06

基于同义词词林和预训练词向量的微调方法

佘琪星[1]，王必聪[1]，刘 铭[1,2]，秦 兵[1,2]，王莉峰[3]

(1. 哈尔滨工业大学 社会计算与信息检索研究中心，黑龙江 哈尔滨 150001；
2. 鹏程实验室，广东 深圳 518055；
3. 腾讯科技(深圳)有限公司，广东 深圳 518000)

摘 要：同义词挖掘是自然语言处理领域中的一个基础任务，而同义词对的判别是该任务的一个重要部分。传统两大类方法，基于分布式表示和基于模板的方法，分别利用了语料的全局统计信息和局部统计信息，只能在精确率和召回率中权衡。随着预训练词向量技术的发展，基于分布式表示的方法存在一种简单高效的方案，即直接对预训练好的词向量计算相似度，将此表示为语义相似度。然而，这样的思路并没有利用到现有的同义词对这一外部知识。该文提出基于《同义词词林》的词向量微调方法，利用同义词对信息，增强预训练词向量的语义表示。经过实验，该微调方法能很好地完成同义词对的判别。

关键词：同义词挖掘；预训练词向量；语义表示；微调

中图分类号：TP391 **文献标识码**：A

A Fine-tuning Method Based on *Tongyi Cilin* and Pre-trained Word Embedding

SHE Qixing[1], WANG Bicong[1], LIU Ming[1,2], QIN Bing[1,2], WANG Lifeng[3]

(1. Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China;
2. Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China;
3. Tencent Technology (Shenzhen) Co., Ltd, Shenzhen, Guangdong 518000, China)

Abstract: Synonym discovery is a typical task in natural language processing, aiming at predicting whether a given word is a synonym of another word. With pre-trained word embedding appeared recently, a simple and effective distributional based approach is available by exploiting the similarity between word embeddings. To further augmenting external knowledge such as synonym tuples, this paper proposes a word embedding fine-tuning approach based on synonym tuples in *Tongyi Cilin*, so as to enhance the semantic representation of embedding. Our experiments show this approach is effective for predicting synonyms.

Keywords: synonym discovery; pre-trained word embedding; semantic representation; fine-tuning

0 引言

同义词挖掘是自然语言处理领域中的基础任务，其目的是在大规模文本中挖掘出同义词信息，挖掘出的同义词信息能够提高许多下游任务的效果。例如，在文本检索中，可以利用同义词信息对请求进行处理，得到更加准确的检索结果[1]。在主题建模中，可以利用同义词信息，把属于同一个同义词组的文本合并，获得更加高质量的主题文本[2]。

同义词挖掘任务的目的是预测一个词对是否是同义词。对于这样的任务，我们通常需要同义词对作为监督信息。然而，现有的同义词对数据很难获得，数量偏少。因此，需要更好地利用这些同义词信息。在过去的工作里，主要有两类方法研究这个问题[3]：

第一种是基于分布式表示的方法[4-7]。基于分布式表示的方法利用了语料级别的全局统计信息，

收稿日期：2019-09-25 **定稿日期**：2019-10-25

基金项目：国家自然科学基金(61772156，61976073)；黑龙江省自然科学基金(F2018013)

认为具有相似上下文的词有更大的可能性是同义词。例如,“哈尔滨工业大学”和“哈工大”具有相似的上下文,它们都是“哈工大”这个实体的同义词。基于这样的假设,我们可以用一个词的上下文信息作为这个词的分布式表示,这样的分布式表示可以用于训练一个同义词对分类器。然而,这样的方法也会引入一些噪声,很多非同义词也具有相似的上下文,所以这个同义词分类器具有较高的召回率,却难以保证较高的精确率。

第二种是基于模板的方法[8-9]。基于模板的方法利用了当前句子的局部上下文信息,根据上下文信息可以推出句中提到的两个词语是同义词。例如,在句子“哈尔滨工业大学,简称哈工大……”中,根据句中的上下文信息可以得出“哈尔滨工业大学”和“哈工大”是同义词。我们可以挖掘出同义词的共现模板,进而利用这些模板挖掘出更多的同义词。同时,这样的方法具有很强的信服度和可解释性。然而,由于许多同义词不会在语料的任何句子中共现,所以这样的方法通常会导致较低的召回率。

最近几年的工作中,预训练词向量已经在许多任务里取得了比较好的效果,例如,Word2Vec[4]、Glove[5]等,这样的背景让第一种方法具有了更加可靠和简单的实现。我们可以通过这些预训练好的词向量,直接计算余弦相似度进行语义相似程度判断。然而,这样的方法虽然简单有效,但通常也难以达到特别好的效果,我们只能通过设置余弦相似度的阈值,让其做同义词判断的精确率和召回率达到一个适当的折中值。上述方法存在的问题在于我们只用到了大规模语料上的统计信息,而没有利用到现有的同义词信息。

本文我们提出一种基于《同义词词林》的预训练词向量微调技术,把《同义词词林》中的同义词信息融入到了预训练的词向量中。《同义词词林》是一个包含同义、同类和独立关系的层次化词典,如图 1 所示。我们提出的方法将《同义词词林》中的同义词数据以同义词对的形式,对预训练好的词向量进行微调,保证让同义词对在向量空间中的距离更短,让相应非同义词对在向量空间中的距离更长。这样的方法保证了我们能够在不破坏具有相似语义信息的词语在向量空间中距离更接近的前提下,引入了同义词对这一外部知识。实验结果表明,我们提出的这一方法能够更好地用于同义词对的判别。

编码位	1	2	3	4	5	6	7	8
符号举例	D	a	1	5	B	0	2	= \ # \ @
符号性质	大类	中类	小类		词群	原子词群		
级别	第 1 级	第 2 级	第 3 级		第 4 级	第 5 级		

图 1 《同义词词林》数据模式

1 方法描述

本文实现了用于同义词语义的微调方法,该方法利用两个词对之间的相似性约束,让词向量包含更加充分的语义相似度信息。对于许多常见的预训练词向量,例如 Word2Vec、Glove 等,我们可以直接使用余弦相似度计算其语义相似度,如式(1)所示。

$$\text{similarity}=\cos(\theta)=\frac{A\cdot B}{AB}=\frac{\sum_{i=1}^{n}A_i\times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\times\sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{1}$$

其中,A 和 B 分别表示词对中两个词的词向量。

我们的目标是希望能够利用到同义词对这一外部知识,进而让预训练好的词向量里同义关系得到增强。通过对词向量进行微调,令词向量中满足同义关系的词语在向量空间中更加接近,如图 2 所示。

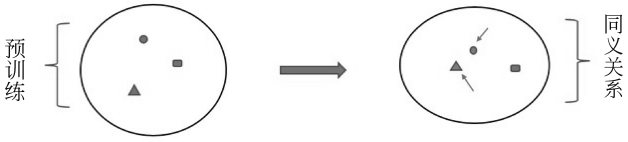

图 2 预训练微调方法简述图

在该方法里,我们将同义信息融入到了预训练好的词向量中。在《同义词词林》中,同义信息的具体形式为多个同义词簇,每个同义词簇的样例如下:

男人 男子 男子汉 男儿 汉子 汉 士 丈夫 男人家 须眉 男士

对于每个同义词簇,其长度为 2 的子集合,是一组同义词对,例如,

男人 男子

根据这些同义词对,我们可以调整预训练好的词向量,基本的指导思想是:

(1) 拉近同义词对中两个词在向量空间中的距离;

(2) 拉远每个词语与其临近的非同义词的

距离；

（3）保证微调后不破坏大规模文本中的语义信息。

基于这样的思路，我们实现了一个能够满足上述要求的损失函数，这个损失函数的原型来自于文献[10]的工作。

1.1　词对距离损失函数

在词对距离损失函数中，我们定义了两种约束方式：吸引约束和排斥约束。

在《同义词词林》数据中，满足吸引约束的实例为所有的同义词对，例如，

男人 男子

满足排斥约束的实例为某个词和其距离最近的非同义词组成的词对，例如，

男人 女人

利用上述约束信息，我们可以定义一个词对距离损失函数，其示意图如图 3 所示。

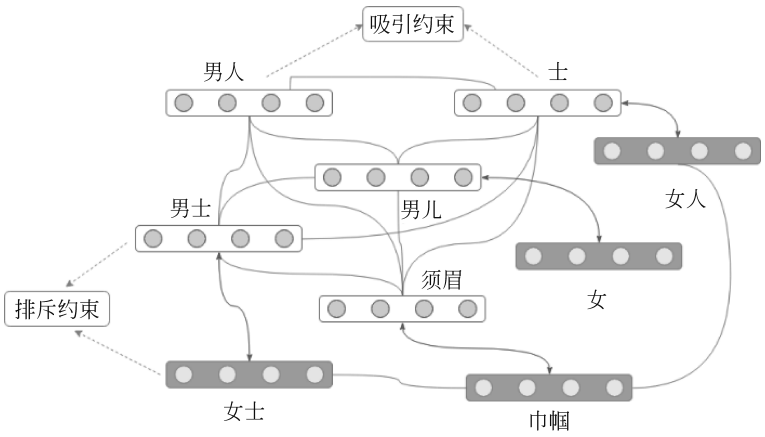

图 3　词对距离损失示意图

其中正例为满足吸引约束的实例，即《同义词词林》数据中的同义词对，负例为当前正例中每个词与其距离最近的非同义词组成的词对。这个思路的形式化描述如式(2)所示。

$$O_C(B_C)=\sum_{(x_l,x_r)\in B_C}[\tau(\delta_{att}+x_l t_l-x_l x_r)+\tau(\delta_{att}+x_r t_r-x_l x_r)] \tag{2}$$

其中，B_C 指的是当前批次的同义词对；τ 是标准的线性整流单元(ReLU)，即 $\tau(x)=\max(0,x)$；δ_{att} 是词对间距，用来标识当前正例和负例的距离差；其中 x_l、x_r 分别表示当前批次的同义词对 (x_l,x_r) 中的左侧和右侧的词，t_i 表示距离词 $x_i(i\in(l,r))$ 最近的非同义词，即没有和词 x_i 在同义词对中共现过的词。

1.2　正则损失函数

为了保证微调后的词向量不破坏原有大规模语料的统计信息，这里采用了L2正则项，用来限制微调前后词向量的距离。其形式化描述如式(3)所示。

$$R(B_C)=\sum_{x_i\in V(B_C)}\lambda_{\text{reg}}\mid \boldsymbol{x}_i^{\text{init}}-\boldsymbol{x}_i\mid_2 \tag{3}$$

其中，$R(B_C)$ 表示 B_C 这个当前批次的词对中所包含的所有词；λ_{reg} 表示L2正则的权重超参数；$\boldsymbol{x}_i^{\text{init}}$ 表示预训练的词向量；$\boldsymbol{x}_i$ 表示微调后的词向量。

1.3　总损失函数

总损失函数表示词对距离损失和正则损失的和，其形式化描述如式(4)所示。

$$O(B_C)=O_C(B_C)+R(B_C) \tag{4}$$

1.4　词向量微调模型

为了保证词向量微调的有效性，这里采用一个包含多层非线性变换的多层感知机(MLP)来表示微调模型，其形式化描述如式(5)所示。

$$\boldsymbol{x}_i=\text{MLP}(\boldsymbol{x}_i^{\text{init}}) \tag{5}$$

2　实验分析

2.1　数据集

同义词词林　《同义词词林》按照树状的层次结构把所有收录的词条组织到一起，把词汇分成大、中、小三类，大类有 12 个，中类有 97 个，小类有 1 400 个。每个小类里都有很多的词，这些词又根据词义的远近和相关性分成了若干个词群(段落)。每个段落中的词语又进一步分成了若干个行，同一行

的词语要么词义相同,要么词义有很强的相关性。对于每行数据,“=”“@”“\#”分别代表同义、独立和同类,数据的样例如下:

Ba01A02= 物质 质 素

Cb02A01= 东南西北 四方

Ba01A03@ 万物

Cb06E09@ 民间

Ba01B08# 固体 液体 气体 流体 半流体

Ba01B10# 导体 半导体 超导体

同义词对 本文的数据集从《同义词词林》中构建,根据数据规模分为通用领域、特定领域两个数据集。其中通用领域的数据规模较大,是《同义词词林》中包含的所有同义词对。特定领域由《同义词词林》词表与腾讯社交广告业务词表的交集构建出,数据规模相对较小。构建步骤为:

(1) 计算出所有的同义词对作为正例;

(2) 随机抽样出词对,如果词对不是正例,则作为负例;

(3) 将同义词的正例对按照一定比例划分为训练集、验证集和测试集;

(4) 在验证集和测试集中加入数量基本相同的负例。

基于本文提到的微调方法,在训练阶段负例是通过计算某词最近的非同义词得到的,因此在训练集中不需要人为构建负例。具体的数据规模如表 1 所示。

表 1 数据集统计信息

数据集	训练集	验证集	测试集
通用领域	279 723	10 000	10 000
特定领域	29 000	3 820	3 713

2.2 词向量

根据本文提出的方法,我们查阅了目前开放的高质量中文词向量,选择了其中几个进行我们的实验验证。

Tencent AI Lab Embedding[7] Tencent AI Lab Embedding 是腾讯公司 AI Lab 开源的 200 维词嵌入向量表示,覆盖了 800 万级别的中文词语和短语。训练的语料来源广泛,包含新闻、网页和小说等,来自不同领域的训练数据可以获得各种类型的词语和短语,并且可以较好地得到新词的语义表示。其训练方法主要为 Directional Skip-Gram 方法,同时考虑了词语共现以及词对之间的方向关系。这些训练上的特点使得最后得到的词嵌入向量具有覆盖率高和正确率高的特点。

FastText[11] FastText 是 Facebook AI Research 开源的一个文本分类器,其中词向量是分类任务的一个产物。在本实验中我们使用开源的 300 维词向量进行实验。

Word2Vec[4] Word2Vec 中文词嵌入向量是采用了由 Mikolov 等人提出的 Word2Vec 模型中的 CBOW 和 Skip-Gram 等语言模型,在中文维基百科上进行训练得到的,是现在较为常见的一种中文词嵌入向量,我们在实验中采用了 300 维的版本进行实验。

表 2 原始词向量实验结果(通用领域)

词向量	*F* 值 (*F*-Measure)	精确率 (Precision)	召回率 (Recall)
Tencent AI Lab Embedding	0.807 8	0.776 1	0.842 2
FastText	0.769 1	0.741 3	0.799 1
Word2Vec	0.797 9	0.795 6	0.800 2
Wikipedia2Vec	0.697 5	0.553 2	0.943 7
Specific Field Embedding	0.730 7	0.6852	0.782 8

表 3 微调词向量实验结果(通用领域)

词向量	*F* 值 (*F*-Measure)	精确率 (Precision)	召回率 (Recall)
Tencent AI Lab Embedding	0.842 8	0.805 8	0.883 3
FastText	0.818 2	0.773 1	0.868 9
Word2Vec	0.842 3	0.805 0	0.883 1
Wikipedia2Vec	0.753 2	0.655 7	0.884 8
Specific Field Embedding	0.752 2	0.679 5	0.842 3

表 4 *F* 值实验结果对比(通用领域)

词向量	原始词向量	微调词向量	净增长
Tencent AI Lab Embedding	0.807 8	0.842 8	0.035 0
FastText	0.769 1	0.818 2	0.049 1
Word2Vec	0.797 9	0.842 3	0.044 4
Wikipedia2Vec	0.697 5	0.753 2	0.055 7
Specific Field Embedding	0.730 7	0.7522	0.021 5

Wikipedia2Vec[6] Wikipedia2Vec采用了 Conventional Skip-Gram 模型进行训练得到词嵌入向量表示,并且被作为众多 NLP 任务的 SOTA 模型

的基础部分。类似于 Word2Vec，Wikipedia2Vec 也是在中文维基百科上进行训练。我们在实验中也采用了 300 维的版本进行实验。

Specific Field Embedding　在腾讯社交广告领域上用大量文本训练出的词向量，与其他词向量的主要区别在于语料来源。

2.3　实验配置

在本实验中，我们使用 Adam 算法[12]进行优化，其中学习率 α 设置为 0.001，β_1 设置为 0.9，β_2 设置为 0.999。同时，由于词对数据明显小于其他类型的文本数据，我们将优化的批次大小设置为 256，训练的轮数设置为 20。对于损失函数中的超参数，我们设置 δ_{att} 为 0.4，设置 λ_{reg} 为 1。

2.4　评价指标

根据过去的工作，我们使用精确率(precision)、召回率(recall)和 F 值(F-measure)作为同义词判别优劣的评价指标。其中，阈值(threshold)由验证集数据根据最优的 F 值进行选择，然后将该阈值作为测试集中需要使用的阈值。

3　实验结果

本文主要针对上述提到的两个数据集，对多个中文词向量进行了实验，其目的是对比经过微调后同义词判别任务的指标是否有明显上升，以下为主要的实验结果。

3.1　通用领域实验结果

通用领域的实验结果如表 2～表 4 所示，表 2 展示了原始词向量的实验结果，表 3 展示了微调词向量的实验结果，表 4 展示了对比实验结果。从对比结果中，我们能够明显看到在多个预训练词向量上，微调后的词向量在同义词判定任务上都具有明显的效果提升。

3.2　特定领域实验结果

特定领域的实验结果如表 5～表 7 所示，表 5 展示了原始词向量的实验结果，表 6 展示了微调词向量的实验结果，表 7 展示了对比实验结果。由于特定领域的词向量更易成簇，质量更高，所以原始词向量和微调词向量的效果相比通用领域基本都偏好。但是，微调词向量的提升效果不如通用领域明显，主要原因是特定领域的同义词对数量相比通用领域少了一个数量级。

表 5　原始词向量实验结果(特定领域)

词向量	F 值(F-Measure)	精确率(Precision)	召回率(Recall)
Tencent AI Lab Embedding	0.841 8	0.806 8	0.879 9
FastText	0.820 3	0.791 3	0.851 4
Word2Vec	0.844 5	0.825 3	0.864 7
Wikipedia2Vec	0.726 1	0.613 0	0.890 2
Specific Field Embedding	0.818 6	0.770 0	0.873 6

表 6　微调词向量实验结果(特定领域)

词向量	F 值(F-Measure)	精确率(Precision)	召回率(Recall)
Tencent AI Lab Embedding	0.859 7	0.795 6	0.935 0
FastText	0.838 8	0.778 1	0.909 8
Word2Vec	0.874 1	0.818 4	0.938 1
Wikipedia2Vec	0.740 3	0.637 7	0.882 2
Specific Field Embedding	0.829 5	0.777 5	0.888 9

表 7　F 值实验结果对比(特定领域)

词向量	原始词向量	微调词向量	净增长
Tencent AI Lab Embedding	0.841 8	0.859 7	0.014 9
FastText	0.820 3	0.838 8	0.018 5
Word2Vec	0.844 5	0.874 1	0.026 9
Wikipedia2Vec	0.726 1	0.740 3	0.014 2
Specific Field Embedding	0.818 6	0.829 5	0.010 9

4　结论

在本文中，我们基于《同义词词林》提出一种预训练词向量的微调方法。该方法基于词语的分布式表示，利用同义词对这一外部知识，让词向量学习到更加符合同义信息的词向量。然后，我们在《同义词词林》中构建出同义词对数据集并进行实验，实验结果表明该方法在多个词向量和不同领域覆盖度上都具有明显的提升效果。

参考文献

[1]　Voorhees E M. Query expansion using lexical-semantic

relations[C]//Proceedings of SIGIR'94. Springer, London, 1994: 61-69.

[2] Xie P, Yang D, Xing E. Incorporating word correlation knowledge into topic modeling[C]//Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015: 725-734.

[3] Qu M, Ren X, Han J. Automatic synonym discovery with knowledge bases[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017: 997-1005.

[4] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv: 1301.3781, 2013.

[5] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1532-1543.

[6] Yamada I, Asai A, Shindo H, et al. Wikipedia2Vec: An optimized tool for learning embeddings of words and entities from Wikipedia[C]//Proceedings of the CoRR, 2018.

[7] Song Y, Shi S, Li J, et al. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018: 175-180.

[8] Hearst M A. Automatic acquisition of hyponyms from large text corpora[C]//Proceedings of the 14th Conference on Computational Linguistics, Association for Computational Linguistics, 1992: 539-545.

[9] Snow R, Jurafsky D, Ng A Y. Learning syntactic patterns for automatic hypernym discovery[C]//Proceedings of Advances in Neural Information Processing Systems, 2005: 1297-1304.

[10] Wieting J, Bansal M, Gimpel K, et al. From paraphrase database to compositional paraphrase model and back[J]. Transactions of the Association for Computational Linguistics, 2015, 3: 345-358.

[11] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 135-146.

[12] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv: 1412.6980, 2014.

佘琪星(1997—),硕士研究生,主要研究领域为知识图谱、信息抽取。
E-mail:qxshe@ir.hit.edu.cn

刘铭(1981—),通信作者,博士,副教授,主要研究领域为文本挖掘、知识图谱。
E-mail: mliu@it.hit.edu.cn

王必聪(1996—),硕士研究生,主要研究领域为知识图谱、信息抽取。
E-mail: bcwang@ir.hit.edu.cn