[论文阅读] (24) 向量表征：从Word2vec和Doc2vec到Deepwalk和Graph2vec，再到Asm2vec和Log2vec (一)

原创 Eastmount 已于 2022-11-28 16:51:52 修改 1539 已收藏 6 编辑 版权
分类专栏： 娜璋带你读论文 文章标签： 论文阅读 word2vec 人工智能

娜璋带你读论文 专栏收录该内容 163 订阅 26 篇文章

《娜璋带你读论文》系列主要是督促自己阅读优秀论文及听取学术讲座，并分享给大家，希望您喜欢。由于作者的英文水平和学术能力不高，需要不断提升，所以还请大家批评指正，非常欢迎大家给我留言评论，学术路上期待与您前行，加油。

前一篇介绍了两个作者溯源的工作，从二进制代码和源代码两方面实现作者去匿名化或识别。这篇文章主要介绍六个非常具有代表性的向量表征算法，它们有特征词向量表示、文档向量表示、图向量表示，以及两个安全领域二进制和日志的向量表征。通过类似的梳理，让读者看看这些大佬是如何创新及应用到新领域的，希望能帮助到大家。这六篇都是非常经典的论文，希望您喜欢。一方面自己英文太差，只能通过最土的办法慢慢提升，另一方面是自己的个人学习笔记，并分享出来希望大家批评和指正。希望这篇文章对您有所帮助，这些大佬是真的值得我们去学习，献上小弟的膝盖~fighting！

文章目录

一.图神经网络发展历程
二.Word2vec：NLP经典工作（谷歌）
三.Doc2vec
四.DeepWalk：网络化数据经典工作（KDD2014）
五.Graph2vec
六.Asm2vec：安全领域经典工作（S&P2019）
七.Log2vec：安全领域经典工作（CCS2019）
八.总结

前文赏析：

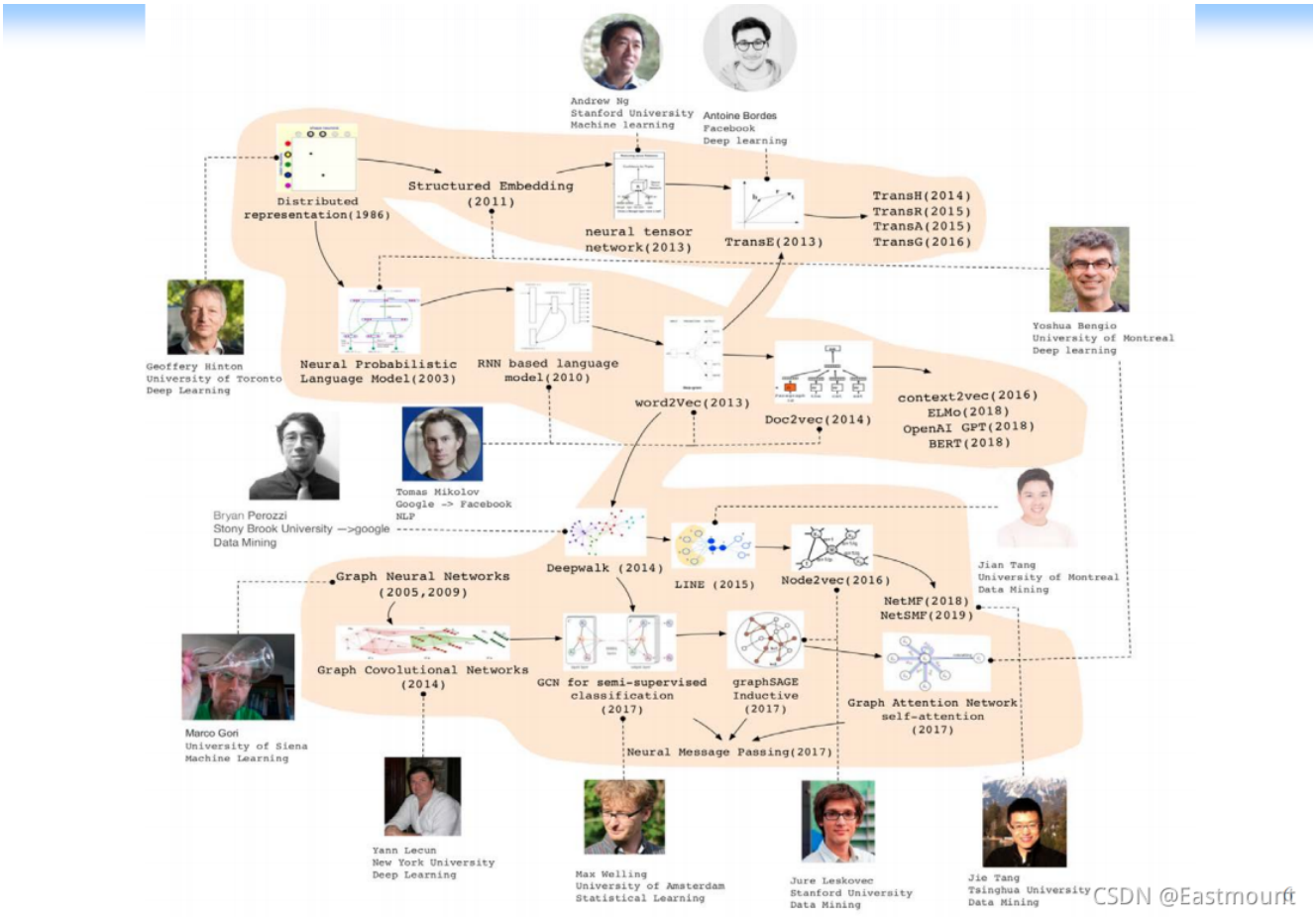
- [论文阅读] (01) 拿什么来拯救我的拖延症？初学者如何提升编程兴趣及LATEX入门详解
- [论文阅读] (02) SP2019-Neural Cleanse: Identifying and Mitigating Backdoor Attacks in DNN

- [论文阅读] (03) 清华张超老师 - GreyOne: Discover Vulnerabilities with Data Flow Sensitive Fuzzing
- [论文阅读] (04) 人工智能真的安全吗？浙大团队外滩大会分享AI对抗样本技术
- [论文阅读] (05) NLP知识总结及NLP论文撰写之道——Pvop老师
- [论文阅读] (06) 万字详解什么是生成对抗网络GAN？经典论文及案例普及
- [论文阅读] (07) RAID2020 Cyber Threat Intelligence Modeling Based on Heterogeneous GCN
- [论文阅读] (08) NDSS2020 UNICORN: Runtime Provenance-Based Detector for Advanced Persistent Threats
- [论文阅读] (09)S&P2019 HOLMES Real-time APT Detection through Correlation of Suspicious Information Flow
- [论文阅读] (10)基于溯源图的APT攻击检测安全顶会总结
- [论文阅读] (11)ACE算法和暗通道先验图像去雾算法（Rizzi | 何恺明老师）
- [论文阅读] (12)英文论文引言introduction如何撰写及精句摘抄——以入侵检测系统(IDS)为例
- [论文阅读] (13)英文论文模型设计（Model Design）如何撰写及精句摘抄——以入侵检测系统(IDS)为例
- [论文阅读] (14)英文论文实验评估（Evaluation）如何撰写及精句摘抄（上）——以入侵检测系统(IDS)为例
- [论文阅读] (15)英文SCI论文审稿意见及应对策略学习笔记总结
- [论文阅读] (16)Powershell恶意代码检测论文总结及抽象语法树（AST）提取
- [论文阅读] (17)CCS2019 针对PowerShell脚本的轻量级去混淆和语义感知攻击检测
- [论文阅读] (18)英文论文Model Design和Overview如何撰写及精句摘抄——以系统AI安全顶会为例
- [论文阅读] (19)英文论文Evaluation（实验数据集、指标和环境）如何描述及精句摘抄——以系统AI安全顶会为例
- [论文阅读] (20)USENIXSec21 DeepReflect：通过二进制重构发现恶意功能（恶意代码ROI分析经典）
- [论文阅读] (21)S&P21 Survivalism: Systematic Analysis of Windows Malware Living-Off-The-Land (经典离地攻击)
- [论文阅读] (22)图神经网络及认知推理总结和普及-清华唐杰老师
- [论文阅读] (23)恶意代码作者溯源(去匿名化)经典论文阅读：二进制和源代码对比
- [论文阅读] (24)向量表征：从Word2vec和Doc2vec到Deepwalk和Graph2vec，再到Asm2vec和Log2vec

一.图神经网络发展历程

在介绍向量表征之前，作者先结合清华大学唐杰老师的分享，带大家看看图神经网络的发展历程，这其中也见证了向量表征的发展历程，包括从Word2vec到Deepwalk发展的缘由。

图神经网络的发展历程如下图所示：



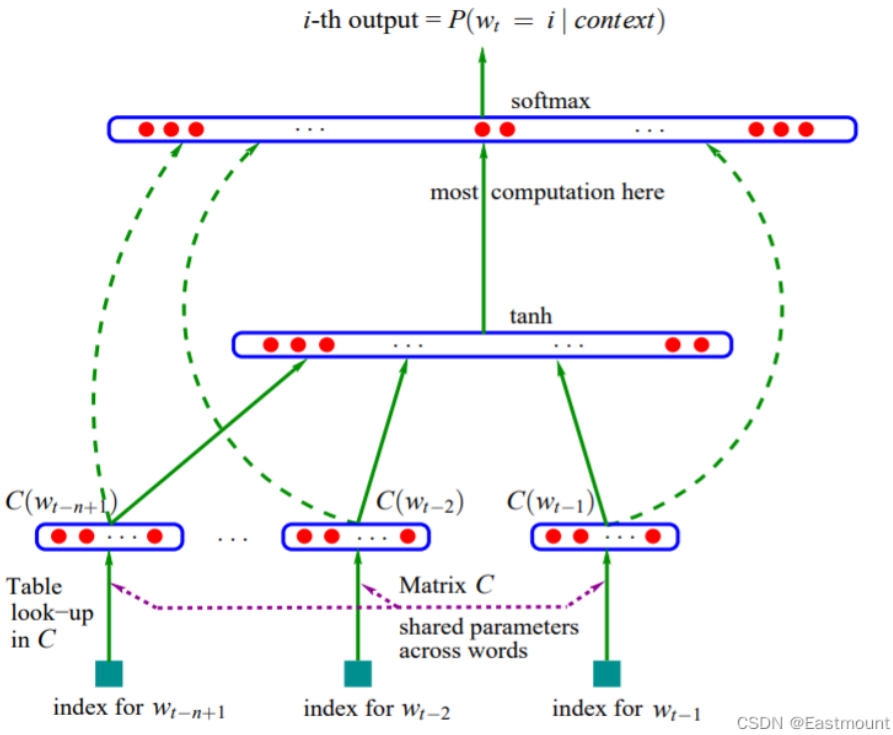
(1) Hinton早期（1986年）

图神经网络最早也不是这样的，从最早期 **Hinton** 做了相关的思路，并给出了很多的ideas，他说“一个样本可以分类成不同的representation，换句话，一个样本我们不应该去关注它的分类结果是什么，而更应该关注它的representation，并且它有很多不同的representation，每个表达的意思可能不同”，`distributed representation` 后接着产生了很多相关的研究。

(2) 扩展（Bengio到Word2Vec）

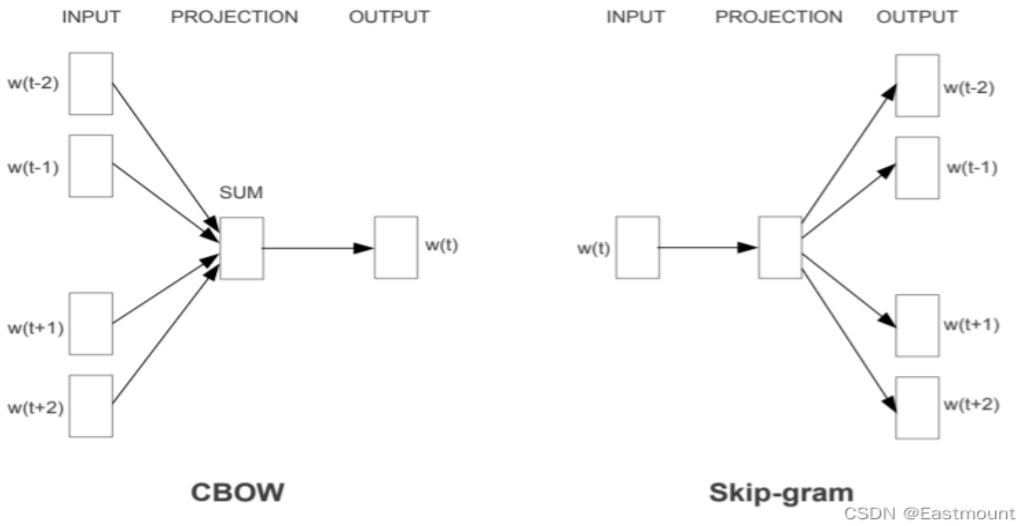
Andrew Ng 将它扩展到网络结构上（结构化数据），另一个图灵奖获得者Yoshua Bengio将它拓展到了自然语言处理上，即NLP领域如何做distributed representation，起初你可能是对一个样本representation，但对自然语言处理来讲，它是sequence，需要表示sequence，并且单词之间的依赖关系如何表示，因此2003年Bengio提出了 `Nerual Probabilistic Language Model` ，这也是他获得图灵奖的一个重要工作。其思路是：每个单词都有一个或多个表示，我就把sequence两个单词之间的关联关系也考虑进去。

- Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. Journal of Machine Learning Research (JMLR), 3:1137–1155, 2003.
- 原文地址：https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf



但是，当时做出来后由于其计算复杂度比较高，很多人无法fellow。直到谷歌2013年提出 `Word2Vec` ，基本上做出来一个场景化算法，之后就爆发了，包括将其扩展到paragraph、文档（Doc2Vec）。补充一句，Word2Vec是非常经典的工作或应用，包括我们安全领域也有相关扩展，比如二进制、审计日志、恶意代码分析的Asm2Vec、Log2Vec、Token2Vec等等。

- Efficient Estimation of Word Representations in Vector Space
- 原文地址：https://arxiv.org/abs/1301.3781v3



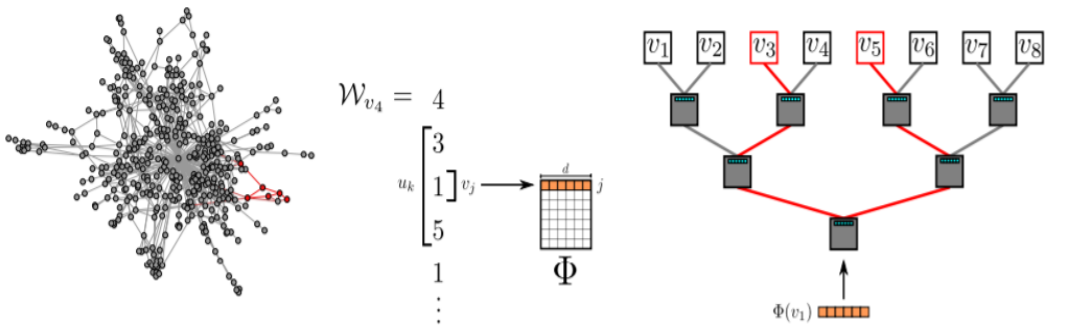
(3) 网络化数据时期（Deepwalk）

此后，有人将其扩展到网络化的数据上，2014年Bryan做了 `Deepwalk` 工作。其原理非常建立，即：原来大家都在自然语言处理或抽象的机器学习样本空间上做，那能不能针对网络化的数据，将网络化数据转换成一个类似于自然语言处理的sequence，因为网络非常复杂，网络也能表示成一个邻接矩阵，但严格意义上没有上下左右概念，只有我们俩的距离是多少，而且周围的点可多可少。如果这时候在网络上直接做很难，那怎么办呢？

通过 **随机游走** 从一个节点随机到另一个节点，此时就变成了了一个序列Sequence，并且和NLP问题很像，接下来就能处理了。

- 原文地址：https://dl.acm.org/doi/10.1145/2623330.2623732

随后又有了LINE（2015）、Node2Vec（2016）、NetMF（2018）、NetSMF（2019）等工作，它们扩展到社交网络领域。唐老师们的工作也给了证明，这些网络本质上是一个Model。



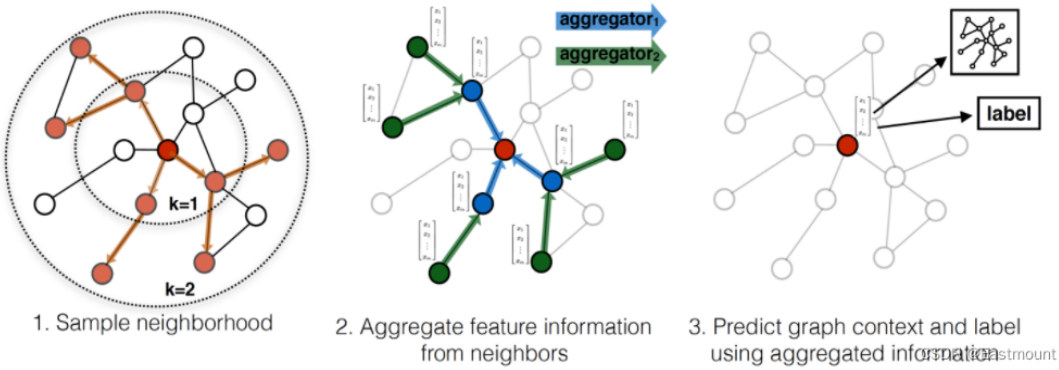
(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

Figure 3: Overview of DeepWalk. We slide a window of length $2w+1$ over the random walk $\mathcal{W}_{v_4}$, mapping the central vertex v_1 to its representation $\Phi(v_1)$. Hierarchical Softmax factors out $\Pr(v_3 \mid \Phi(v_1))$ and $\Pr(v_5 \mid \Phi(v_1))$ over sequences of probability distributions corresponding to the paths starting at the root and ending at v_3 and v_5. The representation Φ is updated to maximize the probability of v_1 co-occurring with its context $\{v_3, v_5\}$.

CSDN @Eastmount

(4) 图卷积神经网络（GCN）时期

2005年，Marco Gori 实现了 `Graph Neural Networks`。2014年，Yann Lecun 提出了图卷积神经网络 `Graph Convolutional Networks`。2017年，Max Welling将图卷积神经网络和图数据结合在一起，完成了 `GCN for semi-supervised classification`，这篇文章引起了很大关注。还有很多不做卷积工作，因此有很多Graph Neural Networks和Neural Message Passing（一个节点的分布传播过去）的工作。Jure针对节点和Transductive Learning又完成了 `Node2vec` 和 `grahpSAGE` 两个经典工作。唐老师他们最近也做了一些工作，包括 `Graph Attention Network`。

> GraphSAGE 是 2017 年提出的一种图神经网络算法，解决了 GCN 网络的局限性: GCN 训练时需要用到整个图的邻接矩阵，依赖于具体的图结构，一般只能用在直推式学习 Transductive Learning。GraphSAGE 使用多层聚合函数，每一层聚合函数会将节点及其邻居的信息聚合在一起得到下一层的特征向量，GraphSAGE 采用了节点的邻域信息，不依赖于全局的图结构。
>
> - Hamilton, Will, Zhitao Ying, and Jure Leskovec. “Inductive representation learning on large graphs.” Advances in neural information processing systems. 2017.
> - 原文地址：https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf
>
> 
>
> 1. Sample neighborhood　2. Aggregate feature information from neighbors　3. Predict graph context and label using aggregated information

Data Mining over Networks

- DM tasks in networks:
  – Modeling individual behavior
  – Modeling group behavioral patterns
  – Reveal anomaly patterns
  – Deal with big scale

第一部分花费大量时间介绍了研究背景，接下来我们正式介绍这六个工作。

二. Word2vec ： NLP经典工作（谷歌）

原文标题：Efficient Estimation of Word Representations in Vector Space
原文作者：Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean

原文链接：https://arxiv.org/pdf/1301.3781.pdf
发表会议：2013 ICLR (Workshop Poster)
参考博客：行歌. Word2Vec论文学习笔记. https://zhuanlan.zhihu.com/p/540680257

> Word2vec是一个用于生成 词向量 (word vectors)并预测相似词汇的高效预测框架，Word2vec是Google公司在2013年开发。

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

1.摘要

本文提出了两种新的“神经网络语言”模型框架，用于计算大规模数据集中单词的连续向量表示。这些表示的质量是在单词相似度任务中测量的，并将结果与以前基于不同类型的神经网络的最佳性能进行对比。

我们观察到，本文所提出的模型拥有更低的计算成本，并大幅提高了准确性。它能从16亿个单词的数据集中学习到高质量的词向量（word vectors），并且只需不到一天的时间。此外，该研究表明，这些向量在评估语法和语义特征词相似性时具有最先进的性能。

2.引言和贡献

先前的自然语言处理系统将单词视为原子单位，单词之间没有相似性的概念。因此通常采用索引的方式来与词汇表建立联系，但这种手段所能处理的数据量远远跟不上复杂任务的大规模数据。如N-gram模型。

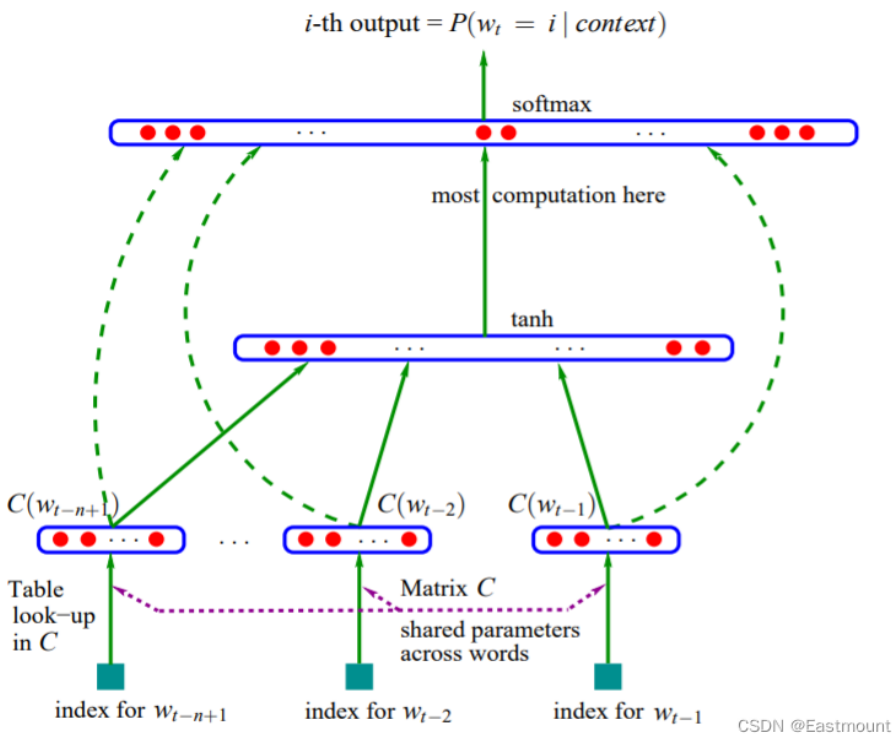
近年来，随着机器学习技术的进步，在更大的数据集上训练更复杂的模型已经成为可能，而且它们通

常优于简单的模型。可能最成功的概念是“distributed representations of words”（单词的分布式表示）。例如，基于神经网络的语言模型明显优于N-gram模型。

> The main goal of this paper is to introduce techniques that can be used for learning high-quality word
> vectors from huge data sets with billions of words, and with millions of words in the vocabulary.

基于此，本文提出了Word2Vec，旨在从大规模词向量中高效学习词向量，并预测与输入词汇关联度大的其他词汇。 在本文中，我们试图通过开发新的模型结构来保持单词之间的线性规律，以及语法和语义的规律，从而来提高这些向量操作的准确性。此外，我们还讨论了训练时间和准确性如何依赖于单词向量的维数和训练数据的数量。

- 例如，向量（King）- 向量（Man）+ 向量（Woman）能推断出一个相近的单词 （Queen）的向量表示。

> 当前，将单词表示为连续向量的诸多模型中，比较受欢迎的是NNLM(Neural Network Language Model)，由Bengio提出，利用线性投影层（linear projection layer）和非线性隐藏层的前馈神经网络，对词向量表示和统计语言模型进行联合学习。
>
> 
>
>
> 其复杂度计算如下，对应输入层、隐藏层和输出层。其中，N-输入单词数量，D-词向量维度，H-隐藏层维度，V-词汇表维度。

$$Q = N \times D + N \times D \times H + H \times V,$$

CSDN @Eastmount

推荐我2016年在CSDN的博客：word2vec词向量训练及中文文本相似度计算

3.系统框架&本文方法

本文提出了两种模型架构，如下图所示。由图可知，本文的模型并没有隐藏层，直接由输入层做一次映射，就进行分类。

- CBOW架构根据上下文预测当前的单词，而Skip-gram根据当前单词预测周围的单词

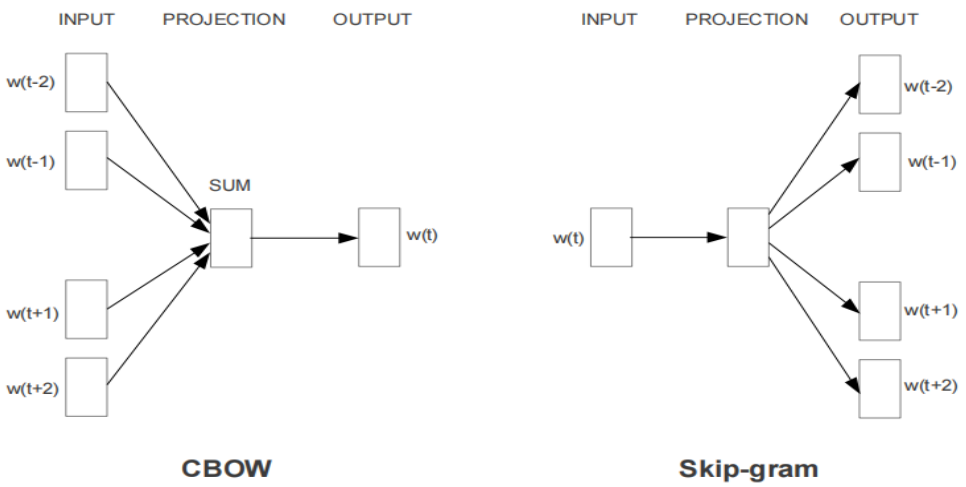


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

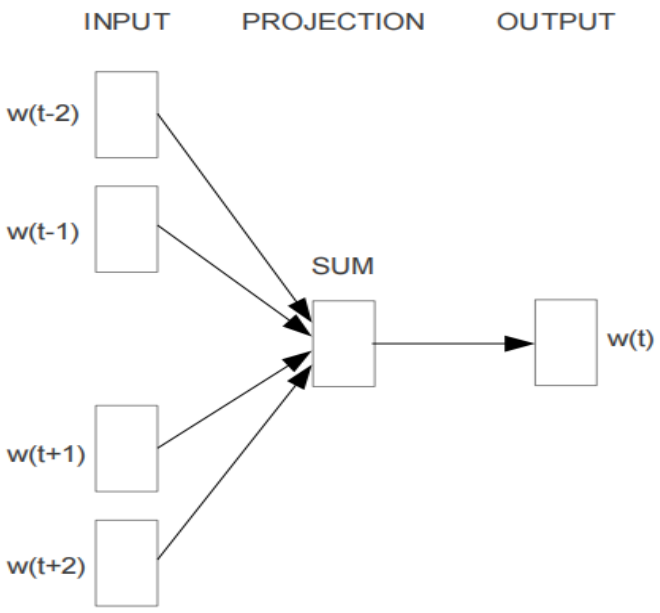
CSDN @Eastmount

(1) 连续词袋模型（CBOW，continuous bag-of-words model）

根据源词上下文词汇来预测目标词汇，可以理解为上下文决定当前词出现的概率。

在CBOW模型中，上下文所有的词对当前词出现概率的影响的权重是一样的，因此叫CBOW词袋模型。如在袋子中取词，取出数量足够的词就可以了，至于取出的先后顺序是无关紧要的，单词在时序中的顺序不影响投影（在输入层到投影层之间，投影层直接对上下文的词向量求平均，这里已经抛去词序信息）。

CBOW模型结构类似于前馈NNLM，去除了非线性隐藏层，并且投影层被所有单词共享（而不再仅仅共享投影矩阵），且输入层和投影层之间的权重矩阵对于所有单词位置都是共享的。因此，所有的单词都被投影到相同的位置。



CBOW

输入层初始化的时候直接为每个词随机生成一个n维的向量，并且把这个n维向量作为模型参数学习，最终得到该词向量，生成词向量的过程是一个参数更新的过程。

- 输入：指向单词的上下文词汇
- 输出：预测该单词出现的概率

模型复杂度如下：

$$Q = N \times D + D \times log_2(V).$$

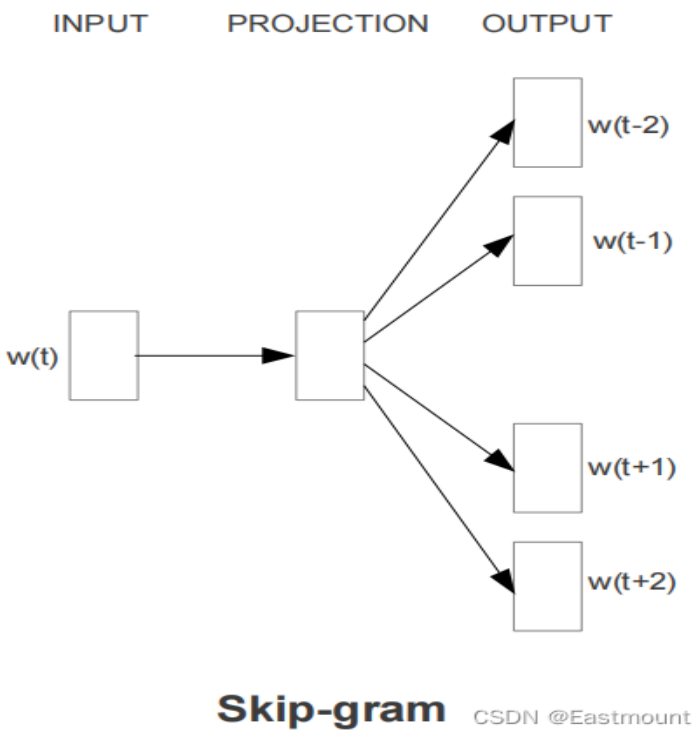
(2) Skip-Gram模型

根据当前单词预测周围的单词。Skip-gram模型类似于CBOW，但它不是基于上下文预测当前单词，而是试图基于同一句子中的另一个单词得到该单词的最大限度分类。

更准确地说，我们将每个当前词作为一个输入，输入到一个带连续投影层的对数线性分类器中，预测当前词前后一定范围内的词。该方法增加范围可以提高词向量的质量，但也增加了计算复杂度。由于距离较远的单词与当前单词之间的联系通常比距离较近的单词更小，因此我们通过在训练示例中对这些单词进行较少的抽样，从而对距离较远的单词给予更少的权重。

- Skip-gram表示“跳过某些符号”。语料的扩展能够提高训练的准确度，获得的词向量更能反映真实

的文本含义，但计算复杂度增加。



模型复杂度如下：

$$Q = C \times (D + D \times log_2(V)),$$

CSDN @Eastmount

优化策略：

- Hierarchical Softmax：Huffman树将较短的二进制代码分配给频繁出现的单词，减少需要评估的输出单元的数量
- 负采样：每次让一个训练样本仅仅更新一小部分的权重

4.对比实验

实验发现：在大量数据上训练高维词向量时，所得到的向量可以用来回答单词之间非常微妙的语义关系，例如一个城市和它所属的国家，例如<法国, 巴黎>，<德国, 柏林>。具有这种语义关系的词向量可以用于改进许多现有的自然语言处理应用，例如机器翻译、信息检索和问答系统，并且可能会使其他尚未出现的未来应用成为可能。

Table 3: *Comparison of architectures using models trained on the same data, with 640-dimensional word vectors. The accuracies are reported on our Semantic-Syntactic Word Relationship test set, and on the syntactic relationship test set of [20]*

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

Table 4: *Comparison of publicly available word vectors on the Semantic-Syntactic Word Relationship test set, and word vectors from our models. Full vocabularies are used.*

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
Our NNLM	20	6B	12.9	26.4	20.3
Our NNLM	50	6B	27.9	55.8	43.2
Our NNLM	100	6B	34.2	**64.5**	50.8
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

5.个人感受

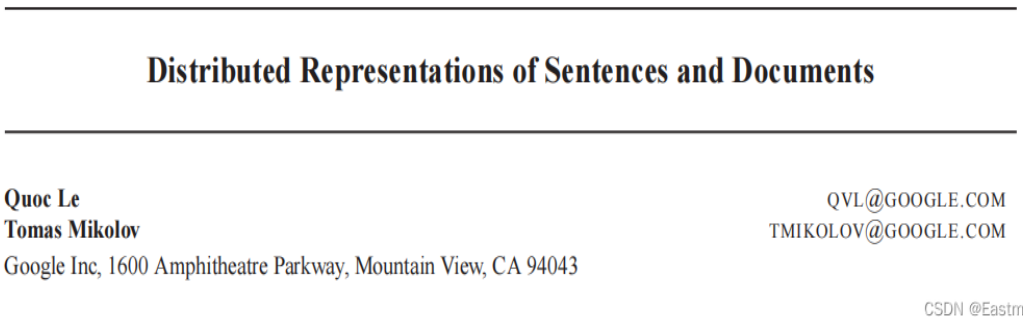
总结：这篇论文研究了在一组在句法和语义语言任务上由各种模型训练出的词向量表示的质量。我们观察到，与流行的神经网络模型（包括前馈神经网路和循环神经网络）相比，使用非常简单的模型结构训练高质量的词向量是可能的。

- Word2Vec有效解决了先前NNLM计算复杂度太高的问题，另一个很重要的意义在于是无监督方法，不需要花额外的功夫去构建数据集来学习模型，只需要给入一个非常大的文本数据集，就可以得到非常好的效果。Word2Vec的提出，有效推动了工业界和学术界的NLP发展。

三.Doc2vec

原文标题：Distributed Representations of Sentences and Documents
原文作者：Quoc V. Le, Tomás Mikolov
原文链接：http://proceedings.mlr.press/v32/le14.pdf
发表会议：2014 ICML （CCF-A）

> 在Word2Vec方法的基础上，谷歌两位大佬Quoc Le和Tomas Mikolov又给出了Doc2Vec的训练方法，也被称为Paragraph Vector，其目标是将文档向量化。



Distributed Representations of Sentences and Documents

Quoc Le QVL@GOOGLE.COM
Tomas Mikolov TMIKOLOV@GOOGLE.COM
Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

1.摘要

许多机器学习算法要求将输入表示为固定长度的特征向量。当涉及文本时，最常见的一种固定长度特征是词袋（bag-of-words）。尽管词袋模型很受欢迎，但它有两个主要弱点：它们失去了单词的顺序，并且忽略了单词的语义。例如，“powerful”, “strong” 和 “Pairs”等距离相同。

在本文中，我们提出了段落向量 `Paragraph Vector` (Doc2vec)，一种无监督算法，它可以从可变长度的文本片段中学习固定长度的特征表示，比如句子、段落和文档。

该算法通过一个密集向量来表示每个文档，该向量被训练来预测文档中的单词。它的构造使我们的算法有可能克服词袋模型的弱点。实验结果表明，我们的技术优于词袋模型和其他文本表示技术。最后，我们在几个文本分类和情感分析任务上取得了最先进的结果。

2.引言和贡献

文本分类和聚类在许多应用中发挥着重要的作用，如文档检索、网络搜索、垃圾邮件过滤。这些应用程序的核心是机器学习算法，如逻辑回归或Kmeans。这些算法通常要求将文本输入表示为一个固定长度的向量，如文本中最常见的固定长度向量表示方法：

- bag-of-words
- bag-of-n-grams

然而，词袋模型存在很多缺点：

- 词序丢失：不同的句子可以有完全相同的表示，只要使用相同的单词
- bag-of-n-grams存在数据稀疏和高维度的问题
- 忽略单词的语义信息

本文提出了段落向量（Doc2vec），这是一种无监督框架，旨在从文本片段中学习连续分布的向量表示。该方法可以应用于可变长度的文本片段，从短语到句子，再到大型文档，均可以使用Doc2vec进行向量表征。

在本文模型中，将段落中要预测的单词用向量表示来训练是很有用的。更准确地说，我们将段落向量与一个段落中的几个单词向量连接起来，并在给定的上下文中预测后续的单词。词向量和段落向量都是通过随机梯度下降和反向传播进行训练的。虽然段落向量在段落中是唯一的，但单词向量是共享的。预测时，通过固定词向量并训练新的段落向量直到收敛来推导段落向量。

Doc2vec优点如下：

- 段落向量能够构造可变长度的输入序列的表示。与以前的一些方法不同，它是通用的，适用于任何长度的文本，包括句子、段落和文档。
- 段落向量不需要对单词加权函数进行特定任务的调整，也不依赖于解析树。
- 本文在几个benchmark数据集上进行实验，证明了段落向量的优势。例如，在情感分析任务中，我们获得了最好的效果，比现有方法更好，其错误率相对提高了16%以上。在文本分类任务中，我们的方法令人惊讶地击败了词袋模型，且提高了约30%。

3.系统框架&本文方法

本文框架的灵感来源于先前的Word2vec工作。Doc2vec包括两种算法：

- 分布记忆的段落向量：PV-DM（the Distributed Memory Model of Paragraph Vector）
- 分布词袋的段落向量：PV-DBOW（the Distributed Bag of Words version of Paragraph Vector）

(1) Paragraph Vector: A distributed memory model

首先介绍单词分布式向量表示的概念。下图是著名的词向量学习的框架。其任务是预测一个上下文中给定的另一个单词。

由图可知，每个Word都被映射成一个唯一的vector编码，并组成矩阵W。其中，每列表示一个Word，对应于单词序列 {w1, w2, …, wT}。列根据该单词在词汇表中的位置进行索引，向量的连接（concatenate）或求和（sum）将被用来预测句子中下一个单词的特征。

- 例如，用三个单词（the、cat、sat）来预测第四个单词（on）。输入单词被映射到矩阵W列中，以预测输出单词。

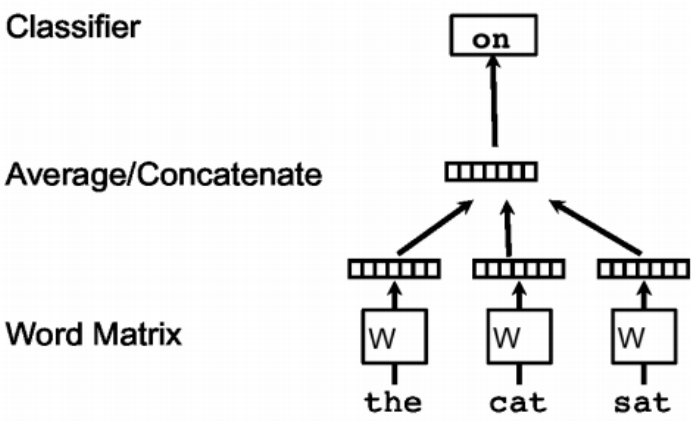


Figure 1. A framework for learning word vectors. Context of three words ("the," "cat," and "sat") is used to predict the fourth word ("on"). The input words are mapped to columns of the matrix W to predict the output word.

词向量模型的目标是最大化平均概率：

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},\ldots,w_{t+k})$$

预测任务通过多分类完成（如softmax），计算如下，其中 yi 表示第 i 个输出的单词未归一化的概率值。

$$p(w_t|w_{t-k},\ldots,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

本文使用和Word2vec相同的hierarical softmax优化策略，从而加快模型的训练速度。

(2) Paragraph Vector: A distributed memory model（分布记忆的段落向量）

段落向量是受词向量的启发而提出。词向量被要求用来预测句子中的下一个单词。尽管词向量是随机初始化的，但它们可以捕获语义信息来作为预测任务的间接结果。我们将以类似的方式在段落向量中使用这个想法。段落向量也被要求用来预测句子中的下一个单词，并且给定从段落中抽样的多个上下文。

本文提出PV-DM和PV-DBOW两种框架，其中分布记忆的段落向量（Distributed Memory Model of Paragraph Vectors,，PV-DM）描述如下。**PV-DM类似于Word2vec中的CBOW模型（连续词袋模型）**。其框架如下图所示，整个框架类似于图1，唯一的区别是：

- 增加了段落标记（paragraph token），通过矩阵D映射到一个向量中

在该模型中，矩阵W为词向量矩阵，矩阵D为段落向量矩阵。向量D与另外三个单词上下文的连接（concatenate）或平均（average）结果被用于预测第四个单词。该段落向量表示了当前上下文中缺失的信息，同时也充当了描述该段落主题的一份记忆。

- 每一个段落被映射为矩阵D中的一个唯一的向量
- 每个单词同样被映射为矩阵W中的一个唯一向量

> Paragraph vector在框架图中扮演一个记忆的角色。在词袋模型中，每次训练只会截取段落的一小部分进行训练，从而忽略本次训练之外的单词，这样仅仅训练出来每个词的向量表示，段落是每个词的向量累加在一起的表征。因此，段落向量可以在一定程度上弥补词袋模型的缺陷。

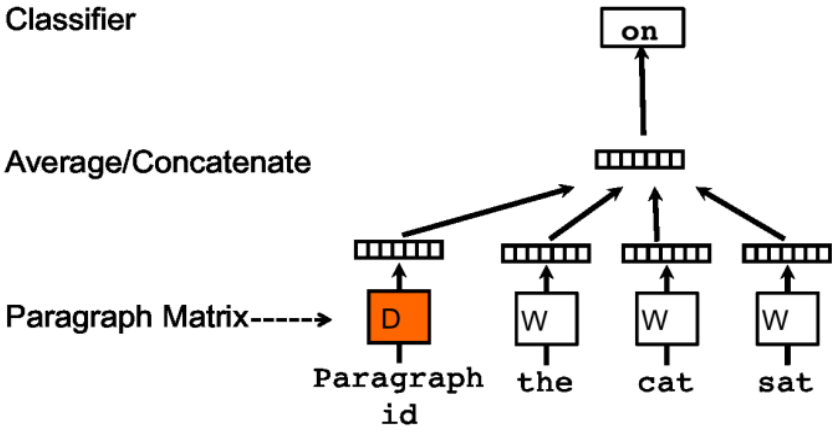


Figure 2. A framework for learning paragraph vector. This framework is similar to the framework presented in Figure 1; the only change is the additional paragraph token that is mapped to a vector via matrix D. In this model, the concatenation or average of this vector with a context of three words is used to predict the fourth word. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

此外，PV-DM模型中的上下文（context）是固定长度的，并从段落上的滑动窗口中采样得到（类似于Word2vec）。段落向量只在同一个paragraph中共享（不在各段落间共享），词向量在paragraph之间共享。换句话说，“powerful”向量对于所有段落都是相同的。

段落向量和词向量都使用随机梯度下降（gradient descent）进行训练，梯度由反向传播（backpropagation）获取。在随机梯度下降的每一步，都可以从随机段落中采样一个固定长度的上下文，从图2网络中计算误差梯度，并使用梯度来更新我们模型中的参数。

在预测期间，模型需要执行一个推理步骤来计算一个新段落的段落向量。这也是由梯度下降得到的。在这个过程中，模型的其它部分，词向量W和softmax权重都是固定的。

> 假设语料库中存在N个段落、M个单词，想要学习段落向量使得每个段落向量被映射到p维，每个词被映射到q维，然后模型总共就有N x p +M x q个参数（不包括softmax的参数）。即使当N很大时，模型的参数也可能会很大，但在训练期间的更新通常是稀疏的，因此模型有效。训练完之后，段落向量可用于表示段落的特征，我们可以将这些特征直接用在传统的机器学习模型中，如逻辑回归、支持向量机或K-means。

总之，整个算法包括以下阶段：

- 无监督训练得到词向量W（word vectors）
- 推理阶段得到段落向量D（paragraph vectors）

- 构造标准的机器学习分类器对特定标签进行预测

段落向量的优点：

- 它们是从未标记的数据中学习出来的，因此可以很好地用于没有足够标记数据的任务。
- 段落向量解决了词袋模型的弱点。它们继承了词向量的一个重要属性——语义。
- 段落向量考虑了单词的顺序，至少在小规模上下文中，能像n-gram模型一样实现任务，保留大量信息（如词序）。Doc2vec比bag-of-n-grams模型更好，因为后者会创建非常高维的特征表示，其泛化能力很差。
- 在训练过程中，段落向量能够记忆整个句子的意义，词向量则能够基于全局部分学习到其具体的含义。

(3) Paragraph Vector without word ordering: Distributed bag of words（分布词袋的段落向量）

上述方法考虑了段落向量与单词向量的连接，以预测文本窗口中的下一个单词。另一种方法是PV-DBOW（分布词袋的段落向量）。PV-DBOW忽略输入中的上下文，强制模型从输出段落中随机抽样来预测单词。

- 和PV-DM不同，PV-DBOW使用段落向量来预测单词

> 通俗而言，PV-DBOW会在随机梯度下降的每次迭代中，采样出一个文本窗口，然后从文本窗口中采样一个随机单词，并形成一个给定段落向量的分类任务。

PV-DBOW类似于Word2vec中的Skip-gram模型，其结构图如下所示，段落向量在一个小窗口中被训练来预测单词。

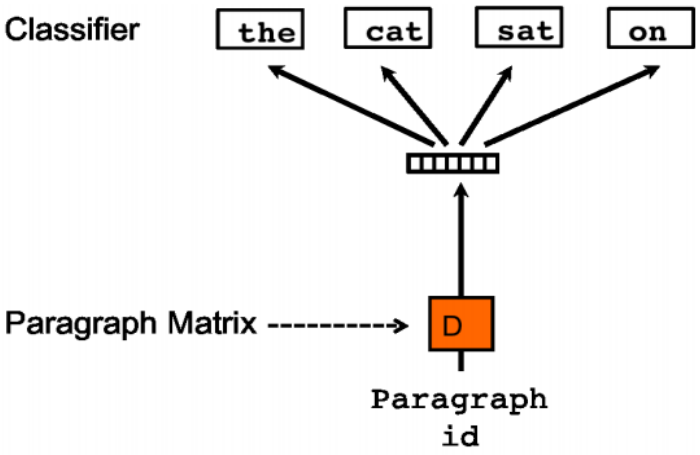


Figure 3. Distributed Bag of Words version of paragraph vectors. In this version, the paragraph vector is trained to predict the words in a small window.

除了概念简单之外，这个模型只需要存储更少的数据。我们只需要存储softmax权值，而不像之前模型那样存储softmax的权值和单词向量。

4.对比实验

在本文实验中，每个段落向量都是PV-DM和PV-DBOW两个向量的组合。实验结果发现，PV-DM在大多数任务上都能取得较好的表现，但如果再与PV-DBOW结合，能在更多的任务中取得始终如一的良好表现，因此强烈推荐使用。

本文在两个需要固定长度的段落向量表示的文本理解问题上进行了段落向量的基准测试，即情感分析和信息检索（推理任务）。数据集：

- 情感分析：Stanford sentiment treebank dataset (Socher et al., 2013b)
- 情感分析：IMDB dataset (Maas et al., 2011)
- 信息检索：information retrieval task
- 下载地址：http://nlp.Stanford.edu/sentiment/

实验参数设置：

- window size设置为8
- vector size设置为400
- word vector聚合使用的是连接

实验结果如下表所示，本文模型能取得较好的效果。

Table 1. The performance of our method compared to other approaches on the Stanford Sentiment Treebank dataset. The error rates of other methods are reported in (Socher et al., 2013b).

Model	Error rate (Positive/ Negative)	Error rate (Fine-grained)
Naïve Bayes (Socher et al., 2013b)	18.2 %	59.0%
SVMs (Socher et al., 2013b)	20.6%	59.3%
Bigram Naïve Bayes (Socher et al., 2013b)	16.9%	58.1%
Word Vector Averaging (Socher et al., 2013b)	19.9%	67.3%
Recursive Neural Network (Socher et al., 2013b)	17.6%	56.8%
Matrix Vector-RNN (Socher et al., 2013b)	17.1%	55.6%
Recursive Neural Tensor Network (Socher et al., 2013b)	14.6%	54.3%
Paragraph Vector	**12.2%**	**51.3%**

Table 2. The performance of Paragraph Vector compared to other approaches on the IMDB dataset. The error rates of other methods are reported in (Wang & Manning, 2012).

Model	Error rate
BoW (bnc) (Maas et al., 2011)	12.20 %
BoW (bΔt'c) (Maas et al., 2011)	11.77%
LDA (Maas et al., 2011)	32.58%
Full+BoW (Maas et al., 2011)	11.67%
Full+Unlabeled+BoW (Maas et al., 2011)	11.11%
WRRBM (Dahl et al., 2012)	12.58%
WRRBM + BoW (bnc) (Dahl et al., 2012)	10.77%
MNB-uni (Wang & Manning, 2012)	16.45%
MNB-bi (Wang & Manning, 2012)	13.41%
SVM-uni (Wang & Manning, 2012)	13.05%
SVM-bi (Wang & Manning, 2012)	10.84%
NBSVM-uni (Wang & Manning, 2012)	11.71%
NBSVM-bi (Wang & Manning, 2012)	8.78%
Paragraph Vector	**7.42%**

信息检索类似于推理任务，实现网页内容和查询的匹配（比较哪两段内容更接近）。实验结果如下：

- **Paragraph 1:** calls from (000) 000 - 0000 . 3913 calls reported from this number . according to 4 reports the identity of this caller is american airlines .

- **Paragraph 2:** do you want to find out who called you from +1 000 - 000 - 0000 , +1 0000000000 or (000) 000 - 0000 ? see reports and share information you have about this caller

- **Paragraph 3:** allina health clinic patients for your convenience , you can pay your allina health clinic bill online . pay your clinic bill now , question and answers...

Table 3. The performance of Paragraph Vector and bag-of-words models on the information retrieval task. "Weighted Bag-of-bigrams" is the method where we learn a linear matrix W on TF-IDF bigram features that maximizes the distance between the first and the third paragraph and minimizes the distance between the first and the second paragraph.

Model	Error rate
Vector Averaging	10.25%
Bag-of-words	8.10 %
Bag-of-bigrams	7.28 %
Weighted Bag-of-bigrams	5.67%
Paragraph Vector	**3.82%**

5.个人感受

本文描述了段落向量Doc2vec，一种无监督学习算法，它可以从可变长度的文本片段中学习固定长度的特征表示，比如句子、段落和文档。向量表示可以被学习来预测段落中上下文周围的单词。本文分别在Stanford和IMDB情感分析数据集上测试，有效证明了方法的性能，以及段落向量能捕获语义信息的优点，且解决词袋模型的许多弱点。

虽然这项工作的重点是文本表示，但本文的方法可以应用于多种领域，比如学习顺序数据的表示。未来，在非文本领域中，我们期望段落向量是词袋和n-grams模型的一个强有力的替代模型。

Doc2vec和Word2vec都是谷歌提出的两个经典工作，Doc2vce是基于Word2vec改进而来，并且继承了后者的许多优点，能在大规模文本数据上捕获文档中的语义和句法信息，加速模型运算。Doc2vec的目标是文档向量化，通过添加段落标记（矩阵D）实现

此外，尽管Doc2vec和Word2vec有效促进了整个NLP领域的发展，但它们也存在缺点。正如机器之心（Hongfeng Ai）总结一样：

> Doc2vec缺乏统计学的运用，如果数据规模较小，一定程度上会影响段落向量质量的好坏。未来，Doc2vecc可能会融入统计学的知识，从而缓解由于数据不足带来的问题。同时，模型计算速度也需要优化。比如2016年Fackbook团队提出了fastText，该模型不像非监督方法如word2vec训练的词向量，fastText得到的词特征能够平均在一起形成好的文本表示，而且模型运算速度很快，使用一个标准多核CPU，在十亿词上只需要不到10分钟便能训练好。而且不到一分钟就可以分类好含有312K个类别的五十万条句子。

四.DeepWalk：网络化数据经典工作（KDD2014）

(待续见后)

五.Graph2vec

(待续见后)

六.Asm2vec：安全领域经典工作（S&P2019）

(待续见后)

七.Log2vec：安全领域经典工作（CCS2019）

(待续见后)

八.总结

写到这里，这篇文章就分享结束了，再次感谢论文作者及引文的老师们。由于是在线论文读书笔记，仅代表个人观点，写得不好的地方，还请各位老师和博友批评指正。下面简单总结下：

这篇文章我从向量表征角度介绍了6个经典的工作，首先是谷歌的Word2vec和Doc2vec，它们开启了NLP的飞跃发展；其次是DeepWalk和Graph2vec，通过随机游走的方式对网络化数据做一个表示学

习，它们促进了图神经网络的发展；最后是Asm2vec和Log2vec，它们是安全领域二进制和日志向量表征的两个经典工作，见解了前面论文的思想，并优化且取得了较好的效果，分别发表在S&P19和CCS19。挺有趣的六个工作，希望您喜欢。其实啊，写博客其实可以从很多个视角写，科研也是，人生更是。

本文主要分享Word2vec和Doc2vec两个经典工作，读者可以思考下面三个问题：

- Word2vec和Doc2vec在NLP领域取得了极大的飞跃。那么，其它计算机领域又将如何作向量表征呢？
- 网络化数据或图数据又将如何实现向量表征呢？又有哪些代表性工作呢？
- 某些具有独特背景知识的领域又将如何借鉴其思想，比如安全领域的二进制、医药生物领域基因等。

代码在gensim中直接可以调用，大家试试，之前我的博客也介绍得很多。

```
model = gensim.models.Word2Vec(size=200, window=8, min_count=10, iter=10, workers=c
ores)
model = gensim.models.doc2vec.Doc2Vec(vector_size = 50, min_count = 2, epochs=40)
```

最后祝大家在读博和科研的路上不断前行。项目学习再忙，也要花点时间读论文和思考，加油！这篇文章就写到这里，希望对您有所帮助。由于作者英语实在太差，论文的水平也很低，写得不好的地方还请海涵和批评。同时，也欢迎大家讨论，继续努力！感恩遇见，且看且珍惜。

(By:Eastmount 2022-09-19 周一夜于武汉 http://blog.csdn.net/eastmount/)

参考文献如下，感谢这些大佬！也推荐大家阅读原文。

- [1] 唐杰老网站：http://keg.cs.tsinghua.edu.cn/jietang
- [2] **Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. ICLR, 2013.**
- [3] 行歌：Word2Vec论文学习笔记. https://zhuanlan.zhihu.com/p/540680257
- [4] **Quoc V. Le, Tomás Mikolov. Distributed Representations of Sentences and Documents. ICML, 2014: 1188-1196.**
- [5] 机器之心. Doc2vec. https://www.jiqizhixin.com/graph/technologies/5d96aebf-926b-4766-89e7-40e68d662e35
- [6] Thinkgamer. 论文｜Doc2vec的算法原理、代码实现及应用启发. https://zhuanlan.zhihu.com/p/336921474
- [7] Eastmount. word2vec词向量训练及中文文本相似度计算. https://blog.csdn.net/Eastmount/article/details/50637476

- [8] -派神-. Doc2Vec的简介及应用(gensim). https://blog.csdn.net/weixin_42608414/article/details/88378984
- [9] DeepWalk和Grahp2vec
- [10] Asm2vec和Log2vec