

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

CSDN

博客 (//blog.csdn.net/?ref=toolbar)学院 (//edu.csdn.net/?ref=toolbar)

下载 (//download.csdn.net/?ref=toolbar)GitChat (//gitbook.cn/?ref=csdn)

更多 ▾

0

0

24083

摘要：本文主要描述了一种文章向量（doc2vec）表示及其训练的相关内容，并列出了相关例子。两位大牛Quoc Le 和 Tomas Mikolov（搞出Word2vec的家伙）在2014年的《Distributed Representations of Sentences and Documents (http://cs.stanford.edu/~quocle/paragraph\_vector.pdf)》所提出文章向量（Documents vector），或者称句向量（Sentences vector），当然在文章中，统一称这种向量为Paragraph Vector，本文也将已doc2vec称呼之。文章中讲述了如何将文章转换成向量表示的算法。

1、Word2vec的基本原理

先简述一下Word2vec (https://code.google.com/p/word2vec/)相关原理，因为本文要讲述的doc2vec是基于Word2vec思想的算法。w2v的数学知识还比较丰富，网络上相关资料也很多。如果要系统的讲述，我可能会涉及包括词向量的理解、sigmoid函数、逻辑回归、Bayes公式、Huffman编码、n-gram模型、浅层神经网络、激活函数、最大似然及其梯度推导、随机梯度下降法、词向量与模型参数的更新公式、CBOW模型和Skip-gram模型、Hierarchical Softmax算法和Negative Sampling算法。当然还会结合google发布的C源码（好像才700+行），讲述相关部分的实现细节，比如Negative Sampling算法如何随机采样、参数更新的细节、sigmoid的快速近似计算、词典的hash存储、低频与高频词的处理、窗口内的采样方式、自适应学习、参数初始化、w2v实际上含有两中方法等，用C代码仅仅700+行实现，并加入了诸多技巧，推荐初识w2v的爱好者得看一看。

Google出品的大多都是精品，w2v也不例外。Word2Vec实际上使用了两种方法，Continuous Bag of Words (CBOW) 和Skip-gram，如下图所示。在CBOW方法中，目的是将文章中某个词的上下文经过模型预测该词。而Skip-gram方法则是用给定的词来预测其周边的词。而词向量是在训练模型中所得到的一个副产品，此模型在源码中是为一个浅层的神经网络（3层）。在训练前，每一个词都会首先初始化为一个N维的向量，训练过程中，会对输入的向量进行反馈更新，在进行大量语料训练之后，便可得到每一个词相应的训练向量。而每一种模型方法都可以使用两种对应的训练方法Hierarchical Softmax算法和Negative Sampling算法，有兴趣的盆友可以自行查阅相关内容。

INPUT

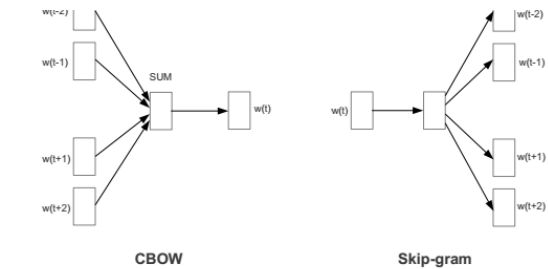
PROJECTION

OUTPUT

INPUT

PROJECTION

OUTPUT

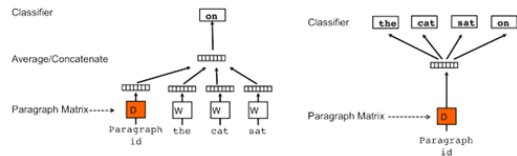


训练出的向量有一定的特性，即相近意义的词在向量空间上其距离也是相近。有一个经典例子就是  $V(\text{'king'}) - V(\text{'man'}) + V(\text{'woman'}) \approx V(\text{'queen'})$

## 2、Doc2Vec的基本原理

基于上述的Word2Vec的方法，Quoc Le 和 Tomas Mikolov又给出了Doc2Vec的训练方法。如下图所示，其原理与Word2Vec非常的相似。分为Distributed Memory (DM) 和Distributed Bag of Words (DBOW)，可以看出Distributed Memory version of Paragraph Vector (PV-DM)方法与Word2Vec的CBOW方法类似，Bag of Words version of Paragraph Vector (PV-DBOW)与Word2Vec的Skip-gram方法类似。不同的是，给文章也配置了向量，并在训练过程中更新。熟悉了w2v之后，Doc2Vec便非常好理解。具体细节可以看原文 ([http://cs.stanford.edu/~quocle/paragraph\\_vector.pdf](http://cs.stanford.edu/~quocle/paragraph_vector.pdf))

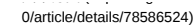
《Distributed Representations of Sentences and Documents ([http://cs.stanford.edu/~quocle/paragraph\\_vector.pdf](http://cs.stanford.edu/~quocle/paragraph_vector.pdf))》



### 3、gensim的实现

使用Doc2Vec进行分类任务，我们使用 IMDB电影评论数据集 (<http://ai.stanford.edu/~amaas/data/sentiment/>)作为分类例子，测试gensim的Doc2Vec的有效性。数据集中包含25000条正向评价，25000条负面评价以及50000条未标注评价。

```
1 #!/usr/bin/python
2 import sys
3 import numpy as np
4 import gensim
5
6 from gensim.models.doc2vec import Doc2Vec, LabeledSentence
```



## ■ 在线课程

[illegible]

Tensorflow一些常用基本概念与函数(2)  
(<http://blog.csdn.net/lenbow/article/details/52181159>)  
📖 32889

基于gensim的Doc2Vec简析 (<http://blog.csdn.net/lenbow/article/details/52120230>)

```

6 from gensim.models.doc2vec import LabeledSentence
7 from sklearn.cross_validation import train_test_split
8
9 LabeledSentence = gensim.models.doc2vec.LabeledSentence

```

23999

安装Tensorflow (Linux ubuntu) (<http://blog.csdn.net/lenbow/article/details/51203526>)

19559

```

1 ##读取并预处理数据
2 def get_dataset():
3     #读取数据
4     with open(pos_file,'r') as infile:
5         pos_reviews = infile.readlines()
6     with open(neg_file,'r') as infile:
7         neg_reviews = infile.readlines()
8     with open(unsup_file,'r') as infile:
9         unsup_reviews = infile.readlines()
10
11     #使用1表示正面情感，0为负面
12     y = np.concatenate((np.ones(len(pos_reviews)), np.zeros(len(neg_reviews))))
13     #将数据分割为训练与测试集
14     x_train, x_test, y_train, y_test = train_test_split(np.concatenate((pos_reviews, neg_reviews)), y, test_size=0.2)
15
16     #对英文做简单的数据清洗预处理，中文根据需要进行修改
17     def cleanText(corpus):
18         punctuation = """,.?!:;(){}[]""
19         corpus = [z.lower().replace("\n",'') for z in corpus]
20         corpus = [z.replace('<br />',' ') for z in corpus]
21
22         #treat punctuation as individual words
23         for c in punctuation:
24             corpus = [z.replace(c, ' %s '%c) for z in corpus]
25         corpus = [z.split() for z in corpus]
26         return corpus
27
28     x_train = cleanText(x_train)
29     x_test = cleanText(x_test)
30     unsup_reviews = cleanText(unsup_reviews)
31
32     #Gensim的Doc2Vec应用于训练要求每一篇文章/句子有一个唯一标识的label.
33     #我们使用Gensim自带的LabeledSentence方法. 标识的格式为"TRAIN_i"和"TEST_i", 其中i为序号
34     def labelizeReviews(reviews, label_type):
35         labeled = []

```

```
35     labeled = []
36     for i,v in enumerate(reviews):
37         label = '%s_%s'%(label_type,i)
38         labeled.append(LabeledSentence(v, [label]))
39     return labeled
40
41     x_train = labelizeReviews(x_train, 'TRAIN')
42
43     x_test = labelizeReviews(x_test, 'TEST')
44     unsup_reviews = labelizeReviews(unsup_reviews, 'UNSUP')
45
46     return x_train,x_test,unsup_reviews,y_train, y_test
```

```
1  ##读取向量
2  def getVecs(model, corpus, size):
3      vecs = [np.array(model.docvecs[z.tags[0]]).reshape((1, size)) for z in corpus]
4      return np.concatenate(vecs)
```

```
1  ##对数据进行训练
2  def train(x_train,x_test,unsup_reviews,size = 400,epoch_num=10):
3      #实例DM和DBOW模型
4      model_dm = gensim.models.Doc2Vec(min_count=1, window=10, size=size, sample=1e-3, negative=5, workers=3)
5      model_dbow = gensim.models.Doc2Vec(min_count=1, window=10, size=size, sample=1e-3, negative=5, dm=0, workers=
6
7      #使用所有的数据建立词典
8      model_dm.build_vocab(np.concatenate((x_train, x_test, unsup_reviews)))
9      model_dbow.build_vocab(np.concatenate((x_train, x_test, unsup_reviews)))
10
11     #进行多次重复训练，每一次都需要对训练数据重新打乱，以提高精度
12     all_train_reviews = np.concatenate((x_train, unsup_reviews))
13     for epoch in range(epoch_num):
14         perm = np.random.permutation(all_train_reviews.shape[0])
15         model_dm.train(all_train_reviews[perm])
16         model_dbow.train(all_train_reviews[perm])
17
18     #训练测试数据集
19     x_test = np.array(x_test)
20     for epoch in range(epoch_num):
21         perm = np.random.permutation(x_test.shape[0])
22         model_dm.train(x_test[perm])
23         model_dbow.train(x_test[perm])
24
25     return model_dm,model_dbow
```

```
1  ##将训练完成的数据转换为vectors
2  def get_vectors(model_dm,model_dbow):
3
4      #获取训练数据集的文档向量
```



0



```
5 train_vecs_dm = getVecs(model_dm, x_train, size)
6 train_vecs_dbow = getVecs(model_dbow, x_train, size)
7 train_vecs = np.hstack((train_vecs_dm, train_vecs_dbow))
8 #获取测试数据集的文档向量
9 test_vecs_dm = getVecs(model_dm, x_test, size)
10 test_vecs_dbow = getVecs(model_dbow, x_test, size)
11 test_vecs = np.hstack((test_vecs_dm, test_vecs_dbow))
12
13 return train_vecs, test_vecs
```

```
1 ##使用分类器对文本向量进行分类训练
2 def Classifier(train_vecs, y_train, test_vecs, y_test):
3     #使用sklearn的SGD分类器
4     from sklearn.linear_model import SGDClassifier
5
6     lr = SGDClassifier(loss='log', penalty='l1')
7     lr.fit(train_vecs, y_train)
8
9     print "Test Accuracy: %.2f" % lr.score(test_vecs, y_test)
10
11     return lr
```



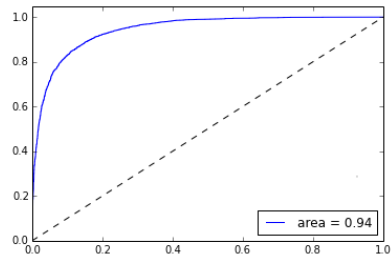
0



```
1 ##绘出ROC曲线，并计算AUC
2 def ROC_curve(lr, y_test):
3     from sklearn.metrics import roc_curve, auc
4     import matplotlib.pyplot as plt
5
6     pred_proba = lr.predict_proba(test_vecs)[:, 1]
7
8     fpr, tpr, _ = roc_curve(y_test, pred_proba)
9     roc_auc = auc(fpr, tpr)
10    plt.plot(fpr, tpr, label='area = %.2f' % roc_auc)
11    plt.plot([0, 1], [0, 1], 'k--')
12    plt.xlim([0.0, 1.0])
13    plt.ylim([0.0, 1.05])
14
15    plt.show()
```



```
1  ##运行模块
2  if __name__ == "__main__":
3      #设置向量维度和训练次数
4      size, epoch_num = 400, 10
5      #获取训练与测试数据及其类别标注
6      x_train,x_test,unsup_reviews,y_train, y_test = get_dataset()
7      #对数据进行训练，获得模型
8      model_dm,model_dbow = train(x_train,x_test,unsup_reviews,size,epoch_num)
9      #从模型中抽取文档相应的向量
10     train_vecs,test_vecs = get_vectors(model_dm,model_dbow)
11     #使用文章所转换的向量进行情感正负分类训练
12     lr=Classifier(train_vecs,y_train,test_vecs, y_test)
13     #画出ROC曲线
14     ROC_curve(lr,y_test)
```



训练结果的，test分类精度为86%，AUC面积为0.94




0



相关链接：

- [1] 安装Tensorflow ( Linux ubuntu ) <http://blog.csdn.net/lenbow/article/details/51203526>  
(<http://blog.csdn.net/lenbow/article/details/51203526>)
- [2] ubuntu下CUDA编译的GCC降级安装 <http://blog.csdn.net/lenbow/article/details/51596706>  
(<http://blog.csdn.net/lenbow/article/details/51596706>)
- [3] ubuntu手动安装最新Nvidia显卡驱动 <http://blog.csdn.net/lenbow/article/details/51683783>  
(<http://blog.csdn.net/lenbow/article/details/51683783>)

[4] Tensorflow的CUDA升级，以及相关配置 <http://blog.csdn.net/lenbow/article/details/52118116>  
(<http://blog.csdn.net/lenbow/article/details/52118116>)


 发表你的评论

([http://my.csdn.net/weixin\\_35068028](http://my.csdn.net/weixin_35068028))

 chengxu28 ([/chengxu28](#)) 2017-09-14 17:32 19楼


([/chengxu28](#))uracy: 0.50, 什么鬼!!!

回复

 qq\_33686272 ([/qq\\_33686272](#)) 2017-09-04 17:24 18楼

([/qq\\_33686272](#))如果用n篇文章做文章的聚类的话，测试集和训练集应该怎么分呢？

回复

 qq\_27075947 ([/qq\\_27075947](#)) 2017-07-30 14:54 17楼

([/qq\\_27075947](#))dataset里面的读文件，给的路径是"/aclImdb/train/urls\_pos.txt"么，还是其他的什么。我看urls\_pos.txt这个文件里都是一些url，本地不是有pos文件夹么，不从pos文件夹内导入数据么？


回复

查看 25 条热评

相关文章推荐


机器学习系列(4)\_机器学习算法一览，应用建议与解决思路 (<http://blog.csdn.net/yaoqiang2011...>)

我们先带着大家过一遍传统机器学习算法，基本思想和用途。把问题解决思路和方法应用建议提前到这里的想法也很简单，希望能提前给大家一些小建议，对于某些容易出错的地方也先给大家打个预防针，这样在理解后续相应机...

 yaoqiang2011 (<http://blog.csdn.net/yaoqiang2011>) 2016年01月06日 15:35 48057

牛顿方法、指数分布族、广义线性模型—斯坦福ML公开课笔记4 (<http://blog.csdn.net/xinzhan...>)

转载请注明：<http://blog.csdn.net/xinzhangyanxiang/article/details/9207047> 最近在看Ng的机器学习公开课，Ng的讲法循循善诱，感觉提高了不少...

 xinzhangyanxiang (<http://blog.csdn.net/xinzhangyanxiang>) 2013年06月30日 16:55 17328



霸气！重磅改革！吴恩达说：女儿识字后就教她学Python！


Python的火爆最近越来越挡不住了，连身边多年工作经验的朋友都开始学Python了！他是这么说的....

(http://www.baidu.com/cb.php?c=lgF\_pyfqHmknjnvPjc0IZ0qnfK9ujYzP1ndPWb10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1YLPjKbP1fzrjRsnADsuhR10AwY5HDdnHnYnjc3njf0lgF\_5y9YIZ0IQzq-

uZR8mLPbUB48ugfElAqspynETZ-YpAq8nWqdIAdxTvqdThP-5yF\_UvTkn0KzujYk0AFV5H00TZcqN0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPHcvnjb)


基于gensim的Doc2Vec简析 (http://blog.csdn.net/junjie20082008/article/details/53696412)

1、Word2vec的基本原理 先简述一下Word2vec相关原理，因为本文要讲述的doc2vec是基于Word2vec思想的算法。w2v的数学知识还比较丰富，网络上相关资料也很多。如果要系统的...

 junjie20082008 (http://blog.csdn.net/junjie20082008) 2016年12月16日 18:20 1300


用gensim.doc2vec 建模、利用相似度做文本分类 (http://blog.csdn.net/a602232180/article/d...

想看看doc2vec的效果怎么说，按照 基于gensim的Doc2Vec简析 上面的实验做了下，发现用随机森林做的模型，二分类的准确率50%，换sklearn的KNN，分类结果也是50%上下。看...

 a602232180 (http://blog.csdn.net/a602232180) 2017年11月20日 21:04 58

用gensim doc2vec计算文本相似度 (http://blog.csdn.net/juanjuan1314/article/details/75124...

最近开始接触gensim库，之前训练word2vec用Mikolov的c版本程序，看了很久才把程序看明白，在gensim库中，word2vec和doc2vec只需要几个接口就可以实现，实在是方便。py...

 juanjuan1314 (http://blog.csdn.net/juanjuan1314) 2017年07月14日 16:48 3208




程序员跨越式成长指南

完成第一次跨越，你会成为具有一技之长的开发者，月薪可能翻上几番；完成第二次跨越，你将成为拥有局部优势或行业优势的专业人士，获得个人内在价值的有效提升和外在收入的大幅跃迁.....

(http://www.baidu.com/cb.php?c=lgF\_pyfqHmknjzrjD0IZ0qnfK9ujYzP1f4PjnY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1dBrHTzryRYnyF-ujTkPyRL0AwY5HDdnHnYnjc3njR0lgF\_5y9YIZ0IQzqMpgwBUvqoQhP8QvGIAPCmgfEmvq\_lyd8Q1R4uWc4uHf3uAckPHRkPWN9PhcsmW9huWqdIAdxTvqdThP-5HDknWFBmhkEusKzujYk0AFV5H00TZcqN0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPjTL)


gensim doc2vec选译 (http://blog.csdn.net/u013339087/article/details/76068312)

Gensim doc2vec 文档选译为了防止自己忘记doc2vec的使用再去花大时间看英文原文，这里挑官网的重点翻译http://radimrehurek.com/gensim/models/doc...

 u013339087 (http://blog.csdn.net/u013339087) 2017年07月25日 14:33 571

gensim doc2vec + sklearn kmeans 做文本聚类 (http://blog.csdn.net/juanjuan1314/article/...

前一篇用doc2vec做文本相似度，模型可以找到输入句子最相似的句子，然而分析大量的语料时，不可能一句一句的输入，语料数据大致怎么分类也不能知晓。于是决定做文本聚类。选择kmeans作为聚类方法。...

 juanjuan1314 (http://blog.csdn.net/juanjuan1314) 2017年07月20日 10:11 879

python 环境下gensim中的word2vec的使用笔记 (http://blog.csdn.net/philosophyatmath/ar...



- 0
- ≡
- 🔖
- 💬
- 🔗
- 👍0
- ≡
- 🔖
- 💬
- 🔗

centos 7, python2.7, gensim (0.13.1)语料 : <http://211.136.8.18/files/10940000015A9F94/mattmahoney.net/dc...>

 philosophyatmath (<http://blog.csdn.net/philosophyatmath>) 2016年08月29日 16:57 16946

python | gensim训练word2vec及相关函数与功能理解 ([http://blog.csdn.net/sinat\\_26917383...](http://blog.csdn.net/sinat_26917383...)

一、gensim介绍gensim是一款强大的自然语言处理工具，里面包括N多常见模型：- 基本的语料处理工具 - LSI - LDA - HDP - DTM - DIM - ...

 sinat\_26917383 ([http://blog.csdn.net/sinat\\_26917383](http://blog.csdn.net/sinat_26917383)) 2017年04月09日 11:23 5561

Python版的Word2Vec -- gensim 学习手札 中文词语相似性度量 V1.1 (<http://blog.csdn.net/M...>

前言相关内容链接：第一节：Google Word2vec 学习手札 昨天好不容易试用了一下Google自己提供的Word2Vector的源代码，花了好长时间训练数据，结果发现似乎Python并不能...

 MebiuW (<http://blog.csdn.net/MebiuW>) 2016年08月24日 20:10 9398

word2vec\_gensim 中文处理 小试牛刀 ([http://blog.csdn.net/qq\\_27824601/article/details/52...](http://blog.csdn.net/qq_27824601/article/details/52...)

word2vec - gensim介绍gensim 是word2vec的python实现。 word2vec是google的一个开源工具，能够计算出词与词之间的距离。 word2vec即是word t...

 qq\_27824601 ([http://blog.csdn.net/qq\\_27824601](http://blog.csdn.net/qq_27824601)) 2016年08月19日 09:33 335

gensim中使用word2vec (<http://blog.csdn.net/a1368783069/article/details/52025764>)

训练语料由于自己在csdn的上传空间不够，暂时将语料放在百度云上 链接: <https://pan.baidu.com/s/1qYKRXOo> 密码: 4psr 文件名是 text8 或者在参考文章...

 a1368783069 (<http://blog.csdn.net/a1368783069>) 2016年07月25日 17:34 6135

Gensim Word2vec 使用教程 ([http://blog.csdn.net/Star\\_Bob/article/details/47808499](http://blog.csdn.net/Star_Bob/article/details/47808499))

本文主要基于Radim Rehurek的Word2vec Tutorial.\*\*准备输入\*\*Gensim的word2vec的输入是句子的序列. 每个句子是一个单词列表代码块例如：>>> # impor...

 Star\_Bob ([http://blog.csdn.net/Star\\_Bob](http://blog.csdn.net/Star_Bob)) 2015年08月20日 15:26 25274



基于 Gensim 的 Word2Vec 实践 ([http://blog.csdn.net/John\\_xyz/article/details/54706807](http://blog.csdn.net/John_xyz/article/details/54706807))

Word2Vec 基于 Gensim 的 Word2Vec 实践，从属于笔者的程序猿的数据科学与机器学习实战手册，代码参考gensim.ipynb。推荐前置阅读Python语法速览与机器学习开发环境...

 John\_xyz ([http://blog.csdn.net/John\\_xyz](http://blog.csdn.net/John_xyz)) 2017年01月24日 12:09 3457



Gensim Word2vec简介 (<http://blog.csdn.net/chivalrousli/article/details/54137706>)

本文转载自:http://ju.outofmemory.cn/entry/80023 本文主要基于Radim Rehurek的Word2vec Tutorial. 准备输入 Ge...

 chivalrousli (<http://blog.csdn.net/chivalrousli>) 2017年01月06日 14:32  1516



**python+gensim | jieba分词、词袋doc2bow、TFIDF文本挖掘 ([http://blog.csdn.net/sinat\\_26917383](http://blog.csdn.net/sinat_26917383))**

分词这块之前一直用R在做，R中由两个jiebaR+Rwordseg来进行分词，来看看python里面的jieba. 之前相关的文章： R语言 | 文本挖掘之中文分词包——Rwordseg包(原理、功...

 sinat\_26917383 ([http://blog.csdn.net/sinat\\_26917383](http://blog.csdn.net/sinat_26917383)) 2017年05月08日 22:24  4190



**用gensim对中文维基百科语料上的word2Vec相似度计算实验 (<http://blog.csdn.net/u013817676>)**

Word2vec 是Google 在 2013 年年中开源的一款将词表征为实数值向量的高效工具, 其利用深度学习的思想, 可以通过训练, 把对文本内容的处理简化为 K 维向量空间中的向量运算, 而向量空间...

 u013817676 (<http://blog.csdn.net/u013817676>) 2016年07月31日 15:51  996



**gensim实现python对word2vec的训练和计算 (<http://blog.csdn.net/xiaopihaierletian/article/details/53433072>)**

词向量 ( word2vec ) 原始的代码是C写的, Python也有对应的版本, 被集成在一个非常牛逼的框架gensim中. 我在自己的开源语义网络项目graph-mind ( 其实是我自己写的小玩具 ) 中使用了...

 xiaopihaierletian (<http://blog.csdn.net/xiaopihaierletian>) 2017年06月22日 20:31  1107



**word2vec词向量训练及gensim的使用 ([http://blog.csdn.net/zi\\_best/article/details/53433072](http://blog.csdn.net/zi_best/article/details/53433072))**

一、什么是词向量 词向量最初是用one-hot representation表征的, 也就是向量中每一个元素都关联着词库中的一个单词, 指定词的向量表示为: 其在向量中对应的元素设置为1, 其他的元素设置为0。采...

 zi\_best ([http://blog.csdn.net/zi\\_best](http://blog.csdn.net/zi_best)) 2016年12月02日 11:35  8405

**win10环境下使用gensim实现word2vec模型训练及测试 (<http://blog.csdn.net/u012614287/article/details/52120230>)**

最近开始从事NLP的实际项目, 需要使用word2vec ( w2v ) 实现语义近似度计算. 本文目的是在windows环境下进行gensim的环境配置和demo训练、测试功能的实现. word2vec是几年...

 u012614287 (<http://blog.csdn.net/u012614287>) 2017年05月15日 21:41  700



0

