





```
16
17 logging.basicConfig(format='%(asctime)s: %(levelname)s: %(message)s')
18 logging.root.setLevel(level=logging.INFO)
19 logger.info("running %s" % ' '.join(sys.argv))
20
21 # check and process input arguments
22 if len(sys.argv) < 4:
23     print globals()[['_doc_']] % locals()
24     sys.exit(1)
25 inp, outp1, outp2 = sys.argv[1:4]
26
27 model = Word2Vec(LineSentence(inp), size=400, window=5, min_count=5,
28                  workers=multiprocessing.cpu_count())
29
30 # trim unneeded model memory = use(much) less RAM
31 #model.init_sims(replace=True)
32 model.save(outp1)
33 model.save_word2vec_format(outp2, binary=False)
```

执行命令：执行“python train\_word2vec\_model.py wiki.en.text wiki.en.text.model wiki.en.text.vector”，大概得好几十个小时，因计算机性能而异。我们得到了一个gensim中默认格式的word2vec model和一个原始c版本word2vec的vector格式的模型: wiki.en.text.vector  
以下，调试结果：

2518

tensorflow保存变量出现错误（提示不能save）(<http://blog.csdn.net/a18852867035/article/details/70941735>)

2518



2





2



```
1 In [2]: import gensim
2
3 In [3]: model = gensim.models.Word2Vec.load_word2vec_format("wiki.en.text.vector", binary=False)
4
5 In [4]: model.most_similar("queen")
6 Out[4]:
7 [(u'princess', 0.5760838389396667),
8  (u'hyoui', 0.5671186447143555),
9  (u'janggyung', 0.5598698854446411),
10 (u'king', 0.5556215047836304),
11 (u'dollalolla', 0.5540223121643066),
12 (u'loranella', 0.5522741079330444),
13 (u'ramphaiphanni', 0.5310937166213989),
14 (u'jeheon', 0.5298476219177246),
15 (u'soheon', 0.5243583917617798),
16 (u'coronation', 0.5217245221138)]
17
18 In [5]: model.most_similar("man")
19 Out[5]:
20 [(u'woman', 0.7120707035064697),
21  (u'girl', 0.58659827709198),
22  (u'handsome', 0.5637181997299194),
23  (u'boy', 0.5425317287445068),
24  (u'villager', 0.5084836483001709),
25  (u'mustachioed', 0.49287813901901245),
26  (u'mcgucket', 0.48355430364608765),
27  (u'spider', 0.4804879426956177),
28  (u'policeman', 0.4780033826828003),
29  (u'stranger', 0.4750771224498749)]
30
31 In [6]: model.most_similar("woman")
32 Out[6]:
33 [(u'man', 0.7120705842971802),
34  (u'girl', 0.6736541986465454),
35  (u'prostitute', 0.5765659809112549),
36  (u'divorcee', 0.5429972410202026),
37  (u'person', 0.5276163816452026),
38  (u'schoolgirl', 0.5102938413619995),
39  (u'housewife', 0.48748138546943665),
40  (u'lover', 0.4858251214027405),
41  (u'handsome', 0.4773051142692566),
42  (u'boy', 0.47445783019065857)]
43
44 In [8]: model.similarity("woman", "man")
45 Out[8]: 0.71207063453821218
46
47 In [10]: model.doesnt_match("breakfast cereal dinner lunch".split())
```



```
48 Out[10]: 'cereal'
49
50 In [11]: model.similarity("woman", "girl")
51 Out[11]: 0.67365416785207421
52
53 In [13]: model.most_similar("frog")
54 Out[13]:
55 [(u'toad', 0.6868536472320557),
56 (u'barycragus', 0.6607867479324341),
57 (u'grylio', 0.626731276512146),
58 (u'heckscheri', 0.6208407878875732),
59 (u'clamitans', 0.6150864362716675),
60 (u'coplandi', 0.612680196762085),
61 (u'pseudacris', 0.6108512878417969),
62 (u'litoria', 0.6084023714065552),
63 (u'raniformis', 0.6044802665710449),
64 (u'watjulumensis', 0.6043726205825806)]
```

3. 中文数据

执行相同命令解压：python process\_wiki.py enwiki-latest-pages-articles.xml.bz2 wiki.en.text  
然后最痛苦的事情来了，如何进行繁简转换？（因为除了加杂一些英文词汇外，还有很多繁体字混迹其中）  
在网上下载开源项目 opencc(opencc-0.4.3.tar.gz 即可，太高版本的无法运行。)，解压，进入到目录下，进入 dos 界面。执行命令：  
opencc -i wiki.zh.text -o wiki.zh.text.jian -c zht2zhs.ini  
（注：不清楚的话，可以在评论区发言，也可通过邮箱：m18852867035@163.com 询问）  
然后进行分词处理，先下载 MeCab（日文分词系统），然后下载 mecab-chinesedict-binary，放在 MeCab bin 目录下，cmd 运行命令：mecab -d mecab-chinesedict-binary wakati wiki.zh.text.jian -o wiki.zh.text.jian.seg -b 10000000  
详细详细可参考《用 MeCab 打造一套实用的中文分词系统》  
(<http://www.52nlp.cn/%E7%94%A8mecab%E6%89%93%E9%80%A0%E4%B8%80%E5%A5%97%E5%AE%9E%E7%94%A8%E7%9A%84%E4%B8%AD%E6%96%87%E5%88%86%E8%AF%8D%E7%B3%BB%E7%BB%9F>)，也可在评论区和邮箱处询问。  
接下来是 utf-8 转码，个人感觉在 windows 下作用不大，但也可试试，反正我是偏向于不用转。  
iconv -c -t UTF-8 < wiki.zh.text.jian.seg > wiki.zh.text.jian.seg.utf-8（你还得下载 iconv 包，还不如不转）

接下来到了训练阶段：  
执行命令：  
python train\_word2vec\_model.py wiki.zh.text.jian.seg ( .utf-8 ) wiki.zh.text.model wiki.zh.text.vector  
大概两小时左右。  
测试命令同英文  
如：`In [1]: import gensim`  
  
In [2]: model = gensim.models.Word2Vec.load("wiki.zh.text.model")  
  
In [3]: model.most\_similar(u"足球")  
Out[3]:  
[(u' 8054 8d5b', 0.6553816199302673),



```
(u'\u7532\u7ea7', 0.6530429720878601),  
(u'\u7bee\u7403', 0.5967546701431274),  
(u'\u4ff1\u4e50\u90e8', 0.5872289538383484),  
(u'\u4e59\u7ea7', 0.5840631723403931),  
(u'\u8db3\u7403\u961f', 0.5560152530670166),  
(u'\u4e9a\u8db3\u8054', 0.5308005809783936),  
(u'allsvenskan', 0.5249762535095215),  
(u'\u4ee3\u8868\u961f', 0.5214947462081909),  
(u'\u7532\u7ec4', 0.5177896022796631)]
```

```
In [4]: result = model.most_similar(u"足球")
```

```
In [5]: for e in result:
```

```
print e[0], e[1]
```

```
.....:
```

```
联赛 0.65538161993
```

```
甲级 0.653042972088
```

```
篮球 0.596754670143
```

```
俱乐部 0.587228953838
```

```
乙级 0.58406317234
```

```
足球队 0.556015253067
```

```
亚足联 0.530800580978
```

```
allsvenskan 0.52497625351
```

```
代表队 0.521494746208
```

```
甲组 0.51778960228
```

```
In [6]: result = model.most_similar(u"男人")
```

```
In [7]: for e in result:
```

```
print e[0], e[1]
```

```
.....:
```

```
女人 0.77537125349
```

```
家伙 0.617369174957
```

```
妈妈 0.567102909088
```

```
漂亮 0.560832381248
```

```
잘했어 0.540875017643
```

```
谎言 0.538448691368
```

```
爸爸 0.53660941124
```

```
傻瓜 0.535608053207
```

```
예쁘다 0.535151124001
```

```
mc刘 0.529670000076
```

```
In [8]: result = model.most_similar(u"女人")
```

```
In [9]: for e in result:
```



2



```
print e[0], e[1]
....:
男人 0.77537125349
我的某 0.589010596275
妈妈 0.576344847679
잘했어 0.562340974808
美丽 0.555426716805
爸爸 0.543958246708
新娘 0.543640494347

谎言 0.540272831917
妞儿 0.531066179276
老婆 0.528521537781

In [10]: result = model.most_similar(u"青蛙")

In [11]: for e in result:
print e[0], e[1]
....:
老鼠 0.559612870216
乌龟 0.489831030369
蜥蜴 0.478990525007
猫 0.46728849411
鳄鱼 0.461885392666
蟾蜍 0.448014199734
猴子 0.436584025621
白雪公主 0.434905380011
蚯蚓 0.433413207531
螃蟹 0.4314712286

In [12]: result = model.most_similar(u"姨夫")

In [13]: for e in result:
print e[0], e[1]
....:
堂伯 0.583935439587
祖父 0.574735701084
妃所生 0.569327116013
内弟 0.562012672424
早卒 0.558042645454
瞻 0.553856015205
胤禔 0.553288519382
陈潜 0.550716996193
愷之 0.550510883331
叔父 0.550032019615
```



2





2



In [14]: result = model.most\_similar(u"衣服")

In [15]: for e in result:

print e[0], e[1]

.....:

鞋子 0.686688780785

穿着 0.672499775887

衣物 0.67173999548

大衣 0.667605519295

裤子 0.662670075893

内裤 0.662210345268

裙子 0.659705817699

西装 0.648508131504

洋装 0.647238850594

围裙 0.642895817757

In [16]: result = model.most\_similar(u"公安局")

In [17]: for e in result:

print e[0], e[1]

.....:

司法局 0.730189085007

公安厅 0.634275555611

公安 0.612798035145

房管局 0.597343325615

商业局 0.597183346748

军管会 0.59476184845

体育局 0.59283208847

财政局 0.588721752167

戒毒所 0.575558543205

新闻办 0.573395550251

In [18]: result = model.most\_similar(u"铁道部")

In [19]: for e in result:

print e[0], e[1]

.....:

盛光祖 0.565509021282

交通部 0.548688530922

批复 0.546967327595

刘志军 0.541010737419

立项 0.517836689949

报送 0.510296344757

计委 0.508456230164



2







水利部 0.503531932831  
国务院 0.503227233887  
经贸委 0.50156635046

```
In [20]: result = model.most_similar(u"清华大学")
```

```
In [21]: for e in result:
print e[0], e[1]
.....:
北京大学 0.763922810555
化学系 0.724210739136
物理系 0.694550514221
数学系 0.684280991554
中山大学 0.677202701569
复旦 0.657914161682
师范大学 0.656435549259
哲学系 0.654701948166
生物系 0.654403865337
中文系 0.653147578239
```

```
In [22]: result = model.most_similar(u"卫视")
```

```
In [23]: for e in result:
print e[0], e[1]
.....:
```

湖南 0.676812887192  
中文台 0.626506924629  
收藏 0.621356606483  
黄金档 0.582251906395  
cctv 0.536769032478  
安徽 0.536752820015  
非同凡响 0.534517168999  
唱响 0.533438682556  
最强音 0.532605051994  
金鹰 0.531676828861



2



```
In [24]: result = model.most_similar(u"习近平")
```

```
In [25]: for e in result:
print e[0], e[1]
.....:
胡锦涛 0.809472680092
江泽民 0.754633367062
李克强 0.739740967751
贾庆林 0.737033963203
```

曾庆红 0.732847094536  
吴邦国 0.726941585541  
总书记 0.719057679176  
李瑞环 0.716384887695  
温家宝 0.711952567101  
王岐山 0.703570842743

In [26]: result = model.most\_similar(u"林丹")

In [27]: for e in result:

print e[0], e[1]

.....:

黄综翰 0.538035452366

蒋燕皎 0.52646958828

刘鑫 0.522252976894

韩晶娜 0.516120731831

王晓理 0.512289524078

王适 0.508560419083

杨影 0.508159279823

陈跃 0.507353425026

龚智超 0.503159761429

李敬元 0.50262516737

In [28]: result = model.most\_similar(u"语言学")

In [29]: for e in result:

print e[0], e[1]

.....:

社会学 0.632598280907

人类学 0.623406708241

历史学 0.618442356586

比较文学 0.604823827744

心理学 0.600066184998

人文科学 0.577783346176

社会心理学 0.575571238995

政治学 0.574541330338

地理学 0.573896467686

哲学 0.573873817921

In [30]: result = model.most\_similar(u"计算机")

In [31]: for e in result:

print e[0], e[1]

.....:

自动化 0.674171924591



2



应用 0.614087462425  
自动化系 0.611132860184  
材料科学 0.607891201973  
集成电路 0.600370049477  
技术 0.597518980503

电子学 0.591316461563  
建模 0.577238917351  
工程学 0.572855889797  
微电子 0.570086717606

```
In [32]: model.similarity(u"计算机", u"自动化")
Out[32]: 0.67417196002404789
```

```
In [33]: model.similarity(u"女人", u"男人")
Out[33]: 0.77537125129824813
```

```
In [34]: model.doesnt_match(u"早餐 晚餐 午餐 中心".split())
Out[34]: u'\u4e2d\u5fc3'
```

```
In [35]: print model.doesnt_match(u"早餐 晚餐 午餐 中心".split())
中心`
```

SogouCA数据的测试：

解压后：合并各个.txt:

copy \*.txt SogouCA.txt

接下来，比较蛋疼，在Linux下运行了一行命令（windows不知道如何弄。）

```
cat SogouCA.txt | iconv -f gbk -t utf-8 -c | grep "" > corpus.txt
```

可参考利用中文数据跑Google开源项目word2vec (<http://www.cnblogs.com/hebin/p/3507609.html>)

接下来，繁简转换，分词等操作同上。

测试也同上，只是换了模型而已。

本人刚接触这一领域，有什么不足的地方欢迎各位指正！

版权声明：本文为博主原创文章，未经博主允许不得转载。

发表评论

hanxiping (hanxiping) 2017-04-21 09:51 1楼

([http://my.csdn.net/weixin\\_350668028](http://my.csdn.net/weixin_350668028))

或者问下在第二个使用中文维基百科的地方, wiki.zh.text和下面生成的wiki.zh.text.jian里面的内容看的话是真正的汉字还是像we2x3e这样的字符串, 我的为什么是这样的呢, 这直接导致了我下面用jieba分词失败, 求指导! 还有在process\_wiki.py的第29行, 必须会将text转化成字符串, 像这样str(text)程序才能够运行, 不然总是报错说错误

欸，错误如下：TypeError: write() argument must be str, not bytes

回复

相关文章推荐

初识nlp word2vec实战1 英语文本分析 (http://blog.csdn.net/qq\_29031319/article/details/65...

Bag of Words Meets Bags of Popcorn https://www.kaggle.com/c/word2vec-nlp-tutorial/data版本 1 未用word2...

qq\_29031319 (http://blog.csdn.net/qq\_29031319) 2017年03月25日 15:57 445

word2vec 之 Deep Learning in NLP （一）词向量和语言模型 (http://blog.csdn.net/jinruoya...

Post navigation ← PreviousNext → Deep Learning in NLP （一）词向量和语言模型 Posted on 2013 年 7 月 29 日...

jinruoyanxu (http://blog.csdn.net/jinruoyanxu) 2016年03月17日 17:34 2179



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\_pyfqnHmknjnvPjc0lZ0qnfK9ujYzP1ndPWb10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dBuhfkmWTYuARkuAFhrH640AwY5HDdnHn1rHDsPWD0lgF\_5y9YlZ0lQzq-uZR8mLPbUB48ugfElAqspynETZ-YpAq8nWqdAdxTvqdThP-5yF\_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqnHRLPjnvnfKEpyfqnHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqn1RdrHR)

NLP with deep learning(一) word2vec——词向量和语言模型 (http://blog.csdn.net/hongbud...

最近在学习斯坦福大学的natural language processing with deep learning课程，将其称之为deep nlp吧，课程链接：点击打开链接，网上公开视频课程：点击打开...

hongbudao (http://blog.csdn.net/hongbudao) 2017年08月08日 09:27 314

Word2Vec之Deep Learning in NLP （一）词向量和语言模型 (http://blog.csdn.net/cuilijuan...

转自licstar，真心觉得不错，可惜自己有些东西没有看懂 这篇博客是我看了半年的论文后，自己对 Deep Learning 在 NLP 领域中应用的理解和总结，在此分享。其中必然有局限性，欢迎...

cuilijuan02 (http://blog.csdn.net/cuilijuan02) 2015年08月26日 15:56 838

Word2Vec (Part 2): NLP With Deep Learning with Tensorflow (CBOW) (http://blog.csdn.n...

Tensorflow上其实本来已经有word2vec的代码了，但是我第一次看的时候也是看得云里雾里，还是看得不太明白。并且官方文档中只有word2vec的skip-gram实现，所以google了一下...

linuxheng (http://blog.csdn.net/linuxheng) 2017年04月14日 12:01 535



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 · 完成第一次跨越 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\\_pyfqHmknjFzrjD0lZ0qnfk9ujYzP1f4PjnY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1Y4uAPhn164P1nsuH04PyR10AwY5HDdnHn1rHDsPWD0lgF\\_5y9YlZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq\\_lyd8Q1R4uWc4uHf3uAckPHRkPWN9PhcsmW9huWqdlAdxTvqdThP-5HDknWFBmhkEusKzujYk0AFV5H00TZcqN0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cwrHnz0AqLUWYs0ZK45HcsP6KWThnqnWT1Pjfj](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjFzrjD0lZ0qnfk9ujYzP1f4PjnY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1Y4uAPhn164P1nsuH04PyR10AwY5HDdnHn1rHDsPWD0lgF_5y9YlZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4uWc4uHf3uAckPHRkPWN9PhcsmW9huWqdlAdxTvqdThP-5HDknWFBmhkEusKzujYk0AFV5H00TZcqN0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cwrHnz0AqLUWYs0ZK45HcsP6KWThnqnWT1Pjfj))

### 中文维基百科上的word2vec实验，python及java版本 (<http://blog.csdn.net/Yan456jie/article...>)

原文地址 <http://www.cnblogs.com/helloever/p/5280891.html> 最近一直把以前放下的NLP收拾起来，刚准备做关系抽取，然后把词变成向量的时候看到了Wor...

 Yan456jie (<http://blog.csdn.net/Yan456jie>) 2016年08月03日 11:57  512



### 英文维基百科语料上的word2vec实验 ([http://blog.csdn.net/baozi\\_/article/details/78295532](http://blog.csdn.net/baozi_/article/details/78295532))

环境准备python先安装python，2.7和3的版本都可以。pip用pip来下载python各种库非常方便，在命令行输入pip，如果不出错说明已经存在。如果没有的话，先下载pip安装文件，在该...

 baozi\_ ([http://blog.csdn.net/baozi\\_](http://blog.csdn.net/baozi_)) 2017年10月20日 15:01  58



### 中文维基百科语料上的Word2Vec实验 (<http://blog.csdn.net/yangyangrenren/article/details/...>)

此文主要参考52nlp-中英文维基百科语料上的Word2Vec实验，按照上面的步骤来做的，略有改动，因此不完全是转载的。这里，为了方便大家可以更快地运行gensim中的word2vec模型，我提供了w...

 yangyangrenren (<http://blog.csdn.net/yangyangrenren>) 2017年02月22日 12:10  5010

### 中英文维基百科语料上的Word2Vec实验 (<http://blog.csdn.net/jinruoyanxu/article/details/50...>)

最近试了一下Word2Vec, GloVe 以及对应的python版本 gensim word2vec 和 python-glove，就有心在一个更大规模的语料上测试一下，自然而然维基百科的语料进...

 jinruoyanxu (<http://blog.csdn.net/jinruoyanxu>) 2016年03月22日 18:25  755



### 中英文维基百科语料上的Word2Vec实验 (<http://blog.csdn.net/helihongzhizhuo/article/detai...>)

出处“我爱自然语言处理”：www.52nlp.cn 原文链接地址：http://www.52nlp.cn/中英文维基百科语料上的word2vec实验 最近试了一下Wor...

 helihongzhizhuo (<http://blog.csdn.net/helihongzhizhuo>) 2015年07月23日 15:28  1188

### wiki\_word2vec\_python实验 (<http://blog.csdn.net/zhangyu132/article/details/51912608>)

1.linux安装python版本 gensim word2vec：依赖库：Numpy和SciPy：首先进行安装以上两个库：ubuntu：sudo apt-get inst a...

 zhangyu132 (<http://blog.csdn.net/zhangyu132>) 2016年07月14日 20:29  2101


### 中英文维基百科语料上的Word2Vec实验 (<http://blog.csdn.net/chivalrousli/article/details/50...>)



2




最近试了一下Word2Vec, GloVe 以及对应的python版本 gensim word2vec 和 python-glove ,就有心在一个更大规模的语料上测试一下,自然而然维基百科的语料进入...

 chivalrousli (<http://blog.csdn.net/chivalrousli>) 2016年01月06日 17:36 1710

**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/u0138176...>)**

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 15:51 995

**中英文维基百科语料上的Word2Vec实验 (<http://blog.csdn.net/NORTHhan/article/details/511...>)**

最近试了一下Word2Vec, GloVe 以及对应的python版本 gensim word2vec 和 python-glove ,就有心在一个更大规模的语料上测试一下,自然而然维基百科的语料进...

 NORTHhan (<http://blog.csdn.net/NORTHhan>) 2016年04月11日 16:47 1458


**使用中文维基百科进行Word2Vec实验 (<http://blog.csdn.net/myonelotus/article/details/7822...>)**

1. 环境及语料1.1 环境 Homebrew Python jieba分词库 gensim库 1.2 下载维基百科语料 从这个链接下载<http://download.wikimedia.com/>...

 myonelotus (<http://blog.csdn.net/myonelotus>) 2017年10月13日 16:07 147


**中英文维基百科语料上的Word2Vec实验 (<http://blog.csdn.net/QFire/article/details/78508218>)**

<http://www.52nlp.cn/%E4%B8%AD%E8%8B%B1%E6%96%87%E7%BB%B4%E5%9F%BA%E7%99%BE%E7%A7%91%E8%AF%AD%E6%96%9...>

 QFire (<http://blog.csdn.net/QFire>) 2017年11月11日 19:41 32


**word2vec实验经验总结 (<http://blog.csdn.net/zhaoh1995/article/details/53242748>)**

最近进行的项目, 前期要求利用word2vec做相关的向量化处理, 于是作为小白从头开始学习word2vec的基本原理, 然后开始实验练手。在这里po一些学习和实验过程中的资料、注意点和经验点。基...


 zhaoh1995 (<http://blog.csdn.net/zhaoh1995>) 2016年11月20日 17:59 505

**[Algorithm & NLP] 文本深度表示模型——word2vec&doc2vec词向量模型 (<http://blog.csdn....>)**

转自: [Algorithm & NLP] 文本深度表示模型——word2vec&doc2vec词向量模型 阅读目录 1. 词向量 2.Distributed ...

 Together\_CZ ([http://blog.csdn.net/Together\\_CZ](http://blog.csdn.net/Together_CZ)) 2017年04月27日 21:38 692

**word2vec C#版本, 亲测可用! (<http://download.csdn.net/download/crac...>)**

 <http://download.csdn.net/download/crac...> 2017年11月22日 08:47 22KB [下载](#)

**word2Vec数学原理 (<http://download.csdn.net/download/poiki110/100443...>)**



2









[/http://download.csdn.net/detail/18852867035/52005030](#) 2017年10月30日 10:05 122KB [下载](#)



2

