用python通过结巴分词对语料库进行分词初步实现word2vec

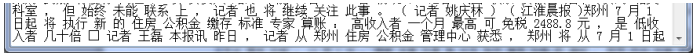
转载 2017年12月05日 16:41:07 297

一、前言

一开始看到word2vec环境的安装还挺复杂的，安了半天Cygwin也没太搞懂。后来突然发现，我为什么要去安c语言版本的呢，我应该去用python版本的，然后就发现了gensim，安装个gensim的包就可以用word2vec了，不过gensim只实现了word2vec里面的skip-gram模型。若要用到其他模型，就需要去研究其他语言的word2vec了。

二、语料准备

有了gensim包之后，看了网上很多教程都是直接传入一个txt文件，但是这个txt文件长啥样，是什么样的数据格式呢，很多博客都没有说明，也没有提供可以下载的txt文件作为例子。进一步理解之后发现这个txt是一个包含巨多文本的分好词的文件。如下图所示，是我自己训练的一个语料，我选取了自己之前用爬虫抓取的7000条新闻当做语料并进行分词。注意，词与词之间一定要用空格：



这里分词使用的是结巴分词。
这部分代码如下：

[python]

```python
import jieba
f1 =open("fenci.txt")
f2 =open("fenci_result.txt", 'a')
lines =f1.readlines()  # 读取全部内容
for line in lines:
    line.replace('\t', '').replace('\n', '').replace(' ','')
    seg_list = jieba.cut(line, cut_all=False)
    f2.write(" ".join(seg_list))

f1.close()
f2.close()
```

还要注意的一点就是语料中的文本一定要多，看网上随便一个语料都是好几个G，而且一开始我就使用了一条新闻当成语料库，结果很不好，输出都是0。然后我就用了7000条新闻作为语料库，分词完之后得到的fenci_result.txt是20M，虽然也不大，但是已经可以得到初步结果了。

三、使用gensim的word2vec训练模型
相关代码如下：

[python]

```python
from gensim.models import word2vec
import logging


# 主程序
logging.basicConfig(format='%(asctime)s:%(levelname)s: %(message)s', level=logging.INFO)
sentences =word2vec.Text8Corpus(u"fenci_result.txt")  # 加载语料
model =word2vec.Word2Vec(sentences, size=200)  #训练skip-gram模型，默认window=5

print(model)
# 计算两个词的相似度/相关程度
try:
    y1 = model.similarity(u"国家", u"国务院")
except KeyError:
    y1 = 0
print u"【国家】和【国务院】的相似度为：", y1
print"-----\n"
#
# 计算某个词的相关词列表
y2 = model.most_similar(u"控烟", topn=20)  # 20个最相关的
print u"和【控烟】最相关的词有：\n"
for item in y2:
    print item[0], item[1]
print"-----\n"

# 寻找对应关系
print u"书-不错，质量-"
y3 =model.most_similar([u'质量', u'不错'], [u'书'], topn=3)
for item in y3:
    print item[0], item[1]
print"----\n"

# 寻找不合群的词
y4 =model.doesnt_match(u"书 书籍 教材 很".split())
print u"不合群的词：", y4
print"-----\n"

# 保存模型，以便重用
model.save(u"书评.model")
# 对应的加载方式
# model_2 =word2vec.Word2Vec.load("text8.model")
```

```

# 以一种c语言可以解析的形式存储词向量
#model.save_word2vec_format(u"书评.model.bin", binary=True)
# 对应的加载方式

# model_3 =word2vec.Word2Vec.load_word2vec_format("text8.model.bin",binary=True)
```

输出如下：

[cpp]

```
"D:\program files\python2.7.0\python.exe" "D:/pycharm workspace/毕设/cluster_test/word2vec.py"
D:\program files\python2.7.0\lib\site-packages\gensim\utils.py:840: UserWarning: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
D:\program files\python2.7.0\lib\site-packages\gensim\utils.py:1015: UserWarning: Pattern library is not installed, lemmatization won't be
  warnings.warn("Pattern library is not installed, lemmatization won't be available.")
2016-12-12 15:37:43,331: INFO: collecting all words and their counts
2016-12-12 15:37:43,332: INFO: PROGRESS: at sentence #0, processed 0 words, keeping 0 word types
2016-12-12 15:37:45,236: INFO: collected 99865 word types from a corpus of 3561156 raw words and 357 sentence
2016-12-12 15:37:45,236: INFO: Loading a fresh vocabulary
2016-12-12 15:37:45,413: INFO: min_count=5 retains 29982 unique words (30% of original 99865, drops 6
2016-12-12 15:37:45,413: INFO: min_count=5 leaves 3444018 word corpus (96% of original 3561156, drops 117138)
2016-12-12 15:37:45,602: INFO: deleting the raw counts dictionary of 99865 items
2016-12-12 15:37:45,615: INFO: sample=0.001 downsamples 29 most-common words
2016-12-12 15:37:45,615: INFO: downsampling leaves estimated 2804247 word corpus (81.4% of prior 344
2016-12-12 15:37:45,615: INFO: estimated required memory for 29982 words and 200 dimensions: 6296220
2016-12-12 15:37:45,746: INFO: resetting layer weights
2016-12-12 15:37:46,782: INFO: training model with 3 workers on 29982 vocabulary and 200 features, using sg=
2016-12-12 15:37:46,782: INFO: expecting 357 sentences, matching count from corpus used for vocabulary surve
2016-12-12 15:37:47,818: INFO: PROGRESS: at 1.96% examples, 267531 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:48,844: INFO: PROGRESS: at 3.70% examples, 254229 words/s, in_qsize 3, out_qsize 1
2016-12-12 15:37:49,871: INFO: PROGRESS: at 5.99% examples, 273509 words/s, in_qsize 3, out_qsize 1
2016-12-12 15:37:50,867: INFO: PROGRESS: at 8.18% examples, 281557 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:51,872: INFO: PROGRESS: at 10.20% examples, 280918 words/s, in_qsize 5, out_qsize 0
2016-12-12 15:37:52,898: INFO: PROGRESS: at 12.44% examples, 284750 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:53,911: INFO: PROGRESS: at 14.17% examples, 278948 words/s, in_qsize 0, out_qsize 0
2016-12-12 15:37:54,956: INFO: PROGRESS: at 16.47% examples, 284101 words/s, in_qsize 2, out_qsize 1
2016-12-12 15:37:55,934: INFO: PROGRESS: at 18.60% examples, 285781 words/s, in_qsize 6, out_qsize 1
2016-12-12 15:37:56,933: INFO: PROGRESS: at 20.84% examples, 288045 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:57,973: INFO: PROGRESS: at 23.03% examples, 289083 words/s, in_qsize 6, out_qsize 2
2016-12-12 15:37:58,993: INFO: PROGRESS: at 24.87% examples, 285990 words/s, in_qsize 6, out_qsize 1
2016-12-12 15:38:00,006: INFO: PROGRESS: at 27.17% examples, 288266 words/s, in_qsize 4, out_qsize 1
2016-12-12 15:38:01,081: INFO: PROGRESS: at 29.52% examples, 290197 words/s, in_qsize 1, out_qsize 2
2016-12-12 15:38:02,065: INFO: PROGRESS: at 31.88% examples, 292344 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:03,188: INFO: PROGRESS: at 34.01% examples, 291356 words/s, in_qsize 2, out_qsize 2
2016-12-12 15:38:04,161: INFO: PROGRESS: at 36.02% examples, 290805 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:05,174: INFO: PROGRESS: at 38.26% examples, 292174 words/s, in_qsize 3, out_qsize 0
2016-12-12 15:38:06,214: INFO: PROGRESS: at 40.56% examples, 293297 words/s, in_qsize 4, out_qsize 1
2016-12-12 15:38:07,201: INFO: PROGRESS: at 42.69% examples, 293428 words/s, in_qsize 4, out_qsize 1
2016-12-12 15:38:08,266: INFO: PROGRESS: at 44.65% examples, 292108 words/s, in_qsize 1, out_qsize 1
2016-12-12 15:38:09,295: INFO: PROGRESS: at 46.83% examples, 292097 words/s, in_qsize 4, out_qsize 1
2016-12-12 15:38:10,315: INFO: PROGRESS: at 49.13% examples, 292968 words/s, in_qsize 2, out_qsize 2
2016-12-12 15:38:11,326: INFO: PROGRESS: at 51.37% examples, 293621 words/s, in_qsize 5, out_qsize 0
2016-12-12 15:38:12,367: INFO: PROGRESS: at 53.39% examples, 292777 words/s, in_qsize 2, out_qsize 2
2016-12-12 15:38:13,348: INFO: PROGRESS: at 55.35% examples, 292187 words/s, in_qsize 5, out_qsize 0
2016-12-12 15:38:14,349: INFO: PROGRESS: at 57.31% examples, 291656 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:15,374: INFO: PROGRESS: at 59.50% examples, 292019 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:16,403: INFO: PROGRESS: at 61.68% examples, 292318 words/s, in_qsize 4, out_qsize 2
2016-12-12 15:38:17,401: INFO: PROGRESS: at 63.81% examples, 292275 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:18,410: INFO: PROGRESS: at 65.71% examples, 291495 words/s, in_qsize 4, out_qsize 1
2016-12-12 15:38:19,433: INFO: PROGRESS: at 67.62% examples, 290443 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:20,473: INFO: PROGRESS: at 69.58% examples, 289655 words/s, in_qsize 6, out_qsize 2
2016-12-12 15:38:21,589: INFO: PROGRESS: at 71.71% examples, 289388 words/s, in_qsize 2, out_qsize 2
2016-12-12 15:38:22,533: INFO: PROGRESS: at 73.78% examples, 289366 words/s, in_qsize 0, out_qsize 1
2016-12-12 15:38:23,611: INFO: PROGRESS: at 75.46% examples, 287542 words/s, in_qsize 5, out_qsize 1
2016-12-12 15:38:24,614: INFO: PROGRESS: at 77.25% examples, 286609 words/s, in_qsize 3, out_qsize 0
2016-12-12 15:38:25,609: INFO: PROGRESS: at 79.33% examples, 286732 words/s, in_qsize 5, out_qsize 1
2016-12-12 15:38:26,621: INFO: PROGRESS: at 81.40% examples, 286595 words/s, in_qsize 2, out_qsize 0
2016-12-12 15:38:27,625: INFO: PROGRESS: at 83.53% examples, 286807 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:28,683: INFO: PROGRESS: at 85.32% examples, 285651 words/s, in_qsize 5, out_qsize 3
2016-12-12 15:38:29,729: INFO: PROGRESS: at 87.56% examples, 286175 words/s, in_qsize 6, out_qsize 1
```

```
2016-12-12 15:38:29,729: INFO: PROGRESS: at 87.56% examples, 281175 words/s, in_qsize 6, out_qsize 1
2016-12-12 15:38:30,706: INFO: PROGRESS: at 89.86% examples, 286920 words/s, in_qsize 5, out_qsize 0
2016-12-12 15:38:31,714: INFO: PROGRESS: at 92.10% examples, 287368 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:38:32,756: INFO: PROGRESS: at 94.40% examples, 288070 words/s, in_qsize 4, out_qsize 2
2016-12-12 15:38:33,755: INFO: PROGRESS: at 96.30% examples, 287543 words/s, in_qsize 1, out_qsize 0
2016-12-12 15:38:34,802: INFO: PROGRESS: at 98.71% examples, 288375 words/s, in_qsize 4, out_qsize 0
2016-12-12 15:38:35,286: INFO: worker thread finished; awaiting finish of 2 more threads
2016-12-12 15:38:35,286: INFO: worker thread finished; awaiting finish of 1 more threads
Word2Vec(vocab=29982, size=200, alpha=0.025)
【国家】和【国务院】的相似度为： 0.387535493256
-----

2016-12-12 15:38:35,293: INFO: worker thread finished; awaiting finish of 0 more threads
2016-12-
12 15:38:35,293: INFO: training on 17805780 raw words (14021191 effective words) took 48.5s, 289037 e
2016-12-12 15:38:35,293: INFO: precomputing L2-norms of word weight vectors
和【控烟】最相关的词有：

禁烟 0.6038454175
防烟 0.585186183453
执行 0.530897378922
烟控 0.516572892666
广而告之 0.508533298969
履约 0.507428050041
执法 0.494115233421
禁烟令 0.471616715193
修法 0.465247869492
该项 0.457907706499
落实 0.457776963711
控制 0.455987215042
这方面 0.450040221214
立法 0.44820779562
控烟办 0.436062157154
执行力 0.432559013367
控烟会 0.430508673191
进展 0.430286765099
监管 0.429748386145
惩罚 0.429243773222
-----

书-不错，质量-
生存 0.613928854465
稳定 0.595371186733
整体 0.592055797577
----

不合群的词： 很
-----

2016-12-12 15:38:35,515: INFO: saving Word2Vec object under 书评.model, separately None
2016-12-12 15:38:35,515: INFO: not storing attribute syn0norm
2016-12-12 15:38:35,515: INFO: not storing attribute cum_table
2016-12-12 15:38:36,490: INFO: saved 书评.model

Process finished with exit code 0
```

```
"D:\program files\python2.7.0\python.exe" "D:/pycharm workspace/环评/cluster_test/word2vec.py"
D:\program files\python2.7.0\lib\site-packages\gensim\utils.py:840: UserWarning: detected Windows, aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows, aliasing chunkize to chunkize_serial")
D:\program files\python2.7.0\lib\site-packages\gensim\utils.py:1015: UserWarning: Pattern library is not installed, lemmatization won't be available.
  warnings.warn("Pattern library is not installed, lemmatization won't be available.")
2016-12-12 15:37:43,331: INFO: collecting all words and their counts
2016-12-12 15:37:43,332: INFO: PROGRESS: at sentence #0, processed 0 words, keeping 0 word types
2016-12-12 15:37:45,236: INFO: collected 99865 word types from a corpus of 3561156 raw words and 357 sentences
2016-12-12 15:37:45,236: INFO: Loading a fresh vocabulary
2016-12-12 15:37:45,413: INFO: min_count=5 retains 29982 unique words (30% of original 99865, drops 69883)
2016-12-12 15:37:45,413: INFO: min_count=5 leaves 3444018 word corpus (96% of original 3561156, drops 117138)
2016-12-12 15:37:45,602: INFO: deleting the raw counts dictionary of 99865 items
2016-12-12 15:37:45,615: INFO: sample=0.001 downsamples 29 most-common words
2016-12-12 15:37:45,615: INFO: downsampling leaves estimated 2804247 word corpus (81.4% of prior 3444018)
2016-12-12 15:37:45,615: INFO: estimated required memory for 29982 words and 200 dimensions: 62962200 bytes
2016-12-12 15:37:45,746: INFO: resetting layer weights
2016-12-12 15:37:46,782: INFO: training model with 3 workers on 29982 vocabulary and 200 features, using sg=0 hs=0 sample=0.001 negative=5 window=5
2016-12-12 15:37:46,782: INFO: expecting 357 sentences, matching count from corpus used for vocabulary survey
2016-12-12 15:37:47,818: INFO: PROGRESS: at 1.98% examples, 267531 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:48,844: INFO: PROGRESS: at 3.70% examples, 254229 words/s, in_qsize 3, out_qsize 1
2016-12-12 15:37:49,871: INFO: PROGRESS: at 5.99% examples, 273509 words/s, in_qsize 3, out_qsize 1
```

```
2016-12-12 15:37:50,847: INFO: PROGRESS: at 6.16% examples, 251857 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:51,872: INFO: PROGRESS: at 10.20% examples, 280918 words/s, in_qsize 5, out_qsize 0
2016-12-12 15:37:52,898: INFO: PROGRESS: at 12.44% examples, 284750 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:53,911: INFO: PROGRESS: at 14.17% examples, 278948 words/s, in_qsize 0, out_qsize 0
2016-12-12 15:37:54,956: INFO: PROGRESS: at 16.47% examples, 284101 words/s, in_qsize 2, out_qsize 1
2016-12-12 15:37:55,934: INFO: PROGRESS: at 18.60% examples, 285781 words/s, in_qsize 6, out_qsize 1
2016-12-12 15:37:56,933: INFO: PROGRESS: at 20.84% examples, 288045 words/s, in_qsize 6, out_qsize 0
2016-12-12 15:37:57,973: INFO: PROGRESS: at 23.03% examples, 289083 words/s, in_qsize 6, out_qsize 2
```

```
【国家】和【国务院】的相似度为： 0.387535493256
```

```
和【控烟】最相关的词有：

禁烟 0.6038454175
防烟 0.585186183453
执行 0.530897378922
烟控 0.516572892666
广而告之 0.508533298969
履约 0.507428050041
执法 0.494115233421
禁烟令 0.471616715193
修法 0.465247869492
该项 0.457907706499
落实 0.457776963711
控制 0.455987215042
这方面 0.450040221214
立法 0.44820779562
控烟办 0.436062157154
执行力 0.432559013367
控烟会 0.430508673191
进展 0.430286765099
监管 0.429748386145
惩罚 0.429243773222
-----
```

```
书-不错，质量-
生存 0.613928854465
稳定 0.595371186733
整体 0.592055797577
----

不合群的词： 很
-----
```

转载自：http://blog.csdn.net/xiaoquantouer/article/details/53583980
(http://blog.csdn.net/xiaoquantouer/article/details/53583980)
原作者：小拳头

发表你的评论
(http://my.csdn.net/weixin_35068028)

相关文章推荐

python初步实现word2vec (http://blog.csdn.net/xiaoquantouer/article/details/53583980)

一、前言 一开始看到word2vec环境的安装还挺复杂的，安了半天Cygwin也没太搞懂。后来突然发现，我为什么要去安c语言

版本的呢，我应该去用python版本的，然后就发现了gensim，安装个ge...

xiaoquantouer (http://blog.csdn.net/xiaoquantouer) 2016年12月12日 16:08 17864

Chunkize warning while installing gensim 疑难杂症 (http://blog.csdn.net/kevinelstri/articl…

UserWarning: detected Windows; aliasing chunkize to chunkize_serial warnings.warn("detected Window...

kevinelstri (http://blog.csdn.net/kevinelstri) 2017年08月16日 19:16 1314

使用文本挖掘实现站点个性化推荐 (http://blog.csdn.net/yours0231/article/details/53689941)

作者：韦玮，重庆韬翔网络科技有限公司（上海）董事长兼总经理，IT作家，CSDN社区专家。 本文为韦玮原创文章，未经允许不得转载，点此查看作者有关《Python数据分析与挖掘经典案例实战》经验分享。...

yours0231 (http://blog.csdn.net/yours0231) 2016年12月16日 10:30 3221

2011-1-1 (http://blog.csdn.net/Lvbags247/article/details/6153747)

我爸是李刚"造句大赛开始了：窗前明月光，我爸是李刚。。。。。 老夫聊发少年狂，我爸爸，是李刚； 试问卷帘人，却道我爸是李刚； 日日思君不见君，我爸是李刚； 假如生活欺骗了你，不要悲...

Lvbags247 (http://blog.csdn.net/Lvbags247) 2011年01月19日 22:23 0

利用 word2vec 训练的字向量进行中文分词 (http://blog.csdn.net/peghoty/article/details/171…

最近针对之前发表的一篇博文《Deep Learning 在中文分词和词性标注任务中的应用》中的算法做了一个实现，感觉效果还不错。本文主要是将我在程序实现过程中的一些数学细节整理出来，借此优化一下自己的...

peghoty (http://blog.csdn.net/peghoty) 2013年12月04日 18:28 23618

【python gensim使用】word2vec词向量处理中文语料 (http://blog.csdn.net/churximi/articl…

word2vec介绍word2vec官网：https://code.google.com/p/word2vec/ word2vec是google的一个开源工具，能够根据输入的词的集合计算出词与词之间的...

churximi (http://blog.csdn.net/churximi) 2016年05月21日 20:57 25189

基于python的Word2Vec从分词到训练数据集详解 (http://blog.csdn.net/TYOUKAI_/article/de…

利用gensim的Word2Vec训练原始语料。 得到分词后的结果和训练出的语料集。

TYOUKAI_ (http://blog.csdn.net/TYOUKAI_) 2017年09月09日 21:13 133

Word2Vec + Jieba 分词实现视频标签距离计算 (http://blog.csdn.net/Mauvemask/article/det…

Word2Vec + Jieba 分词实现视频标签距离计算看[Word2vec][1]有一点时间了，不是很理解里面的算法所以决定先亲手实践试试看。 分词实现 Word2vec学习实现 分词实现首先将文...

Mauvemask (http://blog.csdn.net/Mauvemask) 2017年07月25日 13:54 199

Python：用Word2Vec 和 sklearn 对IMDB评论进行分类训练 (http://blog.csdn.net/github_38…

Python：用Word2Vec 和 sklearn 对IMDB评论进行分类训练之前一直做的是目标跟踪上的东西，这几天在看这本书又看到NLP，两者均作为对数据序列的处理，应该是有共通点的，于是就简单摸...

github_38705794 (http://blog.csdn.net/github_38705794) 2017年07月19日 23:03 326

0

RAR

Word2vec分词工具 (http://download.csdn.net/download/fyuanfena/96467…

2016年10月07日 12:50 203KB 下载

RAR

Word2Vec从分词到训练数据程序+数据集 (http://download.csdn.net/downl…

2017年09月09日 21:26 5.45MB 下载

gensim实现python对word2vec的训练和计算 (http://blog.csdn.net/qdhy199148/article/deta…

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）中使用了这...

qdhy199148 (http://blog.csdn.net/qdhy199148) 2016年06月27日 12:29 8054

gensim实现python对word2vec的训练和计算 (http://blog.csdn.net/xiaopihaierletian/article…

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 1115

word2vec的详细实例介绍（包含jieba分词提供的语料） (http://blog.csdn.net/dongyang_zh…

重要参考博客：http://blog.csdn.net/Eastmount/article/details/50700528 1、下载地址及安装 官网C语言下载地址：http://word2vec.g...

dongyang_zhao (http://blog.csdn.net/dongyang_zhao) 2017年12月03日 20:48 24

R语言丨文本挖掘——jiabaR包与分词向量化的simhash算法（与word2vec简单比较） (http://…

《数据挖掘之道》点评：虽然我比较执着于Rwordseg，并不代表各位看管执着于我的执着，推荐结巴分词包，小巧玲珑，没有那么多幺蛾子，而且R版本和python版本都有，除了词性标注等分词包必备功能以外，...

sinat_26917383 (http://blog.csdn.net/sinat_26917383) 2016年04月05日 21:01 10153

Word2Vec Python源代码 (http://download.csdn.net/download/happymoi…

0

(http://download.c 2017年11月27日 07:53 3KB 下载

word2vec basic python代码详解（配合Wordvec的数学原理使用更佳）(htt...

(http://download.c 2017年12月02日 19:23 19KB 下载

Doc2vec对M10语料库进行多分类 python (http://blog.csdn.net/lily960427/article/details/78...

语料库：是文献引用关系的语料库，将文献分成10类 包含3个txt，一个是文档ID+文档标题信息，一个是文档ID之间的引用关系，一个是文档类别 语料库下载：m10do2ve...

lily960427 (http://blog.csdn.net/lily960427) 2017年10月25日 14:30 79

TensorFlow实战中实现word2vec代码（含中文注释）(http://download.csd...

(http://download.c 2017年11月12日 22:05 13KB 下载

利用word2vec对关键词进行聚类 (http://blog.csdn.net/xiong_mao_1/article/details/2300579...

转载自：http://blog.csdn.net/zhaoxinfan/article/details/11069485

xiong_mao_1 (http://blog.csdn.net/xiong_mao_1) 2014年04月06日 01:00 684