用gensim学习word2vec

在word2vec原理篇中，我们对word2vec的两种模型CBOW和Skip-Gram，以及两种解法Hierarchical Softmax和Negative Sampling做了总结。这里我们就从实践的角度，使用gensim来学习word2vec。

1. gensim安装与概述

gensim是一个很好用的Python NLP的包，不光可以用于使用word2vec，还有很多其他的API可以用。它封装了google的C语言版的word2vec。当然我们可以可以直接使用C语言版的word2vec来学习，但是个人认为没有gensim的python版来的方便。

安装gensim是很容易的，使用"pip install gensim"即可。但是需要注意的是gensim对numpy的版本有要求，所以安装过程中可能会偷偷的升级你的numpy版本。而windows版的numpy直接装或者升级是有问题的。此时我们需要卸载numpy，并重新下载带mkl的符合gensim版本要求的numpy，下载地址在此：http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy。安装方法和scikit-learn 和pandas 基于windows单机机器学习环境的搭建这一篇第4步的方法一样。

安装成功的标志是你可以在代码里做下面的import而不出错：

```
from gensim.models import word2vec
```

2. gensim word2vec API概述

在gensim中，word2vec 相关的API都在包gensim.models.word2vec中。和算法有关的参数都在类gensim.models.word2vec.Word2Vec中。算法需要注意的参数有：

1) sentences: 我们要分析的语料，可以是一个列表，或者从文件中遍历读出。后面我们会有从文件读出的例子。

2) size: 词向量的维度，默认值是100。这个维度的取值一般与我们的语料的大小相关，如果是不大的语料，比如小于100M的文本语料，则使用默认值一般就可以了。如果是超大的语料，建议增大维度。

3) window：即词向量上下文最大距离，这个参数在我们的算法原理篇中标记为c，window越大，则和某一词较远的词也会产生上下文关系。默认值为5。在实际使用中，可以根据实际的需求来动态调整这个window的大小。如果是小语料则这个值可以设的更小。对于一般的语料这个值推荐在[5,10]之间。

4) sg: 即我们的word2vec两个模型的选择了。如果是0， 则是CBOW模型，是1则是Skip-Gram模型，默认是0即CBOW模型。

5) hs: 即我们的word2vec两个解法的选择了，如果是0， 则是Negative Sampling，是1的话并且负采样个数negative大于0， 则是Hierarchical Softmax。默认是0即Negative Sampling。

6) negative:即使用Negative Sampling时负采样的个数，默认是5。推荐在[3,10]之间。这个参数在我们的算法原理篇中标记为neg。

7) cbow_mean: 仅用于CBOW在做投影的时候，为0，则算法中的x_w为上下文的词向量之和，为1则为上下文的词向量的平均值。在我们的原理篇中，是按照词向量的平均值来描述的。个人比较喜欢用平均值来表示x_w,默认值也是1,不推荐修改默认值。

8) min_count:需要计算词向量的最小词频。这个值可以去掉一些很生僻的低频词，默认是5。如果是小语料，可以调低这个值。

9) iter: 随机梯度下降法中迭代的最大次数，默认是5。对于大语料，可以增大这个值。

10) alpha: 在随机梯度下降法中迭代的初始步长。算法原理篇中标记为η，默认是0.025。

11) min_alpha: 由于算法支持在迭代的过程中逐渐减小步长，min_alpha给出了最小的迭代步长值。随机梯度下降中每轮的迭代步长可以由iter，alpha， min_alpha一起得出。这部分由于不是word2vec算法的核心内容，因此在原理篇我们没有提到。对于大语料，需要对alpha, min_alpha,iter一起调参，来选择合适的三个值。

以上就是gensim word2vec的主要的参数，下面我们用一个实际的例子来学习word2vec。

3. gensim word2vec实战

我选择的《人民的名义》的小说原文作为语料，语料原文在这里。

完整代码参见我的github: https://github.com/ljpzzz/machinelearning/blob/master/natural-language-processing/word2vec.ipynb

拿到了原文，我们首先要进行分词，这里使用结巴分词完成。在中文文本挖掘预处理流程总结中，我们已经对分词的原理和实践做了总结。因此，这里直接给出分词的代码，分词的结果，我们放到另一个文件中。代码如下, 加入下面的一串人名是为了结巴分词能更准确的把人名分出来。

```
# -*- coding: utf-8 -*-

import jieba
import jieba.analyse

jieba.suggest_freq('沙瑞金', True)
jieba.suggest_freq('田国富', True)
jieba.suggest_freq('高育良', True)
jieba.suggest_freq('侯亮平', True)
jieba.suggest_freq('钟小艾', True)
jieba.suggest_freq('陈岩石', True)
jieba.suggest_freq('欧阳菁', True)
jieba.suggest_freq('易学习', True)
jieba.suggest_freq('王大路', True)
jieba.suggest_freq('蔡成功', True)
jieba.suggest_freq('孙连城', True)
jieba.suggest_freq('季昌明', True)
jieba.suggest_freq('丁义珍', True)
jieba.suggest_freq('郑西坡', True)
jieba.suggest_freq('赵东来', True)
jieba.suggest_freq('高小琴', True)
jieba.suggest_freq('赵瑞龙', True)
jieba.suggest_freq('林华华', True)
jieba.suggest_freq('陆亦可', True)
jieba.suggest_freq('刘新建', True)
jieba.suggest_freq('刘庆祝', True)

with open('./in_the_name_of_people.txt') as f:
    document = f.read()

    #document_decode = document.decode('GBK')

    document_cut = jieba.cut(document)
    #print  ' '.join(jieba_cut)  //如果打印结果，则分词效果消失，后面的result无法显示
    result = ' '.join(document_cut)
    result = result.encode('utf-8')
    with open('./in_the_name_of_people_segment.txt', 'w') as f2:
        f2.write(result)
f.close()
f2.close()
```

拿到了分词后的文件，在一般的NLP处理中，会需要去停用词。由于word2vec的算法依赖于上下文，而上下文有可能就是停词。因此对于word2vec，我们可以不用去停词。

现在我们可以直接读分词后的文件到内存。这里使用了word2vec提供的LineSentence类来读文件，然后套用word2vec的模型。这里只是一个示例，因此省去了调参的步骤，实际使用的时候，你可能需要对我们上面提到一些参数进行调参。

```
# import modules & set up logging
import logging
import os
from gensim.models import word2vec

logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)

sentences = word2vec.LineSentence('./in_the_name_of_people_segment.txt')

model = word2vec.Word2Vec(sentences, hs=1,min_count=1,window=3,size=100)
```

模型出来了，我们可以用来做什么呢？这里给出三个常用的应用。

第一个是最常用的，找出某一个词向量最相近的词集合，代码如下：

```
req_count = 5
for key in model.wv.similar_by_word('沙瑞金'.decode('utf-8'), topn =100):
    if len(key[0])==3:
        req_count -= 1
        print key[0], key[1]
        if req_count == 0:
            break;
```

我们看看沙书记最相近的一些3个字的词（主要是人名）如下：

高育良 0.967257142067
李达康 0.959131598473
田国富 0.953414440155
易学习 0.943500876427
祁同伟 0.942932963371

第二个应用是看两个词向量的相近程度，这里给出了书中两组人的相似程度：

```
print model.wv.similarity('沙瑞金'.decode('utf-8'), '高育良'.decode('utf-8'))
print model.wv.similarity('李达康'.decode('utf-8'), '王大路'.decode('utf-8'))
```

输出如下：

0.961137455325
0.935589365706

第三个应用是找出不同类的词，这里给出了人物分类题：

```
print model.wv.doesnt_match(u"沙瑞金 高育良 李达康 刘庆祝".split())
```

word2vec也完成的很好，输出为"刘庆祝"。

以上就是用gensim学习word2vec实战的所有内容，希望对大家有所帮助。

（欢迎转载，转载请注明出处。欢迎沟通交流： liujianping-ok@163.com）

分类: 0083. 自然语言处理

标签: 自然语言处理

好文要顶 关注我 收藏该文

刘建平Pinard
关注 - 15
粉丝 - 6332
+加关注

18 推荐 1 反对

« 上一篇: word2vec原理(三) 基于Negative Sampling的模型
» 下一篇: 特征工程之特征选择

posted @ 2017-08-03 14:12 刘建平Pinard 阅读(66042) 评论(82) 编辑 收藏

< Prev 1 2

评论列表

#51楼 [楼主] 2019-01-25 10:14 刘建平Pinard
@ 龙之逆鳞0123
你好，如果你是用的python2.7，那就可以直接用。如果是Python3.6+,那么可能有少量的语法改动，不过主体代码流程基本可以重用。
支持(0) 反对(0)

#52楼 2019-01-25 14:02 龙之逆鳞0123
@ 刘建平Pinard
大佬，您说的问题我都基本解决了。现在想咨询下，我现在找的语料库过大，有15个g的级别，如果用您的方法，就会有memory error的错误，这样应该咋么办啊?
支持(0) 反对(0)

#53楼 [楼主] 2019-01-27 22:22 刘建平Pinard
@ 龙之逆鳞0123
你好，你的数据量太大了，单机跑不了的，所以你要用分布式的word2vec来跑。

比如，你可以用Spark MLlib的 word2vec来跑分布式的训练。这需要你有分布式的spark环境。
算法库使用和gensim的类似，官方文档：
http://spark.apache.org/docs/latest/api/python/pyspark.ml.html?highlight=word2vec#pyspark.ml.feature.Word2Vec
支持(0) 反对(0)

#54楼 2019-01-30 15:21 aaronwang123
你好老师,
像 400万*30万的矩阵；30万是Tf-idf向量长度； 这么大的数据后面一般要怎么处理啊；感觉这个tf-idf特征向量对于大词汇量，普篇都很长啊；这个tf-idf特征几十万很正常吧，因为词汇量几十万应该不算很多；那么把这么长的特征怎么应用到后面的模型中，不管是分类还是聚类。。。。很疑惑，业界是怎么处理这个问题得呢，还请老师指导下。

谢谢
支持(0) 反对(0)

#55楼 2019-01-30 15:24 aaronwang123
另外还有个问题想老师指导下，Tf-idf向量是针对于句子或者段落文章整体的一个特征；而word2vec只是针对某个单词或词组；这两者是完全不同的东西，怎么能混淆一起使用呢？对吧老师
支持(0) 反对(0)

#56楼 [楼主] 2019-01-31 19:50 刘建平Pinard
@ aaronwang123
你好!
你可以对你的TF_IDF向量:
1， 做一些过滤，比如限定词在词汇表出现的频率不能太高和太低,去掉停词，限定每个文本的有特征值的词数量。

比如sklearn.feature_extraction.text.TfidfVectorizer下面就有max_df， min_df， max_features 等上面说的参数，这样用了后每个文本的词向量变少，变稀疏。

2. 做PCA降维，然后再 聚类或者分类
支持(0) 反对(0)

#57楼 [楼主] 2019-01-31 19:52 刘建平Pinard
@ aaronwang123
TF-IDF和Wordvec的确是完全不同的思路，要看你要解决的问题。
如果是你要分类聚类等，就用TF-IDF更简单。如果是要找近义词这样的需求，就用word2vec更方便。
一般不会混淆。
支持(2) 反对(0)

#58楼 2019-04-26 16:15 hllll
博主您好，我想请教一下，对于句子来说，之后要对其聚类，用tf-idf和doc2vec对句子进行投影，用哪个比较好，为什么呢？看到您之前说tfidf适用于聚类，word2vec适用于近义词查找，近义词是依赖于词语间的相似度吧，可是聚类也是依据余弦相似度进行聚类的呀，对于这些困惑，还请博主解答，谢谢！
支持(0) 反对(0)

#59楼 [楼主] 2019-04-27 23:26 刘建平Pinard
@ hllll
你好，首先两种方法肯定都是可以尝试的，理论上都行得通。但是tf-idf会更简单，而doc2vec训练的慢一些。

如果那你的数据量不大，推荐tf-idf法，数据量大的话doc2vec会好一些。或者说你直接使用fasttext库即可。
支持(0) 反对(0)

#60楼 2019-05-21 10:28 xuebao2017
@刘建平Pinard
博主您好！我想问一下，既然word2vec和doc2vec一般用于将词或者文本向量化，那么是否可以将向量化的结果直接用于语种识别任务呢？具体来说就是，先将每一段文本（一般不超过一行）通过doc2vec训练一个句向量出来，然后添加上语言标签，生成一个样本集，之后再划分成训练集与测试集，最后通过SVM或Naive Bayes等机器学习方法来分类和评估，您觉得这种思路可行吗？或者说有更好的建议吗？
支持(0) 反对(0)

#61楼 [楼主] 2019-05-21 10:42 刘建平Pinard
@ xuebao2017
你好，这种方法很好啊，已经有很多人在使用了，而且有不错的效果，应用起来没有任何问题。
支持(1) 反对(0)

#62楼 2019-06-19 18:42 TaoBerica
老师，你好，
现在我在做文本点击率估计（0/1 分类），在用w2v训练后对每行样本求平均得到200000 * 20维的向量数据后还想做相似度特征，请问老师该怎么做，
支持(0) 反对(0)

#63楼 [楼主] 2019-06-20 10:01 刘建平Pinard
@ TaoBerica
你好，简单的做法是，对于文本点击率估计（0/1 分类），使用TF-IDF之类的文本特征会比较简单。

对于文本相似度，则使用w2v得到词向量计算。

如果你想做的比较精细，可以做文本相似度的特征，你可以对所有的样本做一个聚类，然后不同类别的样本得到一个聚类类别离散特征，加入你的CTR模型特征即可。
支持(0) 反对(0)

#64楼 2019-06-20 17:30 TaoBerica
嗯嗯，好的，谢谢老师。
还有一个问题就是，特征相关性分析。
特征相关性分析的方法倒是知道不少，但是得到特征相关性后怎么做不知道，比如某几个特征相关性很高，请问老师该怎么做，是删掉吗?
支持(0) 反对(0)

#65楼 [楼主] 2019-06-21 10:33 刘建平Pinard
@ TaoBerica
你好，这个要看你分析相关性的目的是什么。如果是做普通的机器学习分类回归，那么相关与否关系并不大，不用删除。

如果只是做相似度分析，那么得到相关性的定量值也就可以了。
支持(0) 反对(0)

#66楼 2019-07-02 14:58 TaoBerica
刘老师你好，我想问一下，在训练词向量时，如何根据场景来选择是用Negative Sampling还是 Hierarchical Softmax，是选择用CBOW还是skip_gram
支持(0) 反对(0)

#67楼 [楼主] 2019-07-03 10:39 刘建平Pinard
@ TaoBerica
你好，一般文本数据量比较大的时候选择Negative Sampling好一些。数据量小的时候可以选择 Hierarchical Softmax。

CBOW还是skip_gram在训练的时候则都是可以的，没有特别的倾向性。
支持(0) 反对(0)

#68楼 2019-07-03 22:24 TaoBerica
好的，谢谢老师。
支持(0) 反对(0)

#69楼 2019-07-31 16:59 李子达(DA-South)
老师，您对min_count的设定有什么经验吗，一般设定为多少啊？这个参数对于word2vec的结果影响大吗？
支持(0) 反对(0)

#70楼 [楼主] 2019-08-01 10:21 刘建平Pinard
@ 李子达(DA-South)
你好，看你的文本是长文本还是段文本。如果是段文本，一般1-2就可以了。如果你的文本是中等长度的，使用默认值即可。

如果是超长文章，可以考虑增加到6-10.

这个值影响还是蛮大的，因为word2vec的结果一般会用于后续的分析，所以如果有一些生僻词干扰那么你的后续分析准确性就会差。
支持(0) 反对(0)

#71楼 2019-08-01 11:17 李子达(DA-South)
@ 刘建平Pinard
嗯嗯有道理，我实际工作中一般是多试几个值，然后看在训练集上的oov比率，找一个oov比较小的min_count，一般就是4或者5。刘老师，还有一个问题，就是对iter参数您有什么经验吗，我这边在工作中一般是5/10/15多试几次，找效果比较好的。然而我在看一篇论文的时候(链接在这里:http://licstar.net/archives/tag/词向量),作者的经验是控制iter在30-50之间效果比较好。不过，我按照作者经验试了试，我这边线上效果iter为5的效果比iter为50的要好1%左右，和他论文经验不太一样。您有什么见解吗?
支持(0) 反对(0)

#72楼 [楼主] 2019-08-02 10:22 刘建平Pinard
@ 李子达(DA-South)
你好，一般iter还是大一些比较好，因为少了的话可能梯度下降还没有收敛，这样你的词向量还不是最优的。

你说的情况可能存在，同时1%也很难说明什么，只能说你的文本训练数据比较容易收敛。大多数时候还是到几十比较好。
支持(0) 反对(0)

#73楼 2019-09-15 12:55 zhouxiaoou
大牛，spark下的Word2vec有没有继续训练的接口，意思是我前不久训练好了一个模型，我现在又有一些数据，可不可以接着上次的模型继续训练
支持(0) 反对(0)

#74楼 [楼主] 2019-09-15 14:55 刘建平Pinard
@ zhouxiaoou
你好，spark MLib这个功能似乎没有。

tensorflow是有的，sklearn有部分API也支持增量训练。
支持(0) 反对(0)

#75楼 2019-11-13 21:32 千千世界
您好，请问下如果需要向量化的词在训练的语料库里没有呢，怎么得到这个词的向量表示呢
支持(0) 反对(0)

#76楼 [楼主] 2019-11-14 09:08 刘建平Pinard
@ 千千世界
你好，没有的话就没有办法得到了。只能自己给默认值。
支持(0) 反对(0)

#77楼 2019-12-10 18:11 nivina
@ 刘建平Pinard
您好，在git上并没有'./in_the_name_of_people_segment.txt'这个文件，这个文件的需要做什么指定格式的处理吗
支持(0) 反对(0)

#78楼 [楼主] 2019-12-11 10:28 刘建平Pinard
@ nivina
你好，我的文章中有这个txt的链接： http://files.cnblogs.com/files/pinard/in_the_name_of_people.zip

你下载下来解压即可使用。如果是你自己的文本，那么可以转化成utf8比较好。
支持(0) 反对(0)

#79楼 2020-01-06 23:05 Vessalius
老师您好,关于您的第一个应用,我这里用您的源码运行的结果是
高育良 0.9647451639175415
田国富 0.9515622854232788
易学习 0.9363619089126587
侯亮平 0.9245094060897827
季昌明 0.9234869480133057
第二个应用:
0.9647452
0.92274284
第三个应用:
刘庆祝
结果也对吗?
支持(0) 反对(0)

#80楼 2020-01-30 16:48 nivina
@ 刘建平Pinard
谢谢刘老师
支持(0) 反对(0)

#81楼 2020-04-14 10:24 老猿12138
博主，你的代码在git上好像失效了，您能提供一个其他的下载地址吗?
支持(0) 反对(0)

#82楼 2020-04-14 10:25 老猿12138
好像是我的网络问题。我成功下载了，打扰了博主。
支持(0) 反对(0)

< Prev 1 2

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册， 访问 网站首页。

相关博文：
· 文本分类实战（一）——word2vec预训练词向量
· Gensim进阶教程：训练word2vec与doc2vec模型
· word2vec原理(三) 基于Negative Sampling的模型
· word2vec 入门（二）使用教程篇
· 文本深度表示模型Word2Vec
» 更多推荐...