利用Python实现wiki中文语料的word2vec模型构建

本实例主要介绍的是选取wiki中文语料，并使用python完成词向量模型构建的实践过程，不包含原理部分，旨在一步一步的了解自然语言处理的基本方法和步骤。文章主要包含了开发环境准备、数据的获取、数据的预处理、模型构建和模型测试四大内容，对应的是实现模型构建的四个步骤。

# 1、开发环境准备

## 1.1 python环境

在python官网https://www.python.org/downloads/下载计算机对应的python版本，本人使用的是Python2.7.13的版本。

## 1.2 gensim模块

（1）Word2vec需要使用第三方gensim模块， gensim模块依赖numpy和scipy两个包，因此需要依次下载对应版本的numpy、scipy、gensim。下载地址：

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

（2）下载完成后，在python安装目录下的Scripts目录中执行cmd命令：

pip install numpy\*.whl

pip install scipy\*.whl

pip install gensim.whl

（3）输入python命令进入python命令行，分别输入import numpy; import scipy; import gensim; 没有报错，即安装成功！

# 2、Wiki数据获取

## 2.1 Wiki中文数据的下载

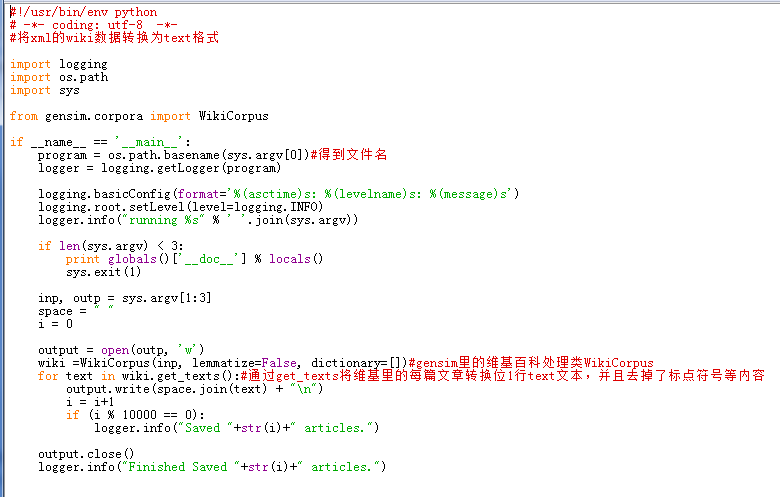
到wiki官网下载中文语料，下载完成后会得到命名为zhwiki-latest-pages-articles.xml.bz2的文件，大小约为1.3G，里面是一个XML文件。下载地址如下：

<https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2>

## 2.2 将XML的Wiki数据转换为text格式

（1）编写python程序将XML文件转换为text格式，使用到了gensim.corpora中的WikiCorpus函数来处理维基百科的数据。

python代码实现如下所示，文件命名为1\_process.py。



（2）在代码文件夹下运行如下cmd命令行，即可得到转换后生成的文件wiki.zh.txt。

D:\PyRoot\iDemo\wiki\_zh>python 1\_process.py zhwiki-latest-pages-articles.xml.bz2 wiki.zh.txt

（3）运行结果

2017-04-18 09:24:28,901: INFO: running 1\_process.py zhwiki-latest-pages-articles

.xml.bz2 wiki.zh.txt

2017-04-18 09:25:31,154: INFO: Saved 10000 articles.

2017-04-18 09:26:21,582: INFO: Saved 20000 articles.

2017-04-18 09:27:05,642: INFO: Saved 30000 articles.

2017-04-18 09:27:48,917: INFO: Saved 40000 articles.

2017-04-18 09:28:35,546: INFO: Saved 50000 articles.

2017-04-18 09:29:21,102: INFO: Saved 60000 articles.

2017-04-18 09:30:04,540: INFO: Saved 70000 articles.

2017-04-18 09:30:48,022: INFO: Saved 80000 articles.

2017-04-18 09:31:30,665: INFO: Saved 90000 articles.

2017-04-18 09:32:17,599: INFO: Saved 100000 articles.

2017-04-18 09:33:13,811: INFO: Saved 110000 articles.

2017-04-18 09:34:06,316: INFO: Saved 120000 articles.

2017-04-18 09:35:01,007: INFO: Saved 130000 articles.

2017-04-18 09:35:52,628: INFO: Saved 140000 articles.

2017-04-18 09:36:47,148: INFO: Saved 150000 articles.

2017-04-18 09:37:41,137: INFO: Saved 160000 articles.

2017-04-18 09:38:33,684: INFO: Saved 170000 articles.

2017-04-18 09:39:37,957: INFO: Saved 180000 articles.

2017-04-18 09:43:36,299: INFO: Saved 190000 articles.

2017-04-18 09:45:21,509: INFO: Saved 200000 articles.

2017-04-18 09:46:40,865: INFO: Saved 210000 articles.

2017-04-18 09:47:55,453: INFO: Saved 220000 articles.

2017-04-18 09:49:07,835: INFO: Saved 230000 articles.

2017-04-18 09:50:27,562: INFO: Saved 240000 articles.

2017-04-18 09:51:38,755: INFO: Saved 250000 articles.

2017-04-18 09:52:50,240: INFO: Saved 260000 articles.

2017-04-18 09:53:57,526: INFO: Saved 270000 articles.

2017-04-18 09:55:01,720: INFO: Saved 280000 articles.

2017-04-18 09:55:22,565: INFO: finished iterating over Wikipedia corpus of 28285

5 documents with 63427579 positions (total 2908316 articles, 75814559 positions

before pruning articles shorter than 50 words)

2017-04-18 09:55:22,568: INFO: Finished Saved 282855 articles.

由结果可知，31分钟运行完成282855篇文章，得到一个931M的txt文件。

# 3、Wiki数据预处理

## 3.1 中文繁体替换成简体

Wiki中文语料中包含了很多繁体字，需要转成简体字再进行处理，这里使用到了OpenCC工具进行转换。

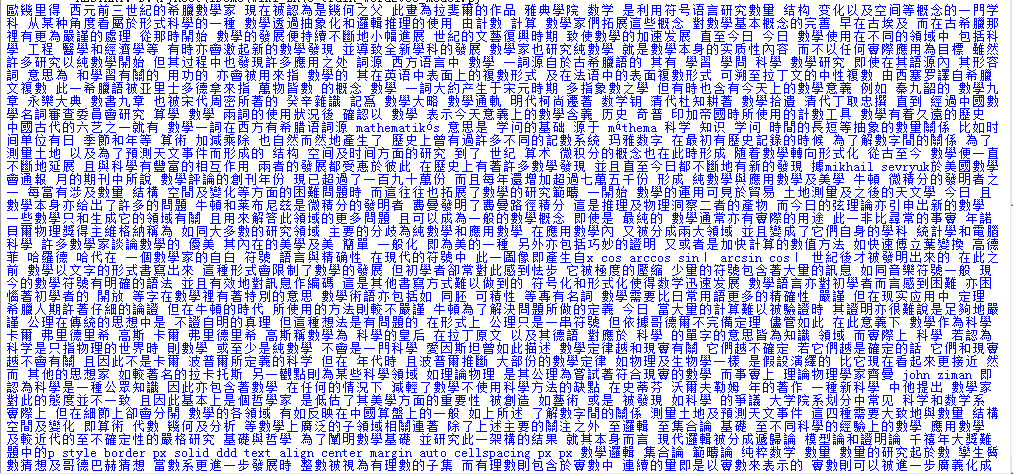
（1）安装OpenCC

到以下链接地址下载对应版本的OpenCC，本人下载的版本是opencc-1.0.1-win32。

<https://bintray.com/package/files/byvoid/opencc/OpenCC>

另外，资料显示还有python版本的，使用pip install opencc-python进行安装，未实践不做赘述。

（2）使用OpenCC进行繁简转换

进入解压后的opencc的目录（opencc-1.0.1-win32），双击opencc.exe文件。在当前目录打开dos窗口（Shift+鼠标右键->在此处打开命令窗口），输入如下命令行：

opencc -i wiki.zh.txt -o wiki.zh.simp.txt -c t2s.json

则会得到文件wiki.zh.simp.txt，即转成了简体的中文。

繁体中文示例截图如下所示：

（3）结果查看

解压后的txt有900多M，用notepad++无法打开，所以采用python自带的IO进行读取。Python代码如下：

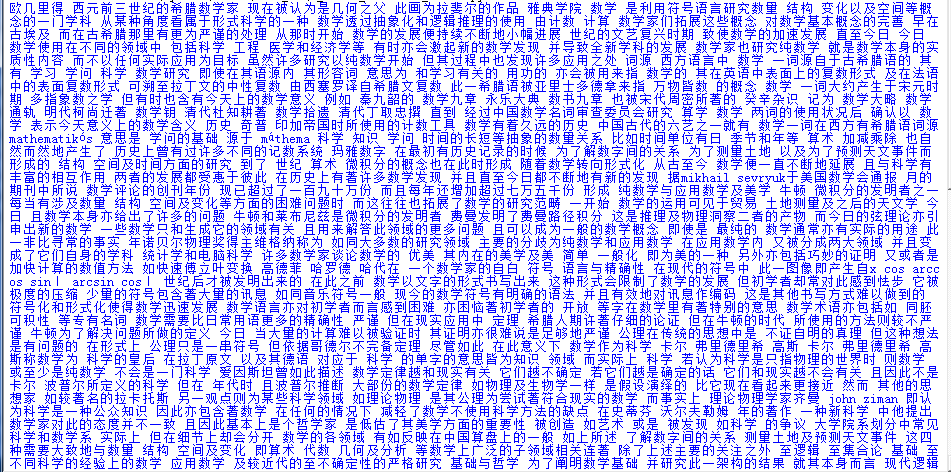
import codecs,sys

f = codecs.open(‘wiki.zh.simp.txt‘,‘r‘,encoding="utf8")

line = f.readline()

print(line)

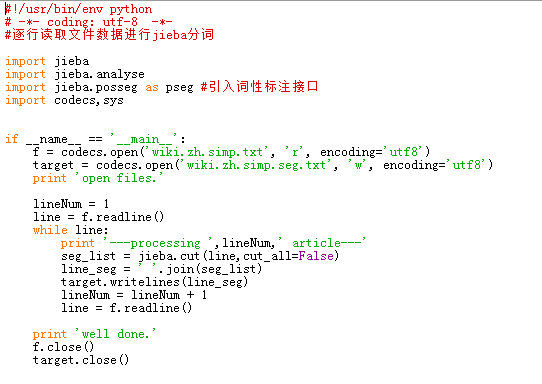
转换后的简体中文截图如下所示：



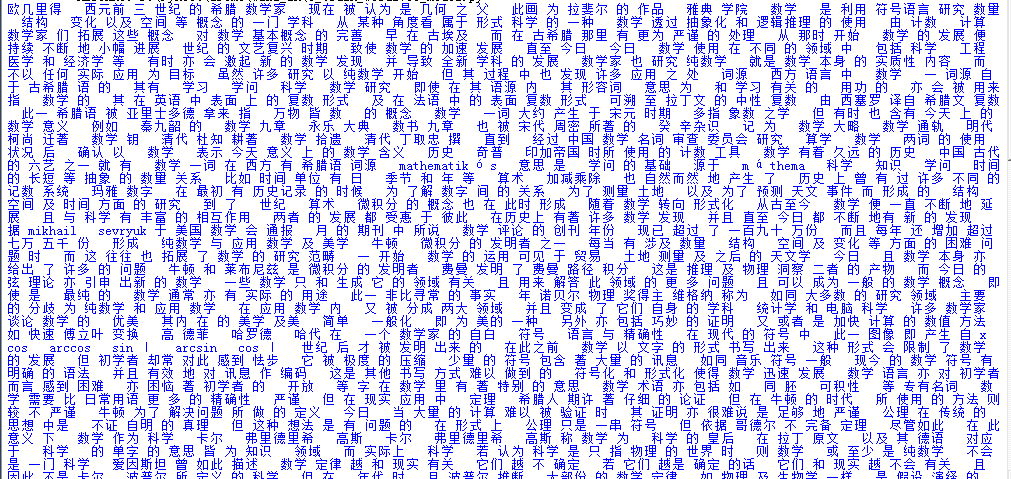
## 3.2 结巴分词

采用结巴分词对字体简化后的wiki中文语料数据集进行分词，在执行代码前需要安装jieba模块。由于此语料已经去除了标点符号，因此在分词程序中无需进行清洗操作，可直接分词。若是自己采集的数据还需进行标点符号去除和去除停用词的操作。

Python实现代码如下：



代码执行完成后得到一个1.12G大小的文档wiki.zh.simp.seg.txt。分词结果截图如下所示：



# 4、Word2Vec模型训练

（1）word2vec模型实现

分好词的文档即可进行word2vec词向量模型的训练了。文档较大，本人在4GWin7的电脑中报内存的错误，更换成8G内容的Mac后即可训练完成，且速度很快。

具体Python代码实现如下所示，文件命名为3\_train\_word2vec\_model.py。



（2）运行结果

2017-05-03 21:54:14,887: INFO: training on 822697865 raw words (765330910 effective words) took 1655.2s, 462390 effective words/s

2017-05-03 21:54:14,888: INFO: saving Word2Vec object under /Users/sy/Desktop/pyRoot/wiki\_zh\_vec/wiki.zh.text.model, separately None

2017-05-03 21:54:14,888: INFO: not storing attribute syn0norm

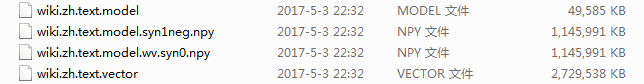
2017-05-03 21:54:14,889: INFO: storing np array 'syn0' to /Users/sy/Desktop/pyRoot/wiki\_zh\_vec/wiki.zh.text.model.wv.syn0.npy

2017-05-03 21:54:16,505: INFO: storing np array 'syn1neg' to /Users/sy/Desktop/pyRoot/wiki\_zh\_vec/wiki.zh.text.model.syn1neg.npy

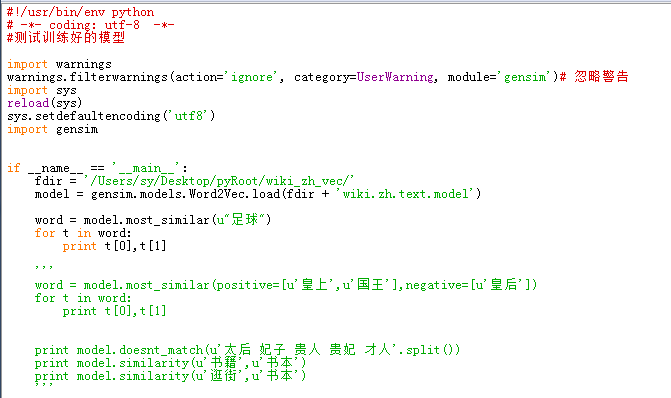
2017-05-03 21:54:18,123: INFO: not storing attribute cum\_table

2017-05-03 21:54:26,542: INFO: saved /Users/sy/Desktop/pyRoot/wiki\_zh\_vec/wiki.zh.text.model

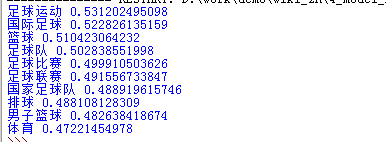
2017-05-03 21:54:26,543: INFO: storing 733434x400 projection weights into /Users/sy/Desktop/pyRoot/wiki\_zh\_vec/wiki.zh.text.vector

代码运行完成后得到如下四个文件，其中wiki.zh.text.model是建好的模型，wiki.zh.text.vector是词向量。

# 5、测试模型

模型训练好后，来测试模型的结果。Python代码如下，文件名为4\_model\_match.py。

运行文件得到结果，即可查看给定词的相关词。



至此，使用python对中文wiki语料的词向量建模就全部结束了，wiki.zh.text.vector中是每个词对应的词向量，可以在此基础上作文本特征的提取以及分类。