wiki中文语料的word2vec模型构建

Chinese Rap项目需要根据topic 生成”联想词”, 通过word2vec 模型将word用vector表示，把word 关联性问题转化为vector距离问题，word2vec模型提供了友好的接口，很容易就能 得到我们需要的”联想词”的集合, 根据项目需要，我们选择了wiki 中文预料，然后一步一步实现模型的构建，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步骤

# 1、开发环境准备

## 1.1 硬件环境

[I5-3320M@2.6GHZ双核](mailto:I5-3320M@2.6GHZ双核)，4G内存

Windows10

(PS: 当然linux 是最佳的选择，但是我只有一个window系统的laptop, 之前都是

用VM 上的linux 系统，但是这次训练的data量太大了，在VM 上简单的文件处理都会挂掉，所以只能选用Window系统，折腾了半天才把环境搭建好)

## 1.2 python环境

Python2.7的版本。

## 1.3 gensim模块

Pip install gensim

# 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 github link:

<https://github.com/hanjing0098/chineserap/blob/master/word2vec/wiki_process.py>

# 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，window版本

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

按照README 进行安装

（2）使用OpenCC进行繁简转换：

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

则会得到文件wiki.zh.jian.txt，即转成了简体的中文。

## 3.2 分词

一开始对比了几个分词工具，发现清华的thulac 分词效果最好，所以选用了清华的的分词工具， 但是跑了一下午，报“index out of range”的错误，网上搜了一下，说是thulac分词对大文件支持不太好，也有人给出了解决方案，将Maxlength调大, 所以我就尝试着把Maxlength调的足够大，结果跑了一晚上也没跑出来结果，sigh！ 就换了结巴分词工具，在执行代码前需要安装jieba模块: pip install jieba, 调用jieba提供的分词接口。

Python 代码github link:

https://github.com/hanjing0098/chineserap/blob/master/word2vec/wiki\_seg.py

# 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。设定参数： hidden size=600,window\_size=20, min\_count=5, 由于文档较大，训练时间为 4~5hours

Python 代码github link:

https://github.com/hanjing0098/chineserap/blob/master/word2vec/train\_word2vec\_model.py

# 5、测试模型

模型训练好后，来测试模型的结果，选取跟topic相近的1000个词。测试模型对环境的要求也很高，试图在虚拟机上测试，结果也是报出memory error 的问题， 一次测试大概需要20分钟才能出结果

Python代码github link

[**https://github.com/hanjing0098/chineserap/blob/master/word2vec/similar\_test.py**](https://github.com/hanjing0098/chineserap/blob/master/word2vec/similar_test.py)