第三次编程作业：汉语词向量

> 模型：Skip-Gram with Negative Sampling (SGNS)
>
> 超参数：前后2窗口
>
> 维数：100
>
> 训练语料：https://dumps.wikimedia.org/backup-index.html 汉语数据
>
> 获得词向量后，利用余弦距离来计算pku_sim_test.txt文件中每行两个词之间的相似度，最终是要输出每行两个词之间的余弦距离值。

方法概要

- 数据预处理
  - 训练语料来源
    - 最新汉语数据：zhwiki-20200501-pages-articles-multistream
    - 实验使用数据：zhwiki-20190320-pages-articles-multistream
    - 备注：最新数据大小为1.9GB，19年3月的数据大小为1.7GB。因为没有找到合适的国内镜像站，所以没有选择最新数据，而是退而求其次，选择了在百度网盘有备份的旧数据。
  - 语料库文章的提取
    - 使用WikiExtractor来直接提取xml.bz2文件中的文章，结果存储于zhwiki/AA。
  - 部分繁体中文的转换
    - 因为据称维基百科语料库中的文章内容里面的简体和繁体是混乱的，所以需要将所有的繁体字转换成为简体字。
    - 这里使用了OpenCC来进行转换，结果存储于zhwiki/BB。
- 正则表达式和分词
  - 过滤标签内容：使用WikiExtractor提取的文章，会包含许多的不相关的内容，所以需要将这些不相关的内容通过正则表达式来去除。
  - 分词：通过jieba对文章进行分词。
  - 合并保存文件：将分割之后的文章保存到文件中，每一行表示一篇文章，每个词之间使用空格进行分隔，结果存储于zhwiki/CC。
- 汉语词向量（耗时1小时+）
  - 使用gensim.models的word2vec按照要求的方法和参数进行模型训练，结果以二进制压缩形式存储于model。
- 计算两个词的相似度:
  - 读取训练得到的模型model，以及待计算相似的pku_sim_test.txt文件，结果保存为2017211416.txt文件

Class ChineseWord2Vec

1. 使用库
   1. `jieba`：分词。
   2. `gensim`：`word2vec`，`KeyedVectors`。
2. 属性
   1. `wiki_path`：wiki路径，默认为`"./zhwiki/BB/"`。

2. `corpus_path`：过滤、分词后的语料路径，默认为`"./zhwiki/CC/"`。
3. `corpus_name`：语料文件名，默认为`"wiki_corpus"`。
4. `model_path`：保存的训练模型的路径，默认为`"./model/"`。
5. `model_name`：模型文件名，默认为`"wiki_corpus_binary.bin"`。

3. 方法
    1. 静态方法
        1. `parse_zhwiki`：从输入路径读取文件，使用正则表达式解析文本，最后将结果保存于所给文件路径。
    2. 实例方法
        1. `parse`：使用正则表达式来去除WikiExtractor提取的文章中无用内容，再通过jieba对文章进行分词，最后合并保存文件（每一行表示一篇文章，每个词之间使用空格进行分隔）。
        2. `train`：使用gensim.models的word2vec按照所给参数进行模型训练，以文件形式保存训练的模型。
        3. `compute`：读取以特定格式保存的待计算相似度的文件，从模型文件中加载模型，将计算结果按题目要求保存于指定文件中。

gensim.models.word2vec

- 简介
    - **Word2Vec**是使用浅层神经网络将单词嵌入到低维向量空间中的模型。结果是一组词向量，其中在向量空间中靠在一起的向量根据上下文具有相似的含义，而彼此远离的词向量具有不同的含义。
    - 该模型有两个版本，Word2Vec类同时实现了这两个版本：
        - **Skip-grams (SG)**
        - Continuous-bag-of-words (CBOW)
    - **SG**模型采用通过跨文本数据移动窗口而生成的对（word1，word2），并根据给定输入单词的合成任务来训练一个1层神经网络，来预测输入附近单词的概率分布。虚拟的一键式单字编码通过“投影层”到达隐藏层。这些投影权重在以后被解释为单词嵌入。因此，如果隐藏层具有100个神经元，则此网络将供100维单词嵌入。
    - 部分模型参数
        - `size`：单词向量的维数，默认为100。
        - `window`：句子中当前词和预测词之间的最大距离，默认为5，即前后2窗口。
        - `sg`：如果为1，则使用SG训练算法；否则为CBOW训练算法。默认为0，即CBOW算法。
        - `hs`：如果为1，则使用层次softmax训练模型。如果为0，且`negative`为非零，则将使用负数采样。
        - `negative`：如果>0，将使用负采样，负数的int指定应绘制多少个“噪声词”（通常在5到20之间）。如果设置为0，则不使用负采样。