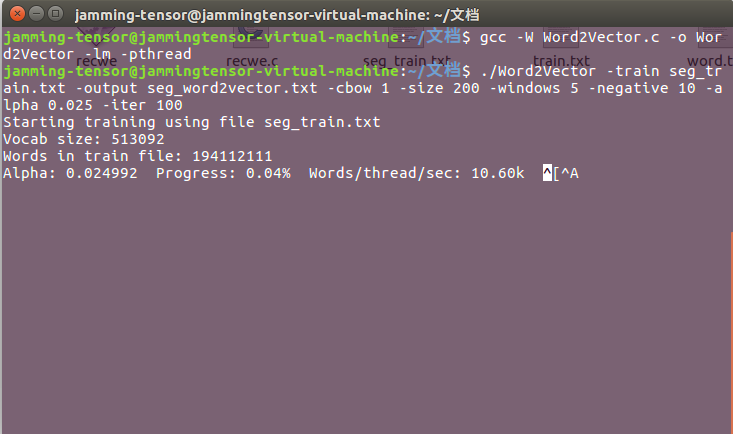
Word2Vector工具多种使用方式，主要包括以下几种：(本文介绍前两种)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 地址 | CBOW | | Skip-Gram | |
| C | 文档末尾会附上源码 | HS | NEG | HS | NEG |
| Python | <http://radimrehurek.com/gensim/> | HS | NEG | HS | NEG |
| Java | <https://github.com/ansjsun/Word2VEC_java> | HS |  | HS |  |
| C++ | <https://github.com/jdeng/word2vec> | 未知 | 未知 | 未知 | 未知 |

一、直接下载编译Google提供的Word2Vector的源码，然后在Linux命令行下进行调用：



然后输入训练参数：



部分训练参数表:

|  |  |
| --- | --- |
| 参数名称 | 参数解释 |
| -train | 训练数据 |
| -output | 向量输出文件 |
| -cbow | 是否使用cbow模型，0表示使用skip-gram模型，1表示使用cbow模型，默认情况下是skip-gram模型，cbow模型快一些，skip-gram模型效果好一些 |
| -size | 表示输出的词向量维数 |
| -window | 为训练的窗口大小，8表示每个词考虑前8个词与后8个词（实际代码中还有一个随机选窗口的过程，窗口大小<=5) |
| -negative | 表示是否使用NEG方，0表示不使用 |
| -hs | 是否使用HS方法，0表示不使用，1表示使用 |
| -sample | 表示 采样的阈值，如果一个词在训练样本中出现的频率越大，那么就越会被采样 |
| -alpha | 学习速率 |
| -min-count | 表示设置最低频率，默认为5，如果一个词语在文档中出现的次数小于该阈值，那么该词就会被舍弃 |
| -classes | 表示词聚类簇的个数，从相关源码中可以得出该聚类是采用k-means |

二、直接在Pycharm中安装gensim工具包，然后导入，执行如下代码

train = "训练文本路径"

vector = r"向量输出文件路径"

# 默认hs等于0时是负采样层，等于1是哈夫曼树，参数negative代表负采样的个数，默认是5，sg默认是0，代表是cbow，等于1是skip-gram

model = Word2Vec(LineSentence(train), size=300,workers=10, window=5, negative=5, iter= 5,alpha=0.015,min\_count=7)

model.wv.save\_word2vec\_format(vector)