genism的word2vec

2020年11月24日 21:39

今天看完数模视频之后稍微搞了一下项目。做到了词向量的一步了。
用的是python里genism包里的word2vec工具

Word2vec训练模型语句：

```
model = word2vec.Word2Vec(data, hs=1, min_count=1, window=10, size=100)
```

原：

class gensim.models.word2vec.Word2Vec(sentences=None,size=100,alpha=0.025,window=5, min_count=5, max_vocab_size=None, sample=0.001,seed=1, workers=3,min_alpha=0.0001, sg=0, hs=0, negative=5, cbow_mean=1, hashfxn=<built-in function hash>,iter=5,null_word=0, trim_rule=None, sorted_vocab=1, batch_words=10000)

参数解析：

- sentences：可以是一个list，对于大语料集，建议使用BrownCorpus,Text8Corpus或LineSentence构建。
- sg： 用于设置训练算法，默认为0，对应CBOW算法；sg=1则采用skip-gram算法。
- size：是指特征向量的维度，默认为100。大的size需要更多的训练数据,但是效果会更好. 推荐值为几十到几百。
- window：表示当前词与预测词在一个句子中的最大距离是多少
- alpha: 是学习速率
- seed：用于随机数发生器。与初始化词向量有关。
- min_count: 可以对字典做截断. 词频少于min_count次数的单词会被丢弃掉, 默认值为5
- max_vocab_size: 设置词向量构建期间的RAM限制。如果所有独立单词个数超过这个，则就消除掉其中最不频繁的一个。每一千万个单词需要大约1GB的RAM。设置成None则没有限制。
- sample: 高频词汇的随机降采样的配置阈值，默认为1e-3，范围是(0,1e-5)
- workers参数控制训练的并行数。
- hs: 如果为1则会采用hierarchical softmax技巧。如果设置为0（defaut），则negative sampling会被使用。
- negative: 如果>0,则会采用negativesamping，用于设置多少个noise words
- cbow_mean: 如果为0，则采用上下文词向量的和，如果为1（defaut）则采用均值。只有使用CBOW的时候才起作用。
- hashfxn： hash函数来初始化权重。默认使用python的hash函数
- iter： 迭代次数，默认为5
- trim_rule： 用于设置词汇表的整理规则，指定那些单词要留下，哪些要被删除。可以设置为None（min_count会被使用）或者一个接受()并返回RU·E_DISCARD,uti·s.RU·E_KEEP或者uti·s.RU·E_DEFAU·T的函数。
- sorted_vocab： 如果为1（defaut），则在分配word index 的时候会先对单词基于频率降序排序。
- batch_words：每一批的传递给线程的单词的数量，默认为10000

一些操作：

操作	含义
Model.sava("word2vec.model")	保存模型
Words_Vector = model.wv	不需要更新只需要查询的保存词向量。KeyedVectors模式
get_latest_training_loss()	获得当前的培训损失值
init_weights()	将所有投影权重置为未经训练的状态，但保留现在的词汇表
Load()	加载以前保存的word2vec模型