成 绩 _____

北京航空航天大學
BEIHANG UNIVERSITY

深度学习与自然语言处理
第四次大作业

词向量汇聚

院（系）名称	自动化科学与电气工程学院
专业名称	电子信息
学生学号	ZY2103812
学生姓名	朱远哲
指导教师	秦曾昌

2022 年 5 月

目　　录

1 问题描述

利用给定语料库（或者自选语料库），利用神经语言模型（如：Word2Vec， GloVe等模型）来训练词向量，通过对词向量的聚类或者其他方法来验证词向量的有效性。

2 问题表达

2.1 词袋模型

词袋模型（bag-of-words model）。这个模型将每个文档变换为一个固定长度的整型向量。例如，给定句子：

John likes to watch movies. Mary likes movies too.

John also likes to watch football games. Mary hates football.

模型输出的向量为：

[1, 2, 1, 1, 2, 1, 1, 0, 0, 0, 0]

[1, 1, 1, 1, 0, 1, 0, 1, 2, 1, 1]

每一个向量有10个元素，其中每个元素为一个特定单词出现在文档中的次数。元素的排序是随机的。在上面的例子中，元素的排序对应单词：["John", "likes", "to", "watch", "movies", "Mary", "too", "also", "football", "games", "hates"]。

词袋模型具有很好的效果，但仍有一些缺点。

首先，它们丢失了关于单词顺序的所有信息：“John likes Mary”和“Mary likes John”对应相同的向量。这里有一个解决方案：为了捕获局部单词顺序，bag of n-grams模型考虑使用长度为n的单词短语来表示作为固定长度向量的文档。但该模型遭受数据稀疏性（data sparsity）和高维性（ high dimensionality）的影响。

第二，这个模型不会试图去学习基础单词的意义，因此，向量间的距离不会总反映它们在词意上的距离。Word2Vec解决了这第二个问题。

2.2 Word2Vec 模型

Word2Vec 是一种较新的模型，它使用浅层神经网络将单词嵌入到低维向量空间中。模型的结果是单词向量集，其中在向量空间中彼此靠近的向量在文本内有相似的意义，且彼此相距遥远的单词向量有不同的含义。例如，strong和powerful彼此间相近，但strong和Paris可能会相当的远。

该模型有两种版本，且Word2Vec类实现了两者：Skip-grams (SG), Continuous-bag-of-words (CBOW)

Word2Vec Skip-gram模型，比如，输入在文本数据上移动的窗口而生成的（word1，word2）对，且基于给定单词的合成任务训练一个只有一个隐藏层的神经网络，从而为我们预测附近单词对输入的概率分布。虚拟的独热编码通过"投影层"到隐藏层：这些投影权重后来被解释为单词嵌入。因此，如果隐藏层有300个神经元，这个网络将给我们300维的单词嵌入。

Continuous-bag-of-words Word2vec和skip-gram模型非常相似。它同样也是一个只包含一个隐藏层的神经网络。合成训练任务现在使用多个输入上下文单词的平均值，而不是像skip-gram那样使用单个单词来预测中心单词。同样，将独热单词转化为平均向量的投影权重（宽度与隐藏层相同）被解释为单词嵌入。

3 具体算法实现

3.1 数据处理

首先选取了相对了解较多的射雕英雄传作为输入进行分析，需要对语料库进行断句分词，使用jieba分词工具，同时对语料进行预处理，去除广告和无意义的字符，得到分词后的文件。在这一步需要注意，在一般的NLP处理中，会需要去停用词。由于word2vec的算法依赖于上下文，而上下文有可能就是停词。因此对于word2vec，我们可以不用去停词，同时加上了人名使jieba分词能够更好地将人名分出来。

```
jieba.suggest_freq('郭靖', True)
jieba.suggest_freq('黄蓉', True)
jieba.suggest_freq('杨康', True)
jieba.suggest_freq('穆念慈', True)
jieba.suggest_freq('黄药师', True)
jieba.suggest_freq('欧阳锋', True)
jieba.suggest_freq('洪七公', True)
jieba.suggest_freq('周伯通', True)
jieba.suggest_freq('柯镇恶', True)
jieba.suggest_freq('梅超风', True)

def read_novel(path_in, path_out):  # 读取语料内容
    content = []
    names = os.listdir(path_in)
    for name in names:
        novel_name = path_in + '\\' + name
        fenci_name = path_out + '\\' + name
        for line in open(novel_name, 'r', encoding='ANSI'):
            line.strip('\n')
            line = content_deal(line)
            con = jieba.cut(line, cut_all=False)# 结巴分词
            # con = content_stopword(con)
            # content.append(con)
            content.append(" ".join(con))
        with open(fenci_name, "w", encoding='utf-8') as f:
            f.writelines(content)
    return content, names
```

3.2 模型训练

模型训练采用gensim中的Word2vec模型进行训练，sentences同样采用gensim中的LineSentence模块进行处理，并设置相应参数

```
model = Word2Vec(sentences=LineSentence(name), hs=1, min_count=10, window=5, vector_size=200, sg=0, epochs=200)
```

3.3 聚类分析

对训练得到的模型进行三个方面的聚类分析测试应用。

首先测试找出某一词向量最相近的词的集合，选取了郭靖、黄蓉、杨康、

欧阳锋四个人名分析其相近词，为了筛选出有意义的人名等，取长度为两个或三个词的排名前五的词。

第二，利用函数寻找指定两个词语的关系。

第三，在一组词中，找出不同类的词语。

代码如下

```
test_name = ['郭靖', '黄蓉', '杨康', '欧阳锋']
name = "output/射雕英雄传.txt"
model = Word2Vec(sentences=LineSentence(name), hs=1, min_count=10, window=5, vector_size=200, sg=0, epochs=200)
for i in range(len(test_name)):
    print(test_name[i])
    req_count = 5
    for key in model.wv.similar_by_word(test_name[i], topn=20):
        if len(key[0]) == 3 or len(key[0]) == 2:
            req_count -= 1
            print(key[0], key[1])
            if req_count == 0:
                break
print(model.wv.similarity('黄药师', '东邪'))
print(model.wv.similarity('欧阳锋', '西毒'))
print(model.wv.doesnt_match(u"郭靖 黄蓉 杨康 铁木真".split()))
```

3.4 结果输出

代码输出结果如下

```
郭靖
黄蓉 0.6389580368995667
欧阳克 0.4623052775859833
陆冠英 0.40385720133781433
周伯通 0.4009549617767334
黄药师 0.38407352566719055
黄蓉
郭靖 0.6389579772949219
洪七公 0.5561643242835999
陆冠英 0.46617186069488525
完颜康 0.4597887098789215
周伯通 0.4484280049800873
杨康
包惜弱 0.33859533071517944
金国 0.3118075728416443
黄蓉 0.29417213797569275
黄药师 0.27674832940101624
郭靖 0.26076531410217285
欧阳锋
周伯通 0.43545523285865784
黄蓉 0.381889283657074
洪七公 0.3783362805843353
黄药师 0.34688302874565125
郭靖 0.3167935907840729
0.021135777
0.029393956
铁木真
```

4 运行结果

4.1 运行结果

1. 选取了郭靖、黄蓉、杨康、欧阳锋四个人名分析其相近词如下：

	郭靖		黄蓉		杨康		欧阳锋	
1	黄蓉	0.644792	郭靖	0.623074	包惜弱	0.332261	洪七公	0.412539
2	陆冠英	0.460886	洪七公	0.554305	郭靖	0.33125	周伯通	0.379936
3	欧阳克	0.450102	周伯通	0.471142	黄蓉	0.309067	黄蓉	0.368079
4	周伯通	0.411171	完颜康	0.448514	杨铁心	0.282329	郭靖	0.319776
5	柯镇恶	0.407128	黄药师	0.441215	裘千仞	0.271198	梅超风	0.307648

从结果中可以看出模型训练取得了较好的效果，算法得到的相近词在文中均具有明显的关系，且排序上也较为合理。

2. 计算了“东邪”和“黄药师”、“西毒”和“欧阳锋”的相近度，结果分别为0.021和0.029，分析原因认为人物的称号在文中和人名一起出现的频率不高，没有建立起联系。

3. 选取郭靖、黄蓉、杨康、铁木真四个词进行不同类词语的筛选，成功识别出了铁木真为不同类词语。

4.2 结果讨论与分析

从结果可以看出，利用word2vec模型得到了较好的训练和测试效果，在预处理中添加了对人名的处理，是的jieba分词能够更准确的分出人名。算法模型直接采用的gensim中的word2vec模型进行训练，并进行了三个方面的应用测试。

5 个人总结与体会

本次大作业实现了利用word2vec进行词向量的聚类，通过本次程序编写，我对python编程变得更加熟悉，同时也学习了解了word2vec模型，，通过算法的学习程序的编写，又让我对自然语言处理有了更加深刻的理解，也通过大作业有了很大的收获。