# 自然语言处理第一次实验

### 实验目的

* 了解分词与词向量的训练。

分词就是将句子、段落、文章这种长文本，分解为以字词为单位的数据结构，方便后续的处理分析工作。词向量，顾名思义就是⽤来表⽰词的向量，也可被认为是词的特征向量或表征。把词映射为实数域向量的技术也叫词嵌入。我们通常在训练语言模型的同时得到词向量。

* 了解Word2Vec词向量的特性。

词向量本质是将一些低维、离散、不带任何意义的序号映射成带有特定任务性质的高维特征。为了研究这些特性，可以显式地通过余弦距离来比较两个词的相似性，也可以通过词间的类比关系来探查词向量的影藏特性，另外还可以通过PCA降维的方式将词向量可视化。

### 实验环境

python 3 + jieba + gensim + sklearn + matplotlib + NumPy + seaborn

* python3

除了高性能外，拥有NumPy、SciPy等优秀的数值计算、统计分析库。TensorFlow、Caffe等著名的深度学习框架都提供了Python接口。

* jieba

jieba是一款优秀的Python第三方中文分词库，支持三种分词模式：精确模式、全模式和搜索引擎模式。

* gensim

gensim是一款开源的第三方Python工具包，用于从原始的非结构化的文本中，无监督地学习到文本隐层的向量表达。它支持包括TF-IDF，LSA，LDA，和word2vec在内的多种算法。

* Sklearn

Sklearn (全称 Scikit-Learn) 是基于 Python 语言的机器学习工具。它建立在 NumPy, SciPy, Pandas 和 Matplotlib 之上，Sklearn 里面有六大任务模块：分别是分类、回归、聚类、降维、模型选择和预处理。

* Matplotlib

Matplotlib 是 Python 的绘图库，可与 NumPy 一起使用，提供了一种有效的 MATLAB 开源替代方案。

* NumPy

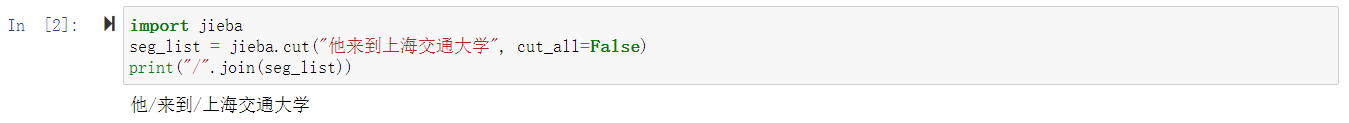
NumPy是Python语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

* Seaborn

Seaborn是基于Matplotlib的图形可视化Python包，在Matplotlib的基础上进行了更高级的封装，使得作图更加容易。

### 实验步骤

1. 使用jieba分词工具进行分词，使用方法：jieba.cut(text)；

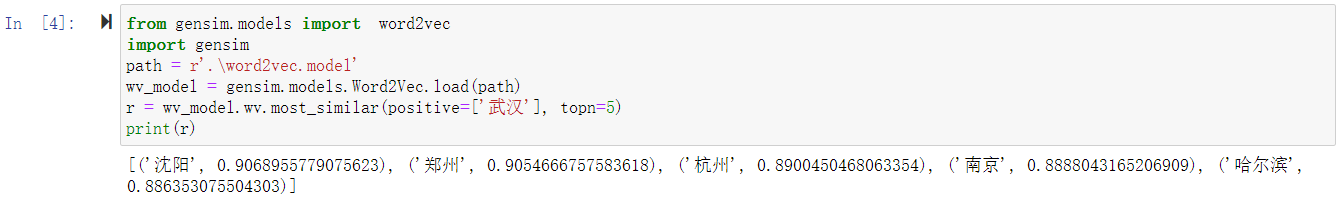


2. 使用gensim中的Word2Vec模型训练词向量：model = Word2Vec(common\_texts, size=100, window=5, min\_count=1, workers=4)；min\_count指定了需要训练词语的最小出现次数，默认为5；size指定了训练时词向量维度，默认为100；worker指定了完成训练过程的线程数，默认为1不使用多线程。

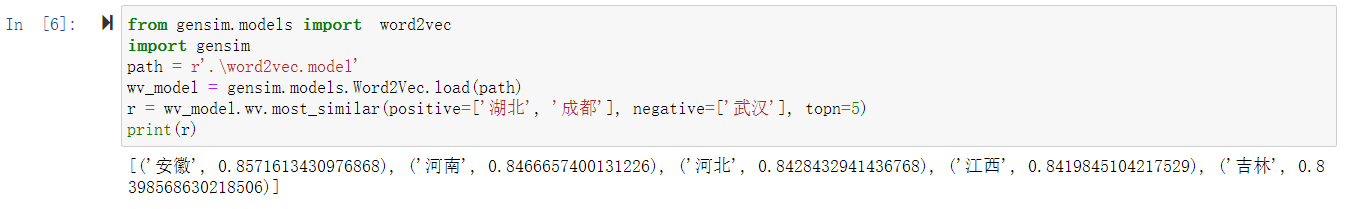
3. 使用训练好的词向量对指定的词（2个例子）进行相关性比较：model.similarity('中国','中华')；



4. 使用训练好的词向量选出与指定词（2个例子）最相似的5个词：model.wv.most\_similar(positive=['武汉'], topn=5)；

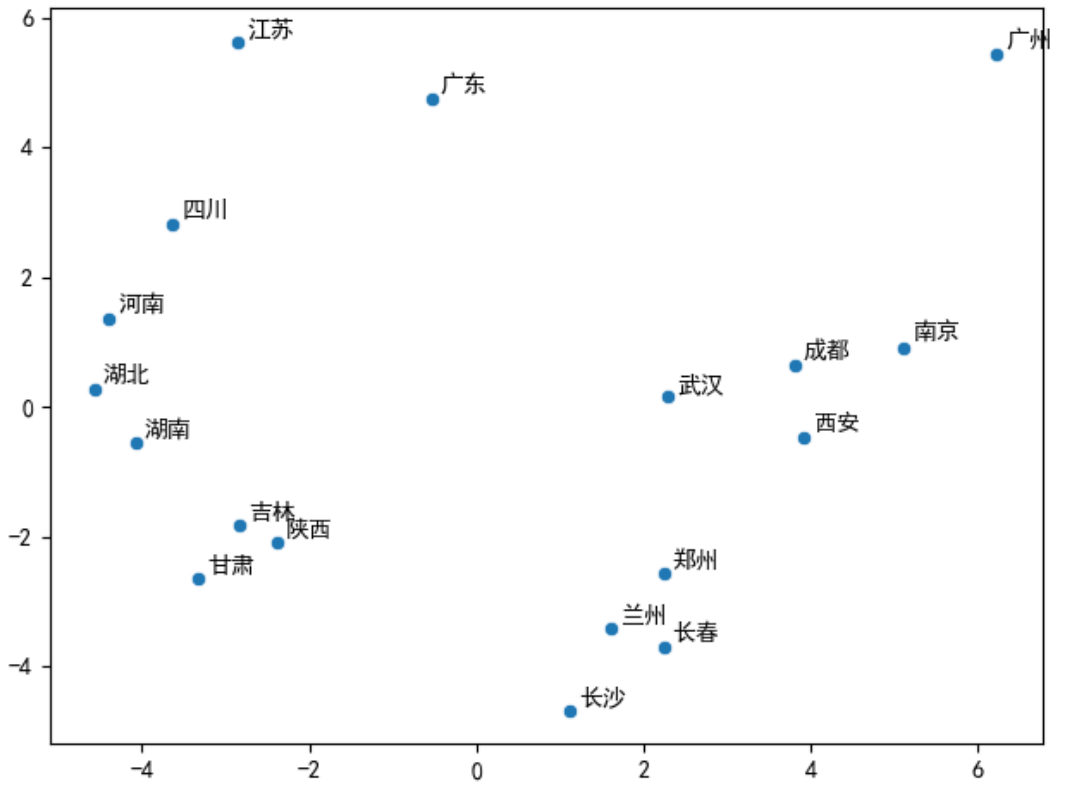


5. 使用训练好的词向量选出与指定词类比最相似的5个词（2个例子），如**湖北 - 武汉 + 成都 = 四川**： model.wv.most\_similar(positive=['湖北', '成都'], negative=['武汉'], topn=5)；



6. 使用sklearn中的PCA方法对列表['江苏', '南京', '成都', '四川', '湖北', '武汉', '河南', '郑州', '甘肃', '兰州', '湖南', '长沙', '陕西', '西安', '吉林', '长春', '广东', '广州', '浙江', '杭州']（可换成其他）中的所有词的词向量进行降维并使用seaborn和matplotlib将其可视化：

|  |
| --- |
| pca = PCA(n\_components=2) |
| results = pca.fit\_transform(embeddings) |
| sns.scatterplot(x=results[:, 0], y=results[:, 1]) |



### 提交时间

**10月23号截止**

各班班长或学习委员收集班内所有同学的实验报告和实验代码后发送到3263979158@qq.com邮箱中。

将所有文件打包为zip格式的压缩包，命名方式：姓名-学号-第1次实验

### 实验要求

* 完成所有实验内容
* 良好的代码风格
* 完整的实验报告

### 参考资料

1. [jieba文档](https://github.com/fxsjy/jieba)

2. [Word2Vec Model in gensim](https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html#sphx-glr-auto-examples-tutorials-run-word2vec-py)

3. [PCA in sklearn](https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html)

4. [Word2Vec中的数学原理详解](https://www.cnblogs.com/peghoty/p/3857839.html)

5. [Distributed Representations of Words and Phrases and their Compositionality](https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf)

6. [Efficient Estimation of Word Representations in Vector Space](https://arxiv.org/pdf/1301.3781.pdf)

**附件1：**

**武汉大学国家网络安全学院**

**实验报告**

**课程名称 自然语言处理**

**专业年级 网络空间安全2019级**

**姓 名 戴挺**

**学 号 2018302060188**

**协 作 者 无**

**实验学期 2021-2022　 学年 第一 学期**

**课堂时数 课外时数**

**填写时间 2021 年 10 月 11 日**

|  |
| --- |
| **实验介绍** |
| **【实验名称】：分词及词向量计算** |
| **【实验目的】：**   * **了解分词与词向量的训练。**   **分词就是将句子、段落、文章这种长文本，分解为以字词为单位的数据结构，方便后续的处理分析工作。词向量，顾名思义就是⽤来表⽰词的向量，也可被认为是词的特征向量或表征。把词映射为实数域向量的技术也叫词嵌入。我们通常在训练语言模型的同时得到词向量。**   * **了解Word2Vec词向量的特性。**   **词向量本质是将一些低维、离散、不带任何意义的序号映射成带有特定任务性质的高维特征。为了研究这些特性，可以显式地通过余弦距离来比较两个词的相似性，也可以通过词间的类比关系来探查词向量的影藏特性，另外还可以通过PCA降维的方式将词向量可视化。**  **【实验环境】：**  **python 3 + jieba + gensim + sklearn + matplotlib + NumPy + seaborn**   * **python3**   **除了高性能外，拥有NumPy、SciPy等优秀的数值计算、统计分析库。TensorFlow、Caffe等著名的深度学习框架都提供了Python接口。**   * **jieba**   **jieba是一款优秀的Python第三方中文分词库，支持三种分词模式：精确模式、全模式和搜索引擎模式。**   * **genism**   **gensim是一款开源的第三方Python工具包，用于从原始的非结构化的文本中，无监督地学习到文本隐层的向量表达。它支持包括TF-IDF，LSA，LDA，和word2vec在内的多种算法。**   * **Sklearn**   **Sklearn (全称 Scikit-Learn) 是基于 Python 语言的机器学习工具。它建立在 NumPy, SciPy, Pandas 和 Matplotlib 之上，Sklearn 里面有六大任务模块：分别是分类、回归、聚类、降维、模型选择和预处理。**   * **Matplotlib**   **Matplotlib 是 Python 的绘图库，可与 NumPy 一起使用，提供了一种有效的 MATLAB 开源替代方案。**   * **NumPy**   **NumPy是Python语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。**   * **Seaborn**   **Seaborn是基于Matplotlib的图形可视化Python包，在Matplotlib的基础上进行了更高级的封装，使得作图更加容易。**  **【参考文献】：**  **[1]:解决plt无法正常显示中文问题:** [**https://zhuanlan.zhihu.com/p/104081310**](https://zhuanlan.zhihu.com/p/104081310)  **[2]:word2vector模型详解**  **https://tedboy.github.io/nlps/generated/generated/gensim.models.Word2Vec.html**  **[3]:jieba分词工具详解**  **https://www.cnblogs.com/en-heng/p/6234006.html**  **[4]:jieba分词github源码**  **https://github.com/fxsjy/jieba**  **[5]:standford-word2vector模型**  **https://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture02-wordvecs2.pdf** |
| **实验内容** |
| **【实验方案设计】：**  1、导入实验数据集，并对实验数据集进行分词处理，并将分词结果写入新的文件“实验一数据集分词.txt”。代码如下所示：  with open('实验一数据集.txt',encoding='utf-8') as f:  document = f.read()  document\_cut = jieba.cut(document)  result = ' '.join(document\_cut)  with open('实验一数据集分词.txt', 'w',encoding="utf-8") as f2:  f2.write(result)  2、将分词处理后的文本数据作为训练数据，训练word2vector词向量模型model = Word2Vec(texts, size=100, window=5, min\_count=1, workers=4)；min\_count指定了需要训练词语的最小出现次数，默认为5；size指定了训练时词向量维度，默认为100；worker指定了完成训练过程的线程数，默认为1不使用多线程。  注意：Word2Vector模型的训练数据texts具有一定的格式需求，例子如下：  texts = [["Python", "深度学习", "机器学习"], ["NLP", "深度学习", "机器学习"]]  因此，需要用word2vec.LineSentence函数对训练数据进行预处理。  sentences = word2vec.LineSentence('实验一数据集分词.txt')  然后用处理好的数据对模型进行训练  model = word2vec.Word2Vec(sentences, window=5, min\_count=1, workers=4)  model.save('./w2v.model')  为了防止每次执行代码重新训练模型，model.save()函数保存模型文件到对应文件夹。  3、用训练好的模型进行测试，对指定的词（2个例子）进行相关性比较：  代码如下：  model.wv.similarity('中国','中华')  4、使用训练好的词向量选出与指定词（2个例子）最相似的5个词：  代码如下：  model.wv.most\_similar(positive=['武汉'], topn=5)  5、使用训练好的词向量选出与指定词类比最相似的5个词（2个例子），如**湖北 - 武汉 + 成都 = 四川**：  代码如下：  model.wv.most\_similar(positive=['湖北','成都'], negative = ['武汉'], topn = 5)  6、使用sklearn中的PCA方法对列表['江苏', '南京', '成都', '四川', '湖北', '武汉', '河南', '郑州', '甘肃', '兰州', '湖南', '长沙', '陕西', '西安', '吉林', '长春', '广东', '广州', '浙江', '杭州']（可换成其他）中的所有词的词向量进行降维并使用seaborn和matplotlib将其可视化。  代码如下：  province\_list = ['江苏', '南京', '成都', '四川', '湖北', '武汉', '河南', '郑州', '甘肃', '兰州', '湖南', '长沙', '陕西', '西安', '吉林', '长春', '广东', '广州', '浙江', '杭州']  province\_vec = model.wv[province\_list]  pca = PCA(n\_components=2)  results = pca.fit\_transform(province\_vec)  plt.scatter(x=results[:, 0], y=results[:, 1])  for i in range(len(province\_list)):  plt.annotate(province\_list[i], xy=(results[i, 0], results[i, 1]),  xytext=(results[i, 0]+0.1, results[i, 1]+0.1))  plt.show()  **【实验结果分析】：**  1、训练数据分词结果如下图所示。相较于原始文本数据，每一行的词用空格分开，标点符号与数字均被视为词用空格分开。  因为jieba.cut函数在默认条件下执行精准模式的分词，因此不会出现一个句子中，一个字同时处于两个分词中的情况。**关于jieba.cut分词模型的相关会在实验总结中给出。**    2、训练数据预处理后，训练模型，结果存放在当前文件目录下，测试模型计算单个词的词向量。  Province\_list为输入测试样例，province\_vec保存测试结果。  province\_list = ['江苏', '南京', '成都', '四川', '湖北', '武汉', '河南', '郑州', '甘肃', '兰州', '湖南', '长沙', '陕西', '西安', '吉林', '长春', '广东', '广州', '浙江', '杭州']  province\_vec = model.wv[province\_list]  输出province\_vec,则为对应测试词对应的词向量。    上述测试结果显示province\_vec为20\*100维矩阵，也应证了训练词向量模型时，定义的训练参数size =100保证训练词向量维度为100.  **注意，这里没有细细深究词向量模型维度的定义是什么，如何选取模型的维度，也同样是问题。猜测词向量维度同样是一些词，或者某些具有代表性的特征词。**  **后续思考与资料查证会在实验总结中给出。**  3、用训练好的模型进行测试，对指定的词（2个例子）进行相关性比较。  计算出‘中国’，‘中华’两个词对应词向量的相关性系数为0.46657196    4、使用训练好的词向量选出与指定词（2个例子）最相似的5个词：    **结果显示均为省会**，**说明模型训练效果比较良好。训练文本数据能够识别省会。**  5、使用训练好的词向量选出与指定词类比最相似的5个词（2个例子），如**湖北 - 武汉 + 成都 = 四川**：    **结果显示最类似的词不包含四川，但是得出的词均为省份，说明模型训练效果比较良好，但是仍然缺少准确性。原因可能是训练文本数据能够表明省会与省份的关系，但是不能表明省会与对应省份的关系。**  6、可视化显示最终的词向量降维处理的结果  中文显示异常  在最终生成词向量降维可视化结果时，发现中文标注显示异常，如下图所示    查询相关材料，找到解决方法，在执行plt.show之前，找到本机上的字体库，代码如下：（详见font\_list.py）  from matplotlib.font\_manager import FontManager  import subprocess  mpl\_fonts = set(f.name for f in FontManager().ttflist)  print('all font list get from matplotlib.font\_manager:')  for f in sorted(mpl\_fonts):  print('\t' + f)  执行这部分代码后，显示本机对应的字体库。选择导入相应的中文字体。  import matplotlib  matplotlib.rc("font",family='PingFang HK')  之后再执行plt.show().则中文显示正常，结果如下 |
| **【实验总结】：**  **1、jieba.cut函数相关**  jieba分词有三种模式：全模式、精确模式、搜索引擎模式。全模式和精确模式通过jieba.cut实现，搜索引擎模式对应cut\_for\_search，且三者均可以通过参数HMM决定是否使用新词识别功能。  官方例子如下所示：  # encoding=utf-8  import jieba  seg\_list = jieba.cut("我来到北京清华大学", cut\_all=True)  print("Full Mode: " + "/ ".join(seg\_list)) # 全模式  # 【全模式】: 我/ 来到/ 北京/ 清华/ 清华大学/ 华大/ 大学  seg\_list = jieba.cut("我来到北京清华大学", cut\_all=False)  print("Default Mode: " + "/ ".join(seg\_list)) # 精确模式  # 【精确模式】: 我/ 来到/ 北京/ 清华大学  seg\_list = jieba.cut("他来到了网易杭研大厦") # 默认是精确模式  print(", ".join(seg\_list))  # 【新词识别】：他, 来到, 了, 网易, 杭研, 大厦 (此处，“杭研”并没有在词典中，但是也被Viterbi算法识别出来了)  seg\_list = jieba.cut\_for\_search("小明硕士毕业于中国科学院计算所，后在日本京都大学深造") # 搜索引擎模式  print(", ".join(seg\_list))  # 【搜索引擎模式】： 小明, 硕士, 毕业, 于, 中国, 科学, 学院, 科学院, 中国科学院, 计算, 计算所, 后, 在, 日本, 京都, 大学, 日本京都大学, 深造  Jieba分词的算法主要通过计算所有分词结果中，概率最大的作为最终分词结果。优化算法的过程中，通过词典将句子进行分词，构建对应的DAG。例子如下图所示：  text = '我来到北京清华大学'  print(jieba.get\_DAG(text))  {0: [0],  1: [1, 2],  2: [2],  3: [3, 4],  4: [4],  5: [5, 6, 8],  6: [6, 7],  7: [7, 8],  8: [8]}  其中，5->8表示清华大学。  此时，将两点之间的权值设定为其代表词汇的词频的相关系数，则最大概率问题可以转化为求最优路径问题。Jieba用动态规划（DP）来求解最大路径问题。最终选出最佳的分词模式  2、Word2Vector中词向量模型  Word2Vector的主要作用是将语料中的单词映射到一个固定低维度的稠密向量空间中，使得相近的词在这个空间中的距离较近。  它的做法是解决一个“假”的相邻词查找任务，在解决完这个任务的同时，得到“副产物”就是所有的词向量。另一个类似的技术是Auto Encode，通过编码将原始数据压缩到一个低维的向量空间，然后通过解码将其还原为原始数据，但是最终只需要中间的隐藏向量。这个“假”任务有两种形式:Skip-Gram和CBOW。这两个形式原理比较类似，但是Skip-Gram的形式更为简单，所以主要Skip-Gram的形式下介绍相关原理。    上面是一个比较经典的例子。此时设置的window\_size为2。当两个词的词间距小于等于2时，则两词共现的频数增加1.遍历一个文本，可以得到若干这样的词对（word\_i,word\_j）。  而skip-gram算法的目的则是得到一个权重系数矩阵W，使得输入一个词wordx的独热向量v \* W ，经过softmax层后得到一个与v维度相同的向量，该向量每个值表示依据输入wordsx，输出为对应位置的词的概率。  以下为模型的结构  而我们基于成对的单词来对神经网络进行训练，训练样本是 (input word, output word ) 这样的单词对，input word和output word都是one-hot编码的向量。最终模型的输出是一个概率分布。这是一个有监督学习的过程，最终结果为对应的权重系数矩阵。  权重矩阵W与词wordx的独热向量想乘则可以最终计算出词对应的词向量。取决于独热向量的特殊性（只有一个维度为1，其余均为0），这样模型中的权重矩阵便成了一个查找表，进行矩阵计算时，直接去查输入向量中取值为1的维度下对应的那些权重值。 |
| **评语及评分（指导教师）** |
| **【评语】：**      **评分：**  **日期：** |

**附件2：**

**实验报告说明**

**1．实验名称：**要用最简练的语言反映实验的内容。

**2．实验目的**：目的要明确，要抓住重点。

**3．实验环境**：实验用的软硬件环境（配置）。

**4．实验方案设计（思路、**步骤和方法等**）**：这是实验报告极其重要的内容。包括概要设计、详细设计和核心算法说明及分析，系统开发工具等。应同时提交程序或设计电子版。

对于**设计型和综合型实验**，在上述内容基础上还应该画出流程图、设计思路和设计方法，再配以相应的文字说明。

对于**创新型实验**，还应注明其创新点、特色。

**5．实验结果分析：**即根据实验过程中所见到的现象和测得的数据，进行对比分析并做出结论（可以将部分测试结果进行截屏）。

**6．实验总结：**对本次实验的心得体会，所遇到的问题及解决方法，其他思考和建议。

**7．评语及评分：**指导教师依据学生的实际报告内容，用简练语言给出本次实验报告的评价和价值。