实验七：嵌入计算

2021 年 5 月 27 日

学　　校:	华中农业大学
学院班级:	信息学院生信 1801 班
姓　　名:	邓启东
学　　号:	2018317220103
指导教师:	夏静波

目录

1 实验目的

2019 年的 12 月 8 日开始出现不明原因的肺炎，在 2020 年的 2 月 11 日命名为新型冠状病毒肺炎。截止 2021 年 5 月 27 日，国内累计新冠肺炎确诊已达 109,016 人，全球已达一千四百万人。本次实验是想要处理 litcovid 的文献文本，使用 Pytorch 的功能实现经典的词嵌入模型 Word2Vec 得到其中高频词的嵌入向量并进行降维可视化以寻求其中的词汇语义关联。

2 研究方法

2.1 Word2vec 与 NLP

自然语言处理最基本的单位就是词语。但是词语本身比如中文、英文都是符号形式的，如果想要构建数学模型，必须要转化为数值型的输入。或者说——嵌入到一个数学空间里，这种嵌入方式，就叫词嵌入（word embedding)，而 Word2vec，就是词嵌入（word embedding) 的一种。

Word2Vec 是一种有效创建词嵌入的方法，它是从大量文本预料中以无监督方式学习语义知识的模型，这个模型为浅层双层的神经网络，用来训练以重新建构语言学之词文本。

Word2Vec 是轻量级的神经网络，其模型仅仅包括输入层，隐藏层和输出层，模型框架根据输入输出的不同，可以分为 CBOW 模型和 skip-gram 模型，CBOW 模型是通过上下文的内容预测中心词的可能情况，而 skip-gram 模型与其相反，它是通过中心词预测上下文词。

2.2 Word2Vec 词嵌入模型

2.3 词语意义表示类型

2.3.1 词典

我们想要表示一个词语，首先想到的是建立一个词典，而已经有这样一个词库 WordNet 根据不同的词性建立起词和词的关系。实现词语分类，它是一种离散表征，反应不出词汇之间的差别。

这种方式存在以下两个弊端：

1. 缺少新词的含义；

2. 并不能完全的整合所有词，即便可以，它的数量也是十分庞大的。

由于词语是符号形式的，计算机也无法理解，因此需要转换成数值形式。

2.3.2 独热编码

使用独热编码（one-hot）进行表示。One-Hot 在特征提取上属于词袋模型（bag of words）。

例如，使用 one-hot 即如果我想表示的词典有“你”，“我”，“他”这么大，那么我令每一个字都对应一个三维向量，所以：“你”→[1, 0, 0]; “我”→[0, 1, 0]；“他”→[0, 0, 1]。

独热编码有明显的缺陷：

1. 它不能表示所有的词, 即从未出现过的合成词语
2. 无法表示和学习词语之间的相关性，反映不出文字天然的内在含义，每个词都是正交的，即所有的词语点积均为 0，找不出相似词语
3. 会产生数据稀疏的问题，会浪费很多的存储空间

解决它的方法是利用派生词法 (derivational morphology)，也就是使用词根词缀的方式来避免一味增加词典的问题。但是派生词法本身也存在致命的问题，就是有的词根意义实在太多，重复的意思同样也会伤害词的表意。

2.3.3 矩阵分解

通过 SVD 或者 PCA 降维等方式，将稀疏矩阵进行浓缩，得到一个低纬度稠密的类似矩阵。能更好的精炼词向量，并减少运算量。但是，这样的方法太过暴力，所得到的词向量效果也不好。

2.3.4 语义分布表示 (distributed representation)

最早由 Hinton 提出，可以克服 one-hot representation 的上述缺点，基本思路是通过训练将每个词映射成一个固定长度的短向量，所有这些向量就构成一个词向量空间，每一个向量可视为该空间上的一个点 [1]。此时向量长度可以自由选择，与词典规模无关。这是非常大的优势。

使用密集型向量表示词语的含义，比如通过上下文，查看和它一起出现的词理解这个词语的意思。所谓的词语含义的意思是词语代表的具体事物。

在我们的假设中，词得相关性与词和词互相作为语境的出现次数有关。则在固定的语料库（corpus）（即一个超大型的句子库，作为训练的数据集）中，我们可以利用统计，得到词 A 与词 B 的概率模型，我们可以得到良好的词向量。

2.4 t-SNE

2.4.1 SNE

SNE 是通过仿射 (affinitie) 变换将数据点映射到概率分布上，主要包括两个步骤：

1. SNE 构建一个高维对象之间的概率分布，使得相似的对象有更高的概率被选择，而不相似的对象有较低的概率被选择。

2. SNE 在低维空间里在构建这些点的概率分布，使得这两个概率分布之间尽可能的相似。

我们看到 t-SNE 模型是非监督的降维，它与 kmeans 等方法不同，它不能通过训练得到一些模型参数后再适用于其它数据（比如 kmeans 可以通过训练得到 k 个点，再用于其它数据集，而 t-SNE 只能单独的对数据做操作，也就是说它只有 fit_transform，而没有 fit 操作）。

SNE 是先将欧几里得距离转换为条件概率来表达点与点之间的相似度。

2.4.2 t-SNE

尽管 SNE 提供了很好的可视化方法,但是他很难优化,而且存在”crowding problem”(拥挤问题)。后续中，Hinton 等人又提出了 t-SNE 的方法。与 SNE 不同，主要如下:

1. 使用对称版的 SNE，简化梯度公式

2. 低维空间下，使用 t 分布替代高斯分布表达两点之间的相似度

对称 SNE 实际上在高维度下另外一种减轻”拥挤问题”的方法：在高维空间下，在高维空间下我们使用高斯分布将距离转换为概率分布，在低维空间下，我们使用更加偏重长尾分布的方式来将距离转换为概率分布，使得高维度下中低等的距离在映射后能够有一个较大的距离。

对于不相似的点，用一个较小的距离会产生较大的梯度来让这些点排斥开来。这种排斥又不会无限大 (梯度中分母)，避免不相似的点距离太远。

3 数据格式

3.1 数据及输入输出

语料库：本次实验数据在 litcovid.sentence.txt 中，文件里每行都是一个有关新冠的文献句子，并且已经提前去除了标点符号，以空格分隔开。

输入是 One-Hot Vector，Hidden Layer 没有激活函数，也就是线性的单元。Output Layer 维度跟 Input Layer 的维度一样，用的是 Softmax 回归。当这个模型训练好以后，我们并不会用这个训练好的模型处理新的任务，我们真正需要的是这个模型通过训练数据所学得的参数，即我们的权重矩阵。它恰好就是我们想要得到的词嵌入。

4 代码运行

4.1 包的使用

使用 Pytorch 构造神经网络的方式来实现 Word2Vec。torch 是 python 构造神经网络的包，torch.utils.data 中的 DataLoader, Dataset 实现数据的自由读取；torch.optim 是一个实现了各种优化算法的库；matplotlib 进行画图。TSNE 进行将为可视化。我们的字典是选取前 50,000 个高频词。

4.2 报错及解决

```
05/27/2021 13:40:35 - INFO - __main__ -   epoch: 7, step: 150, loss: 7.7585
05/27/2021 13:40:44 - INFO - __main__ -   epoch: 7, step: 200, loss: 7.2324
Traceback (most recent call last):
  File "E:/Desktop/NLP/作业7/实验报告/Embedding-experiment-in-BioNLP-course-main/src/Skip_Gram_basic.py",
 line 277, in <module>
    loss.backward()
  File "D:\Anaconda3\lib\site-packages\torch\tensor.py", line 221, in backward
    torch.autograd.backward(self, gradient, retain_graph, create_graph)
  File "D:\Anaconda3\lib\site-packages\torch\autograd\__init__.py", line 130, in backward
    Variable._execution_engine.run_backward(
RuntimeError: [enforce fail at ..\c10\core\CPUAllocator.cpp:73] data. DefaultCPUAllocator: not enough
 memory: you tried to allocate 25600000 bytes. Buy new RAM!
```

图 1:

出现这个问题是因为显存不够，把其他占用显存的进程全部关闭后问题得到解决。

5 实验结果

5.1 运行时间

整个代码的运行时间从 14:00:52→15:21:23。足足运行了 1 个小时 20 分钟。想要降低时间可以增大学习率、减少训练次数、减小词汇规模大小等等方式。

Bio-BERT 则快很多，从 15:39:20→15:46:12 仅 16 分钟便输出结果 (见图2)。因为 Bio-BERT 本身是更好符合生物医药语料库的已经学好的嵌入。不需要在自己的本地进行学习。因此不论是速度还是最终呈现的效果都要比前者好很多。

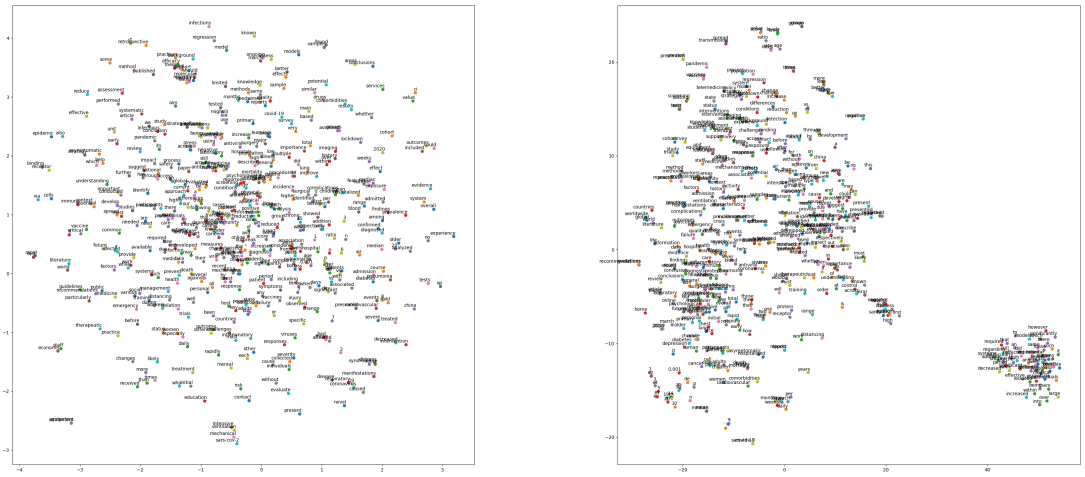
```
05/27/2021 15:45:46 - INFO - root -   Loading Embedding from ../model/
05/27/2021 15:46:07 - INFO - root -   Visualizing.
05/27/2021 15:46:12 - INFO - root -   TSNE visualization is completed,
 .bio-bert.png.

Process finished with exit code 0
```

图 2: 可视化完成图片已保存

5.2 结果对比

使用 t-SNE 进行降维可视化，变成二维空间，依旧能够保留词语之间的相似度信息（见下图3）。



(a) Skip-gram 结果　　(b) Bio-BERT 结果

图 3: AGAC 词干 wordcloud 结果

从上图对比可以看出，Bio-BERT 的聚簇效果更好，通过放大图片仔细对比发现一些有趣的相似词语靠近现象。

5.2.1 Skip-gram

使用 Word2Vec 的 Skip-gram 模型自己学习到的词语嵌入见上图，明显看到呈现一个零散的球形，没有明确的聚簇的感觉。

不过也可以窥见一些相似的地方，比如一些数字靠的比较近，虽然没有紧紧地聚在一起，但是还是可以看出来。

还有就是一些副词，以 ly 作为词根的一些词语，比如 daily,especially,rapidly,likely 等也离得比较近。

还有就是新冠肺炎疾病相关的，比如数字 19，单词 novel,cronavirus,manifestations(临床表现),disease,respiratory,syndrome 等等。

5.2.2 Bio-BERT

Bio-BERT 分为两个簇，一个簇是一些名词之类，另一簇小的更多的是一些借此还有不多的代词还有动词（见下图4）。而对于左边大簇而言，可以人工直观感受出其中的语义关联嵌入表示还是非常成功的。

表示时间的：比如 2019 和 2020 在该二维空间中几乎是完全重叠在一起的。也就是说在新冠肆虐的这两年里，年份出现的上下文非常相似。

表示时间的还有 month,weeks,daily,per。

表示器官：chest 胸口和 lung 肺的距离很近，以及 face。都表示人的一些器官部位。

再看一些代词：those,they 和 their。都表示多个人。

情态动词：should,would,must,will 也聚在一团。

还有数字：one,two,three,four 阶梯式呈现，表明嵌入计算一定程度上能够反映其语义逻辑内容。阿拉伯数字同样如此。

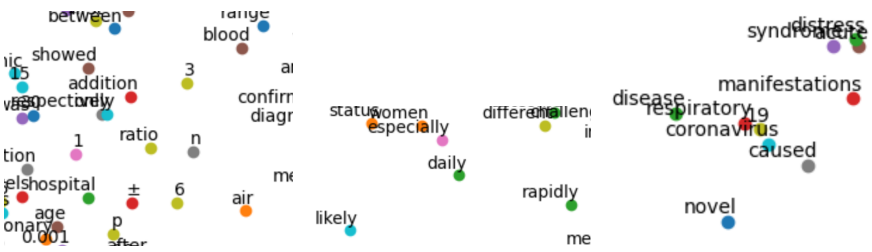
表示数值程度的：value,rate,ratio,rates,score,levels,lavel 也聚在一起。

图 4: 另一小簇

6 github 链接

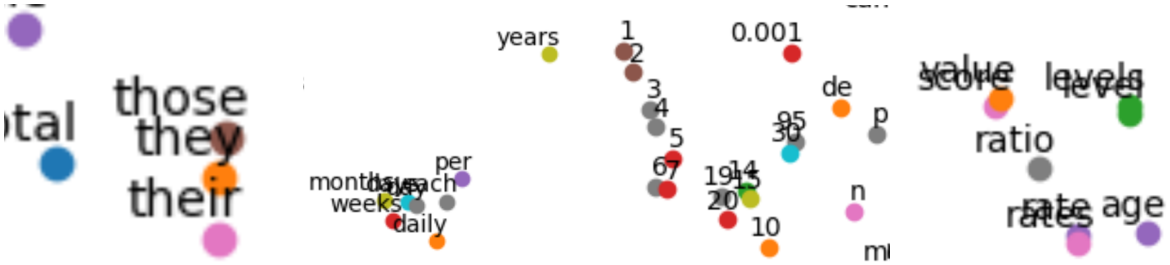
github 链接：https://github.com/LianzePuppet/nlp.homework7

7 附录



(a) 数字　　(b) 以“ly”结尾副词　　(c) 新冠肺炎相关

图 5: 上文提到 Skip-gram 得到结果的一些局部展示



(a) 代词 (b) 年份 (c) 数字 (d) 数值程度

图 6: 上文提到 Bio-BERT 得到结果的一些局部展示

8 参考链接

1. 疫情实时大数据报告
2. torch.optim
3. pytorch 实现自由的数据读取－ torch.utils.data 的学习
4. NLP（自然语言处理）：词的表示（Word representation）
5. 深度学习之词向量 Word Embedding 总结
6. 通俗理解 word2vec
7. t-SNE 原理与推导
8. RuntimeError