作业 3：利用神经语言模型训练词向量

黄家懿 ZY2303203

zy2303203@buaa.edu.cn

Abstract

使用给定语料库，利用基于 Word2Vec 的神经语言模型来训练词向量，通过计算词向量之间的语意距离、某一类词语的聚类、某些段落直接的语意关联、或者其他方法来验证词向量的有效性。

Introduction

- 词向量

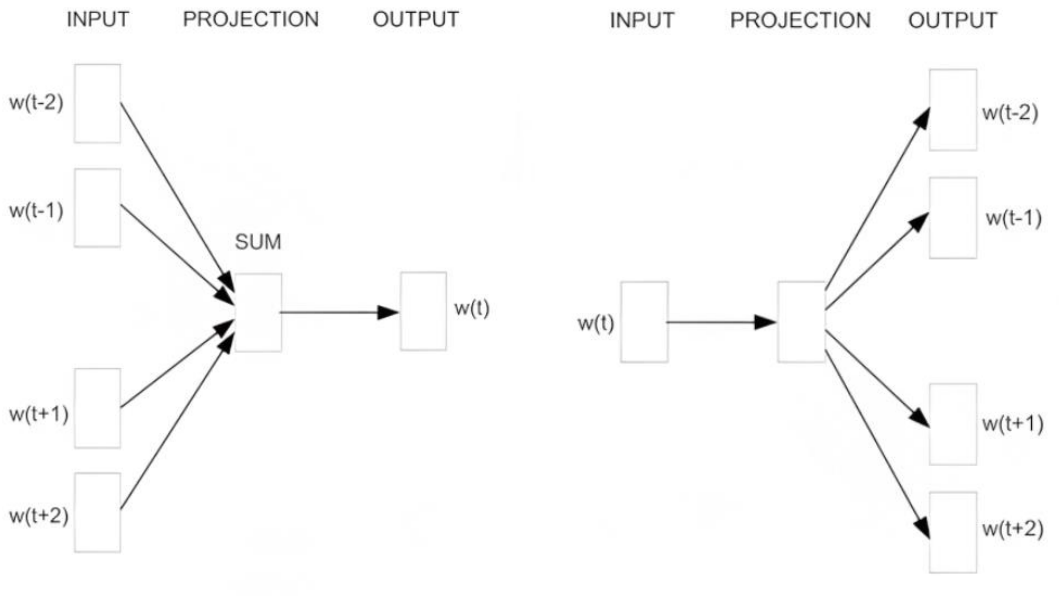
词向量是用来表示词的特征向量，将实体的词映射到实属域向量的技术即为词嵌入（word embedding）。最早的词向量模型是独热编码（One-Hot Representation），这种编码方式通过对每个词汇表中的词对应的向量位置置 1 得到词向量。这样会得到一个非常大的稀疏向量，甚至达到百万计，可能会造成维度灾难且效率不高。为了解决独热编码的问题，Distributed Representation 被提出，这种编码方式通过思路，将每个词映射到较短的词向量，所有词向量构成向量空间，因此可以通过统计学的方法来研究词与词之间的关系。

- Word2Vec

Word2Vec 的方法是最早是在 2013 年的论文《Efficient Estimation of Word Representations inVectorSpace》中提出的。主要包括 CBOW（Continuous Bag-of-Word）模型和 Skig-gram 模型，接下来分别对两种模型进行介绍。

CBOW 模型：根据上下文词汇预测当前词，即使用 w_{t-2}，w_{t-1}，w_{t+1}，w_{t+2} 去预测 w_t。

Skip-gram 模型：根据当前词预测其上下文词汇，即使用 w_t 去预测 w_{t-2}，w_{t-1}，w_{t+1}，w_{t+2} 。



1）CBOW 模型　　2）Skip-gram 模型

图 1 Word2Vec 的两种模型

- k-means 聚类算法

聚类算法是一种无监督的学习算法，主要用于将相似的样本自动归类到一个类别中。K-means 聚类算法是一种基于最小误差平方和准则的聚类算法，在众多聚类算法中，引起简单高效而备受青睐。算法步骤如下：

Step1：随机设置 K 个特征空间内的点作为初始的聚类中心；

Step2：对于其他每个点计算到 K 个中心的距离，未知的点选择最近的一个聚类中心点作为标记类别；

Step3：接着对着标记的聚类中心之后，重新计算出每个聚类的新中心点（平均值）；

Step4：如果计算得出的新中心点与原中点行样（质心不再移动），那么结束，否则重新进行第二步过程。

Methodology

M1：基于 Word2Vec 的神经网络模型

- 实验环境：

Python 3.7

PyCharm Community Edition 2023.1

gensim 4.2.0

- 实验数据：

  中文语料库：包括《白马啸西风》、《碧血剑》等小说的 txt 格式文档

  标点符号库：cn_punctuation.txt

  中文停词库：cn_stopwords.txt

- 实验代码：

Word2Vec(sentences=PathLineSentences('train_data'), hs=1, min_count=10, window=5, vector_size=200, sg=1, workers=16, epochs=10)：

- sentences 是一个语料文本的列表，PathLineSentences 函数读入文件路径，将文本处理成“一行一文本”的格式；
- hs 参数若为 1，则该模型的训练采用 hierarchica softmax，如果设置为 0(default)则使用 neqative sampling 的方式；
- min_count 参数可以调整文本处理时可忽略的词频树。在不同大小的语料集中，我们对于基准词频需求也是不一样的，譬如在较大的语料集中，我们希望忽略那些只出现两次的单词，这里设置为 5；
- window 是一个句子中当前单词和被预测单词的最大距离，滑动窗口的大小；
- vector size 为词向量的维度，这里设置为 200；
- sg 参数取值为{0，1}，其决定了模型的训练算法：1：skip-gram；0：CBOW；
- epochs:调用 word2Vec(sentences，epoches=1)会调用句子迭代器运行两次（一般来说，会运行 iter+1 次，默认情况下 iter=5）。第一次运行收集单词和它们的出现频率，从而构造一个内部字典树;第二次运行负责训练神经模型。这里设置为 10。

M2: k-means 聚类模型

- 实验环境：

  Python 3.7

  PyCharm Community Edition 2023.1

  sklearn 0.23.2

- 实验代码：

使用 sklearn 包中的 TSNE 类实现对数据特征数量的降维以便于实现可视化，KMeans 类进行 k-means 聚类。

KMeans(n_clusters=16)：n_clusters 是聚类的主题数量。

TSNE.fit_transfromer(x)：将 x 放到嵌入空间中，返回转换后的输出。

Experimental Studies

（1） 词向量之间的语义距离

词向量语义距离可以衡量两个词在语义空间中的相近程度，使用 word2vec_model.wv.similar_by_word(name, topn=10)可以获得数据集中与 name 相近程度最高的前十个词语，使用 word2vec_model.wv.similarity(name1,name2)可以比较两个词语 name1 和 name2 的语义距离。表 1，表 2 所示是分别使用 CBOW 模型和 Skig-gram 模型的实验结果。

表 1 CBOW 模型下的语义距离排序结果

CBOW 模型			
郭靖	萧峰	桃花岛	蛤蟆功
黄蓉 0.810	乔峰 0.618	嘉兴 0.420	降龙十八掌 0.517
欧阳锋 0.712	段誉 0.610	古墓 0.401	西毒 0.472
杨过 0.683	耶律洪基 0.580	回疆 0.399	掌法 0.469
欧阳克 0.678	段正淳 0.571	晋阳 0.390	心法 0.463
黄药师 0.674	游坦之 0.540	九阴真经 0.383	阴柔 0.462
洪七公 0.658	虚竹 0.527	嵩山少林寺 0.379	绝招 0.454
周伯通 0.655	慕容复 0.527	牛家村 0.378	纯阳 0.450
柯镇恶 0.603	鸠摩智 0.504	洛阳 0.377	阳刚 0.444
完颜康 0.595	木婉清 0.484	全真教 0.374	邪术 0.443
丘处机 0.578	王语嫣 0.478	恩师 0.369	一阳指 0.443

表 2 Skig-gram 模型下的语义距离排序结果

Skig-gram 模型			
郭靖	萧峰	桃花岛	蛤蟆功
黄蓉 0.828	耶律洪基 0.674	黄药师 0.562	欧阳锋 0.584
黄药师 0.714	游坦之 0.610	岛主 0.540	降龙十八掌 0.540

杨过 0.714	段誉 0.600	桃花 0.530	一阳指 0.516
欧阳锋 0.690	虚竹 0.581	岛上 0.497	西毒 0.507
洪七公 0.679	乔峰 0.568	嘉兴 0.455	内功 0.488
周伯通 0.635	契丹 0.560	黄老邪 0.449	猛劲 0.477
柯镇恶 0.594	阿紫 0.557	老毒物 0.438	纯阳 0.465
杨康 0.589	辽国 0.535	黄蓉 0.438	九阴真经 0.464
裘千仞 0.585	慕容复 0.520	九阴真经 0.436	洪七公 0.463
鲁有脚 0.582	阿朱 0.505	洪七公 0.434	修习 0.454

从实验结果可以看出，一方面，两种模型选取的与目标词相似度前十的词语都与实际认知不相冲突，说明实现了词向量语义距离的度量，验证了词向量的有效性；另一方面，两种模型在具体的距离值和选词结果上有出入，这是因为二者采用的建模思路不同导致了结果差异。

（2） 某一类词语的聚类

图 2，图 3 所示是对 16 本金庸小说，分别使用 CBOW 模型和 Skig-gram 模型，在 n_cluster=16 下的聚类结果。

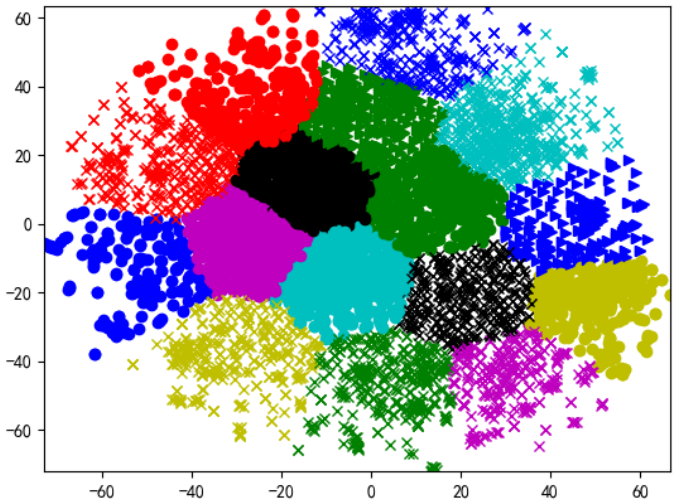


图 2 基于 CBOW 模型的聚类结果

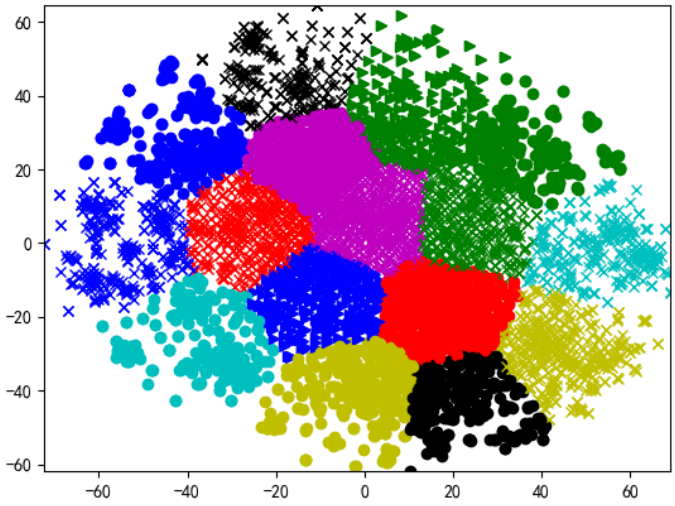


图 3 基于 Skig-gram 模型的聚类结果

从实验结果可以看出，使用 k-means 对两种模型进行聚类都有不错的效果，但是在细节的划分上有差异。

（3） 某些段落之间的语义关联

使用 word2vec_model.wv.wmdistance(test_paragraph1, test_paragraph2)可以比较两个段落 test_paragraph1 和 test_paragraph2 的相似程度，数值越大表示相似程度越低。使用的段落文本如下。表 3 所示是分别使用 CBOW 模型和 Skig-gram 模型的实验结果。

第一组：

test_paragraph1：郭杨二人见他背上负着一个包裹，甚是累赘，斗了一会，一名武官钢刀砍去，削在他包裹之上，当啷一声，包裹破裂，散出无数物事。曲三乘他欢喜大叫之际，右拐挥出，啪的一声，一名武官顶门中拐，扑地倒了。余下那人大骇，转身便逃。他脚步甚快，顷刻间奔出数丈。曲三右手往怀中一掏，跟着扬手，月光下只见一块圆盘似的黑物飞将出去，托的一下轻响，嵌入了那武官后脑。那武官惨声长叫，单刀脱手飞出，双手乱舞，仰天缓缓倒下，扭转了几下，就此不动，眼见是不活了。

test_paragraph2：郭杨二人见跛子曲三于顷刻之间连毙三人，武功之高，生平从未所见，心中都是怦怦乱跳，大气也不敢喘上一口，均想：“这人击杀命官，犯下了滔天大罪。我们若是给他发觉，只怕他要杀人灭口，我兄弟俩可万万不是敌手。”

第二组：其中，test_paragraph2 是 test_paragraph1 的改写版。

test_paragraph1：郭杨二人见他背上负着一个包裹，甚是累赘，斗了一会，一名武官钢刀砍去，削在他包裹之上，当啷一声，包裹破裂，散出无数物事。曲三乘他欢喜大叫之际，

右拐挥出，啪的一声，一名武官顶门中拐，扑地倒了。余下那人大骇，转身便逃。他脚步甚快，顷刻间奔出数丈。曲三右手往怀中一掏，跟着扬手，月光下只见一块圆盘似的黑物飞将出去，托的一下轻响，嵌入了那武官后脑。那武官惨声长叫，单刀脱手飞出，双手乱舞，仰天缓缓倒下，扭转了几下，就此不动，眼见是不活了。

test_paragraph2：郭和杨两个人看到他肩上背着一个包裹，十分累赘，和武官过招了一会，其中一名武官使用钢刀向他砍去，当啷一声，砍在了包裹上，包裹破裂，掉落出很多东西。曲三乘他欣喜大叫的时候，挥出右拐，一名武官的顶门中拐，扑地倒了。剩下那个人大惊失色，转身就逃。曲三右手往怀中一掏，顺手一扬，只见一块圆盘似的黑物飞出去，托的一下轻响，嵌入了逃跑的那位武官后脑。那位武官惨声长叫，仰天缓缓倒下，痉挛了几下，就此不动，眼见是死得彻底了。

表 3 段落之间的语义关联的实验结果

	CBOW 模型	Skig-gram 模型
第一组	0.9316516257281959	0.862317042301451
第二组	0.329662248641289	0.304278527694875

从实验结果可以看出，无论是 CBOW 模型还是 Skig-gram 模型，第一组段落的语义关联度较低，第二组段落的语义关联都较高，这与人的观测分析结果是一致的，说明两种模型都有效地估计了段落间的语义关联。

Conclusions

- 通过计算词向量的语义距离、某一类词语的聚类、某些段落之间的语义关联，验证了词向量的有效性；

References

[1] https://blog.csdn.net/v_JULY_v/article/details/102708459

[2] https://zhuanlan.zhihu.com/p/683029343