Report III of Deep Learning and Natural Langauge Processing

牛华坤
ZY2303315

Abstract

这份报告主要包括两部分内容。第一部分，从金庸小说语料库中抽取段落并进行分词，并基于 Word2Vec，GloVe 模型来训练词向量。第二部分，通过计算词向量之间的相似度、词向量类比、词向量降维后可视化来验证词向量的有效性。

Introduction

I1：神经语言模型

神经语言模型（Neural Language Model，NLM）是由 Bengio 等人在 2003 年提出的，共享一个词（及其上下文）和其他类似词和上下文之间的统计强度。模型为每个词学习的分布式表示，允许模型处理具有类似共同特征的词来实现这种共享。

神经语言模型是一类用来克服维数灾难的语言模型，它使用词的分布式表示对自然语言序列建模。不同于基于类的 n-gram 模型，神经语言模型在能够识别两个相似的词，并且不丧失将每个词编码为彼此不同的能力。神经语言模型共享一个词（及其上下文）和其他类似词。

I2：词向量与词嵌入

词向量通常指通过语言模型学习得到的词的分布式特征表示，也被称为词编码，可以非稀疏的表示大规模语料中复杂的上下文信息。为了更好的表征词向量，Hinton 于 1986 年提出了一种分布式表示(distribution representation)的方法，通俗的讲，假如你想了解某个人，你可以通过了解他身边最好关系的几个人即可推断这个人如何。分布式词向量可以表示为多维空间中的一个点，而具有多个词向量的单词在空间上表示为数个点的集合，也可以看作在一个椭球分布上采集的数个样本。

词嵌入是自然语言处理（NLP）中语言模型与表征学习技术的统称。概念上而言，它是指把一个维数为所有词的数量的高维空间嵌入到一个维数低得多的连续向量空间中，每个单词或词组被映射为实数域上的向量。词嵌入的方法包括人工神经网络、对词语同现矩阵降维、概率模型以及单词所在上下文的显式表示等。在底层输入中，使用词嵌入来表示词组的方法极大提升了 NLP 中语法分析器和文本情感分析等的效果。

Methodology

M1: Word2Vec 模型概述

Word2Vec 是 google 在 2013 年推出的一种用于生成词向量的算法，它的特点是将所有的词转化为词向量，以便于定量去度量词之间的语义关系。Word2Vec 模型包含两种主要的训练方法：CBOW（Continuous Bag of Words Model）和 Skip-gram（Continuous Skip-gram Model）。

Skip-gram 模型的基本思想是根据当前词来预测其上下文中的词。具体来说，给定一个中心词，模型会尝试预测该词前后一定范围内的词（即上下文词）。通过这种方式，模型可以学习到词语之间的共现关系，并将这些关系编码到词向量中。在训练过程中，模型会优化一个目标函数（如负采样或层次 softmax），以最小化预测错误。通过不断地调整词向量的参数，模型能够逐渐学习到词语之间的语义关系。

与 Skip-gram 模型不同，CBOW 模型是通过上下文词来预测中心词。具体来说，给定一个词的上下文（即前后一定范围内的词），模型会尝试预测该中心词本身。CBOW 模型的训练过程与 Skip-gram 类似，也是通过优化目标函数来最小化预测错误。不同的是，CBOW 模型更注重上下文信息对中心词的影响，因此它在某些任务中可能表现出不同的性能特点。

M2: GloVe 模型概述

GloVe 模型是由斯坦福教授 Manning、Socher 等人于 2014 年提出的一种词向量训练模型，该模型的提出正是在 Skip-Gram 和 CBOW 模型出来一年之后。区别于 Skip-Gram 和 CBOW，GloVe 模型的训练并没有使用神经网络，而是计算共线矩阵通过统计的方法获得。

GloVe 模型训练词向量的第一个步骤是根据语料库构建共现矩阵 X。假设语料库的大小为 N（即有 N 个单词），则共现矩阵是一个形状为(N, N)的二维向量，其中每个元素 X_{ij} 当窗口滑动遍历完整个语料库之后，即可更新得到共线矩阵 X，比如代表单词 i 和单词 j 共同出现在一个窗口(window)中的频数。

第二个步骤是构造损失函数并训练模型，以单词 i 和单词 j 共同出现在一个窗口(window)中的频数作为真实标签，以词向量的点乘作为预测值。

Glove 模型的简化版损失函数：

$$J = \sum_{i,j}^{N} (v_i^T v_j - log(X_{i,j}))^2$$

其中，v_i 和 v_j 是在初始阶段单词 i 和单词 j 的词向量（在后续的训练中会逼近真实的词向量），$X_{i,j}$ 为根据共现矩阵得到的单词 i 和单词 j 共同出现在一个窗口(window)中的频数。

向量的点乘结果类似余弦相似度，可以在一定程度上反映两个向量的相近程度，此处就是使用两个词向量的点乘来反映预测相似度，以两个单词共同出现的频数来反映实际相似度。通过对平方误差损失函数的不断训练，可以让预测相似度不断地逼近实际相似度，当预测相似度和实际相似度非常的接近时，也就得到了期望的词向量。

由于 $X_{i,j}$ 有可能为 0，即两个单词从来都没有一起出现过。则 $log^{(X_{ij})}$ 变为 log^0 是未定

义的，所以我们需要加上一个加权项 $f(x_{ij})$，即：

$$J=\sum_{i,j}^{N}f(x_{ij})(v_i^Tv_j-log(X_{i,j}))^2$$

使得当 $X_{i,j}$ 为 0 时， $f(x_{ij})$ 即跳过这一项。同时 $f(x_{ij})$ 还有一个作用是使得常出现的词如(this, is, he,I)等不会有太大的权重值，而一些很少出现的词比如(incessant,retention)等不会有太小的权重。通俗来说，就是告诫模型不要一直盯着那些一直出现的词，也要照顾一下那些很少出现的词。

最后，为了增强模型的鲁棒性，作者像在线性回归中加入偏置项一样在 GloVe 模型中也加入了偏置项，得到了最终的损失函数模型：

$$J=\sum_{i,j}^{N}f(x_{ij})[(v_i^Tv_j+bi+bj)-log(X_{i,j})]^2$$

通过训练这个损失函数使得 J 最小，我们也就得到了训练好的词向量。

Experimental Studies

E1：文本预处理

对于金庸小说构成的语料库，需要对读取的文本进行预处理：

1）删除开头无用信息。

2）删除中文停词及标点符号（利用 cn_stopwords.txt）。

3）删除空白符号，比如空格、换行符等。

4）在 jieba 分词库中添加金庸小说专有名词（利用金庸小说全人物.txt、金庸小说全武功.txt、金庸小说全门派.txt），对段落分词后用空格隔开，以段落形式保存在 corpus.txt。

E2：语言模型训练

1.Word2Vec 模型训练

1）读取 corpus.txt 为二维嵌套列表，分别以段落和分词为元素。

2）利用 gensim 中的 Word2Vec 得到训练后的模型。

3）按 CBOW 和 skip-gram 的训练方式保存模型。

```
# mode: CBOW,skip-gram
if(mode == "CBOW"):
    model = models.Word2Vec(sentences=sentences, vector_size=200, window=5, min_count=5, sg=0)
    model.save('w2v_model/CBOW.model')  # 保存模型
elif(mode == "skip-gram"):
    model = models.Word2Vec(sentences=sentences, vector_size=200, window=5, min_count=5, sg=1)
    model.save('w2v_model/skip-gram.model')  # 保存模型
```

Figure 1：保存 Word2Vec 模型

2.GloVe 模型训练

1）将 corpus.txt 放在 stanford GloVe 官方代码的文件夹下，并修改 demo.sh 文件（修改

文件源和向量维数，注释下载 text8 文件的代码）。

```
CORPUS=corpus.txt
VOCAB_FILE=vocab.txt
COOCCURRENCE_FILE=cooccurrence.bin
COOCCURRENCE_SHUF_FILE=cooccurrence.shuf.bin
BUILDDIR=build
SAVE_FILE=vectors
VERBOSE=2
MEMORY=4.0
VOCAB_MIN_COUNT=5
VECTOR_SIZE=200
MAX_ITER=15
WINDOW_SIZE=15
BINARY=2
NUM_THREADS=8
X_MAX=10
```

Figure 2：设置 GloVe 参数

2）到 linux 环境下（此处使用 Ubuntu18.04）编译并执行 demo.sh 得到 vector.txt，即词向量文件。

```
05/31/24 - 09:55.50PM, iter: 001, cost: 0.054101
05/31/24 - 09:56.21PM, iter: 002, cost: 0.044717
05/31/24 - 09:56.52PM, iter: 003, cost: 0.040873
05/31/24 - 09:57.24PM, iter: 004, cost: 0.037603
05/31/24 - 09:57.55PM, iter: 005, cost: 0.034639
05/31/24 - 09:58.26PM, iter: 006, cost: 0.031864
05/31/24 - 09:58.58PM, iter: 007, cost: 0.029386
05/31/24 - 09:59.30PM, iter: 008, cost: 0.027259
05/31/24 - 10:00.02PM, iter: 009, cost: 0.025468
05/31/24 - 10:00.34PM, iter: 010, cost: 0.024012
05/31/24 - 10:01.06PM, iter: 011, cost: 0.022795
05/31/24 - 10:01.37PM, iter: 012, cost: 0.021747
05/31/24 - 10:02.09PM, iter: 013, cost: 0.020887
05/31/24 - 10:02.40PM, iter: 014, cost: 0.020122
05/31/24 - 10:03.12PM, iter: 015, cost: 0.019482
```

Figure 3：训练 GloVe 模型

3）将词向量文件转换为 Word2Vec 格式，利用 gensim 的 KeyedVectors 模块加载并保存模型。

E3：验证词向量的有效性

验证词向量有效性主要采用内部对比分析的方式，根据已知资料验证词之间的关系。

主要分析了《鹿鼎记》和神雕三部曲四部小说中词之间的语义关系，W2v_CBOW 和 W2v_skip-gram 即运用 Word2Vec 在 CBOW 和 skip-gram 方式训练得到的模型。

1）首先已知四部小说中的主角，选取七个相关角色的词向量计算与主角词向量的余弦相似度，对比不同模型下的余弦相似度如下。

Table 1 相似度测试

主角或模型	角色 1	角色 2	角色 3	角色 4	角色 5	角色 6	角色 7
韦小宝	双儿	阿珂	沐剑屏	方怡	曾柔	建宁公主	苏荃
W2v_CBOW	0.524	0.430	0.514	0.499	0.331	0.448	0.449
W2v_skip-gram	0.451	0.332	0.386	0.427	0.343	0.483	0.395
GloVe	0.487	0.423	0.399	0.434	0.291	0.231	0.308
郭靖	黄蓉	洪七公	黄药师	周伯通	梅超风	欧阳克	丘处机
W2v_CBOW	0.670	0.575	0.548	0.580	0.540	0.571	0.469

W2v_skip-gram	0.572	0.431	0.455	0.339	0.287	0.334	0.328
GloVe	0.710	0.551	0.528	0.495	0.420	0.441	0.443
杨过	小龙女	黄蓉	李莫愁	金轮法王	欧阳锋	公孙止	陆无双
W2v_CBOW	0.831	0.823	0.755	0.535	0.681	0.615	0.686
W2v_skip-gram	0.656	0.428	0.465	0.553	0.411	0.484	0.407
GloVe	0.754	0.497	0.536	0.478	0.386	0.410	0.539
张无忌	赵敏	张翠山	殷素素	谢逊	周芷若	小昭	成昆
W2v_CBOW	0.788	0.823	0.685	0.655	0.760	0.551	0.461
W2v_skip-gram	0.629	0.411	0.421	0.513	0.597	0.488	0.328
GloVe	0.714	0.299	0.294	0.489	0.640	0.510	0.274

2）然后选择七个与主角的词向量最邻近的词向量，即余弦相似度最大的七个词向量如下。

Table 2 最邻近测试

主角或模型	邻近词 1	邻近词 2	邻近词 3	邻近词 4	邻近词 5	邻近词 6	邻近词 7
韦小宝	康熙	袁承志	陈家洛	乾隆	公主	张无忌	费要多罗
W2v_CBOW	0.721	0.684	0.682	0.680	0.656	0.651	0.641
韦小宝	索额图	费要多罗	施琅	康熙	多隆	罗刹	佟国纲
W2v_skip-gram	0.642	0.622	0.619	0.608	0.602	0.572	0.571
韦小宝	康熙	目录	皇上	海老公	笑	双儿	问道
GloVe	0.571	0.497	0.489	0.489	0.488	0.487	0.485
郭靖	杨过	张无忌	段誉	小龙女	郭襄	令狐冲	胡斐
W2v_CBOW	0.769	0.731	0.723	0.708	0.705	0.686	0.686
郭靖	黄蓉	蓉儿	华筝	武三通	靖	金轮法王	欧阳锋
W2v_skip-gram	0.572	0.499	0.473	0.466	0.465	0.461	0.458
郭靖	黄蓉	洪七公	欧阳锋	黄药师	目录	周伯通	杨过
GloVe	0.710	0.551	0.548	0.528	0.518	0.495	0.478
杨过	段誉	郭襄	小龙女	黄蓉	胡斐	石破天	张无忌
W2v_CBOW	0.845	0.840	0.831	0.823	0.822	0.782	0.782
杨过	小龙女	郭襄	金轮法王	法王	神雕	郭芙	武三通
W2v_skip-gram	0.656	0.650	0.553	0.532	0.516	0.489	0.488
杨过	小龙女	第二十六回	麽	郭芙	陆无双	李莫愁	郭襄
GloVe	0.754	0.671	0.559	0.553	0.539	0.536	0.527
张无忌	令狐冲	虚竹	张翠山	胡斐	段誉	慕容复	赵敏
W2v_CBOW	0.890	0.846	0.823	0.817	0.809	0.807	0.788
张无忌	赵敏	周芷若	盈盈	蛛儿	杨逍	谢逊	殷天正
W2v_skip-gram	0.629	0.597	0.526	0.519	0.518	0.513	0.489
张无忌	赵敏	周芷若	小昭	杨逍	谢逊	义父	赵姑娘
GloVe	0.714	0.640	0.510	0.508	0.489	0.474	0.462

3）最后选择三个问题做词向量类比。

问题一：杨过之于小龙女，相当于郭靖之于？

问题二：降龙十八掌之于洪七公，相当于蛤蟆功之于？

问题三：武当派之于张翠山，相当于丐帮之于？

Table 3 词类比测试

问题或模型	角色 1	角色 2	角色 3	角色 4	角色 5	角色 6	角色 7
问题一	石清	穆念慈	慕容复	林平之	郭襄	王语嫣	俞岱岩
W2v_CBOW	0.651	0.638	0.628	0.624	0.623	0.622	0.622
问题一	华筝	蓉儿	韩小莹	过儿	穆念慈	终南山	夫妇俩
W2v_skip-gram	0.475	0.431	0.429	0.422	0.420	0.404	0.401
问题一	黄蓉	靖哥哥	周伯通	欧阳锋	丘处机	黄药师	手
GloVe	0.587	0.489	0.473	0.453	0.445	0.430	0.419
问题二	周伯通	欧阳锋	乌老大	黑白子	石破天	袁紫衣	丁典
W2v_CBOW	0.900	0.870	0.844	0.828	0.827	0.812	0.811
问题二	欧阳锋	西毒	老毒物	周伯通	朱长龄	走火	闭气
W2v_skip-gram	0.747	0.618	0.603	0.602	0.599	0.599	0.599
问题二	欧阳锋	西毒	周伯通	口述	老毒物	九阴真经	侄儿
GloVe	0.686	0.463	0.452	0.424	0.418	0.401	0.384
问题三	萧峰	俞岱岩	完颜洪烈	包惜弱	马春花	杨康	陈家洛
W2v_CBOW	0.789	0.754	0.753	0.741	0.738	0.706	0.703
问题三	乔峰	殷素素	萧峰	帮主	阿朱	徐长老	韩林儿
W2v_skip-gram	0.531	0.506	0.504	0.452	0.439	0.417	0.413
问题三	殷素素	乔峰	萧峰	谢逊	帮主	阿朱	帮众
GloVe	0.468	0.457	0.436	0.420	0.393	0.388	0.386

4）综合评判后发现 GloVe 模型表现最好，利用 TSNE 模块将词向量转换到二维并可视化，对象是 50 个角色名，可以发现相同小说的聚类效应，放大可以发现《射雕英雄传》的角色词向量聚类和《书剑恩仇录》的角色词向量聚类，类似还有如杨过和小龙女，张无忌、赵敏和周芷若等。

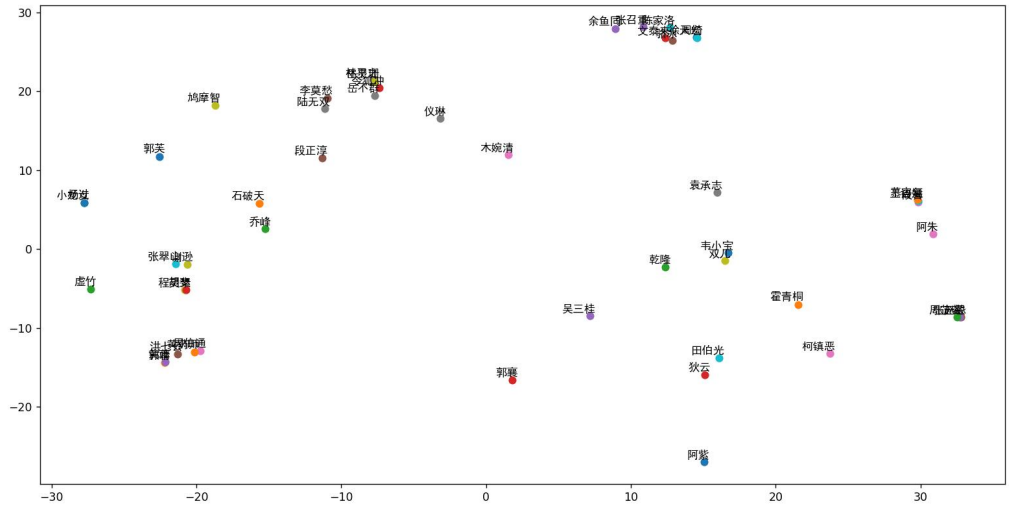

Figure 4 50 个角色词向量可视化

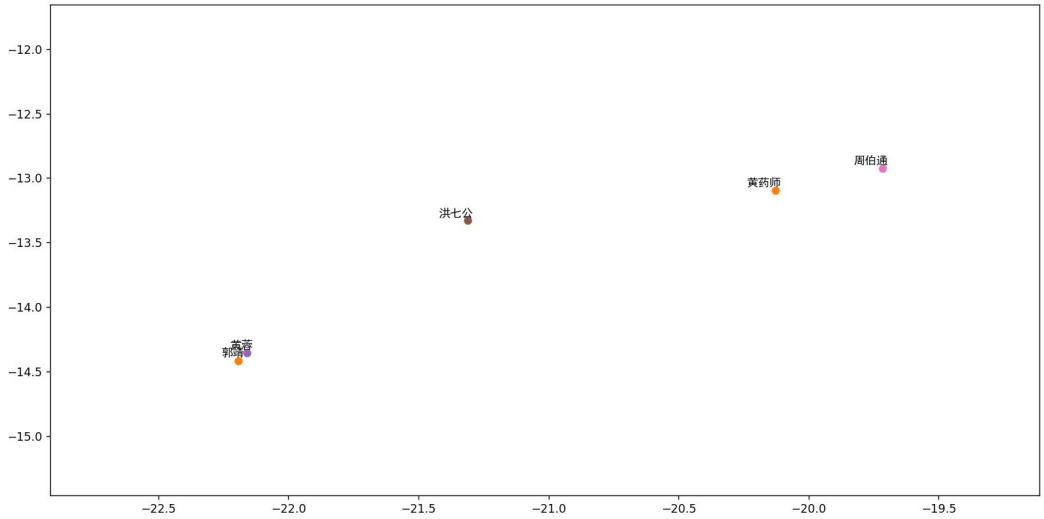

Figure 5 《射雕英雄传》角色词向量聚类

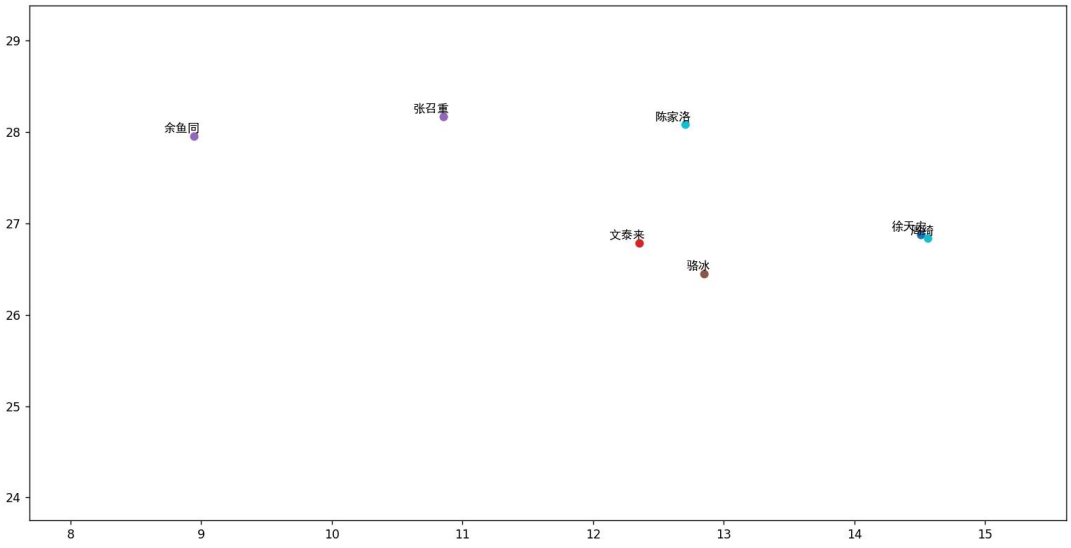

Figure 6 《书剑恩仇录》角色词向量聚类

Conclusions

C1: 词向量有效性分析

从 Table1 中可以看出，三种模型均能较好反映出人物的之间的相近关系。

从 Table2 中可以看出，Word2Vec 模型的 CBOW 和 skip-gram 方法有着明显的差别，直观上可以观察到 CBOW 中与主角相似的角色有很大其他小说的主角，这可能是由于各个主角在一些维度上有着相似的特征，比如出现频率高，说明 CBOW 更注重上下文对中心词的影响，而 skip-gram 由于更注重中心词对其他词的影响，可以看到描写时仅与主角联系紧密的角色容易有更高的相似度，容易使得高曝光词汇权重过高；GloVe 模型的表现则显著由于 Word2Vec 模型，其出现的相似度最高的角色多与主角有着紧密的社会关系，由于基于共现矩阵和窗口滑动构造词向量的方法，GloVe 能更好捕捉语法的信息。

从 Table3 中可以看出，Word2Vec 模型的 CBOW 方法还是很容易定位到其他小说中，skip-gram 方法定位在同一小说但是容易“认错人”，而且很快发散到其他小说中，GloVe 模型则表现出较好的稳定性和准确性，可以较好描述词向量的类比关系。

最后的可视化同样是 GloVe 的表现最好，一方面将相同小说的角色聚类在一起，另一方面也拉开了和其他小说的距离，不足之处在于对于郭襄这种在多部小说中出现过的角色，可能会出现误判。

References

[1] https://blog.csdn.net/weixin_45069761/article/details/107859872

[2] https://zhuanlan.zhihu.com/p/58916233

[3] https://zhuanlan.zhihu.com/p/129295944

[4] https://zhuanlan.zhihu.com/p/52705512

[5] https://blog.csdn.net/weixin_43819408/article/details/113608380