Report of Homework Three

张海尼
ZY2303215

Abstract

报告中主要研究了使用神经语言模型训练词向量的方法，在给定语料库（金庸小说集）上，使用 Word2Vec 和 GloVe 两种神经语言模型训练词向量，并通过词汇聚类，词向量之间语意距离对比，以及 T-SNE 词向量降维可视化等方法验证词向量的有效性。

Introduction

I1: 神经语言模型

为了描述每个词汇的特征，Hinton 在 1986 年提出了使用词向量来表示词，即抽取一组数据特征形成一组向量，用向量来表示词汇并计算词汇相互之间的关系。

神经语言模型是自然语言处理中典型方法，其主要功能是根据当前单词进行相关单词的预测。神经语言模型的输出为其所知单词的概率评分，通常使用百分比表示。模型在经过训练后会生成表示所有单词的映射矩阵，在预测中，需要再该映射矩阵中查询输入单词，然后计算对应的预测值。

Methodology

M1: Word2Vec

Word2Vec 是 google 于 2013 年发表的计算词向量的算法，其将能够将词汇转换为词向量，并在此基础上定量地度量不同词汇之间的关系。Word2Vec 包括两种训练模式，CBOW（Continuous Bag-of-Words Model）和 Skip-gram（Continuous Skip-gram Model）。

CBOW 使用提供上下文对目标词进行预测的方法学习词向量的表达，即对缺失目标词的词袋模型乘以 embedding 矩阵，以得到连续的 embedding 向量。

Skip-gram 使用提供目标词对上下文进行预测的方法学习词向量的表达，即给定中心词 w，计算生成背景词 c 的条件概率 $p(c \mid w)$。假定给定中心词时背景词相互独立，则其目标函数为：

$$l(\theta) = \arg\max_{\theta} \prod_{w \in Text} \prod_{C \in Context(w)} P(c \mid w; \theta)$$

取对数后为：

$$L(\theta) = \arg\max_{\theta} \sum_{w \in Text} \sum_{C \in Context(w)} \log P(c \mid w; \theta)$$

其中，$\theta = [u, v]$，u 为上下文词向量，v 为中心词词向量。

由于 Skip-gram 模型中上下文为词库，因此计算的时间复杂度很高。为解决该问题，Word2Vec 使用了两种优化算法，负采样与层次 Softmax，其核心思想均为降低模型训练过程的计算量。负采样通过计算中心词与背景词不同时出现的概率降低计算量；层次 Softmax 使用 Huffman 树代替从隐藏层到输出层 softmax 的映射，使用二元逻辑回归的方式进行判别，将计算量从 V 降低到 log2V，且高频词靠近根节点，高频词的查询时间更短。

M2: GloVe

GloVe 是 Jeffrey Pennington 等人与 2014 年提出的神经网络模型，该方法基于全局词汇共现的统计信息训练词向量，将统计信息与局部上下文窗口的优点相结合，提高了词向量对全局语义关系的理解。

GloVe 使用词汇共现矩阵提取语义关系。词汇共现矩阵是描述上下文两个词汇间共现的方阵，即描述两个单词在同一句话（同一个窗口）中的共现次数，该矩阵反映了词汇间的共生关系。模型使用 k-skip-n-gram 方法建立词汇共现矩阵，将上下文范围扩大到 k+n 大小的滑动窗口求取共现矩阵。

$P_{ij} = P(j \mid i) = X_{ij} / X_i$ 表示词汇 j 出现在词汇 i 上下文的概率，其中，$X_i = \sum_k X_{ik}$ 表示在词汇 i 的上下文中所有出现的单词的次数之和，X_{ij} 表示词汇 j 出现在词汇 i 上下文中的次数之和。

模型训练的目标是得到描述词向量与词汇共现概率间关系的函数 F，即：

$$F(w_i, w_j, \tilde{w}_k) = P_{ik} / P_{jk}$$

其中，$w_i, w_j, \tilde{w}_k$ 为词汇 i，j，k 对应的词向量。

考虑到不同共现词汇具有不同的权重，模型训练的损失函数定义如下：

$$J = \sum_{i,k=1}^{V} f(X_{ik})(w_i^T \tilde{w}_k + b_i + b_k - \log X_{ik})^2$$

其中，$f(X_{ik})$ 表示词汇的权重，该函数由词汇共现统计信息确定。

GloVe 模型根据词汇共现概率建模，相较于仅依赖局部上下文关系的 Word2Vec 模型，其能够更好地捕捉全局语义关系，但其训练过程依赖高质量的共现矩阵，在小型或特定数据集上可能表现不佳。此外，GloVe 模型与大多数词嵌入模型相同，无法捕捉词汇间的顺序关系与语法结构。

Experimental Studies

E1：语料库处理

1.1 语料库处理算法流程

1) 所用数据库为 16 部金庸小说；
2) 删除部分文件开头的无用信息；
3) 使用 jieba 算法库进行分词；
4) 根据 cn_stopwords.txt 删除中文停词；
5) 删除无意义符号，包括标点符号和空白符号等；
6) 对语料库整体以 100 个词汇为单位进行分句，并保存为 txt 文件，其中句子间用\n 分割，词汇间用空格分割。

1.2 部分代码

```
words = []
sent_len = 100
for name in names:
    filePath = dir + name + ".pickle"
    with open(filePath, 'rb') as f:
        data = pickle.load(f)

    # 分句
    words_len = len(data)
    sent_num = math.ceil(words_len/sent_len)
    for i in range(sent_num):
        if i == sent_num-1:
            tmp = data[i*sent_len:-1]
        else:
            tmp = data[i*sent_len:(i+1)*sent_len]
        words.append(tmp)

with open('NLPmodel/data.txt', 'w', encoding='utf-8') as f:
    data_str = '\n'.join([' '.join(row) for row in words])
    f.write(data_str)
```

E2：神经语言模型训练

2.1 训练 Word2Vec 模型流程

1) 所用语料库从 16 部金庸小说中提取；
2) 使用 gensim 提供的 word2vec 训练语言模型；
3) 保存语言模型。

2.2 训练 GloVe 模型流程

1) 所用语料库从 16 部金庸小说中提取；
2) 使用 stanford 提供的 GloVe 模型训练词向量并保存；

3) 使用 gensim 读取词向量构建语言模型；
4) 保存语言模型。

2.3 部分代码

```
# 训练
sentences = list(word2vec.LineSentence('NLPmodel/data.txt'))

model = Word2Vec(sentences, min_count=5, window=5,
                 sg=0, workers=multiprocessing.cpu_count())
model.save('NLPmodel/word2vec.model')

# 读取模型
model = Word2Vec.load('NLPmodel/word2vec.model')
# print(model)
```

E3：验证词向量有效性

3.1 验证词向量有效性算法流程

1) 计算不同词汇的词汇聚类，以前 10 个为例；
2) 计算不同类型词汇间的语义距离；
3) 使用 t-SNE 对词向量降维，通过可视化方式说明词向量的有效性，以前 100 个词汇为例。

3.2 部分代码

```
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号

vector = model.wv.vectors[:100]
label = model.wv.index_to_key[:100]

tsne = TSNE(n_components=2)
vector2 = tsne.fit_transform(vector)

x_vals = [v[0] for v in vector2]
y_vals = [v[1] for v in vector2]  # 创建一个 trace
trace = go.Scatter(x=x_vals, y=y_vals, mode='text', text=label)
data = [trace]

plot(data, filename='word-embedding-plot.html')
```

3.2 实验结果

Table 1：词汇聚类分析结果

词汇	Word2Vec	GloVe
杨过	('小龙女', 0.8346632122993469) ('石破天', 0.8295251131057739) ('胡斐', 0.8267642855644226) ('张无忌', 0.8179601430892944) ('黄蓉', 0.8120946884155273) ('周芷若', 0.7989614605903625) ('石清', 0.7982308864593506) ('郭襄', 0.7946046590805054) ('张翠山', 0.78195720911026) ('苗人凤', 0.7812521457672119)	('小龙女', 0.9017375111579895) ('李莫愁', 0.8260535001754761) ('麼', 0.8132018446922302) ('後', 0.7978254556655884) ('陆无双', 0.7806150317192078) ('著', 0.7775416374206543) ('黄蓉', 0.7715674042701721) ('於', 0.7513590455055237) ('郭靖', 0.7477588653564453) ('郭', 0.7441518306732178)
剑法	('刀法', 0.8979476094245911) ('拳法', 0.8728920221328735) ('掌法', 0.8574995994567871) ('剑术', 0.8548632264137268) ('招数', 0.8521313071250916) ('招式', 0.8356708884239197) ('功夫', 0.8276431560516357) ('几招', 0.8206008076667786) ('内功', 0.812402069568634) ('棒法', 0.7990604639053345)	('精妙', 0.7970240712165833) ('剑术', 0.7679197788238525) ('剑', 0.7676780819892883) ('剑招', 0.7495372891426086) ('冲灵', 0.7384160161018372) ('招数', 0.7188246846199036) ('这套', 0.700410008430481) ('精妙绝伦', 0.696157693862915) ('辟邪', 0.6921288967132568) ('华山派', 0.6895010471343994)
客栈	('城西', 0.9522685408592224) ('南门', 0.9514778852462769) ('投店', 0.9492562413215637) ('店铺', 0.9486979246139526) ('十里', 0.9450049996376038) ('杨柳', 0.9447057247161865) ('房子', 0.9446754455566406) ('风雪', 0.9432793259620667) ('一所', 0.9418918490409851) ('大户人家', 0.941627562046051)	('南安', 0.7805456519126892) ('城西', 0.7674578428268433) ('仙安', 0.7641555070877075) ('高升', 0.7635140419006348) ('镇甸', 0.7401076555252075) ('走廊', 0.7119114398956299) ('爬出来', 0.7105628252029419) ('岘', 0.704447329044342) ('羊太傅', 0.6750568151473999) ('村庄', 0.6745142936706543)

Table 2：语义距离分析结果

输入词汇	Word2Vec	GloVe
（杨过，小龙女）	0.825825	0.9017375
（杨过，东方）	-0.050278	-0.0815114
（华山派，弟子）	0.709729	0.68419206
（华山派，剑法）	0.677864	0.68950105

Figure 1：Word2Vec 模型降维结果

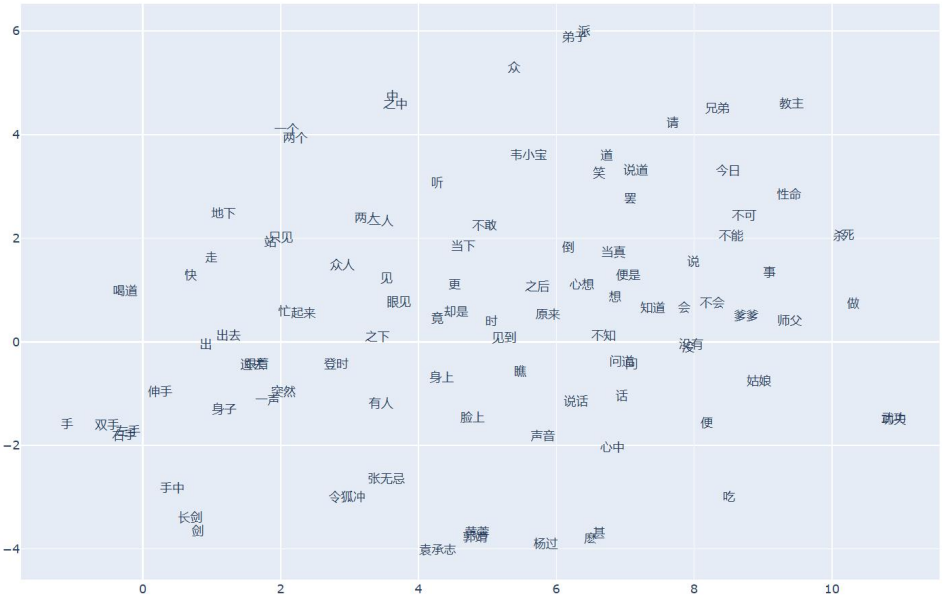

Figure 2：GloVe 模型降维结果

Conclusions

C1：词向量有效性分析

从 Table 1 中可以看出，两种语言模型都能够实现相关词汇聚类的功能，但两种模型的侧重点不同。Word2Vec 模型根据上下文词汇进行训练，例如，输入词汇为“剑法”时，得到的排名前 10 的词汇大部分都是武功招式；GloVe 模型使用共现矩阵了解语义关系，在全

局语义关系上表现较好，例如，输入词汇为“剑法”时，得到的排名前 10 的词汇大部分与“剑法”本身相关，包括对其的形容词、量词，以及相关的门派等。

从 Table 2 中可以看出，两种语言模型在判断输入词汇的语义距离时都具有较好的性能，但 Word2Vec 模型的效果更好一些。例如，输入为（华山派，弟子）和（华山派，剑法）时，Word2Vec 判断前一个组合的相似度更高，这是因为该模型更注重词汇的上下文关系；而 GloVe 模型输出的相似度基本相同，因为该模型跟注重全局语义表现关系。

从 Figure 1 和 Figure 2 中可以很直观地看出，在两个模型训练出的词向量中，关联词汇基本聚集在同一区域，例如“左手”、“右手”、“伸手”和“师父”、“弟子”、“派”等，说明了词向量的有效性。但两个模型的聚类结果都不是非常聚拢，整体词汇的分布较为分散，可能是因为语料库规模有限，模型训练不充分的缘故。

References

[1] https://zhuanlan.zhihu.com/p/170292703
[2] https://blog.csdn.net/weixin_44649780/article/details/127365081