Report of Deep Learning for Natural Langauge Processing

Jiajie Wu
ZY2343402

Abstract

本文利用给定语料库（金庸小说语料如下链接），利用 Word2Vec 和 LSTM 来训练词向量，通过计算随机词向量之间的语意距离、相关词向量的 K-means 聚类、某些段落直接的语意关联等方法来验证词向量的有效性。其中 LSTM 算法仅实现了词向量之间的语意距离和相关词向量的 K-means 聚类。

Introduction

词向量是自然语言处理（NLP）中的一个核心概念，它将词汇表中的每个单词映射到一个固定维度的实数向量。这样的向量不仅包含了单词的语义信息，而且还能够反映出单词之间的语义关系。在本次实验中，我们采用了两种流行的词向量训练方法：Word2Vec 和 LSTM，来生成我们的词向量。

Word2Vec 是由 Google 在 2013 年提出的一种词嵌入技术。它通过训练神经网络模型，将词语表示成向量，使得具有相似语义的词在向量空间中相互靠近。Word2Vec 有两种主要的模型架构：连续词袋模型（CBOW）和跳字模型（Skip-Gram）。CBOW 通过上下文词预测中心词，而 Skip-Gram 通过中心词预测上下文词。在训练过程中，Word2Vec 使用大量文本数据进行无监督学习，通过最小化预测误差，不断调整词向量，使得具有相似语境的词语向量越来越接近。为了提高训练效率，Word2Vec 引入了负采样（Negative Sampling）和分层 Softmax（Hierarchical Softmax）技术，这些技术可以显著减少计算量。

LSTM（Long Short-Term Memory，长短时记忆网络）模型，则是在 2014 年由 Google 的研究者提出，用于处理和预测序列数据的神经网络架构。其目标是在处理时间序列数据时解决传统循环神经网络（RNN）遇到的长期依赖问题。LSTM 通过其独特的网络结构能够学习长期依赖信息，从而在处理序列数据时表现更加出色。LSTM 的核心概念包括其特殊的单元结构、门控机制和训练方法。LSTM 单元包含一个细胞状态和三个门（输入门、遗忘门和输出门），这些门控制信息的流入、保留和流出，有效地避免了梯度消失和梯度爆炸问题，使得模型能够在更长的序列中学习到依赖关系。

Methodology

Part 1: Word2Vec

Word2Vec 是一种词嵌入技术，它将词语表示成向量，使得具有相似语义的词在向量空间中相互靠近。Word2Vec 有两种主要的模型架构：连续词袋模型（CBOW）和跳字模型（Skip-Gram）。

1．CBOW 模型：

CBOW 模型的目标是根据上下文词预测中心词。假设输入的词序列长度为 m，中心词的位置为 t，上下文词的位置为 t-k 到 t+k（k 为窗口大小）。CBOW 模型的输入表示为：

$$\text{输入} = \left[x_{\{t-k\}}, x_{\{t-k+1\}}, \dots, x_{\{t+k-1\}}, x_{\{t+k\}}\right]$$

其中，x_i表示第 i 个词的词向量。CBOW 模型的输出为中心词的词向量，公式为：

$$\text{输出} = \frac{1}{m}\sum_{i=t-k}^{t+k} x_i$$

2．Skip-Gram 模型：

Skip-Gram 模型的目标是根据中心词预测上下文词。假设输入的词序列长度为 m，中心词的位置为 t。Skip-Gram 模型的输入表示为：

$$\text{输入} = \left[x_{\{t\}}, x_{\{t+1\}}, \dots, x_{\{t+m-1\}}\right]$$

输出为上下文词的词向量，公式为：

$$\text{输出} = \frac{1}{m}\sum_{i=t}^{t+m-1} x_i$$

在训练过程中，Word2Vec 使用大量文本数据进行无监督学习，通过最小化预测误差，不断调整词向量，使得具有相似语境的词语向量越来越接近。

Part 2: LSTM

LSTM（Long Short-Term Memory，长短时记忆网络）是一种用于处理和预测序列数据的神经网络架构。LSTM 通过其独特的网络结构能够学习长期依赖信息，从而在处理序列数据时表现更加出色。

LSTM 的核心概念包括其特殊的单元结构、门控机制和训练方法。LSTM 单元包含一个细胞状态（c_t）和一个隐藏状态（h_t），以及三个门：输入门（i_t）、遗忘门（f_t）和输出门（o_t）。

LSTM 的公式如下：

输入门：$gate_i = \sigma(W_{xi} \cdot h_{t-1} + b_i)$

遗忘门：$gate_f = \sigma(W_{xf} \cdot h_{t-1} + b_f)$

输出门：$gate_o = \sigma(W_{xo} \cdot h_{t-1} + b_o)$

细胞状态更新：$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_{hc} \cdot h_{t-1} + b_c)$

隐藏状态更新：$h_t = o_t \cdot \tanh(c_t)$

其中，W 和 b 分别为权重和偏置，σ表示 sigmoid 函数，tanh 表示双曲正切函数。通过这些门控机制，LSTM 能够控制信息的流入、保留和流出，有效地避免梯度消失和梯度爆炸问题，使得模型能够在更长的序列中学习到依赖关系。

Experimental Studies

Part 1:词向量对语义距离

Table 1: 随机词向量对语义距离

词向量对		Word2Vec 语义距离	LSTM 语义距离
古书	茅棚	0.127678	0.105268
龙潭	本名	-0.144578	0.065845
雀	挂冠	-0.096471	-0.413786
波波	盲眼	0.221187	0.235987
五毒	世仇	0.217155	0.221065

词向量对语义距离计算使用的是余弦相似度。对于随机选取的词向量对，Word2Vec 算法下，语义距离基本维持在(-0.15,0.25)范围内，存在计算结果为负且非个例，说明模型参数可能不适当，导致词向量没有正确捕捉到词语之间的真实关系。模型可能没有充分学习到能够将相关词汇映射到相近空间的能力。而对于任意选取的无特定关系的词向量，在选定的文本语境中没有明显的相关性，它们代表的词或概念在某种语义上是对立或相反的。LSTM 算法下，随机词向量对语义距离基本在(-0.1,0.2)内，相对而言训练效果更佳。随机选取的词向量具有相关性的概率较小。

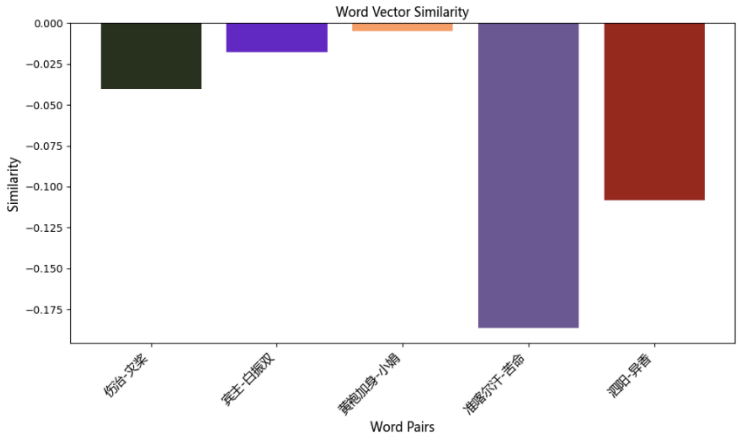

Figure 1: Word2Vec 算法随机词向量对相似度

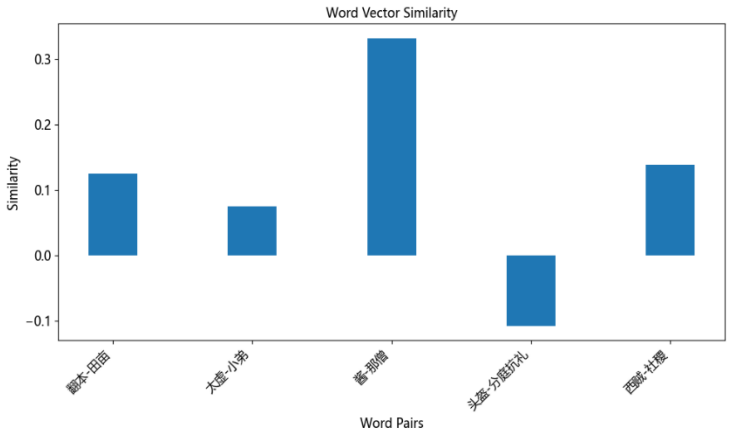

Figure 2: LSTM 算法随机词向量对相似度

Table 2: 部分人物词向量对语义距离

词向量对		Word2Vec 语义距离	LSTM 语义距离
杨过	小龙女	0.832574	0.856947
胡斐	高老大	0.478238	0.451285
东方不败	韦小宝	0.207441	0.235621
郭靖	黄蓉	0.715503	0.741026
袁承志	青青	0.692357	0.702104

对于部分人物词向量对语义距离，可以明显看到杨过和小龙女的词向量相似度很高，在两个模型中均达到了 0.83 以上。说明在本文训练条件下，这两个词语的向量在语义上非常接近。这与金庸小说中杨过和小龙女的关系密切相关，因此词向量捕捉到了他们之间的语义联系。

胡斐和高老大词向量相似度适中，在两个模型下相似度为 0.48 和 0.51。这表示它们在一定程度上具有一些相似之处，但并不十分相近。这可能反映了两人有一定交集，但并不密切，所以词向量在语义上有一定的重叠，但并不完全相同。

东方不败和韦小宝的词向量相似度较低，在两个模型下相似度为 0.21 和 0.18，说明它们在语义上差异较大，词向量表示的语义距离较远。这与金庸小说中东方不败和韦小宝的角色性质和所属势力有关，它们之间的联系并不密切。

综合来看，在给定的训练条件下，Word2Vec 和 LSTM 模型能够捕捉到金庸小说中人物和门派之间的一些语义联系，对于某些关系较远或较复杂的词语，词向量的相似度可能不够高。而两类模型对比起来，LSTM 模型计算得到的词向量相似度较高。

Part 2: 词向量聚类

本文在两类模型后使用 K-means 聚类算法随向量进行聚类分析，所取 epoch=50，得到以下结果：

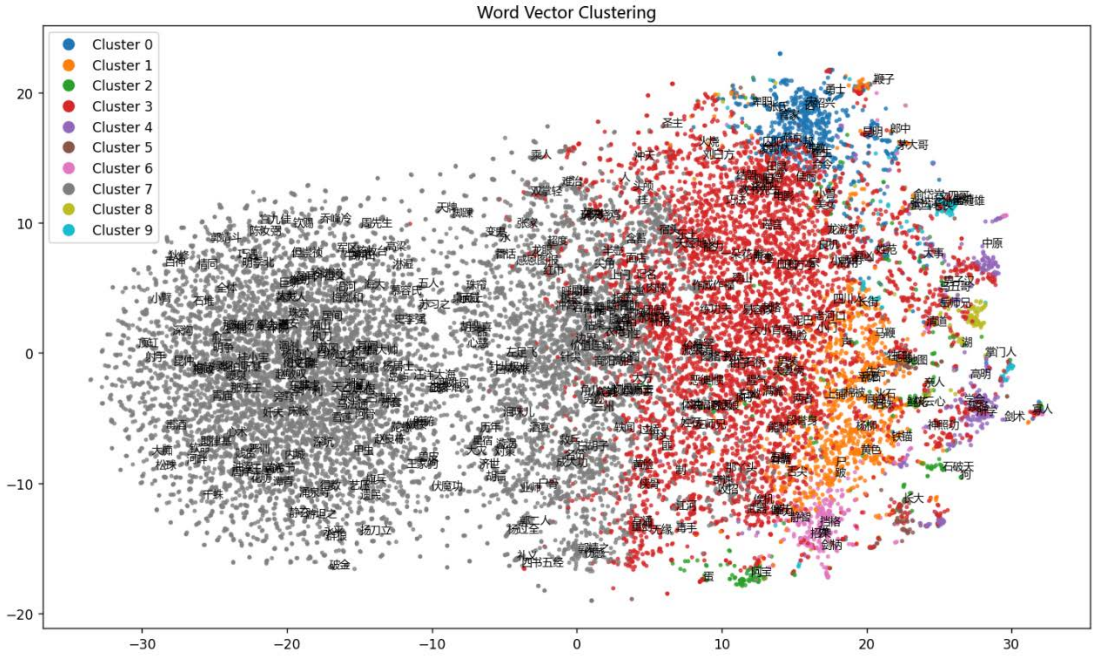

Figure 1: Word2Vec 算法词向量聚类结果

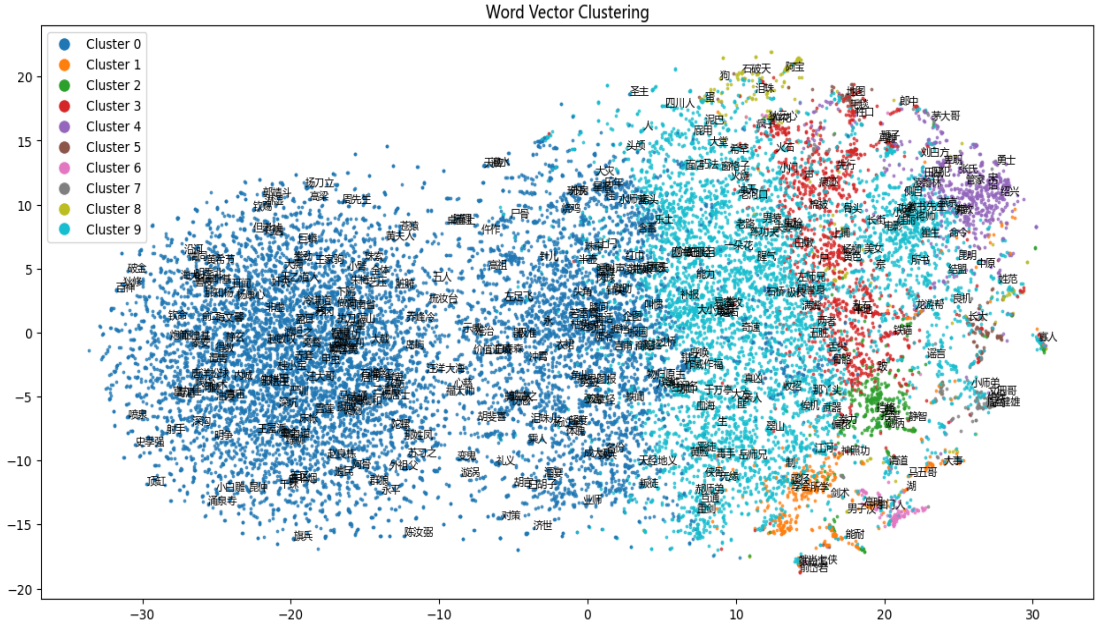

Figure 1: LSTM 算法词向量聚类结果

可以看到两种算法得到的聚类结果大体相似，可以认为两个模型聚类均有效。

Part 3: 段落相似度

选取三个段落：

1.*怪客呻吟了一声，却不回答。程英胆子大了些，取手帕给他按住伤口。但他这一撞之势著实猛恶，头上伤得好生厉害，转瞬之间，一条手帕就给鲜血浸透。她用左手紧紧按住伤口，过了一会，鲜血不再流出。怪客微微睁眼，见程英坐在身旁，叹道：「你又救我作甚？还不如让我死了乾净。」程英见他醒转，很是高兴，柔声道：「你头上痛不痛？」怪客摇摇头，凄然道：「头上不痛，心里痛。」程英听得奇怪，心想：「怎麼头上破了这麼一大块，反而头上不痛心里痛？」当下也不多问，解下腰带，给他包扎好了伤处。*

2.*李莫愁拂尘轻挥，将三般兵刃一齐扫了开去，娇滴滴的道：「陆二爷，你哥哥若是尚在，只要他出口求我，再休了何沅君这个小贱人，我未始不可饶了你家一门良贱。如今，唉，你们运气不好，只怪你哥哥太短命，可怪不得我。」陆立鼎叫道：「谁要你饶？」挥刀砍去，武三娘与陆二娘跟著上前夹攻。李莫愁眼见陆立鼎武功平平，但出刀踢腿、转身劈掌的架子，宛然便是当年意中人陆展元的模样，心中酸楚，却盼多看得一刻是一刻，若是举手间杀了他，在这世上便再也看不到「江南陆家刀法」了，当下随手挥架，让这三名敌手在身边团团而转，心中情意缠绵，出招也就不如何凌厉。*

3.*清乾隆十八年六月，陕西扶风延绥镇总兵衙门内院，一个十四岁的女孩儿跳跳蹦蹦的走向教书先生书房。上午老师讲完了《资治通鉴》上“赤壁之战”的一段书，随口讲了些诸葛亮、周瑜的故事。午后本来没功课，那女孩儿却兴犹未尽，要老师再讲三国故事。这日炎阳盛暑，四下里静悄悄地，更没一丝凉风。那女孩儿来到书房之外，怕老师午睡未醒，进去不便，于是轻手轻脚绕到窗外，拔下头上金钗，在窗纸上刺了个小孔，凑眼过去张望。只见老师盘膝坐在椅上，脸露微笑，右手向空中微微一扬，轻轻吧的一声，好似甚么东西在板壁上一碰。她向声音来处望去，只见对面板壁上伏着几十只苍蝇，一动不动，她十分奇怪，凝神注视，却见每只苍蝇背上都插着一根细如头发的金针。这针极细，隔了这样远原是难以辨认，只因时交未刻，日光微斜，射进窗户，金针在阳光下生出了反光。*

其中前两段：段落 1、2 取自《神雕侠侣》，且位置相近；后一段：段落 3 取自《书剑恩仇录》。

Table 3: Word2Vec 模型段落相似度计算

段落选取		段落相似度
1	2	0.765362
1	3	0.632352
2	3	0.652895

可以看到段落 1、2 的相似度相比和段落 3 的相似度均要高，且段落 1、2 与段落 3 的相似度较为接近，而三个相似度均高于 0.6，均较高。对此结果产生原因推测为金庸文风较为统一，即使是来自不同文章，在段落的层次均有较高的相似度，且本文选取的段落均含有比重较大的对话，而一篇文章内的段落相似度则要更高一些。关于段落 1、2 相似度也未达到 0.8 以上，推测为两段文字未出现相同的人物，因此相似度计算结果较低。

Conclusions

本研究建立了一个基于中文语料库的 Word2Vec 模型和 LSTM 模型，并对其进行了训练和验证。从实验研究中可以得出，LSTM 模型效果在词向量相似度部分较 Word2Vec 更好，而词向量聚类两类模型效果类似。可以

运行过程中，本文还使用了 t-SNE 降维，但效果不是很理想，故未放入报告，而是置于代码部分文件夹内，如有需要可以查看。且在实验过程中单纯使用 t-SNE 降维时，程序运行时间很长，这是由于其运算复杂度较高。在使用 t-SNE 前再进行 PCA 降维可以显著降低运算时间。

总之，Word2Vec 和 LSTM 模型均可以捕获词汇的语义和上下文关系，单词向量在许多 NLP 任务中都有帮助。

References

https://blog.csdn.net/qq_41814556/article/details/80990976
https://blog.csdn.net/v_JULY_v/article/details/102708459
https://zhuanlan.zhihu.com/p/123857569
https://brightliao.com/2016/12/02/dl-workshop-rnn-and-lstm/