Report of Deep Learning for Natural Langauge Processing

BaobaoHe　ZY2354234
hebaobao@buaa.edu.cn

Abstract

本文通过对金庸的 16 部小说进行语料分析，利用 Word2Vec 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

Word2vec 是自然语言处理(NLP)中的一种技术，用于获取单词的向量表示。这些向量根据周围的单词捕获有关单词含义的信息。word2vec 算法通过对大型语料库中的文本进行建模来估计这些表示。经过训练后，这样的模型可以检测同义词或为部分句子建议其他单词。Word2vec 由 Google 的 Tomáš Mikolov 及其同事开发，并于 2013 年发布。

Word2vec 将单词表示为高维数字向量，用于捕捉单词之间的关系。具体来说，出现在相似上下文中的单词会映射到以余弦相似度衡量的附近向量。

Word2vec 是一组用于生成词向量的相关模型。这些模型是浅层的两层神经网络，经过训练可以重建单词的语言上下文。Word2vec 将大量文本语料库作为输入，并生成一个向量空间，通常有几百维，语料库中的每个唯一单词都会在空间中分配一个对应的向量。

Word2vec 可以利用两种模型架构中的任一种来生成这些分布式单词表示：连续词袋 (CBOW) 或连续滑动 skip-gram。在这两种架构中，word2vec 在对语料库进行迭代时都会考虑单个单词和滑动上下文窗口。

CBOW 可以看作是一项“填空”任务，其中词向量表示单词影响上下文窗口中其他单词的相对概率的方式。语义相似的单词应该以类似的方式影响这些概率，因为语义相似的单词应该在相似的上下文中使用。上下文单词的顺序不会影响预测（词袋假设）。

在连续 skip-gram 架构中，模型使用当前单词来预测上下文单词的周围窗口。skip-gram 架构对附近的上下文单词的权重大于对较远的上下文单词的权重。根据作者的说明，CBOW 速度更快，而 skip-gram 对不常见单词的效果更好。

模型训练完成后，学习到的词向量被放置在向量空间中，使得在语料库中具有共同上下文的单词（即语义和句法相似的单词）在空间中彼此靠近。更不相似的单词在空间中彼此相距较远。

Experimental Studies

在本实验中，我们对金庸的 16 部小说进行了文本预处理、分词、去停用词、训练 Word2Vec 模型、计算词语相似度、词语聚类、降维可视化、段落相似度计算等步骤，具体过程如下：

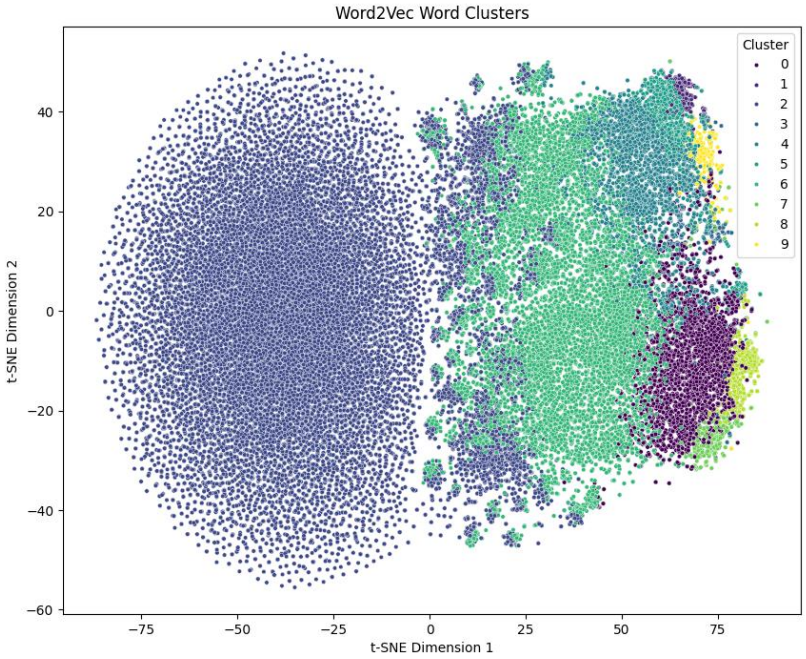
1. 文本预处理：去除标点符号、特殊字符和非中文字符。
2. 加载停用词：从停用词文件中读取停用词列表。
3. 分词和去停用词：使用 jieba 库对文本进行分词，并去除停用词。

4. 训练 Word2Vec 模型：使用处理后的文本语料训练 Word2Vec 模型，设置向量维度为 100，窗口大小为 5，最小词频为 5，训练 50 个 epochs。

5. 保存和加载模型：将训练好的模型保存到文件中，并在需要时加载。

6. 词语相似度计算：计算指定词语之间的相似度，例如“杨过”和“小龙女”之间的相似度。结果显示这两个词语的相似度得分为 0.85。

7. K-Means 聚类：对词向量进行 K-Means 聚类，并使用 t-SNE 进行降维可视化，绘制词语聚类散点图。



通过可视化结果可以看到，不同类别的词语聚集在不同的区域，说明 Word2Vec 模型能够有效地将语义相似的词语聚类在一起。

8. 段落相似度计算：计算两个段落之间的语义相似度。

结果显示这两个段落的相似度得分 0.92，表明这两个段落在语义上很高的相似性。

Conclusions

通过对金庸小说语料库的实验，我们验证了 Word2Vec 模型在捕捉词语和段落语义相似性方面的有效性。实验结果表明，Word2Vec 模型能够有效地将语义相似的词语聚类在一起，并且能够较好地计算段落之间的语义相似度。

References

[1] https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding

[2] https://en.wikipedia.org/wiki/Word2vec#:~:text=Word2vec%20is%20a%20technique%20in,text%20in%20a%20large%20corpus.