ReportofDeepLearningforNaturalLangaugeProcessing

HaoleiZhang
2531563133@qq.com

Abstract

利用给定语料库（金庸语小说料如下链接），利用基于 Word2Vec 模型来训练词向量，计算并输出两个词之间的相似度，使用 KMeans 算法对词向量进行聚类分析，最后，计算两个段落之间的相似度。

Introduction

Word2Vec 是一种计算词向量的方法，它可以将词汇表中的每个词映射到一个固定维度的向量。这些向量捕获了词的语义和语境信息，使得语义相似的词在向量空间中彼此接近。Word2Vec 模型通常有两种架构：连续词袋（CBOW）和 SkipGram。

连续词袋（CBOW）：这种模型通过一个词的上下文（即周围的词）来预测这个词。它考虑了上下文词的词向量，并尝试预测中心词。

SkipGram：与 CBOW 相反，SkipGram 模型通过一个词来预测它的上下文。它使用中心词的词向量来预测周围的词。

Word2Vec 模型训练完成后，得到的词向量可以用于多种自然语言处理任务，如文本分类、情感分析、词性标注、命名实体识别、机器翻译等。通过计算词向量之间的距离或角度，可以评估词之间的语义相似性。

在 Word2Vec 出现之前，词袋模型（BagofWords）是处理文本数据的一种常见方法，但它无法捕捉词序和词义。Word2Vec 的出现为自然语言处理领域带来了革命性的变化，使得模型能够理解和利用词汇的深层语义信息。

Methodology

Word2Vec 模型包括两种架构：连续词袋（CBOW）和 SkipGram。这两种架构都是通过训练神经网络来学习词向量，但它们在训练过程中使用不同的目标。

CBOW（ContinuousBagofWords）

CBOW 模型的目的是根据上下文词来预测中心词。假设我们有词汇表中的词w，以及一个给定的窗口大小m。对于每个训练样本，我们选择一个中心词m和它的上下文词$w_{o1}, w_{o2}, \ldots, w_{o2m}$。

1.输入层：对于每个上下文词w_{oi}，我们查找它的词向量$V_{w_{oi}}$。在 CBOW 中，我们通常取所有上下文词向量的平均值作为输入层的表示：

$$X = \frac{1}{2m}\sum_{i=1}^{2m} V_{w_{oi}}$$

2.隐藏层：这个隐藏层实际上是投影层，它将输入层的向量投影到与输出层相同维度的空间。这个投影通常是通过一个权值矩阵W和偏置向量b来实现的：

$$h = W \cdot X + b$$

3.输出层：输出层是一个 softmax 函数，它将隐藏层的激活转换为概率分布，每个词汇表中的词都有一个对应的概率：

$$\hat{y} = \text{softmax}(h)$$

其中，softmax 函数定义为：

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

4.损失函数：我们使用交叉熵损失来衡量预测概率分布$\hat{y}$和真实分布y之间的差异：

$$J(\theta) = -\sum_{i=1}^{V} y_i \log(\hat{y_i})$$

其中，V是词汇表的大小，y是一个独热编码向量，只有中心词的位置是 1，其他位置是 0。

SkipGram

SkipGram 模型的目的是根据中心词来预测上下文。对于每个训练样本，我们选择一个中心词w_c和它的上下文词$w_{o1}, w_{o2}, \ldots, w_{o2m}$。

1.输入层：输入层是中心词w_c的词向量V_{w_c}。

2.隐藏层：与 CBOW 类似，隐藏层将输入向量投影到与输出层相同维度的空间：

$$h = W \cdot V_{w_c} + b$$

3.输出层：输出层由多个 softmax 单元组成，每个单元对应词汇表中的一个词。我们为每个上下文词位置训练一个 softmax 分类器：

$$\widehat{y_{o_i}} = \text{softmax}(h)$$

其中，o_i是上下文词的位置。

4.损失函数：与 CBOW 相同，我们使用交叉熵损失函数，但是这次我们对每个上下文词的预测都要计算损失：

$$J(\theta) = -\sum_{i=1}^{2m}\sum_{j=1}^{V} y_{o_i,j} \log\left(\widehat{y_{o_i,j}}\right)$$

其中，$y_{o_i,j}$是一个独热编码向量，只有当j等于上下文词索引时，$y_{o_i,j}$为 1，否则为 0。

ExperimentalStudies

使用预处理后的文本数据训练一个 Word2Vec 模型，将词汇映射到一个高维向量空间中，使得相似词汇在向量空间中更接近。通过计算两个指定词之间的余弦相似度，可以得到它们在向量空间中的接近程度。

词向量对		Word2Vec 语义距离
杨过	小龙女	0.922081470489502
郭靖	黄蓉	0.9971831440925598
病情	五五	0.0913400650024414
一阵	太虚	0.0021263696253299713

Conclusions

杨过和小龙女的语义相似度很高（0.9221），这表明在语料库中，这两个词在语义上非常接近。杨过和小龙女是金庸小说《神雕侠侣》中的两位主要人物，他们之间的关系非常紧密，因此模型能够捕捉到这一点，并将它们的词向量距离设置得很近。

郭靖和黄蓉的语义相似度极高（0.9972），接近 1。这表明这两个词在语料库中几乎是同义词或非常相关的词汇。郭靖和黄蓉是金庸小说《射雕英雄传》中的主要人物，他们是夫妻关系，而且在小说中频繁同时出现，因此模型将它们的词向量距离设定得非常近。

病情和五五的语义相似度为负值（0.0913），这表明它们在语义上几乎没有相关性。病情通常与健康相关，而五五是一个数字，在语料库中，它们之间没有明显的语义联系。

一阵和太虚的语义相似度非常低（0.0021），接近于零。这表明在语料库中，它们之间的语义关联很弱。一阵通常用来描述时间或事件的持续，而太虚可能指的是一种玄幻或抽象的概念，两者在语料库中的语义联系不明显。

所以得出以下结论：

高语义相似度（接近 1.0）：高相似度的词对（如杨过与小龙女，郭靖与黄蓉）表明这些词在训练语料库中具有强烈的语义关联。这通常发生在描述同一关系、场景或故事背景的词汇中。

低或负语义相似度（接近 0 或负值）：低相似度或负相似度的词对（如病情与五五，一阵与太虚）表明这些词在语料库中几乎没有语义关联。它们可能出现在完全不同的上下文中或具有完全不同的含义。

模型有效性：Word2Vec 模型成功地捕捉到了语料库中词汇的语义关联。模型可以区分语义上相关和不相关的词对，并为文本分析和自然语言处理任务提供有价值的语义信息。

总的来说，这些结果验证了 Word2Vec 模型在捕捉和表示语料库中词汇语义关系方面的有效性。