第 3 次作业报告

Siyuan Jin
20373680@buaa.edu.cn

Abstract

本报告探讨了自然语言处理中词向量的表示形式，重点关注独热编码、词袋模型和分布式表述三种方法。随后，报告详细介绍了 Word2Vec 模型，包括其两种架构（CBOW 和 Skip-Gram）以及训练过程。最后，报告通过实验验证了基于 Word2Vec 的词向量在词语相关度和聚类任务中的有效性，并分析了 CBOW 和 Skip-Gram 模型的优缺点。

Introduction

计算机无法看懂人类的自然语言，也无法对其进行直接处理，因此自然语言处理第一步需要将自然文字转换为计算机能够“看懂”的数字，因此需要将自然文字进行编码，转换由数字组成的词向量。

1. 常见词向量的表示形式

(1) 独热编码（One-Hot Encoding）

独热编码是一种将分类数据转换为二进制向量的方法。每个类别都表示为一个长度为类别数的二进制向量，其中只有一个位置为 1，其他位置均为 0。假设有一个词汇表 [”cat”, ”dog”, ”fish”]，对这三个词进行独热编码：”cat” 的独热编码是 [1, 0, 0]，”dog” 的独热编码是 [0, 1, 0]，”fish” 的独热编码是 [0, 0, 1]。如果有一个新的句子”cat dog dog fish”，可以将其转换为如下形式：[[1, 0, 0], [0, 1, 0], [0, 1, 0], [0, 0, 1]]。

独热编码的优势在于，通过将每个词映射到一个唯一的、维度等于词汇表大小的向量，独热编码确保了词之间的相互区分性，从而避免了潜在的歧义问题。然而，独热编码也存在显著的局限性。首先，由于每个词向量都是唯一的且与其他词向量正交，独热编码无法捕捉词之间的任何语义相似性或相关性。其次，随着词汇表规模的增大，独热编码产生的词向量维度也会相应增加，这可能导致模型过拟合，尤其是在训练数据有限的情况下。此外，由于独热编码产生的向量通常是高度稀疏的，这会对计算效率产生负面影响，尤其是在处理大规模数据集时。因此，尽管独热编码在特定场景下有其应用价值，但在需要捕捉词之间语义关系的自然语言处理任务中，它通常不是最佳选择。

(2) 词袋模型（Bag of Words, BoW）

词袋模型（Bag of Words, BoW）是一种用于文本数据向量化表示的模型，它将文本看作是无序的词集合，忽略了文本中词的顺序和语法结构。在词袋模型中，每个文档被表示为一个向量，其维度等于词汇表的大小，而向量中的每个元素代表对应词在文档中出现的次数或频率。

词袋模型的优点在于它的简单性和处理大规模文本数据的能力。它不依赖于词的顺序，因此对于许多分类和聚类任务来说是非常有用的。此外，词袋模型可以很容易地与其他机器学习算法结合使用。然而，词袋模型也有其局限性：一是丢失上下文信息，由于词袋模型不考虑词的顺序，因此它无法捕捉到文本中的语义关系和上下文信息，例如，”我喜欢狗”和”狗喜欢我”在词袋模型中会有相同的表示，尽管它们的含义不同；二是维度灾难，随着词汇表大小的增加，特征向量的维度也会相应增加，这可能导致过拟合和计算效率低下；三是稀疏性，大多数文档只会使用词汇表中的一小部分词，因此得到的特征向量通常是高度稀疏的。

尽管有这些局限性，词袋模型仍然是一个广泛使用的工具，特别是在处理文本分类和情感分析等问题时。在深度学习和词嵌入技术出现之前，词袋模型是自然语言处理领域中最常用的文本表示方法之一。

(3) 分布式表述（Distributed Representations）

分布式表述是一种在机器学习和自然语言处理中广泛采用的数值向量表示方法。与传统的离散符号表示（如词袋模型）不同，分布式表述将数据编码为连续的、低维的实数向量，从而实现对数据内在结构和语义信息的捕捉。分布式表述的主要特点包括：

连续性：分布式表述通过连续的向量空间来建模数据，允许对数据中的相似性和差异性进行精细的量化。

低维性：分布式表述通常采用较低维度的向量，这有助于减少计算复杂度，并提高模型的泛化能力。

语义保持：分布式表述能够保留数据的语义信息，使得在向量空间中相近的元素在语义上也是相似的。

泛化能力：分布式表述支持向量运算，从而可以实现对未见过数据的泛化处理，例如，通过向量加减法可以推断出词语之间的类比关系。

在自然语言处理领域，词嵌入（Word Embeddings）是分布式表述的一个典型应用，它将词汇映射到固定维度的向量空间中。常见的词嵌入技术包括 Word2Vec、GloVe 和 FastText 等，这些技术通过大规模语料库的训练，学习得到能够反映词语语义和用法特征的向量表示。分布式表述在多个领域都有广泛应用，如推荐系统、计算机视觉、语音识别等，它为处理高维数据和模型降维提供了一种有效的手段。此外，分布式表述在提高模型性能、减少过拟合风险以及增强模型解释性等方面也展现出显著的优势。

2. Word2Vec

Word2Vec 是一种计算模型，它将词汇表中的每个词映射到一个固定维度的向量。Word2Vec 模型由 Tomas Mikolov 在 Google 开发，并在 2013 年发布。这个模型的目标是通过词的上下文来学习词的向量表示，使得语义上相似的词在向量空间中靠近。

Word2Vec 模型有两种架构：连续词袋（CBOW）和 Skip-Gram。

CBOW：CBOW 模型通过一个词的上下文（即周围的词）来预测这个词。具体来说，它使用一个词的上下文词汇的词向量平均值来预测这个词。CBOW 模型在大型数据集上训练速度较快，并且在处理频繁词方面表现较好。

Skip-Gram：Skip-Gram 模型与 CBOW 相反，它通过一个词来预测其上下文。也就是说，给定一个词，Skip-Gram 模型会尝试预测该词周围的词。Skip-Gram 模型在处理罕见词和复杂模式方面表现更好，但需要更多的训练时间。

Word2Vec 模型的核心思想是，如果两个词在上下文中具有相似性，那么它们的向量表示也

应该相似。这样，通过大量的文本数据训练，Word2Vec 能够学习到词的语义和用法信息。

Word2Vec 模型的训练通常涉及以下步骤：

构建训练数据：从文本数据中提取词对作为训练样本。对于 CBOW 模型，输入是上下文词，标签是中心词；对于 Skip-Gram 模型，输入是中心词，标签是上下文词。

初始化词向量：为词汇表中的每个词随机初始化一个向量。

训练模型：使用梯度下降等优化算法来更新词向量，使得模型能够更好地预测词的上下文。

向量表示：训练完成后，每个词的向量表示就可以用于下游任务，如文本分类、情感分析等。

Word2Vec 模型在自然语言处理领域产生了深远的影响，它为词的向量表示提供了一种有效的计算方法，并且在多个 NLP 任务中都取得了很好的效果。此外，Word2Vec 模型也为后来的词嵌入技术（如 GloVe、FastText）的发展奠定了基础。

Methodology

M1：训练基于 Word2Vec 的词向量

1. 导入语料文件；
2. 对语料库的数据去除停用词；
3. 使用 jieba 进行分词；
4. 保存分词后的语料结果;
5. 读取保存的语料，进行 Word2Vec 的模型训练；
6. 保存训练好的 CBOW 模型和 Skip-Gram 模型。

M2：词语相关度展示

1. 读取训练好的模型；
2. 指定某个具体的词，展示与该词最相关的 5 个词。

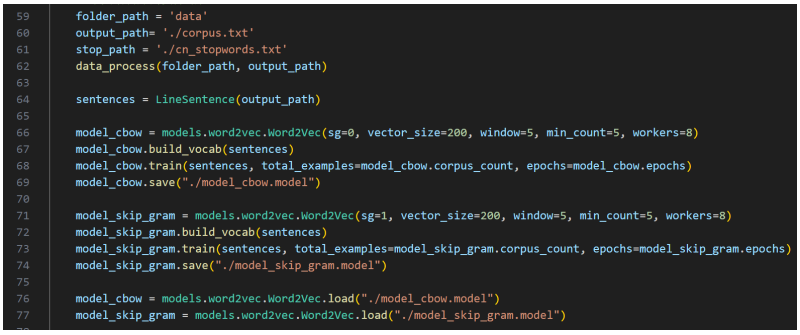
M3：通过聚类验证词向量的有效性

1. 从 16 本小说组成的语料库的所有词语中，选取出现次数总共超过 50 次的词作为高频词汇，并过滤高频词中的停用词；
2. 对于所有剩余的高频词，通过训练好的 skip-gram 模型得到其词向量；
3. 采用 K-means 聚类方法对这些词向量进行聚类, 聚类类别数设置为 16；
4. 聚类结果通过 tSNE 方法进行可视化。

Experimental Studies

M1：训练基于 Word2Vec 的词向量

实验设定：词向量的维度：200；上下文窗口：5；训练过程忽略出现次数少于 5 次的词。



```
folder_path = 'data'
output_path= './corpus.txt'
stop_path = './cn_stopwords.txt'
data_process(folder_path, output_path)

sentences = LineSentence(output_path)

model_cbow = models.word2vec.Word2Vec(sg=0, vector_size=200, window=5, min_count=5, workers=8)
model_cbow.build_vocab(sentences)
model_cbow.train(sentences, total_examples=model_cbow.corpus_count, epochs=model_cbow.epochs)
model_cbow.save("./model_cbow.model")

model_skip_gram = models.word2vec.Word2Vec(sg=1, vector_size=200, window=5, min_count=5, workers=8)
model_skip_gram.build_vocab(sentences)
model_skip_gram.train(sentences, total_examples=model_skip_gram.corpus_count, epochs=model_skip_gram.epochs)
model_skip_gram.save("./model_skip_gram.model")

model_cbow = models.word2vec.Word2Vec.load("./model_cbow.model")
model_skip_gram = models.word2vec.Word2Vec.load("./model_skip_gram.model")
```

图 1: 实验 1 代码截图

实验代码截图如下：

M2：词语相关度展示

模型训练后，分别读取训练好的 CBOW 模型和 Skip-Gram 模型，然后指定某一个词，展示与该词最相关的 5 个词。指定的词包括：黄蓉、杨过、张无忌、令狐冲、韦小宝、峨嵋派、屠龙刀、蛤蟆功、葵花宝典。其涵盖了人名、门派名、武器名、功夫名、重要物品名等。结果如下所示。

实验结果如下：

表 1: 实验 2 CBOW 模型结果统计表

指定词	相关词
黄蓉	(郭襄, 0.8827192187309265), (郭靖, 0.8676692247390747), (杨过, 0.8651381731033325), (陆无双, 0.862692654132843), (胡斐, 0.8600690960884094)
杨过	(郭 靖, 0.8815273642539978), (黄 蓉, 0.8651382327079773), (张 无 忌, 0.8537903428077698), (小龙女, 0.8526134490966797), (石破天, 0.8438446521759033)
张无忌	(令 狐 冲, 0.9142075181007385), (杨 过, 0.8537903428077698), (袁 承 志, 0.8527193665504456), (石破天, 0.8436607718467712), (虚竹, 0.8406840562820435)
令狐冲	(张 无 忌, 0.9142071604728699), (张 翠 山, 0.8184325098991394), (虚 竹, 0.8109227418899536), (杨过, 0.8076299428939819), (胡斐, 0.7966643571853638)
韦小宝	(袁 承 志, 0.7353826761245728), (康 熙, 0.7025651931762695), (张 无 忌, 0.6816034317016602), (令狐冲, 0.6611337661743164), (徐天宏, 0.6565374732017517)
峨嵋派	(华 山 派, 0.9182558655738831), (武 当 派, 0.8938098549842834), (本 派, 0.867118239402771), (青城派, 0.8656613230705261), (武当, 0.8648597598075867)
屠龙刀	(宝 刀, 0.7974214553833008), (倚 天 剑, 0.7779286503791809), (打 狗 棒, 0.7677837014198303), (铜牌, 0.752639651298523), (宝剑, 0.7139824032783508))
蛤蟆功	(一 阳 指, 0.861961841583252), (六 脉, 0.8498128652572632), (神 剑 掌, 0.8470044136047363), (空明拳, 0.8424929976463318), (打狗棒法, 0.8378633856773376)
葵花宝典	(宝 典, 0.8428520560264587), (可 兰 经, 0.839589536190033), (贵 寺, 0.8334234952926636), (兵法, 0.8300827741622925), (至高无上, 0.8283761739730835)

表 2: 实验 2 Skip-Gram 模型结果统计表

指定词	相关词
黄蓉	(郭 靖, 0.7138510942459106), (杨 过, 0.6545323133468628), (洪 七 公, 0.6451625227928162), (陆无双, 0.6310157775878906), (郭襄, 0.6282601952552795)
杨过	(黄 蓉, 0.6545323133468628), (郭 靖, 0.6537489891052246), (小 龙 女, 0.6445891261100769), (郭襄, 0.6104824542999268), (武三通, 0.5716080069541931)
张无忌	(张 翠 山, 0.6851072311401367), (令 狐 冲, 0.6595088839530945), (石 破 天, 0.6244333982467651), (赵敏, 0.6230799555778503), (范遥, 0.6183814406394958)
令狐冲	(张 无 忌, 0.6595087647438049), (盈 盈, 0.5965223908424377), (岳 不 群, 0.5559675097465515), (林平之, 0.5555816888809204), (宋青书, 0.5534787774085999)
韦小宝	(康 熙, 0.6395106315612793), (苏 菲 亚, 0.6076319813728333), (乾 隆, 0.607426106929779), (费要多罗, 0.6050223708152771), (索额图, 0.6010286808013916)
峨嵋派	(峨 嵋, 0.7744516730308533), (韦 陀 门, 0.7717207074165344), (青 海, 0.7524594664573669), (华山派, 0.7424366474151611), (武当派, 0.7383488416671753)
屠龙刀	(倚 天 剑, 0.7869552969932556), (宝 刀, 0.7455915808677673), (屠 龙, 0.7155578136444092), (打狗棒, 0.6932938694953918), (宝剑, 0.6742925047874451)
蛤蟆功	(玄 冥 神, 0.878624439239502), (寒 冰 绵, 0.8405723571777344), (一 阳 指, 0.8336117267608643), (纯阳, 0.8301880955696106), (斗转星移, 0.8273778557777405)
葵花宝典	(宝 典, 0.8777536749839783), (神 照 经, 0.8476731181144714), (真 经, 0.8306441307067871), (九阴真经, 0.8209508657455444), (下卷, 0.8177026510238647)

在词语分类的准确性方面，CBOW 模型和 Skip-Gram 模型均展现出了对词类或词性的精确识别能力。具体来说，在两种模型中，与特定人名（例如“黄蓉”）最相关的词汇同样为人名（如“郭靖”、“杨过”、“岳灵珊”、“胡斐”、“陆无双”等）；与门派名称（如“峨嵋派”）相关的词汇则都属于门派名称；与武功名称（如“蛤蟆功”）相关的词汇同样为武功名称（例如“玄冥神掌”、“一阳指”等）；与物品名称（如“葵花宝典”）相关的词汇也都属于物品名称（如“九阳真经”、“玉箫剑法”等）。

从人物关系的表现来看，Skip-Gram 模型相较于 CBOW 模型在捕捉人物关联方面表现更为优异。例如，在 CBOW 模型中，与“杨过”最相关的词汇是“黄蓉”，然而这两个人物并未出现在同一部小说中。这可能是因为“黄蓉”同样作为主要人物，在训练语料库中频繁出现，从而被错误地判断为与“杨过”具有较高相关性。而在 Skip-gram 模型的结果中，与人名最相关的词汇通常与该人物具有紧密的联系。例如，“黄蓉”最相关的词汇“郭靖”是她的配偶；“杨过”最相关的词汇是“小龙女”，他们之间是师徒关系；“张无忌”最相关的词汇“张翠山”是他的生父。由此可见，CBOW 模型更倾向于反映文本的整体关联性，其中许多高度相关的词汇跨越了多个时代和不同的小说，而 Skip-gram 模型则更加专注于局部的关联性，高度相关的词汇主要集中出现在同一部小说中。

M3：通过聚类验证词向量的有效性

从 16 本小说组成的语料库的所有词语中，选取出现次数总共超过 50 次的词作为高频词汇，

并根据停用词表，过滤掉这些高频词中的停用词。对于所有剩余的高频词，通过训练好的 skip-gram 模型得到其词向量。然后采用 K-means 聚类方法对这些词向量进行聚类，并利用 tSNE 方法将聚类结果进行可视化。聚类类别数设置为 16。可视化结果如图 2 所示。

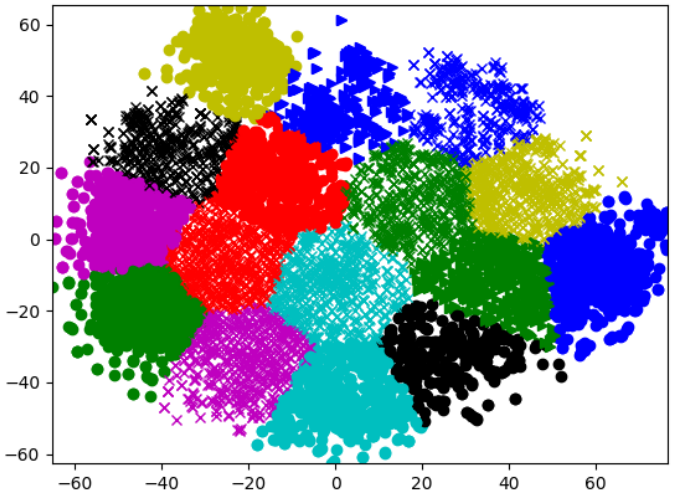


图 2: 实验 3 聚类结果

Conclusions

通过实验可以得到以下结论：1. Word2Vec 模型能够有效地学习词的语义和用法信息，并在多个任务中取得了良好的效果。2. CBOW 和 Skip-Gram 模型在词语分类的准确性方面表现相当，但 Skip-Gram 模型在捕捉人物关联方面更为优异。3. 词向量可以有效地反映词语之间的语义关系，并通过聚类等方法进行可视化分析。