实验名称

word2vec模型实现与效果测试

数据准备与处理

数据集

数据集来源：'dataset/news.txt'

数据集预处理：使用jieba进行中文分词，去除停用词，生成分词后的文件`cutdata.txt`

预处理步骤

1. 利用`dataprocess.py`进行数据预处理，包括分词、去除停用词，并保存结果。

2. 统计词频，保留词频大于等于最小词频阈值`min\_count`的词。

3. 将处理后的数据保存在`datasave/cutdata0.txt`。

模型与算法介绍

word2vec模型

word2vec是一种生成词嵌入（word embeddings）的模型，它使用浅层神经网络从大量文本数据中学习词的连续向量表示。

算法

SkipGram模型：该模型预测一个词在给定上下文的情况下出现的概率。

负采样（Negative Sampling）：一种优化技术，用于提高训练效率，通过随机采样生成负样本。

实验流程介绍

训练word2vec模型

1. 使用`train.py`脚本训练word2vec模型，生成词嵌入文件`word\_embedding.txt`。

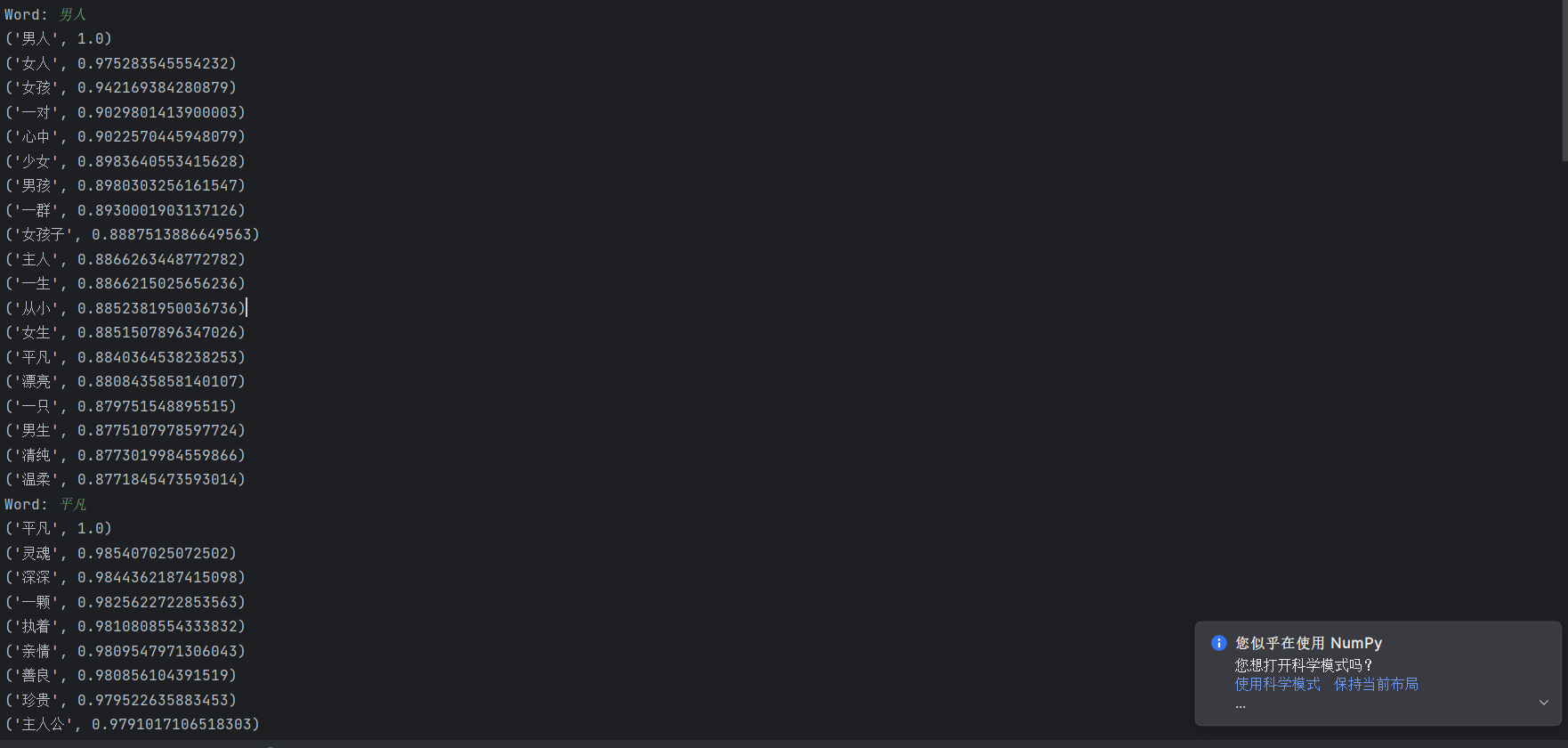
2. 配置模型参数，如嵌入维度`emb\_dimension`、窗口大小`window\_size`、批次大小`batch\_size`等。

测试word2vec模型

1. 使用`test.py`脚本，通过计算余弦相似度测试词嵌入的效果。

2. 输入一个词，输出与之最相关的前20个词。

实验结果



实验通过`test.py`脚本进行，输入不同的词，得到与之最相关的一系列词。以下是部分结果摘要：

输入“男人”，相关词汇包括“女人”、“女孩”、“男孩”等，显示出模型能够捕捉性别相关的语义信息。

输入“平凡”，相关词汇包括“灵魂”、“深深”、“一颗”等，反映出模型能够关联到情感和抽象概念。

输入“思考”，相关词汇如“勇气”、“反思”、“渴望”等，表明模型可以关联到心理活动和内在状态。

输入“沉着”，相关词汇包括“畸形”、“后半段”、“挺身而出”等，这可能表明模型在某些情况下会生成不相关的词汇，这可能是由于数据集的偏差或模型训练的限制。

输入“考虑”，相关词汇如“慎重”、“专家建议”、“首要”等，显示出模型能够关联到决策和重要性。

输入“冷漠”，相关词汇包括“无助”、“烦躁”、“偏偏”等，表明模型能够捕捉到情感状态和行为特征。

输入“马列”，未能找到相关词汇，可能是因为这个词在训练数据集中出现的频率较低，或者模型未能很好地学习到该词的语义信息。

结果分析

模型在大多数情况下能够生成语义上相关的词汇，表明word2vec模型能够学习到中文词汇的语义信息。

模型在某些情况下生成了不相关的词汇，这可能是由于训练数据的局限性或模型参数设置的不足。

模型对于出现频率较低的词汇（如“马列”）可能无法生成相关词汇，这提示我们在应用模型时应考虑其局限性。

项目个人心得体会

通过本项目，我深入理解了word2vec模型的工作原理和实现过程。在实验中，我遇到了数据预处理、模型参数调优等挑战，通过查阅资料和不断尝试，我逐步解决了这些问题。此外，我也认识到了模型的局限性，比如对低频词的捕捉能力较弱。未来，我希望能够探索更先进的词嵌入模型，并尝试改进模型以提高对低频词的处理能力。总的来说，这是一个非常有价值的学习经历，它不仅提升了我的技术能力，也加深了我对自然语言处理领域的理解。