# 实验报告

## 实验名称

神经机器翻译系统（模型）的实现

## 数据准备与处理

### 数据集来源

实验数据集来源于[TED2013](https://opus.nlpl.eu/download.php?f=TED2013/v1.1/moses/en-zh.txt.zip)，是中文和英文之间的平行语料。

### 数据集划分

- 训练集：大部分句子，用于模型训练。

- 验证集：倒数第1001句至倒数第2000句，用于模型的超参数调整。

- 测试集：倒数第1句至倒数第1000句，用于评估模型的最终性能。

### 数据预处理流程

1. \*\*训练BPE分词模型\*\*：使用BPE（Byte Pair Encoding）算法对数据进行分词处理。

2. \*\*BPE分词\*\*：将训练、验证和测试集的句子通过BPE模型进行分词。

3. \*\*生成词表\*\*：根据BPE分词结果生成词汇表，用于模型的输入和输出。

4. \*\*去除长句子\*\*：为了提高训练效率，去除超出模型最大长度限制的句子。

## 模型与算法介绍

### Transformer模型

本实验实现了基本的Transformer模型，该模型由Encoder和Decoder组成，使用了多头注意力机制（Multi-Head Attention）和位置感知的前馈网络（Positionwise Feedforward）。

### 解码算法

实验中实现了任意一种常用的解码算法，例如贪心解码、Beam Search解码、Top-k sampling或Top-p sampling。

### 实验环境

- \*\*PyTorch\*\*：用于构建和训练神经网络模型。

- \*\*sentencepiece\*\*：用于BPE分词。

- \*\*sacreBLEU\*\*：用于评估翻译质量。

## 实验流程介绍

1. \*\*数据预处理\*\*：按照上述步骤对数据集进行处理。

2. \*\*模型构建\*\*：根据Transformer模型架构构建Encoder、Decoder和整体模型。

3. \*\*模型训练\*\*：使用训练集数据，通过反向传播算法训练模型。

4. \*\*验证与测试\*\*：在验证集上调整超参数，最终在测试集上评估模型性能。

5. \*\*结果分析\*\*：使用sacreBLEU等工具对翻译结果进行评测。

## 实验结果

实验结果通常包括模型在测试集上的性能指标，如BLEU分数等，以及可能的翻译示例。

## 结果分析

分析实验结果，讨论模型性能的可能影响因素，如数据集大小、模型结构、训练时长等。

## 项目个人心得体会

在本项目中，我学习到了神经机器翻译的基本原理和实现方法。通过亲自动手实践，我对Transformer模型有了更深刻的理解。同时，我也体会到了数据预处理在机器学习项目中的重要性。此外，调参过程锻炼了我的问题解决能力。总之，这是一个既有挑战又充满成就感的项目。

