



自然语言处理课程设计

课程设计题目 XXXXXXXXXX的设计与实现

学生姓名/学号 XXXX/181501123

所 在 学 院 信息工程学院

班级

完 成 日 期 2022年\*月

# 目 录

[目 录 - 4 -](#_Toc120261691)

[一. 系统的目标与意义 - 1 -](#_Toc120261692)

[二. 开发环境与工具 - 2 -](#_Toc120261693)

[三. XXX算法的原理介绍 - 3 -](#_Toc120261694)

[四. 系统概要设计 - 4 -](#_Toc120261695)

[4.1 总体架构设计 - 4 -](#_Toc120261696)

[4.2 训练数据 - 4 -](#_Toc120261697)

[4.3 自动评估指标 - 4 -](#_Toc120261698)

[五. 系统详细设计 - 5 -](#_Toc120261699)

[5.1 模型的设计 - 5 -](#_Toc120261700)

[5.2 语料库搭建 - 5 -](#_Toc120261701)

[5.3 模型构建 - 5 -](#_Toc120261702)

[5.4 训练与测试 - 6 -](#_Toc120261703)

[5.5 定义可调用接口 - 6 -](#_Toc120261704)

[5.6 结果显示接口 - 7 -](#_Toc120261705)

[六. 实验及展示 - 9 -](#_Toc120261706)

[6.1 实验结果 - 9 -](#_Toc120261707)

[6.2 系统展示 - 9 -](#_Toc120261708)

[七. 小组介绍 - 10 -](#_Toc120261709)

[7.1 小组成员介绍 - 10 -](#_Toc120261710)

[7.2 我的工作介绍 - 10 -](#_Toc120261711)

[八. 结 论 - 11 -](#_Toc120261712)

[参考文献 - 12 -](#_Toc120261713)

# 

# 一. 系统的目标与意义

对开发的系统进行一个总结（介绍开发一个什么系统，能够完成什么任务，使用了什么技术。该系统具有什么意义）

# 二. 开发环境与工具

**主要系统开发过程中使用的开发语言、框架、工具包等**

XXX

# 三. XXX算法的原理介绍

对完成该任务的算法的原理展开介绍，如使用的Transformer模型，必须对Transformer进行详细介绍。

# 四. 系统概要设计

## 4.1 总体架构设计

可以画一个系统架构图，并对图的各个模块展开介绍。

## 4.2 训练数据

**对训练模型需要的数据进行介绍，如数据集的来源，数据集的训练集数目、验证集数目和测试机数目，再给出几个训练集的样例。**

## 4.3 自动评估指标

对自动评估系统指标进行介绍，可以是多个指标，如机器翻译常用的是BLEU。

# 五. 系统详细设计

## 5.1 模型的设计

Transformer模型架构是本系统的核心部分。系统基于Transformer架构实现对古汉语到现代文的Seq2Seq翻译，根据其工作的基本原理，将分为XXX个步骤实现，分别为：XXX，XXX，XXX。

## 5.2 语料库搭建

读取语料库，怎么实现的，展开介绍

关键代码如下所示，简单介绍（下面表格中给出核心代码）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | src\_tokenizer, trg\_tokenizer = (VernacularTokenTokenizer(), AncientTokenTokenizer(), )     random.seed(args.seed)     random.shuffle(all\_text\_pair)     if args.dataset\_size > 0:      all\_text\_pair = all\_text\_pair[:args.dataset\_size]     test\_size, valid\_size = args.test\_size, args.valid\_size     if test\_size + valid\_size >= 1:      raise ValueError("Sum of test size and valid size must be less than 1")     # tokenize text     num\_test\_dataset = round(len(all\_text\_pair) \* test\_size)     num\_valid\_dataset = round(len(all\_text\_pair) \* valid\_size)     num\_train\_dataset = len(all\_text\_pair) - num\_test\_dataset - num\_valid\_dataset     training\_text\_pairs = all\_text\_pair[:num\_train\_dataset]     valid\_text\_pairs = all\_text\_pair[ num\_train\_dataset: num\_train\_dataset + num\_v alid\_dataset |

## 5.3 模型构建

**算法模型的构建介绍**。

程序代码如下所示。其主要功能为XXX。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | class Transformer(tf.keras.Model):     def \_\_init\_\_(self, n\_layers, d\_model, n\_heads, diff,       input\_vocab\_size, target\_vocab\_size,                   def call(self, inputs, targets, training, encode\_padding\_mask,      look\_ahead\_mask, decode\_padding\_mask):           return final\_out, att\_weights |

## 5.4 训练与测试

训练集大小设置为12000，batch大小为4，进行100个epoch。

程序代码如下所示。功能为通过框架的Trainer和ModelCheckPoint实现模型的训练以及检查点保存功能。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | model = ModelInterface(         src\_vocab=data\_module.src\_vocab,         trg\_vocab=data\_module.trg\_vocab,         num\_epoch=args.max\_epochs,         steps\_per\_epoch=len(data\_module.train\_dataloader()),         model\_config=get\_args\_by\_parser(args, ModelInterface.model\_cls.parser),         \*\*get\_args\_by\_parser(args, ModelInterface.parser),     )      # 初始化Trainer的相关回调     lr\_monitor = LearningRateMonitor(logging\_interval=‘step’)     model\_checkpoint = ModelCheckpoint(         monitor="train/loss",         dirpath=args.default\_root\_dir,         mode="min",         every\_n\_train\_steps=15000,)     trainer = pl.Trainer.from\_argparse\_args(         args, logger=logger, callbacks=[lr\_monitor, model\_checkpoint],)     trainer.fit(model, datamodule=data\_module)     trainer.test(model, datamodule=data\_module) |

## 5.5 定义可调用接口

接口输入输出：用户输入目标语句后，能够判断语句长度依据模型维度将语句分割为合适的长度，将语句映射为向量后调用目标词表的映射方法，返还sent\_list。

程序代码如下所示。功能为将用户输入的文本映射为向量，通过模型翻译后拼接入输出序列列表，最终生成一个翻译结果。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | batch\_token\_ids = self.model.inference(                 src.to(self.device), src\_size.to(self.device),                 trg\_eos\_idx=self.trg\_vocab.eos\_idx,                 max\_seq\_len=max\_seq\_len,             ).unsqueeze(0)         else:          raise ValueError(f"Unsupported model: {self.model\_name}")         sent\_list = []         for i in range(batch\_token\_ids.size(0)):          token\_ids = batch\_token\_ids[i]          token\_list = []          for tid in token\_ids:                  if tid == self.trg\_vocab.eos\_idx:                      break                 token\_list.append(self.trg\_vocab.itos(tid.item()))           sent\_list.append("".join(token\_list)) return sent\_list |

## 5.6 结果显示接口

基于用户界面输入的内容，如果显示算法生成的结果到界面上。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | def tran\_implement():     #预测用户的输入，猜测通过点击猜测的内容，修改用户当前输入的该行内容     if request.method=="POST":         #根据用户是否使用的翻译类型进行返还         text=request.form.get(‘text’).strip().replace(‘ ’,’’)         atm=request.form.get(‘atm’)         print(text+str(atm))         text\_list=text.split(‘\n’)         res\_list=[]         try:             if atm==‘true’:                 #text中根据回车分行 然后拼接                 for text in text\_list:                     res=atmModel.inference(text)                     res\_list.append(res)             else:                 for text in text\_list:                     res=mtaModel.inference(text)                     res\_list.append(res)             res=‘\n’.join(res\_list)         except RuntimeError:             print(‘cant translate’)             result = {‘code’: 401, ’data’: "错误"}             return jsonify(result)         result={‘code’:200,’data’:res}         return jsonify(result) |

|  |  |
| --- | --- |
|  |  |

# 六. 实验及展示

## 6.1 实验结果

给出一个表展示系统在评估指标上的结果，并分析。

在系统功能测试过程中，系统各个功能模块均能正常运行，并能返回预期结果，因此本系统较为稳定。系统所测试的数据来源于不同于语料库资源的网络，文本较为典型、并且可以查询到较一致的白话文翻译。系统在进行测试时达到预期效果，速度和稳定性上表现良好。

## 6.2 系统展示

本系统采用前后端分离架构，前端界面精简地提供用户所需的翻译、阅读、统计等功能。系统界面如图7.1所示。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成图6.1 翻译界面

再给出3个到5个测试用例的截图

# 七. 小组介绍

## 7.1 小组分工

介绍每个成员，分别是谁，具体的分工是什么。

## 7.2 我的工作

这一部分每个人应该是不一样的，对根据上面的分工，自己具体做了什么事情，进行详细介绍，比如学习了什么技术，查阅了什么资料，开发了哪一模块，报告的撰写，汇报PPT的撰写，等。

# 八. 结 论

系统的介绍

开发系统中，遇到了什么挑战。通过该课程设计，学习到了什么。

# 参考文献

（给出查询的教材、论文、文档等资料）

1. 肖桐，朱靖波.机器翻译基础与模型[M].东北大学自然语言处理实验室,2020
2. Ashish Vaswani, Noam Shazeer. Attention Is All You Need [J].Eprint Arxiv,2017.
3. Zaremba W , Sutskever I , Vinyals O . Recurrent Neural Network Regularization[J]. Eprint Arxiv, 2014.
4. Minuzzi F C , Farina L . A deep learning approach to predict significant wave height using long short-term memory[J]. 2022.
5. Cho K , Merrienboer B V , Gulcehre C , et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.