華中科技大學

自然语言处理课程

第一次作业-拼写纠正

院　　系　人工智能与自动化学院

组员信息 魏萌博（60%） 人工智能 01 班 U202114966

组员信息 张伟业（40%） 本硕博 2101 班 U202115203

指导教师　陈伟

日　　期　2024 年 7 月 8 日

1 实验原理

1.1 n-gram 语言模型

n-gram 语言模型是一种基于概率统计的语言模型，通过计算 n 个连续词出现的概率来预测下一个词。其核心思想是通过上下文中前 n-1 个词来预测第 n 个词，公式如下：

$$P(w_i \mid w_{i-n+1}, \ldots, w_{i-1}) = \frac{\text{count}(w_{i-n+1}, \ldots, w_{i-1}, w_i)}{\text{count}(w_{i-n+1}, \ldots, w_{i-1})} \tag{1}$$

1.2 噪声信道模型

噪声信道模型是一种用于拼写纠正的经典模型，其核心思想是将拼写错误视为在传输过程中引入的噪声，通过最大化后验概率来找到最可能的正确词。该模型基于贝叶斯公式：

$$P(w \mid e) = \frac{P(e \mid w)P(w)}{P(e)} \tag{2}$$

$$P(w) = \max\left(P_{\text{unigram}}\left(w\right), P_{\text{bigram}}\ \left(w \mid pw_2\right), P_{\text{trigram}}\ \left(w \mid pw_1, pw_2\right)\right) \tag{3}$$

1.3 Transformer 模型

本实验采用 Hugging Face 的 Transformers 库，导入 GPT2 模型到本地:

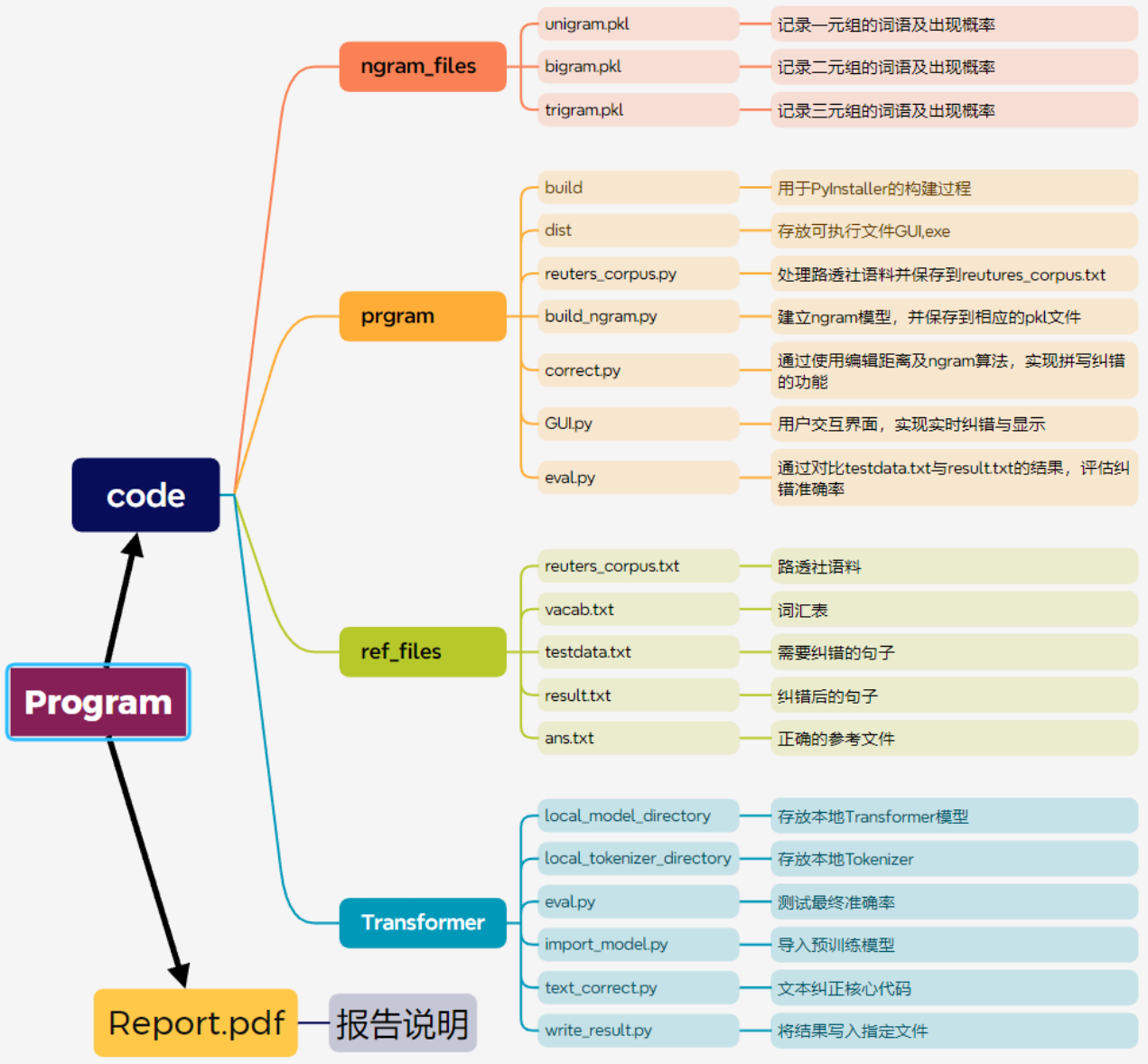
```
from transformers import GPT2LMHeadModel, GPT2Tokenizer
```

基于 text-generation 任务设计了句子生成函数，结合语义信息和 LD 编辑距离信息对句子进行生成，通过生成句子的方式来达到文本纠错的目的，核心思路如下：

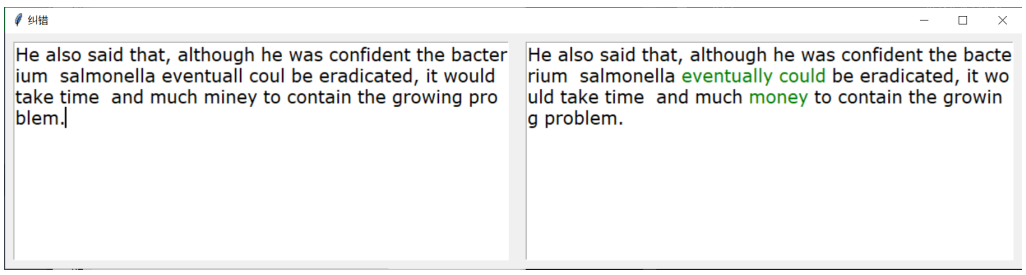
```
def sentence_generate(sentence):#输入含有错误拼写的句子
    prompt = sentence[0]#以句子的第一个词作为输入初始化
    probs, tokens= GPT_Calculate(input, device)#计算topk个可能出现的token以及概率
    for iter in range(1,len(sentence)):#依据句子长度进行循环
        for i in range(k):#对k个token进行筛选
            dis = LD_dis(token[i], sentence[iter])#计算token和当前输入词的编辑距离
            if dis > 3 or dis/len(sentence[iter]) > 0.3:
                continue#编辑距离太长或者错误比例太高舍去该token
            else:
                candidate.append(token[i])#将token加入候选
        candidate.avg(prob)#对候选token的prob归一化并排序
        score = calculate(prob, error)#根据编辑距离和原始概率重新计算复合得分
        candidate.sort(score)#根据新的复合得分对候选重新排序
        if candidate[0].multi_prob < threshold:#排名第一得分小于阈值
            prompt += sentence[iter]#不进行修改
        else:
            prompt += candidate[0]#否则修改为最可能出现的词
```

2 实验内容

2.1 项目结构



2.2 UI 设计



3 实验结果

3.1 结果展示

本实验的模型由两部分组成：传统模型以及 Transformer 模型。

传统模型由 n-gram 算法和噪声信道模型组成，在 1000 个测试样本中正确率为 91.4%，其中有 49 个句子修改错误，另外还有 37 个句子无法识别错误。我们发现不能识别的 37 个句子中的错误类型主要是 real word 类型，这样的错误结合语义信息进行修改更为有优势。

对这 37 个样本，我们又构建了 Transformer 模型来解决语义问题，结果 Transformer 模型能正确修改其中 23 个错误。对于剩余的错误分为两种类型，一种是 Transformer 计算的复合得分低于阈值不进行修改，一种是正确修改了拼写错误的单词，但同时也修改了其他低频词汇。

结合两种方法，对可以识别错误的句子使用传统算法，对不能识别的错误使用新算法，我们最终得到的正确率为 93.7%。

```
Accuracy in N-gram model is : 91.4%.    914 correct in 1000 examples(49 wrong and 37 can't recognize).

Accuracy in Transformer model is :  62.16216216216216 %.    23 correct in 37 examples.

Accuracy in total is : 93.70% .    937 correct in 1000 examples.
```

3.2 错误结果示例

3.2.1 传统-编辑距离问题

209 Citibank lost 490,000 crowns in Norway in 1985, but Sejerstad said a profit was likely this year because of planned liberalisation and better economic performance, helped by a steadier oil price of around 18 dlrs a barler.(barler—>barley)

为了加快运行速度，优先考虑了编辑距离为 1 的词语。所以只考虑了'barley', 'barber' 这两个导致的错误。因为大部分都是编辑距离为 1 的，如果让全部都同时考虑编辑距离为 2，不仅运行速度会慢，还会把更多原本正确的词改错。

3.2.2 传统-单复数问题

439 Two months of strikes in the sector began on January 19 in protest at employers' proposas for 350 redundancies from the 4,000-strong workforce this year.(proposas—>proposal)

语料库中单数的词出现的多，自动计算频率就大，模型就会选择出现频率更高的单数词汇或者复数词汇。

3.2.3 Transformer-低频词汇替换问题

139 Demands that Japan open its farm products market, will tell U.S. Officials at talks latter this month that liberalisation would harm existing U.S. Farm exports to Japan, a senior ministry official said.(latter—>later,farm—>arm)

Transformer 虽然能正确地修改 latter 的错误，但同时也会把原本正确的 farm 改成出现频率更高的 arm，最终的结果仍然是错误的。

4 实验总结

4.1 实验特点

1. 传统模型使用了路透社的大型语料库，建立 ngram 语言模型具有全面性和准确性
2. 由于给的 vocab.txt 基本上包含 testdata.txt 中的所有正确词汇，所以在此任务中准确率较高
3. 候选词优先考虑了编辑距离为 1 的词语，加快了运行速度还提高了准确率，实现了实时纠正
4. Transormer 模型不依赖任何语料库，直接通过输入 token 并计算 attention 即可计算下一个值
5. 将 text-generation 任务迁移到 sentence correction 任务，提出了一个新的算法
6. 编写了简明易用的图形化用户界面，实现有提示的实时文本的纠错功能

4.2 仍然存在的问题

1. 传统 ngram 模型无法捕捉到更长距离的依赖关系和上下文信息
2. 模型的泛化能力较差，难以处理在词典之外的情况
3. ngram 模型只关注词的共现频率，而不考虑词的语法和语义信息，所以可能会出现纠错异常情况
4. 没有特定训练和微调 Transformer 模型，而是直接用预训练好的模型，如果专门训练效果应该更好
5. Transformer 的计算资源消耗太大，不能很好地满足实时性的要求
6. Transformer 会把原本正确的词汇替换为更高频和他更相近的词汇
7. GPT2 是单向预测模型，不能有效的利用全部的语义信息，如果换成双向的 Bert 效果可能更好

4.3 实验心得

通过本次实验，我们对 n-gram 语言模型和噪声信道模型有了更深的理解。n-gram 语言模型的优点在于实现简单且效率较高，但在处理长距离依赖时效果较差，通过合理的预处理、平滑技术和候选词选择策略，我们能够在一定程度上克服这些问题。而噪声信道模型则能够更好地处理拼写错误，提升拼写纠正的准确性。

在 Transformer 模型的迁移中，我们也遇到了一些挑战。例如，如何构建评估函数，如何保证生成的词语与原输入直接的关系，如何有效地平衡模型的复杂度和计算资源。这些问题促使我们不断思考和改进模型，从而提升模型的性能。

总的来说，本次实验不仅让我们掌握了拼写纠正模型的基本原理和实现方法，还培养了我们解决实际问题的能力。在未来的研究和应用中，我们可以尝试结合更多的模型并结合 Transformer 模型，进一步提升拼写纠正的效果。