人智导大作业报告：拼音输入法

张凯文 计12 2021010729

算法原理

本次作业的基本思路是使用隐马尔科夫概率模型（HMM）估计汉字序列的概率，并使用 viterbi 算法搜索概率最大的序列。具体而言，给定拼音序列 $x_i|i=1,2,\cdots,n$ ，我们希望输出概率最大的汉字序列（一句话） $w_i|i=1,2,\cdots,n$ ，其中每个汉字 w_i 来自拼音汉字表中 x_i 对应的汉字集合 $C(x_i)$（注意：拼音 x_i 是给定的固定值而非随机变量，起到限制汉字搜索空间的作用）。也就是说，我们希望最大化 $P(w_1w_2\cdots w_n), \text{where } w_i \in C(x_i)$ 。根据全概率公式：

$$P(w_1w_2\cdots w_n)=P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\cdots P(w_n|w_1w_2\cdots w_{n-1})$$

为了简化概率模型的复杂度，基于对汉字序列的马尔可夫假设，我们可以分别给出上述表达式的基于字的二元或三元估计：

- 二元模型（Binary Model）：

$$P(w_1w_2\cdots w_n)\approx P(w_1)P(w_2|w_1)P(w_3|w_2)\cdots P(w_n|w_{n-1})$$

- 三元模型（Triple Model）：

$$P(w_1w_2\cdots w_n)\approx P(w_1)P(w_2|w_1)P(w_3|w_1w_2)P(w_4|w_2w_3)\cdots P(w_n|w_{n-2}w_{n-1})$$

为了更准确地估计句首 w_1 的概率 $P(w_1)$ 以及优化对句尾的估计，我**额外引入两个特殊字符：`<start>` 和 `<end>`**（可以将这两个字符当作汉字看待，词频表中也会估计这两个字符出现的频率，具体方法详见数据处理部分），记 w_0 = `<start>`，w_{n+1} = `<end>`，则两个模型优化目标可改写为**最小化**：

- 二元模型（Binary Model）：

$$\sum_{i=1}^{n+1}-\log P(w_i|w_{i-1})$$

- 三元模型（Triple Model）：

$$-\log P(w_1|w_0)+\sum_{i=2}^{n+1}-\log P(w_i|w_{i-2}w_{i-1})$$

实际中，我们无法精确获知上述表达式中的条件概率 P，因此需要从语料库中统计词频，用频率估计概率。为了解决零概率问题，我们还需要对估计的概率进行一定的平滑处理（这在三元模型或更多元的模型中是极为重要的，可能的多元组总数指数型上升但语料库中覆盖的多元组有限，覆盖率的下降将导致对大量三元组概率的估计是0，这将对模型有毁灭性的影响，对此在后续实验部分将有所验证）：

- 二元模型（Binary Model）：

$$P(w_i|w_{i-1})\approx\alpha\frac{\text{count}(w_{i-1}w_i)}{\text{count}(w_{i-1})}+(1-\alpha)\frac{\text{count}(w_i)}{\text{Total}}$$

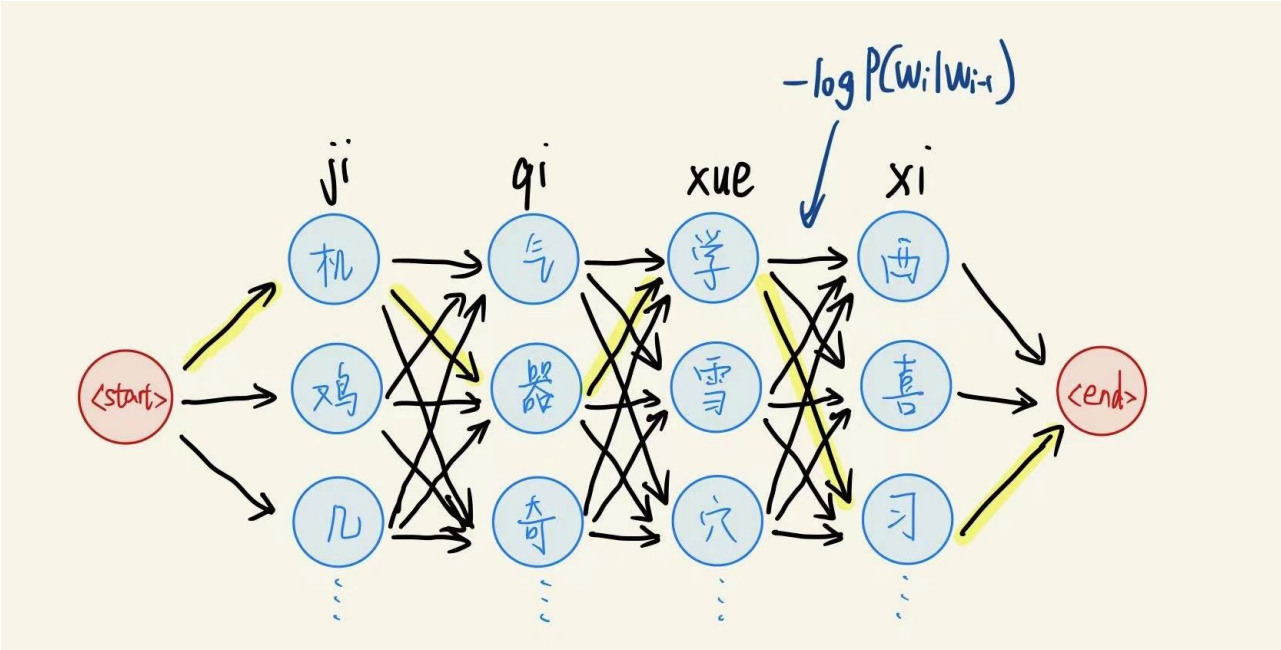
- 三元模型（Triple Model）：

$$P(w_i|w_{i-2}w_{i-1}) \approx \beta \frac{\text{count}(w_{i-2}w_{i-1}w_i)}{\text{count}(w_{i-2}w_{i-1})} + (1-\beta)\left(\alpha \frac{\text{count}(w_{i-1}w_i)}{\text{count}(w_{i-1})} + (1-\alpha)\frac{\text{count}(w_i)}{\text{Total}}\right)$$

其中 α、β 均为接近 1 的超参数，Total 为对语料库总字符数的估计。

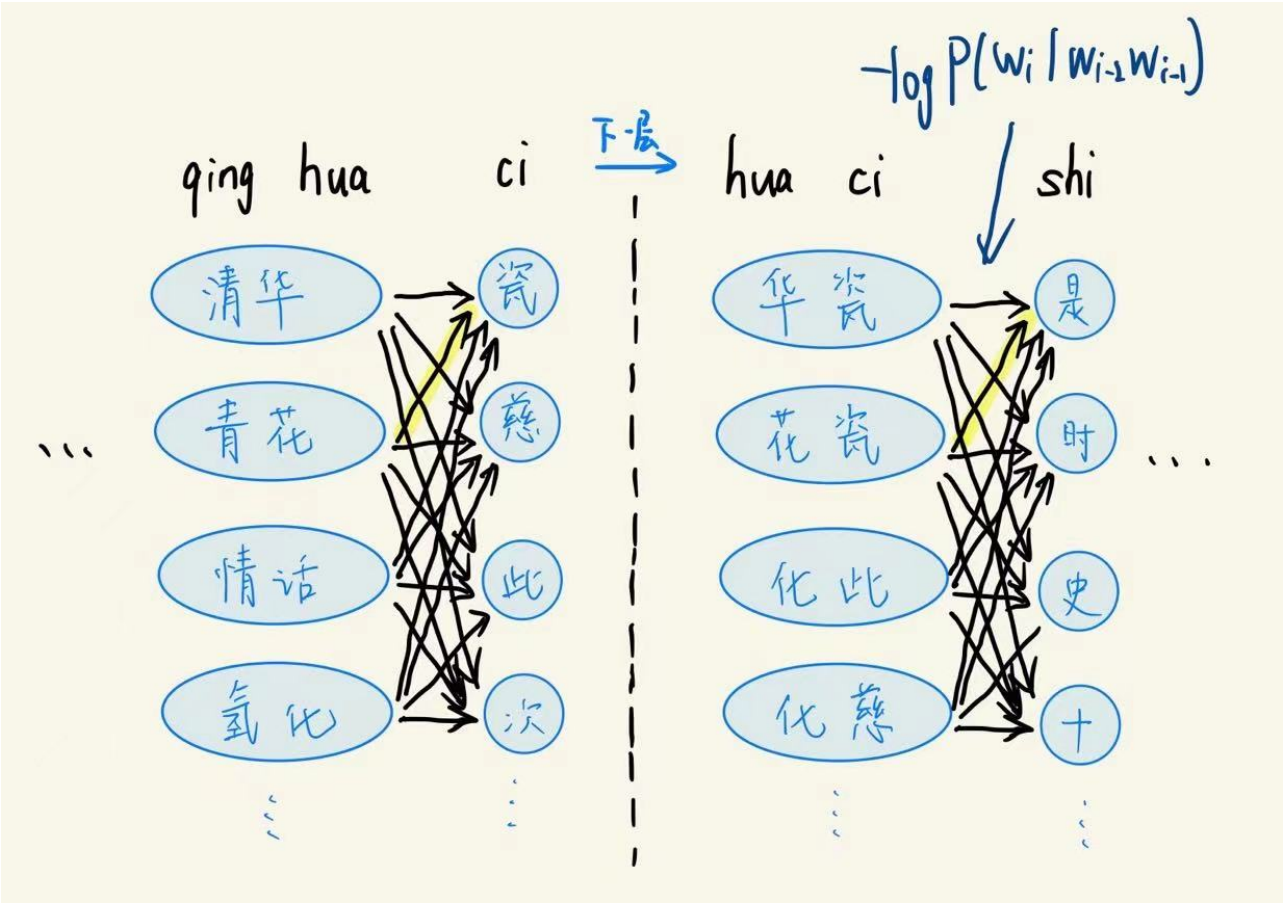
基于这一优化目标，我们采用 viterbi 算法对最优序列进行搜索：

- 二元模型（Binary Model）：将每个汉字视作结点，$-\log P(w_i|w_{i-1})$ 视作结点到结点间的路径长度，通过动态规划逐层推进，最优化目标即为找到从 w_0 = `<start>`到 w_{n+1} = `<end>` 的最短路径。



- 三元模型（Triple Model）：通过动态规划逐层推进，上一层的每个结点内为二元组合（$w_{i-2}w_{i-1}$），下一层的每个结点内为单字（w_i），转移路径长度为$-\log P(w_i|w_{i-2}w_{i-1})$（除了句首句尾处需要特

判），经过一层转移后更新当前层结点为新的二元组合（$w_{i-1}w_i$），同样是搜索全局最短路径。



算法实现

核心算法实现在 `src/models.py` 文件中。为了统一不同模型的接口和实现，我首先定义了一个结点类 `CharNode`，维护图中每个结点对应的字符值（可能为单个字或两个字）和到当前结点的 Top k 个最优路径，以及每个路径对应的总长度；并定义了一个抽象类 `PinyinIMEModel`，完成对单字频数表、拼音汉字表和 参数 `k`、`total`的基本初始化，且规定了拼音输入法概率模型的通用数据结构和接口：

- `node_layer`：维护当前层结点（`CharNode`） 的数据结构；
- `calc_path_cost`：计算语言模型中从一个节点转换到另一个节点的代价；
- `inference`：以一个拼音音节列表作为输入，返回一个与输入拼音对应的 Top k 个最可能的汉字列表。

`src/models.py` 中还对 `PinyinIMEModel` 提供了两个具体实现：二元模型 `BinaryModel` 和三元模型 `TripleModel`。`BinaryModel` 通过二元词频表计算二元条件概率的负对数作为路径长度，`TripleModel` 通过三元词频表计算三元条件概率的负对数作为路径长度，整体实现较为清晰、简洁、优雅。

评估实验

实验环境

- 系统配置：Windows 10 Intel(R) Core(TM) i5-8250U CPU
- python 版本：需要在 3.10 以上
- python 库：主要包含 `tqdm`、`argparse`、`pathlib` 等，除辅助进度可视化的 `tqdm` 外均为 python 自带库

数据预处理

在 `src/dataprocess.py` 中我实现了一个中文语料库词频提取的通用接口，并用它分别处理得到了基于给定的新浪新闻数据集（sina_news_gbk）和百度百科问答数据集（baike_qa2019）的二元、三元词频表，以及整合两个数据集的大词频表，每个词频表有其相应的压缩版本（删去频数小于等于2的项），均以上传至清华云盘。其具体使用方法可参见 `README.md` 中的详细介绍。需要特别说明的是，我将语料中出现的 `，` `。` `：` `、` `\n` 五种特殊符号和空格均视作句子间的分割符，以此来估计 `<start>` `<end>` 出现的频数和各个字符出现在句首、句尾的频数。与同学对比表明，加入这一先验后在预测句子首字符的准确率大有提高。

评价指标

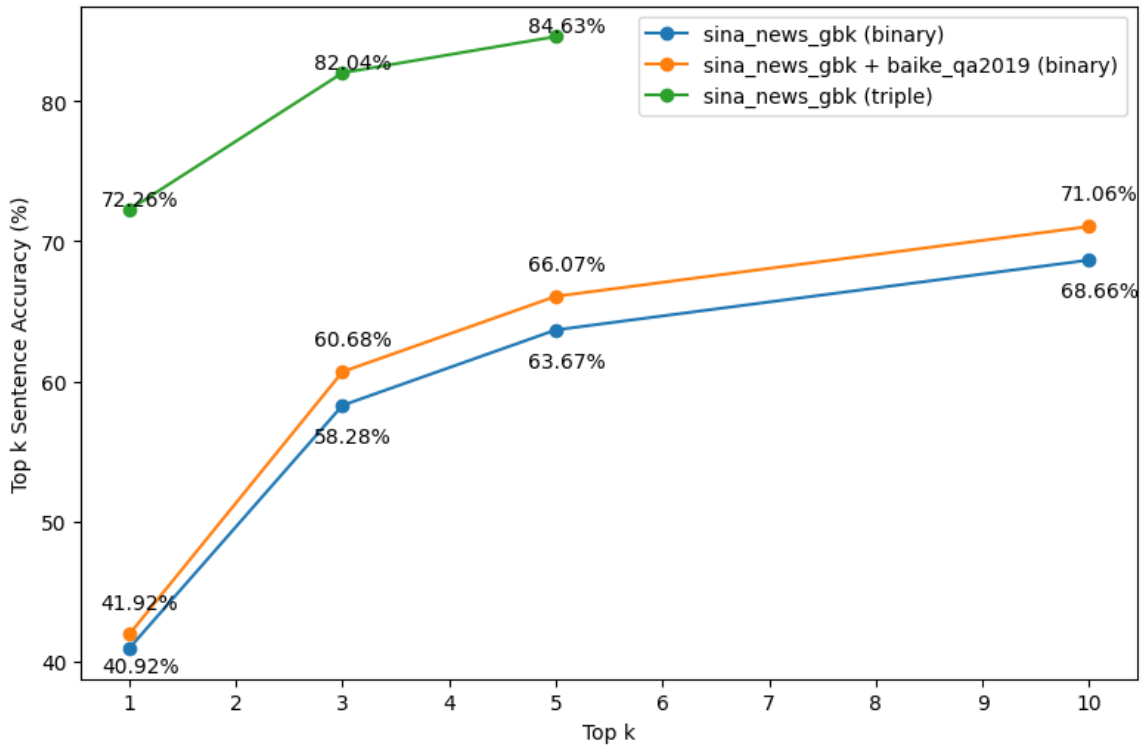
除了标准的字准确率和句准确率外，我还加入了 **Top k 句准确率** 作为评价指标，即模型给出的概率最大的前 k 个句子中有一个正确即算作正确。我认为这在拼音输入法中是非常实际合理的指标，在真实场景中的拼音输入法也会给出多个备选项，一般而言，只要在前 5 项中有一项正确结果，这个拼音输入法的结果都是相当不错的。

实验结果

字准确率对比：

训练数据集与模型	sina_news_gbk (binary)	sina_news_gbk + baike_qs2019 (binary)	sina_news_gbk (triple)
字准确率	84.47%	84.97%	93.45%

Top k 句准确率对比（Top 1 即为标准句准确率）：



> 注：由于三元模型时间复杂度较高（完成500句话的 Top 1 推理大约需要 5 分钟， Top 5 推理大约需要 20 分钟），且可以看到 Top 5 和 Top 10差距不大，故没有进行三元模型的 Top 10 实验。与三元模型相比，二元模型的主要优势在于时间复杂度低，Top 5 约在 15~20s 即可完成推理，Top 10 约在 1~2 分钟

> 完成推理，但准确率与三元模型有显著差距，且为模型先验假设带来的限制，增加数据集或增大 k 值改进能力也有限。

案例分析

> 注：此处均以标准新浪新闻语料库训练的结果进行分析

二元模型 Top 1 成功案例分析

```
ji qi xue xi shi dang xia fei chang huo re de ji shu
机器学习是当下非常火热的技术

zou zhong guo te se she hui zhu yi dao lu
走中国特色社会主义道路

de guo zong li mo ke er ri qian fa biao yan shuo
德国总理默克尔日前发表演说

chun jiang chao shui lian hai ping
春江潮水连海平
```

可以看出，由大量常见二元词语组成的句子是二元模型最擅长的（包括正确推理的古诗基本也符合这一特点）；此外由于训练语料的新闻特点，在新闻中常出现的词和句子具有更高的先验概率，也更容易成功推理。

二元模型 Top 10 成功案例分析

```
cheng nian ren de sheng huo li mei you rong yi er zi
成年人的生活里没有容易二字 (Top 4)

yong gan mao mao bu pa kun nan
勇敢猫猫不怕困难 (Top 6)
```

第一句话中"里"容易被写成"力"，"二字"容易被写成"儿子"；第二句话中"猫"容易被写成"毛""冒"，对此三元模型也没能成功处理好，这是语料库中概率分布的固有问题。但是词语本身仍是二元形式的，因此给二元模型一定容错率，允许它输出更多结果也能做好，且效率相较三元模型高很多。

二元模型 Top 10 失败案例分析

什么样的句子会让二元模型感到棘手？以下是一些三元模型 Top 1 就能做对而二元模型 Top 10 全都失败的案例：

```
wei ji bai ke shi yi ge wang luo bai ke quan shu xiang mu
维基百科是一个网络百科全书项目
['违纪伯克是一个网罗伯克全数项目', '违纪伯克是一个网络百科拳术项目', '违纪伯克是一个网罗伯克权属项目', '违纪伯克是一个网络百科全数项目', '为几百科室一个网罗伯克全数项目', '为几百科是一个网罗伯克全数项目', '为几百科室一个网络百科拳术项目', '为几百科室一个网罗伯克权
```

```
属项目', '为几百科室一个网络百科全数项目', '违纪伯克是一个网罗伯克全书项目']

nian nian bu wang bi you hui xiang
念念不忘必有回响
['年年不忘必有回想', '年年不忘必有回乡', '年年不忘必有回响', '年年不枉必有回想', '年年不忘比优惠享', '年年不枉必有回乡', '年年不忘比优惠祥', '年年不往比优惠享', '年年部网比优惠享', '年年不枉必有回响']

bu shi suo you ren dou neng gong cheng ming jiu
不是所有人都能功成名就
['不是所有人都能工程名酒', '不是所有人都能工程名救', '不是所有人都能工程明久', '不是所有人都能工程明就', '不是所有人都能工程鸣久', '不是所有人都能工程命旧', '不是所有人都能工程明酒', '不是所有人都能工程明究', '不是所有人都能工程命救', '不是所有人都能工程名就']
```

可见二元模型对于四字词语的把控能力非常差，其本身已经超出二元模型的先验假设边界。且由于二元模型的“视野框”较小，更容易受到多音字影响。一个有意思的现象是，由于一个字会同时与前一个字和后一个字产生概率关联但不会隔一个字产生关联，因此二元模型能够勉强通过 Top k 应对三字词语（如“青花瓷”），但对四字词语就几乎束手无策了；相应地，三元模型处理四字成语的能力就非常强。

三元模型成功案例分析

仅在新浪2016年新闻数据集上训练的三元模型 Top 1 准确率就能达到 72%，Top 5 准确率更是达到近 85%，效果是非常惊人的。除了上述提到的涉及“青花瓷” “维基百科” “念念不忘” “功成名就” 等三字、四字词语的句子三元模型能够在 Top 1 就正确解答之外，三元模型还能正确解答出 “新型冠状病毒” “坚决打赢疫情防控的人民战争” 这样让人感到 “超出语料库时间界限” 的句子。究其原因，我认为是三元模型能够给予语料库中出现过的、尽管频数较少的真实词句组合更高的概率偏好，以此惩罚了常见字在句子中的不常见用法的影响，一定程度上可以认为是增强的对长序列语义的建模理解。

三元模型失败案例分析

```
bei jing shi shou ge ju ban guo xia ao hui yu dong ao hui de cheng shi
北京是首个举办过夏奥会与冬奥会的城市
北京市首个举办过夏奥会与冬奥会的城市

yong du du du she hui bu hui bei she du si a
用毒毒毒蛇会不会被毒死啊
拥堵堵堵社会不会被堵死啊

ju jue nei juan
拒绝内卷
拒绝内圈
```

可以看出，概率过大但不合语境的三元组如“北京市”、语料库中可能完全没有出现过的新词“内卷”、语义过于复杂的句子是三元模型也无能为力的。

拓展思考

要进一步提升模型准确率，最好的办法还是从当前三元模型的局限性上入手革新模型。针对三元模型对复杂语义和句子整体理解能力不够的问题，可以尝试通过引入基于词的二元或三元模型改进，或引入语言模型。一种可能的方式是，先通过二元或三元模型输出大量 Top k 的可能结果，然后让分词模型或语言模型从句子整体角度计算句子的概率或 Perplexity 对结果重排，以此提高 Top 1 准确率。另外，目前的模型仅仅将拼音作为推理时限制搜索空间的信息，在训练时没有考虑字音信息，因此在处理多音字和某些歧义问题上可能存在天然的困难，可以尝试通过在训练集中引入字音标注进行改进。