**（4）基于流式文本生成模型的文本安全性检测方法**

流式文本生成模型可以理解为一类支持在处理文本流过程中实时返回识别结果的自动文本识别模型。该模型可应用于数据文本的扫描工作。因此，本项目可基于流式文本生成模型的安全检测方法，以检测输出文本的安全性。

流式文本生成模型一般可以定义为：持续接收文本流进行识别，并根据已经接收到的一部分 ，获取对应的后验概率最大的 Token 序列 ，因此其数学表达可以简单写为：



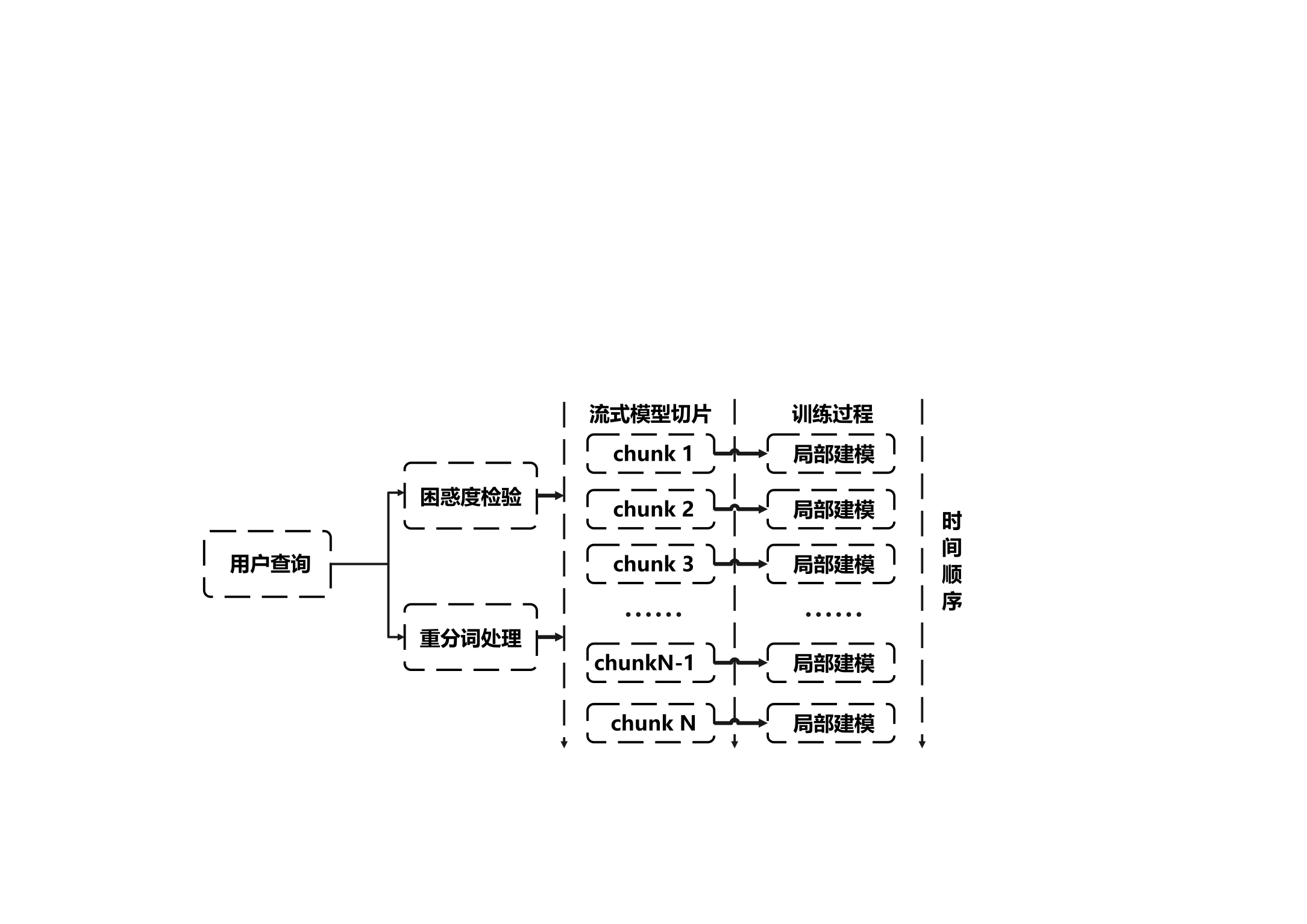
其中为已接收的文本特征流的当前帧索引。在实现中，允许模型有一定的延迟，此时虽然已经接收了帧输入，但实际解码到的是第帧，为有限的下文长度。或者对上文长度也有一定限制，即在对第帧后验概率建模的时候，只考虑指定长度的上下文内容：



和分别是上文（left-context）和下文（right-context）长度，则代表使用所有上文。本项目拟使用端到端流式文本生成模型实现文本数据扫描需求，端到端流式文本生成模型直接建模序列后验概率，序列整体的后验概率通过Token的后验概率累积得到，并假设当前 Token 只依赖于之前出现的 Token ，即：



在序列建模框架下，通过使用chunk方式可以间接实现局部建模，从而实现流式自动文本识别。具体操作是控制模型每次处理的上下文窗口大小，使得模型能够对序列进行局部建模，然后在计算损失函数时将局部计算的结果拼接起来。与语言识别不同，文本识别并不过分强调识别延迟，因此对于单个chunk的长度并没有过多的要求。因此，本项目设置一个较大的chunk size以扩充单个chunk信息量。建模过程及训练过程如图3所示：



**图3 流式文本生成模型建模及训练过程示意图**

大语言模型（LLM）掀起了自然语言处理（NLP）领域的一场革新。模型参数和预训练语料库的扩大，赋予LLM在各类NLP任务中卓越的能力。随着流式文本生成模型在各类自然语言处理任务中的广泛应用，其安全性和稳健性受到了越来越多的关注。尽管流式文本生成模型取得了显著成功，但它们有时也可能违背人类的价值观和偏好。此外，精心设计的对抗性提示可能引发LLM产生有害响应。即使没有遭受对抗性攻击，当前的LLM也可能生成不真实、有毒、有偏见甚至非法的内容。这些不良内容可能被滥用，产生不良的社会影响。因此，学术界和工业界有许多工作致力于缓解这些问题。但之前的研究多聚焦于评估模型的文本安全性，忽视了输出的稳健性。

在此基础上，本文提出了一种基于流式窗口文本检测的输出模块防御措施。

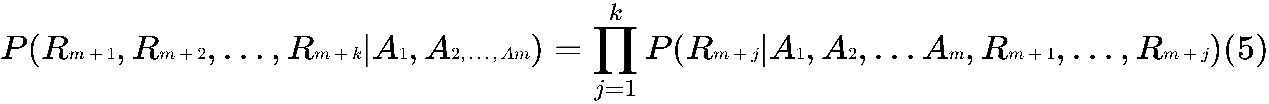
公式：

在输出端，利用第三方大模型对输出文本进行实时检测，进行分级的安全性检测。（考虑了文本的歧义性，仅仅进行单个token或者几个token的检测容易产生错误）

将文本输出端的文本标记为A = {A1, A2, . . . , An}，第三方大模型的回应记为R = {R1, R2, . . . , Rn}，则第三方开源大模型的输出可表示为如下概率公式：



由于任务目标的特殊性，我们锚定第三方大模型的输出为 “The text is harmful!” 当第三方大模型输出此文本时，我们认为原输出内容具有不安全性。此时第三方大模型的输出概率可表示为



当大模型的输出目标可以表示为式5，我们就可以用一个函数来计算输出的损失，这里我们采用Zou et al. (2023)引入的对数似然作为模型的损失函数，损失的计算过程可表示为



根据损失函数可求出每一部分相对于锚定输出的损失，当损失函数小于0.693，即模型预测概率大于0.5，认为输出文本的安全性较差，此时则返回s上一部分的安全内容，将不安全的输出中断。

方法

损失函数的计算将从文本级，句子级以及token级三个层级进行。

Token级对每个token进行损失计算返回loss值。若loss值符合要求，将token返回到文本中

句子级上对token进行重要度排序后对前n个token进行同义词替换，将替换后的token整合到句子中，对每个同义词组成的进行loss值的计算，取平均值后返回平均值与阈值比较，若处在阈值内，将原句返回文本

文本级上，使用第三方大语言模型进行句子替换，同样得到loss值后取平均值，若loss值符合阈值范围，则返回文本。

以上过程中，任意过程不满足损失函数值域阈值，则中断源文本的输出。

为了保证我们任务在黑盒模型上的有效性，我们使用分数表示单词在句子中的重要度。给定一个n个单词的句子记为，将删除某一词的句子记为，记预测结果，则可表示为删除前后预测分数的变化，定义如下：

