5.3文本纠正样例

|  |  |
| --- | --- |
| 原始文本 | 那年冬天竹母死了父亲的差事也交谢了正是祸部单行的日子 |

|  |  |
| --- | --- |
| 纠正文本 | 那年冬天祖母死了父亲的差事也交谢了正是祸不单行的日子 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| target | candidate | cos\_sim | pinyin\_sim | ngram\_sim |
| 竹 | 祖 | 0.6100631 | 0.7692307692 |  |
| 父 | 0.5683848 | 0. |  |
| 异 | 0.42654333 |  |  |
| 母 | 0.55227005 |  |  |
| 亲 | 0.5160396 |  |  |
| 部 | 不 | 0.6238163 |  |  |
| 补 | 0.61413115 |  |  |
| 女 | 0.48703188 |  |  |

[{'

char': '祖', 'cos\_sim': 0.6100631, 'pinyin\_sim': 0.7692307692307692,

'ngram\_sim': 3.2518814545525276},

{'char': '父', 'cos\_sim': 0.5683848, 'pinyin\_sim': 0.3846153846153846, 'ngram\_sim': 3.9463048549934747},

{'char': '异', 'cos\_sim': 0.42654333, 'pinyin\_sim': 0.2777777777777778, 'ngram\_sim': 2.315970345456918},

{'char': '母', 'cos\_sim': 0.55227005, 'pinyin\_sim': 0.4347826086956522, 'ngram\_sim': 1.0791812460476249},

{'char': '亲', 'cos\_sim': 0.5160396, 'pinyin\_sim': 0.30303030303030304, 'ngram\_sim': 1.591064607026499}]

[{'char': '不', 'cos\_sim': 0.6238163, 'pinyin\_sim': 10.0, 'ngram\_sim': 2.69284691927723},

{'char': '补', 'cos\_sim': 0.61413115, 'pinyin\_sim': 3.3333333333333335, 'ngram\_sim': 0.3010299956639812},

{'char': '女', 'cos\_sim': 0.48703188, 'pinyin\_sim': 0.4347826086956522, 'ngram\_sim': 2.5224442335063197}]

：

5.4 标点符号添加

语义识别模块输出的识别结果为原始的文本序列，只有汉语文字，不包含标点符号，而本文的自动字幕功能是需要将生成的文本字幕显示到用户客户端供用户浏览的，若字幕语句中没有标点符号停顿，会极大的影响阅览体验，所以本文在后处理阶段对文本进行标点添加操作，提高字幕结果的可读性。

标点添加功能具体的实现流程为，搭建一个深度学习模型预测标点符号的添加位置以及具体添加哪种标点符号，之后通过标点添加逻辑对语音识别模块的流式文本输出进行标点添加。

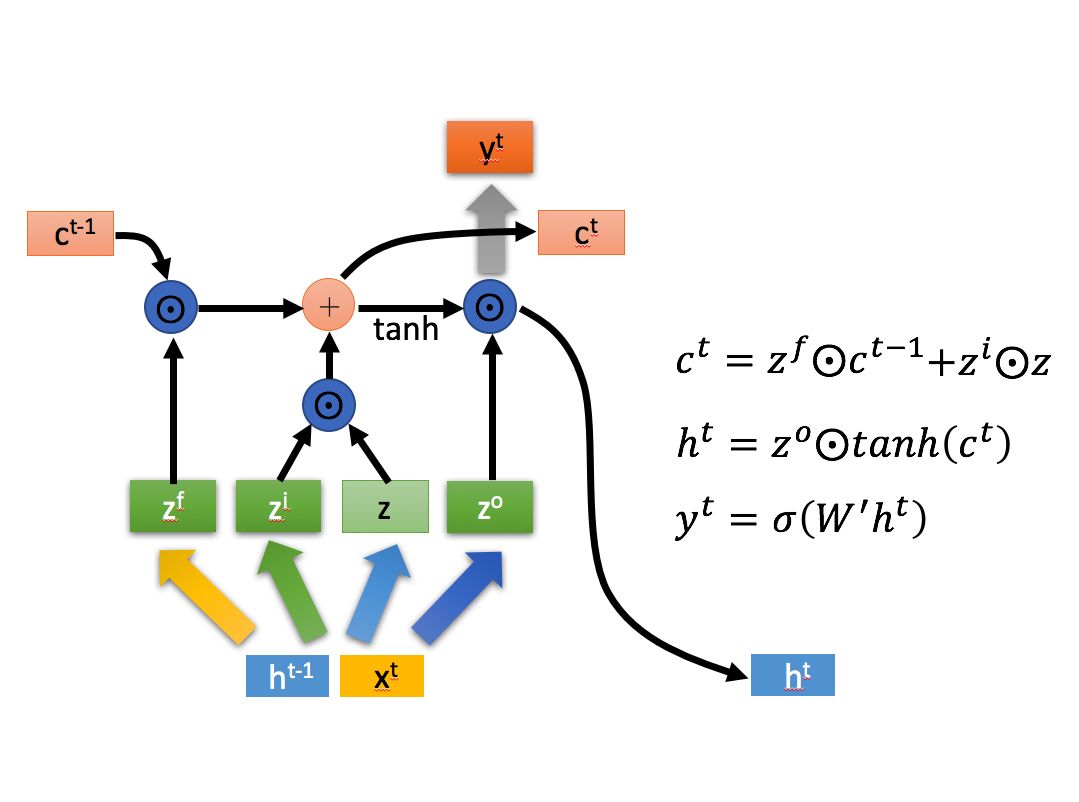
5.4.1 标点预测模型

首先是构建标点符号标注数据集，本文使用的语料数据同5.3章节用到的wiki中文语料相同，只是在其基础上进行了分词和标点符号标签标注工作。本文将标点符号添加任务抽象成序列标注问题，因为中文语句中是有字和词的区别，一般标点符号都是存在于一个完整词语的后面，所以首先对语句进行分词操作，一个词作为一个输入序列的token，对于每一个token，都有一个对应的符号标签，若这个token之后是一个标点符号，则将此标点符号作为此token的标签，若其之后不是标点符号，则将其标注为空白标签X，本文提供支持的标签序列为[**'X'**, **'，'**, **'。'**, **'！'**, **'？'**, **'、'**, **'：'**]，其标注数据集格式样例如表5.1所示。

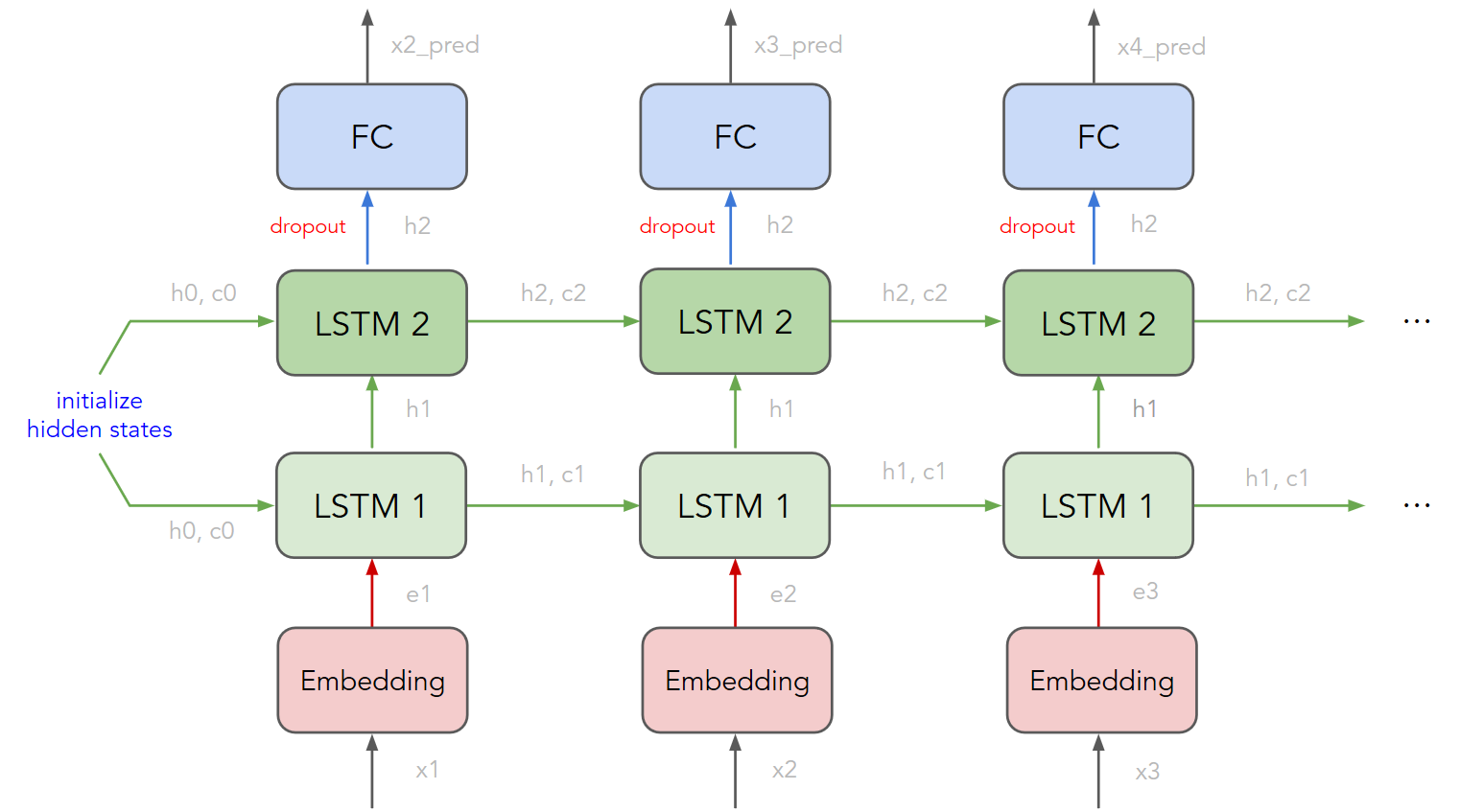
|  |  |
| --- | --- |
| 输入x | 标签y |
| 中国 河北省 一 烟花场 发生 爆炸 造成 多人 死亡 | X X X X X ， X X 。 |
| 第七届 世界 华商 大会 在 马来西亚 开幕 | X X X X X X 。 |
| 美军 占领 伊拉克 美军 今天 在 搜捕行动 中 遇袭 两人 受 重伤 | X X : X X X X X ，X X 。 |

本文使用的深度学习预测模型为BiLstm，对于序列标注问题双向循环神经网络RNN可以记忆提取上下文信息，考虑各个token之间的相关性，从而提高对当前时间步token的理解预测能力。但是传统的循环神经网络在考虑远距离历史信息时，由于训练过程中反向传播梯度消失问题，很容易将长期记忆特征丢失掉，不能考虑到全局语义，所以本文使用引入逻辑门控结构的长短期记忆单元Bilstm网络模型，其单元内部结构和计算公式如图5.7所示，主要是通过三个逻辑控制门输入门，遗忘门和输出门来控制时间步单元对当前信息，历史信息和当前输出的考虑权重，缓解RNN中各个时间步参数累乘造成的梯度消失问题，

提高模型的长期记忆能力和对序列的拟合理解能力。

本文基于wiki语料分词统计构造训练词典，共有352219个词条，如果和预测模型统一训练embedding，收敛会很慢，所以本文使用了开源项目Chinese Word Vectors（2018 Analogical Reasoning on Chinese Morphological and Semantic Relations）提供的预训练词向量sgns.wiki，其基于wiki中文语料使用word2vec模型训练所得，每一个词用300维向量表示，在本文的标点添加模型训练过程中，通过加载此预训练词向量，可以极大提高训练收敛速度。

本文的标点添加模型结构如下图5.8所示。

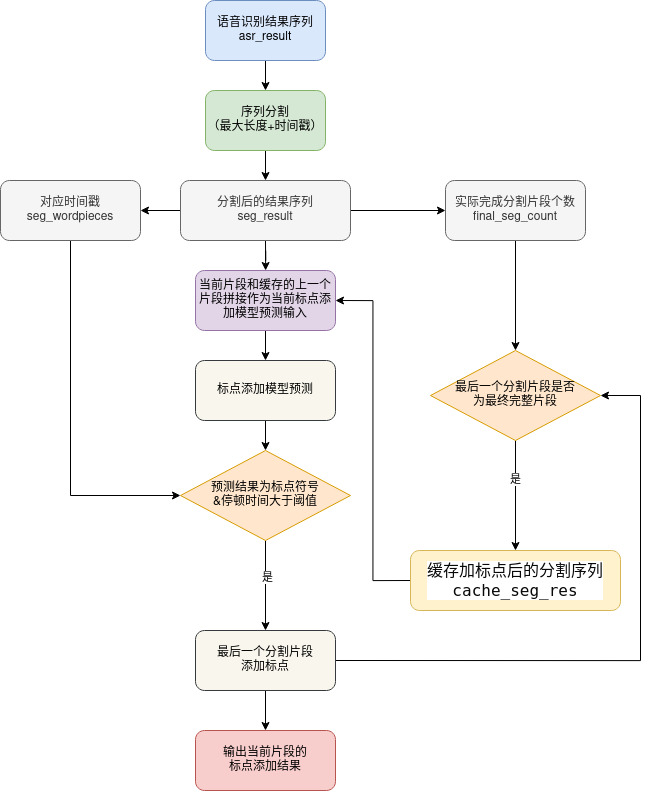


原始的输入token序列经过embedding层获得对应的嵌入向量表示，再经过两层Lstm层抽取上下文信息，输出高层稠密信息表示的隐状态向量，最终通过一个全连接层将输出映射到任务标签维度，得到预测结果。

5.4.2标点添加逻辑

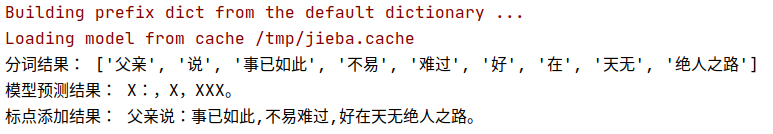
因为语音识别模块的输出方式是流式输出，每次返回结果会和上一次结果有重复的片段序列，而标点添加模型需要尽可能长的序列信息，才能考虑全局的上下文语境 ，得到较好的标点预测效果，但同时考虑模型推理的时间性能问题，输入也不能太长，所以折衷考虑两方面因素，本文设计了一些规则逻辑来实现流式序列标点添加功能。

具体的添加逻辑流程如图5.9所示。



首先根据语音识别模块返回的时间戳信息计算每个字之间的时间间隔，根据此间隔大小和最大长度阈值等规则对识别结果进行分割，得到分割后片段的序列结果和对应的时间间隔信息，以及当前已经识别的完成的分割片段个数。然后将当前片段和缓存结果中最后一个片段拼接，目的是在标点添加模型预测过程中能够考虑到更多的语境信息，然后将此拼接结果序列通过模型预测得到预测结果，同时在预测为标点的位置加入停顿间隔时长判断，若停顿时长小于最低阈值，证明此处预测结果不可信的概率较大，则去除添加此标点，较少错误识别。接下来根据完整分割片段个数确定是否将当前片段加入缓存。最终将当前添加标点符号之后的序列片段返回输出。

下图5.9为测试样例的标点识别结果。



5.5 生成字幕帧

5.6 本章小结