# **Project 思路计划**

## **API case整理：**

每一份代码，标注好里面包含的API、实现的功能。

标注情况如下表格：

|  |  |
| --- | --- |
| 代码case\_1 | 实现的功能(描述) |
| API\_1 |
| API\_2 |
| API\_… |
| 代码case\_2 | 实现的功能(描述) |
| API\_1 |
| API\_2 |
| API\_… |
| …… | 实现的功能(描述) |
| API\_1 |
| API\_2 |
| API\_… |
| 代码case\_n | 实现的功能(描述) |
| API\_1 |
| API\_2 |
| API\_… |

## **Project：**

**总体思路：**

**输入程序的功能描述，生成相应的python代码。**

**实现细节：**

1）基于深度学习，使用Encoder-Decoder的模型架构

Encoder：Bi-LSTM，对自然语言进行编码

Decoder：LSTM，通过多步解码出结果

2）引入编译原理中的“Abstract Syntax Tree（AST）”概念，将原本“自然语言->程序代码”的 直接过程变为“自然语言->AST->程序代码”的两步过程

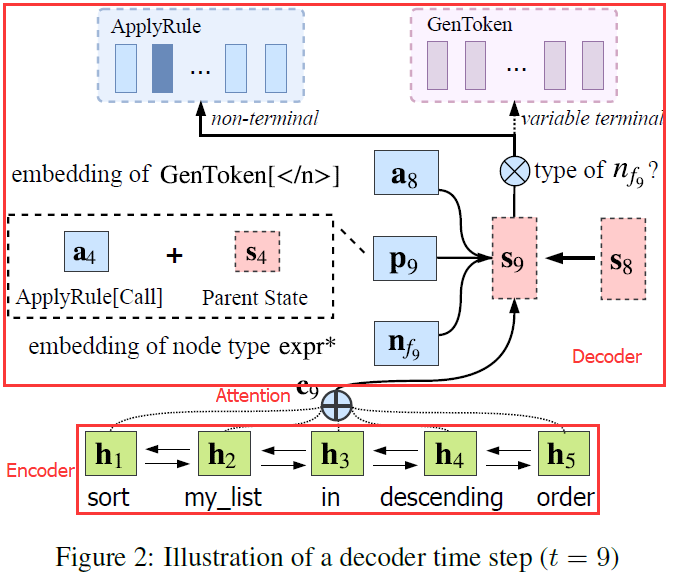
3）借助编译器相关的工具，AST与程序代码之间可以无损、确定性地互相转换

学习目标减小为“自然语言->AST”，编程语言的语法作为先验信息融入到模型中

模型不需要额外去学习编程语言的特定语法，模型的参数搜索空间被缩减了，降低优化难度

模型能专注于学习如何把自然语言抽象成AST

得到AST，就能保证得到语法正确的代码



## **组员：**

邓梓君，李可可，于田菲，晏明昊，贺海，吴章勇