摘要：

随着计算机的不断普及发展，编程能力已经成为了人们日常生活中，学生老师科研学习工作中一项不可或缺的能力与工具。编写程序能够帮助我们有效的，随心所欲的利用计算机，极大的提高我们工作学习的效率。然而，也不得不承认编写代码对很多人来说是一项相对困难而且耗时的工作，对于那些并非从事计算机相关行业的人来说，往往希望快速得到解决方案，编写自己需要的程序而不耗费额外而不必要的精力。

以JAVA语言为例，JAVA语言中有丰富的函数库API，可以帮我们解决很多复杂的问题，当有问题需要解决时，找到合适的API可以让我们对问题的解决事半功倍甚至直接解决我们当前的问题。因此，当我们有编程困难时，第一件事便是到网上查询API和code片段，帮助我们打开解决问题的思路。

传统的搜索方法通过自然语言来搜索自然语言，不适合检索代码。因此，我们针对Java编程语言，设计了编程助手，通过自然语言找到与之关联性最高的GitHub代码片段，并检索出需要使用的方法和API, 采用Seq2Seq神经网络模型，分别对代码和自然语言进行encoder,并训练自然语言与代码的对应关系，使我们可以通过自然语言，找到与之相关性最高的5组代码及代码使用的API反馈给用户。

Code\_Searcher采用神经网络训练模型,搜索结果相关性高，且虽然使用神经网络，但将后端代码数据训练与前端输入自然语言的训练分离，实现在搜索准确度高的同时搜索速度快。并为其搭建了Web客户端，方便客户使用。

关键词：编程助手，深度神经网络，GitHub，API，Web应用

绪论：

我们在编程时，常常有遇到问题不知道该如何去解决的困境，有类似困难时，我们做的第一件事往往是到网上找其他人的解决过类似问题是，或者

二：整体设计：

我们的整体设计主要针对代码数据和自然语言数据分别建立神经网络，并将二者分别映射到高维向量空间中去，以达到将两者均转化为数值向量的目的，最后，我们建立两者之间的联系，在抽去的encoder层，即代码层上增加线性映射层，将两个向量集合映射到同一向量空间，并通过对神经网络的训练。具体实现主要分三步：

1. 数据收集：我们从GitHub上获取真实的项目代码，为确保我们代码的有效性，我们对GitHub中低Star的代码进行了筛除，最终得到我们的代码数据库。
2. 得到代码数据库后，为了提取出代码特征，我们采取了seq2seq模型中encoder部分来将代码映射到高维向量空间，以达到获取代码特征的目的。
3. 为了获取查询使用的自然语言的语义特征，
4. 在第二部抽取的encoder层之上增加线性映射层通过训练这个网络，最终实现从code向量空间到自然语言向量空间的映射。得到该网络后，我们便将所有代码进行映射并将映射的结果储存到数据库中，最终，当用户进行检索之后，我们可以通过寻找在空间中距离用户搜索内容所对应向量距离最近的结果，并将结果返还给用户。从而实现我们的目的。

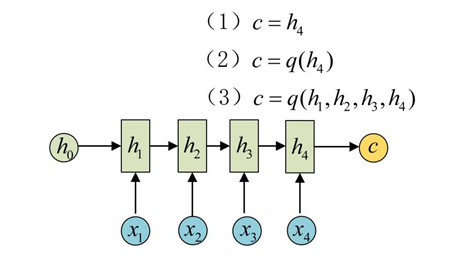
三：分步设计

（一）……

（二）code->vector

为了完成代码序列与自然语言序列的比对，我们首先要将它们都分别转化为高维向量空间中的向量，这样之后再通过向量之间的距离的比较来匹配代码和自然语言序列。

而在从代码到向量的过程中，我们采用的是深度神经网络中的seq2seq模型的encoder部分。seq2seq是RNN模型的一个变种，RNN（Recurrent Neural Network）是一类专门用于处理序列数据的神经网络，而seq2seq模型的特点则是输入与输出可以不等长，这也就方便了我们实现从代码到向量的转变。



如图所示，是我们的输入层，我们将代码从此层输入，经过等隐层的处理之后从c层输出，得到的就是我们需要的向量。