



# Intelligent Code Editor with Code Correction

Ben Gonner (Report Manager), Evan Christensen (Meeting Scribe),  
Jacob Puetz (Lead Engineer), Jordan Silvers (Meeting Facilitator),  
Cory Smith (Test Engineer)

Client: Hung Phan

Advisor: Dr. Ali Jannesari



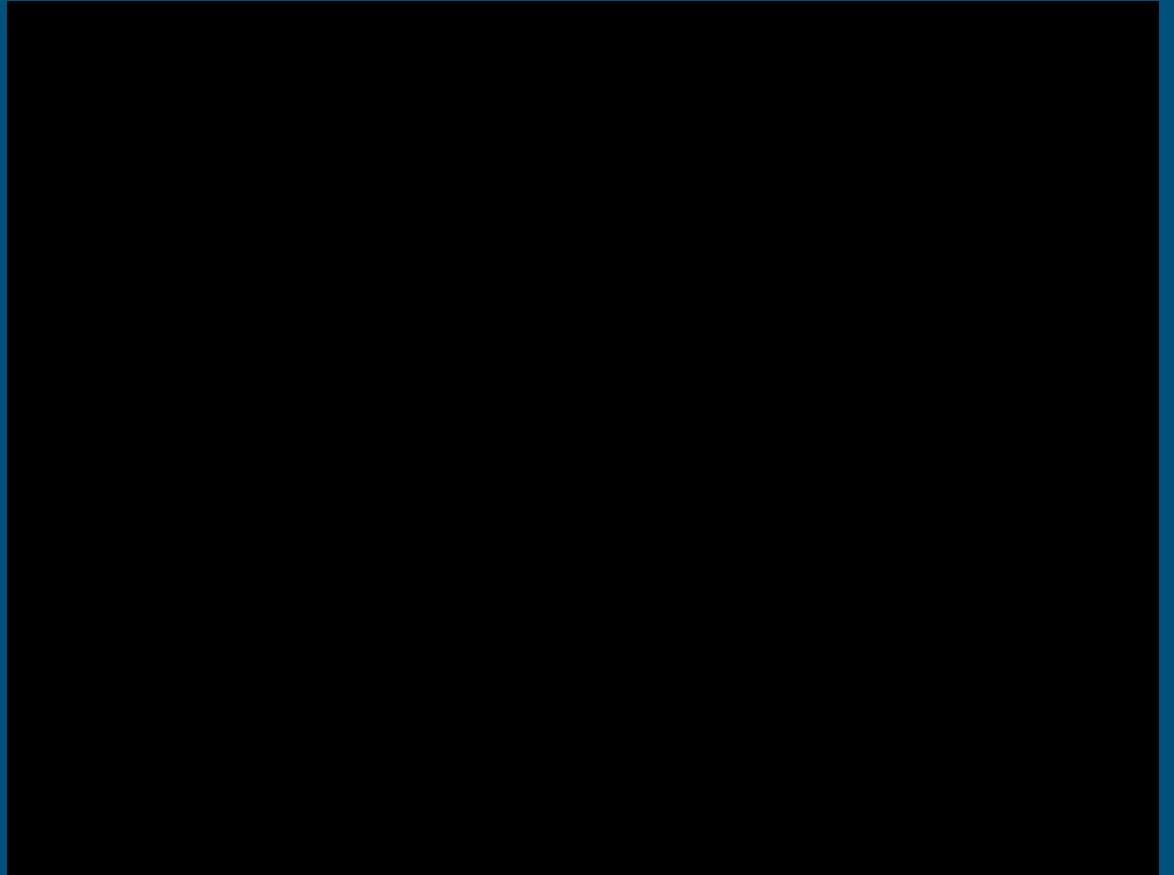
# Project Description

- Program that translates English to C++
- Program is designed to help people who struggle with converting ideas into a functional program
- Plugin for Visual Studio
  - Takes in a string from the editor
  - Classifies the string with a neural network
  - Fills in variable and function names from the original string
  - Replaces the text in the editor with a translated version

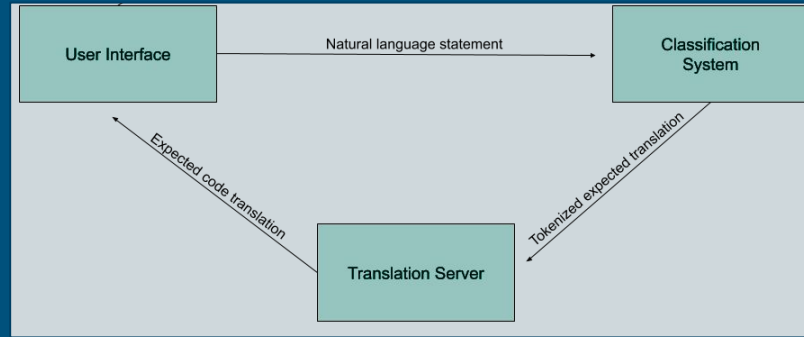
# Design Changes

- Switched from Java to C++
  - We had a readily available dataset for C++
- Updated our target platform from IntelliJ to Visual Studio
  - Language change from Java To C# for extension
- Adjusted our approach to use text classification instead of Neural Machine Translation
  - We had a bad time with Spektral

# DEMO - 1

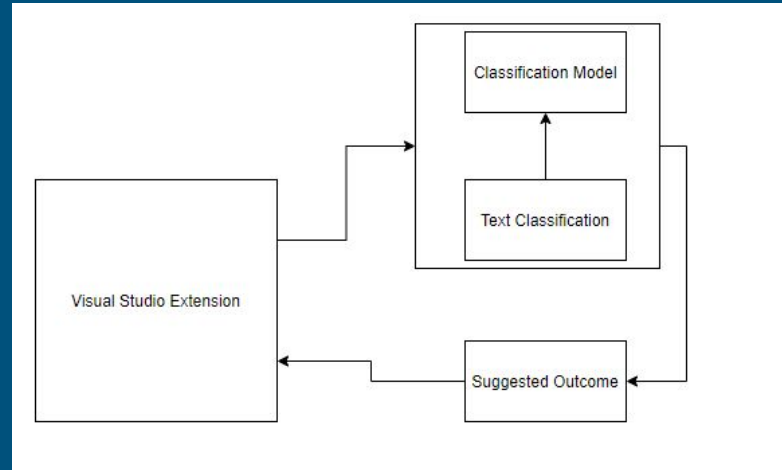


# Conceptual Design Diagram



# System Design

- Design Diagram



# Functional Requirements

- Ability for user to input code and natural language into an editor
- Translation should be fast as to not annoy the user
- Ability for plugin to classify and translate natural language to code
- Translation should only be initiated by the user
- Ability for code to be executed once translated
- Capability for multiple natural language statements to be translated to code at once

# Non-functional Requirements

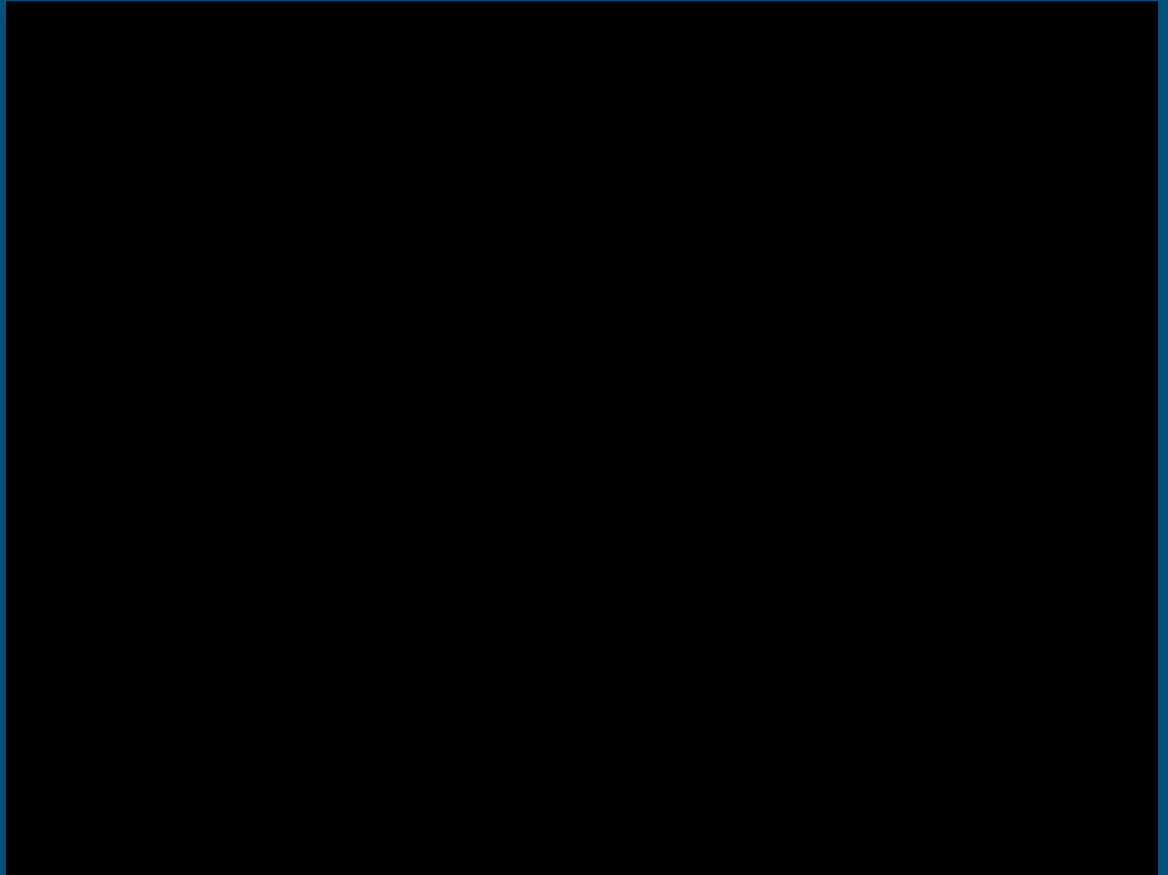
- Must have translational accuracy greater than 50%
- Can give 1 - 3 different possible translations
- The translation process should take less than 5 seconds
- Should be usable from anywhere - I.E. no internet required
- The plugin should be easy to use once added to the IDE



# Technical Requirements and other constraints

- Must use Visual Studio Extension development
- Graph Convolutional Networks will be used for natural language classification

# DEMO - 2



# System Design - Standards

- IEEE P14764
- IEEE 29119-2-2013
- IEEE P15026-2
- Agile Workflow

# Cyber Security

- Extension uses a local server
  - Low to no threat is involved here
- We are not stealing information
  - We used best practices to not take/leak information from the user's computer

# Implementation - Visual Studio

- Implemented through C# and .NET
- C# is friendly and works well with other code
  - Important if we wanted to implement python directly
- Extension easily accessible from the context menu
- Extension passes highlighted text to our translation model
  - The resulted translated line then replaces the highlighted line

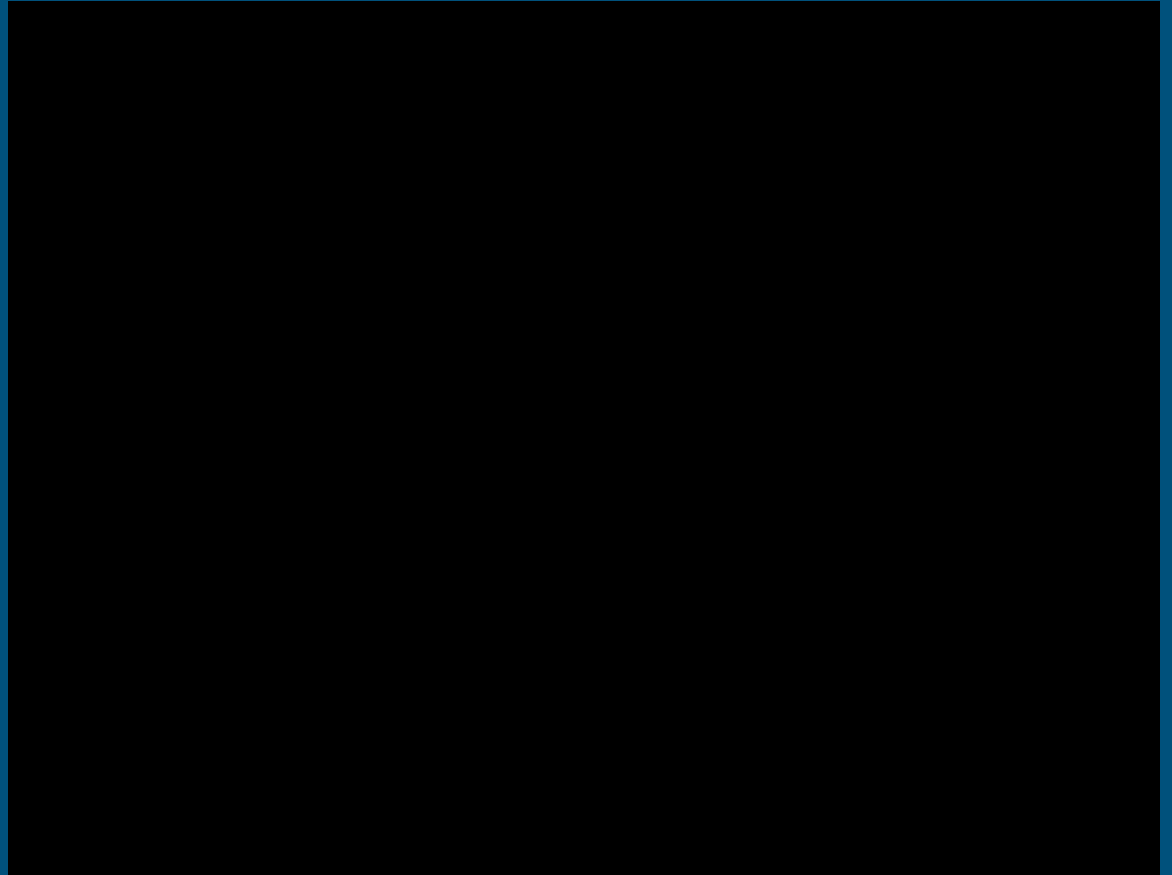
# Implementation - Training Data

- SPoC dataset - from SPoC paper in related literature (at the end)
  - Contains roughly 300,000 lines of pseudocode and its equivalent code
- Preprocessing system uses regex to parse code
- Then determines classification of each line of equivalent pseudocode
- It separates the data based on classification and places them into files to be used for training by neural network

# Implementation - Python Model/Neural Network

- Originally developed for the paper "Graph Convolutional Networks for Text Classification"
- Adapted for use with our project. It is a classification GNN based off of the TensorFlow library.

# DEMO - 3





# Related Projects

- Google Translate
  - While not directly related to our project, Google Translate does something similar to ours with taking one language and changing it to another language.
- Other Neural Projects
  - As we have seen in other papers on the subject, other projects out there exist with a similar task as ours. The difference between ours and theirs is that ours is basically open source. The other projects that have been talked about in other academic papers give no insight as to what is being done in the project and how it is being applied. There is also no code provided from these other projects so it is hard to tell if they are actually completed or more theory based.

# Related Literature/Bibliography

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, & Maosong Sun. (2019). Graph Neural Networks: A Review of Methods and Applications.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, & Yang Liu. (2016). Semi-Supervised Learning for Neural Machine Translation.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, & Marc'Aurelio Ranzato. (2018). Unsupervised Machine Translation Using Monolingual Corpora Only.

Miltiadis Allamanis, Marc Brockschmidt, & Mahmoud Khademi. (2018). Learning to Represent Programs with Graphs.

Liang Yao, Chengsheng Mao, Yuan Luo. "Graph Convolutional Networks for Text Classification." In 33rd AAAI Conference on Artificial Intelligence (AAAI-19), 7370-7377

SPoC: Search-based Pseudocode to Code from Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, Percy Liang