



Effect of Ontology-Based Data Augmentation on Code Generation with LLM and Automatic Generation

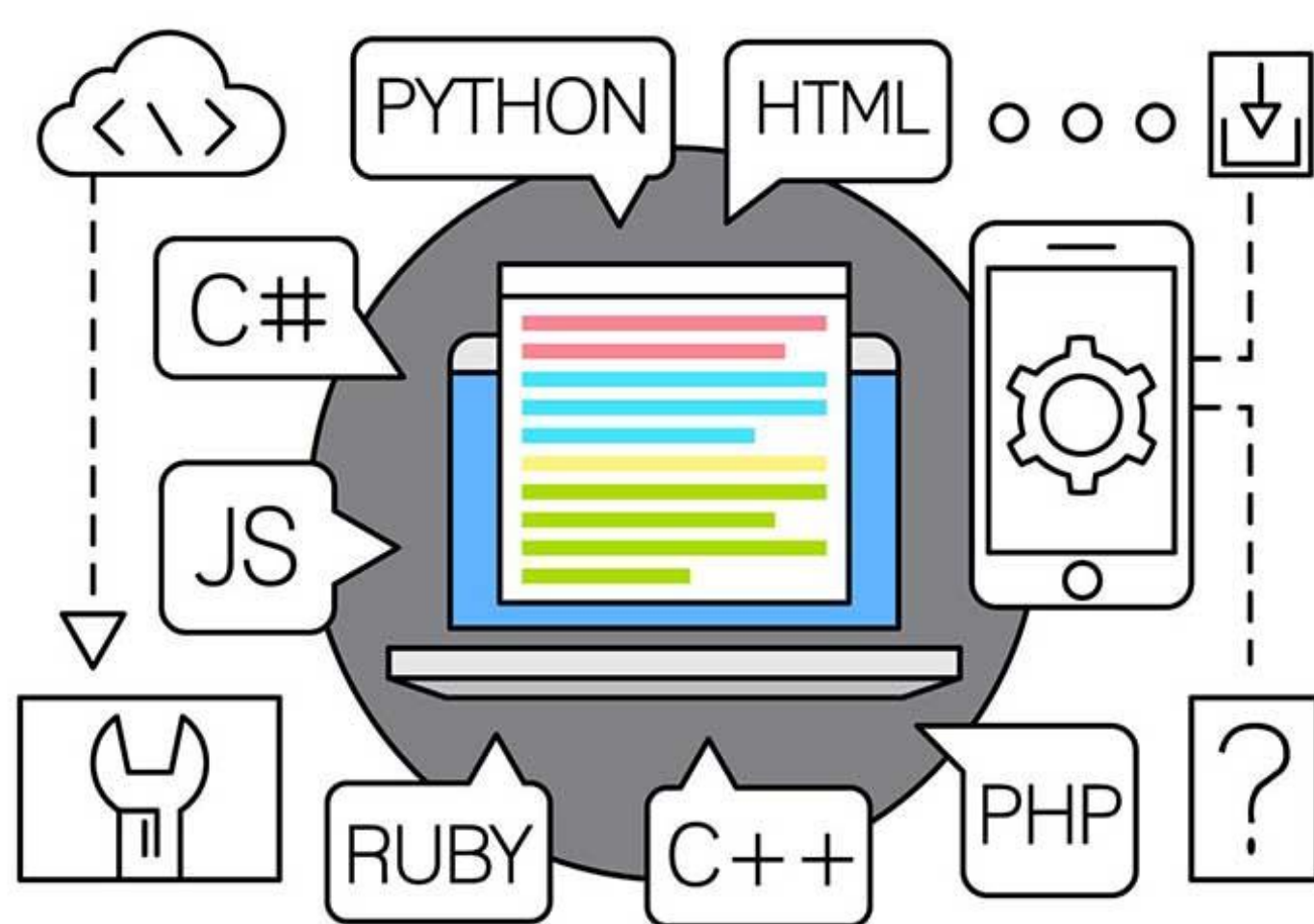
Poster ID: 9

m5271051 Keisuke Utsumi Supervisor Prof. Incheon Paik

In recent years, AI and deep learning have been used in many areas of society.

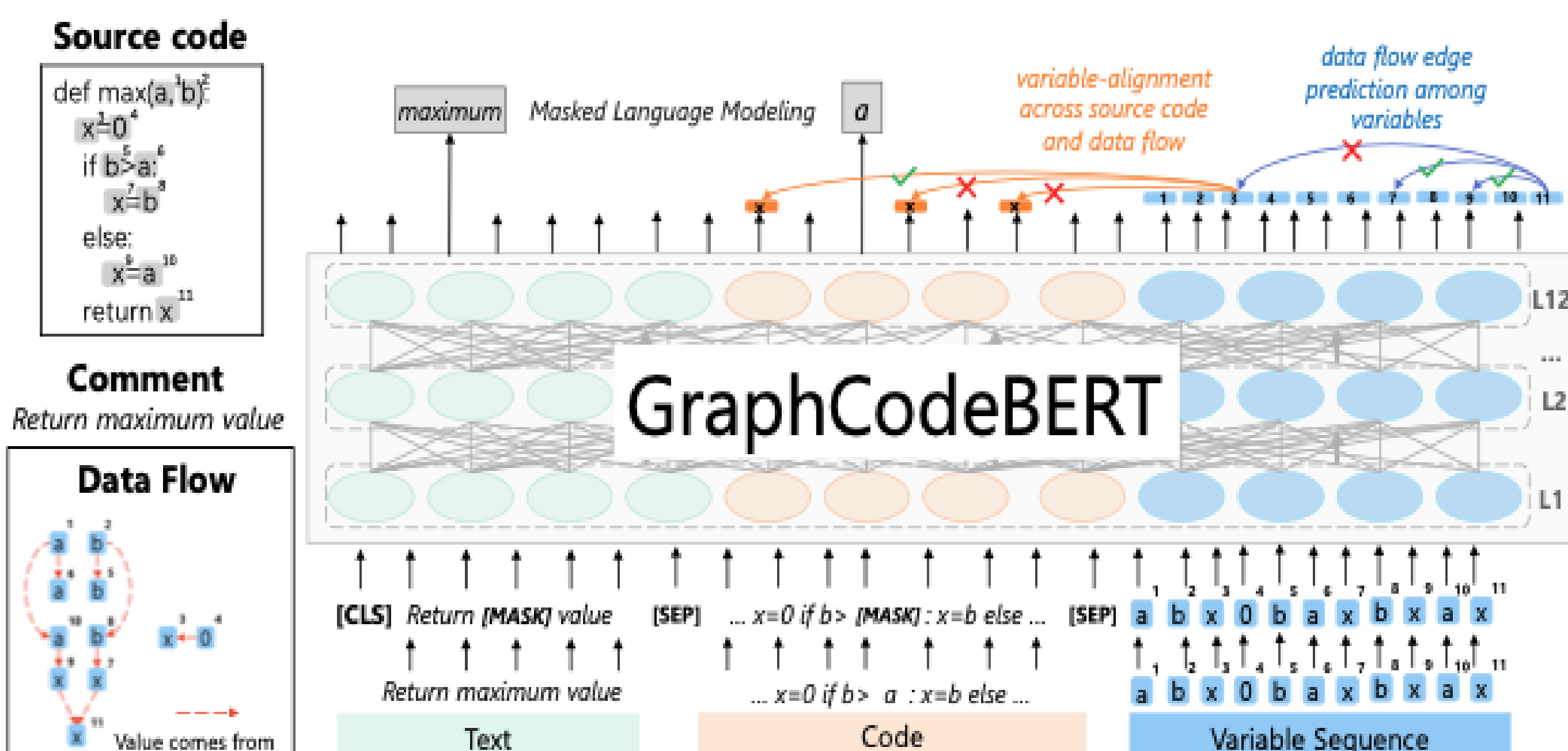
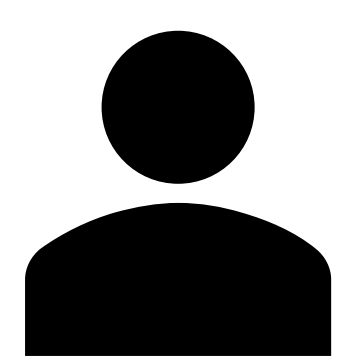
Applications such as Copilot and ChatGPT have helped engineers by improving their productivity by recommending and generating code.

Research will combine existing code generation using GPT and BERT with ontologies to study more accurate code generation and improve society's productivity.



Train on CodeSearchNet dataset (Java).

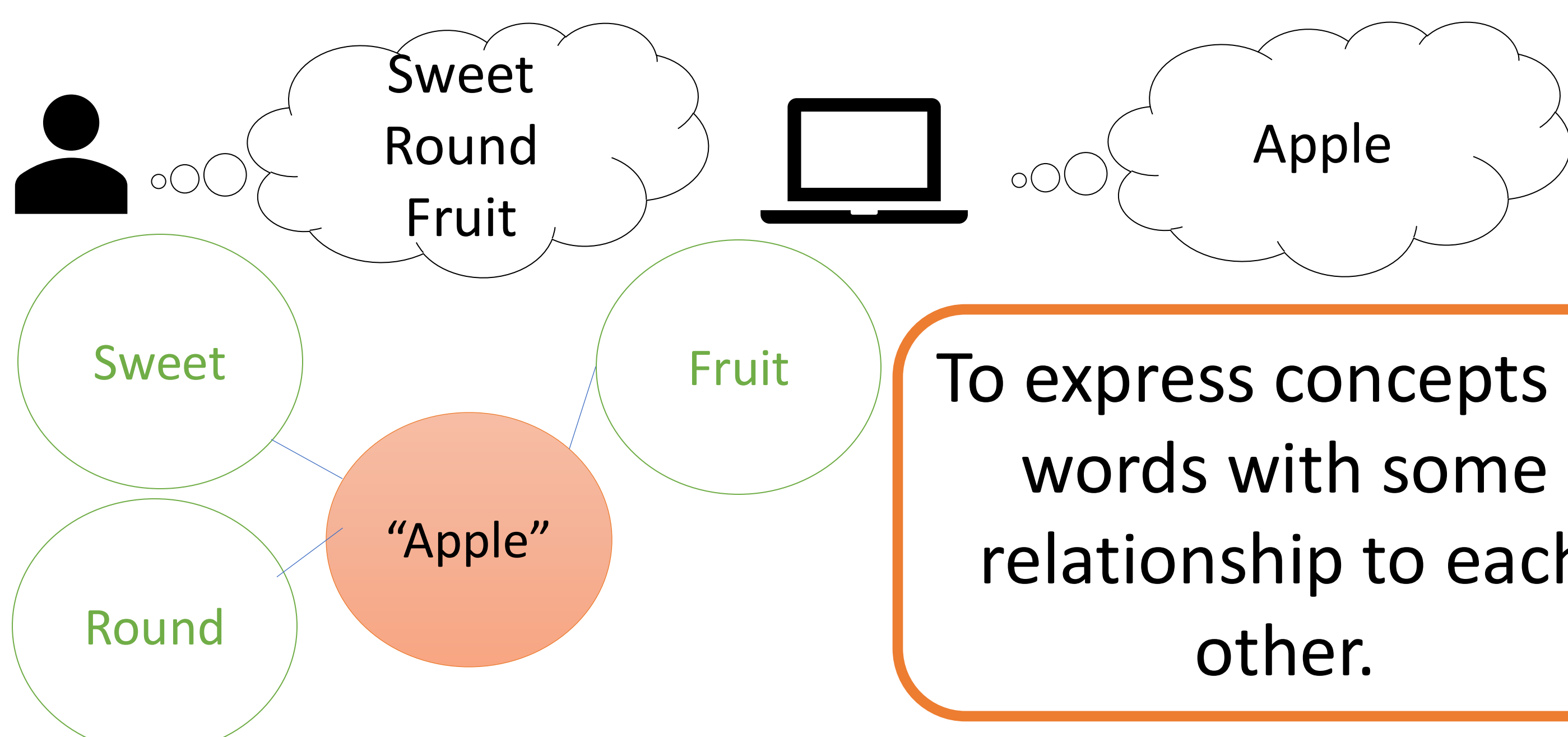
The study is trained on BERT. It also combines Code Graph, a prior study, to compare results. The combination of BERT and Code Graph collects code features by reference to an abstract syntax tree, which helps the model understand the relationships between variables and variable types in the code.



What is Ontology?

To express concepts or words with some relationship to each other.

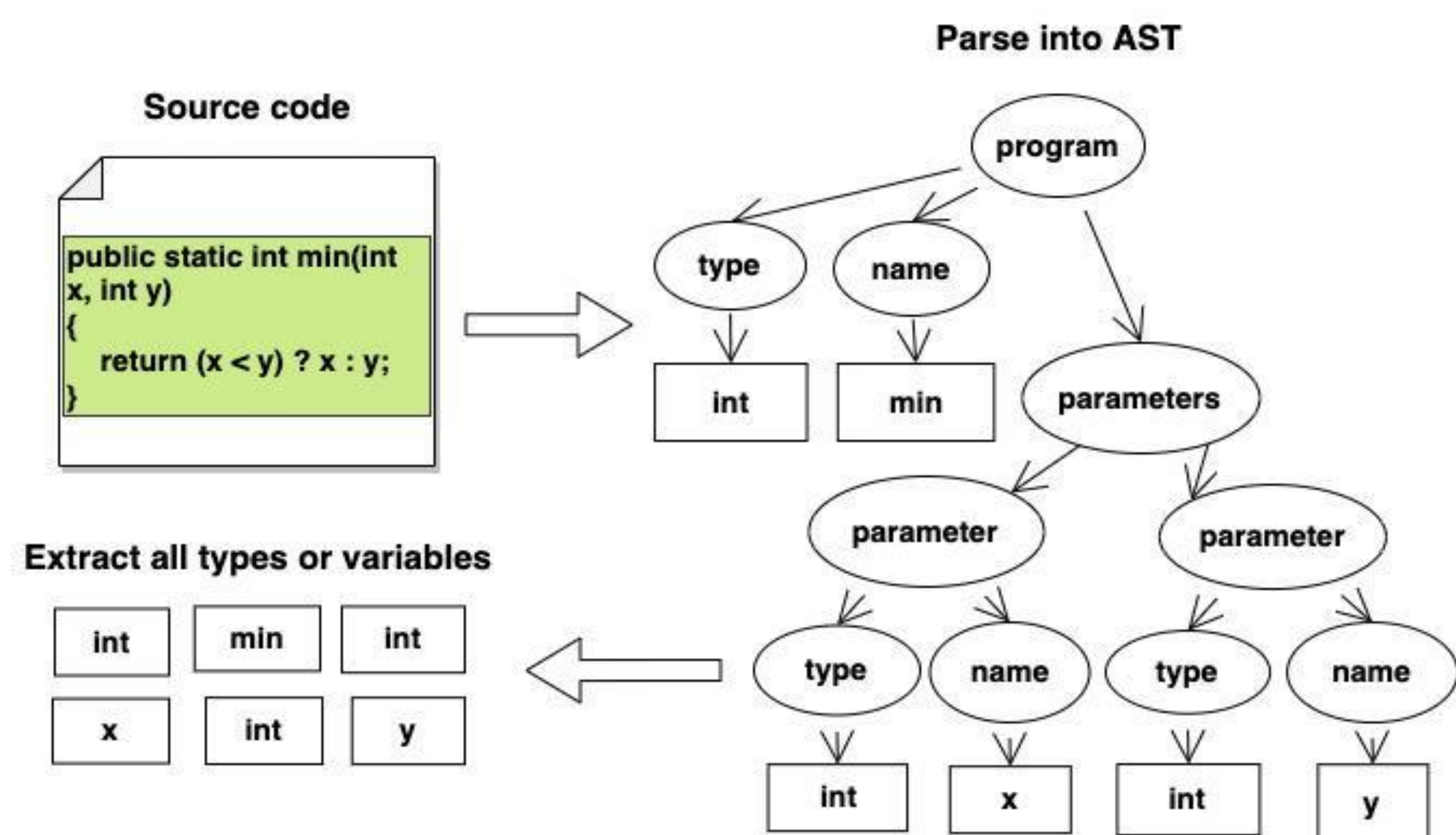
For example, if a human sees the word “apple,” it can automatically pull from memory the taste, color, and the fact that an apple is a fruit, but a computer can only see the word “apple” as a word.



In the previous research, all noun phrases were extracted from the SQuAD v2.0 dataset and word relations were mapped 1to1 as shown in the figure below.

An encoder-decoder model is used to reveal the part-of-speech and relations of words within comments.

This model is applied to each context of each sentence to help understand the structure of the sentence and generate the appropriate syntax tree.



The generated ontology is incorporated into the dataset by adding pairs of NPs and special tokens to the dataset and tokenizer. Related NPs are searched for in the comments and special tokens are added to represent them. This allows relationships such as superordinate-subordinate relationships and synonymous relationships to be represented.

Example

Create or replace [S:sub-sup] generate [E:sub-sup] [S:sub-sup] update [E:sub-sup].

Insert these ontologies using LLM

----Problem----

- Inserts the ontology into the prompt, but not into all data.

----Few shot learning----

- Few shot is an approach that gives multiple examples.

～Output Example～

No ontology inserted

```
public SchemaColumn asSchemaColumn () {  
    String columnAlias = ( m _ alias == null ) ? m _ columnName : m _ alias ;  
    return new SchemaColumn ( m _ tableName , m _ tableAlias , m _  
        columnName , columnAlias , m _ expression , m _ differentiator ) ;  
}
```

Insert ontology

```
public SchemaColumn asSchemaColumn () {  
    if ( m _ alias == null )  
        return this ;  
    else {  
        return new SchemaColumn ( m _ columnName , m _ alias , m _ tableName ,  
            m _ tableAlias , m _ columnName , columnAlias , m _ expression , m _  
            differentiator ) ;  
    }  
}
```