CODE ANALYSES USING NATURAL LANGUAGE QUESTIONS

Software Composition Seminar

Student: Michael Zbinden
Supervisor: Pooja Rani

29. May 2018

Problem

- Huge code base
- Finding classes

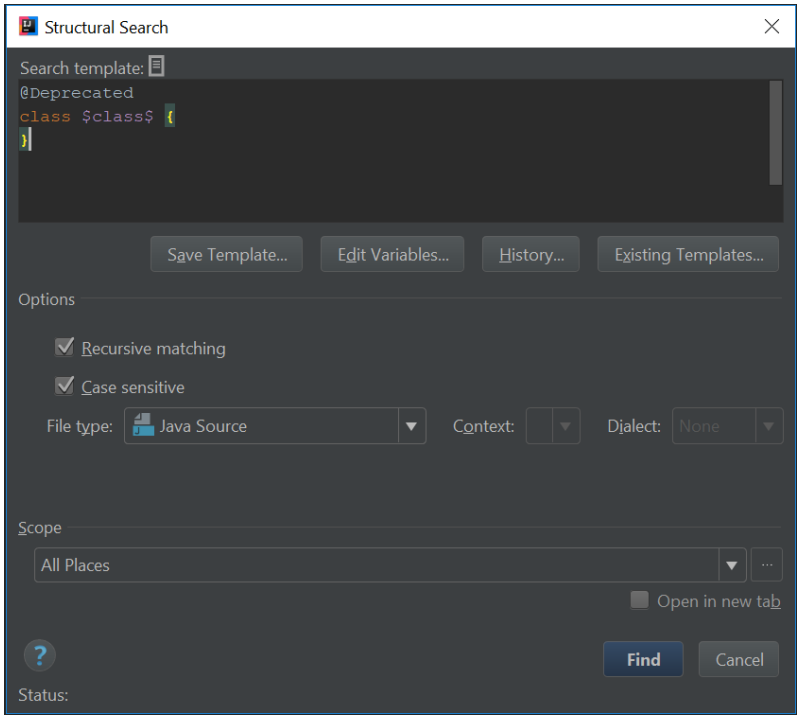
```
 * Represents the Singleton object AddressBook
 */
public final class AddressBook {

    private static AddressBook book;
    private List<IAddress> entries;

    /*
     * Load a CSV-String into the AddressBook
     */
    public void load(String csv){
        entries.addAll(CsvParser.csv2iAddress(csv));
    }
```

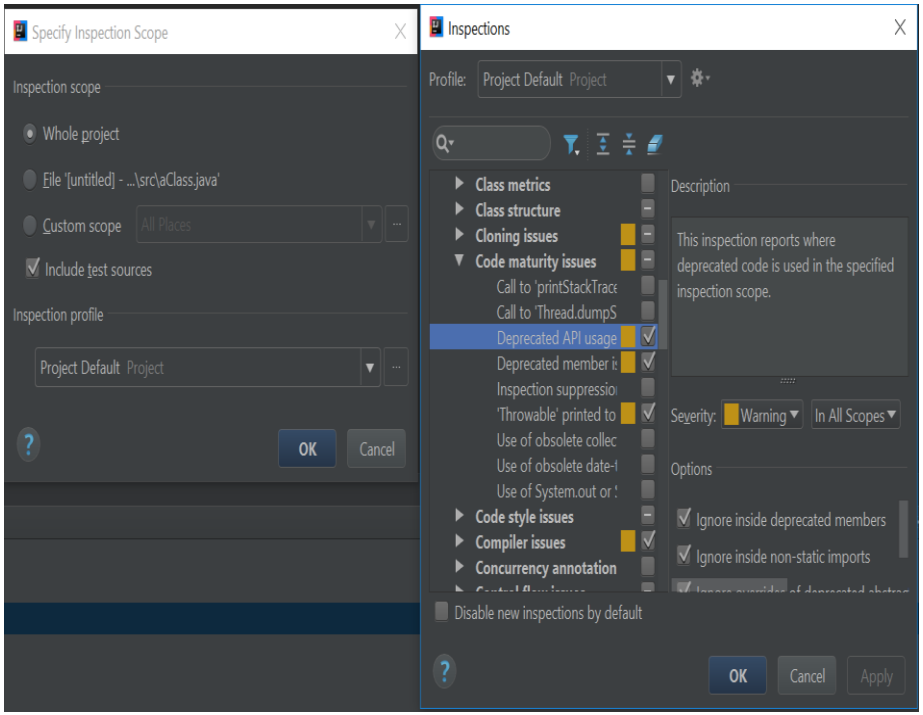
- Example: *What are the deprecated classes*?

```
AccessibleResourceBundle ar = new AccessibleResourceBundle();
Date date = new Date( year: 2018, month: 3, date: 14);
```

Tools Support

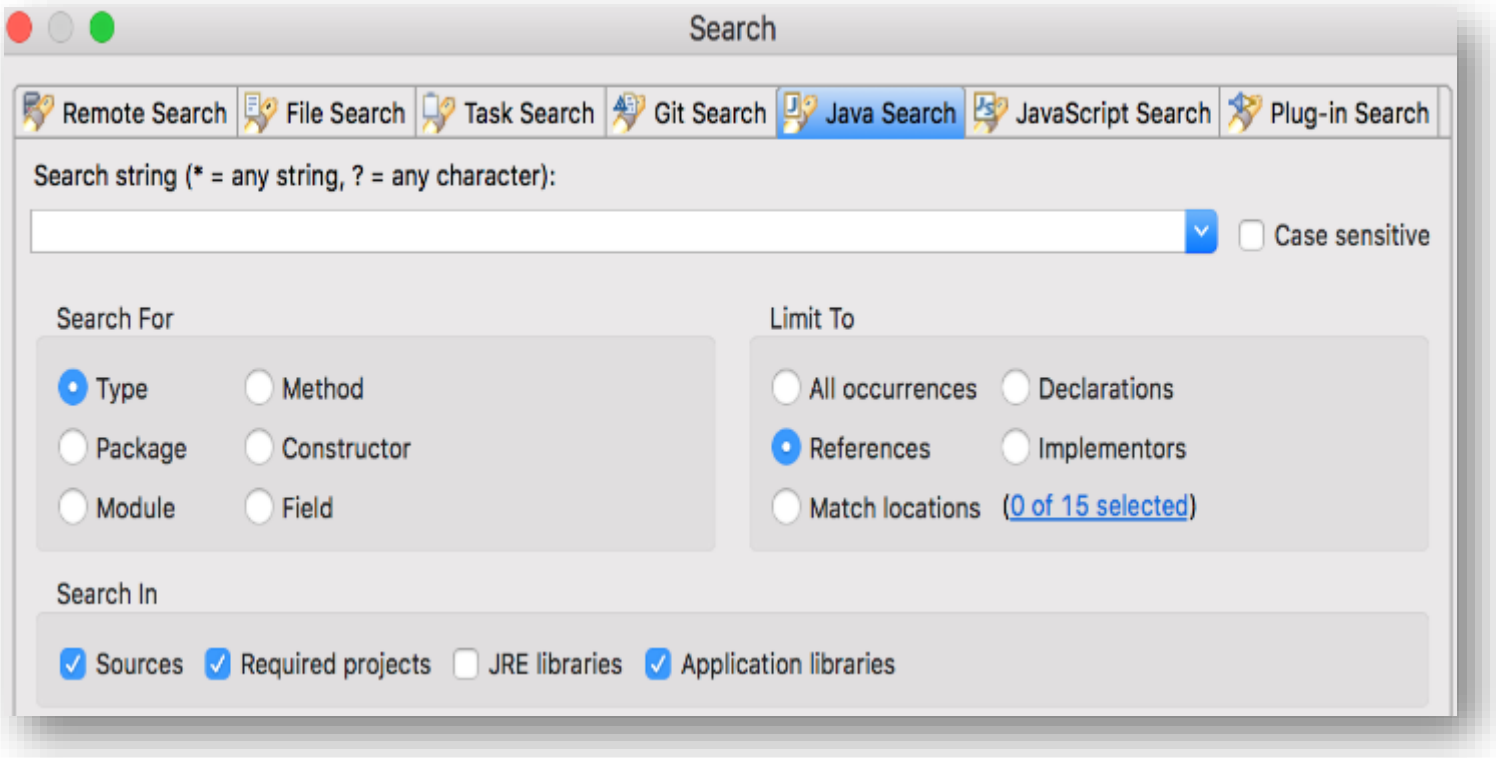
- IntelliJ Search



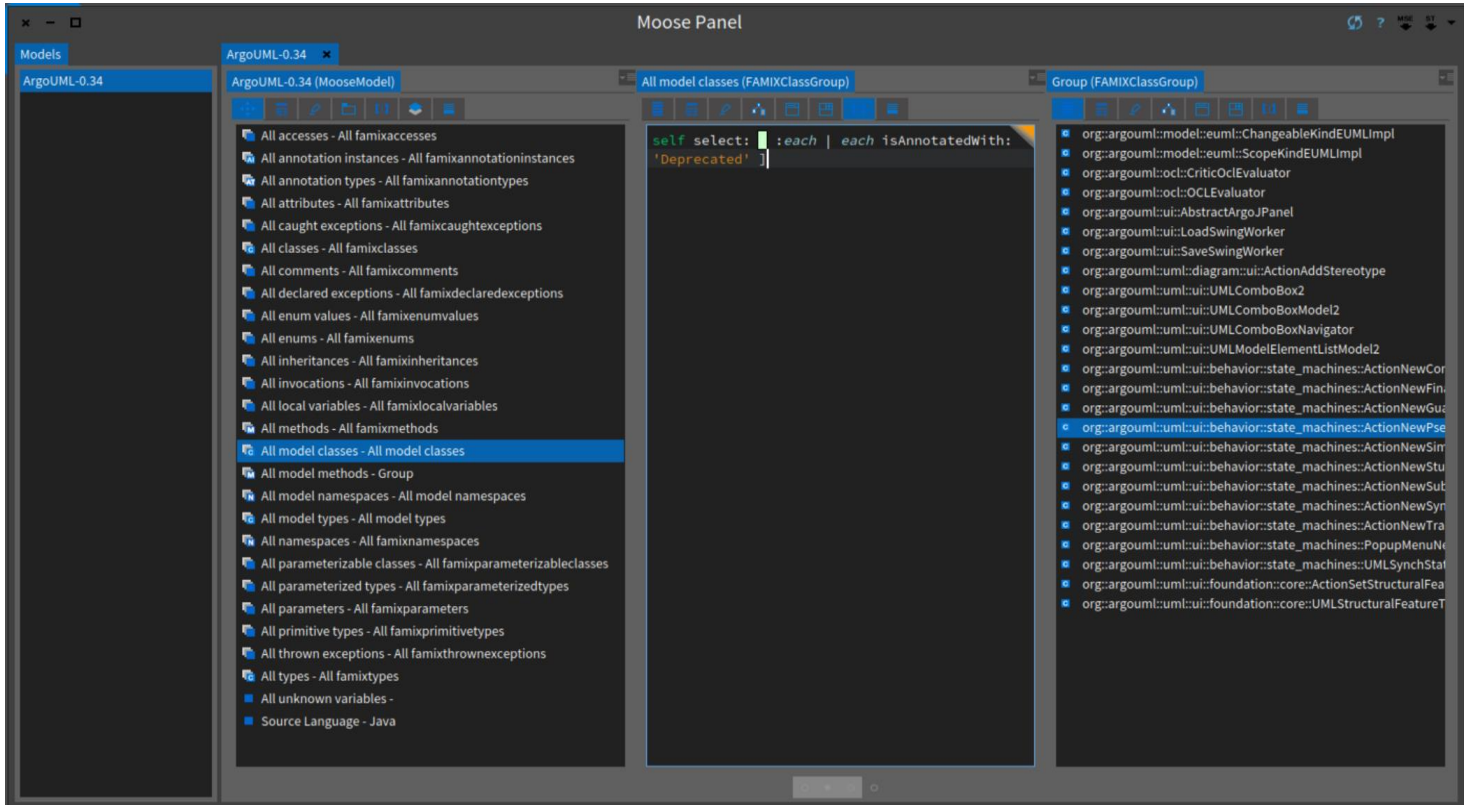


Tools Support

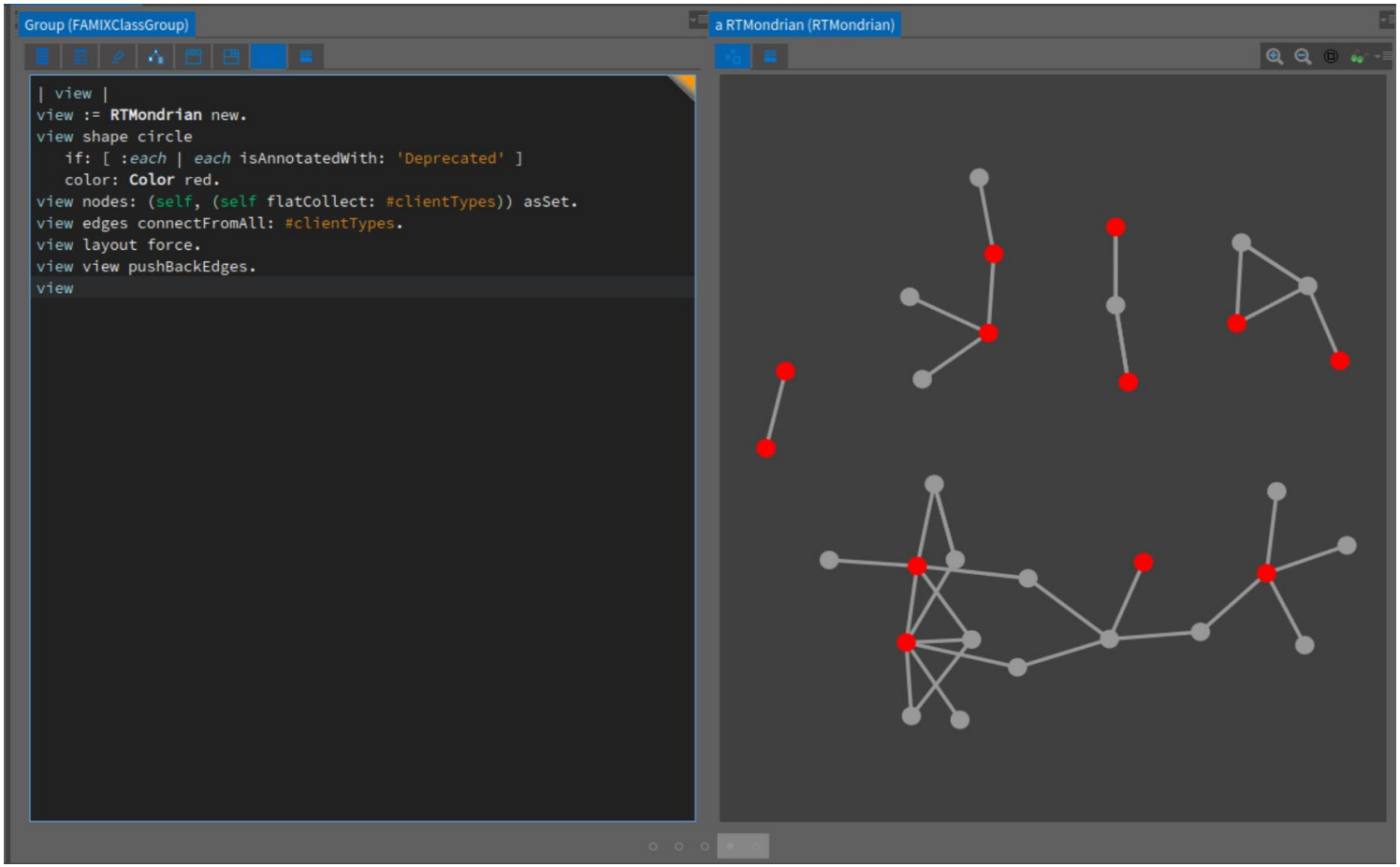
- Eclipse Search



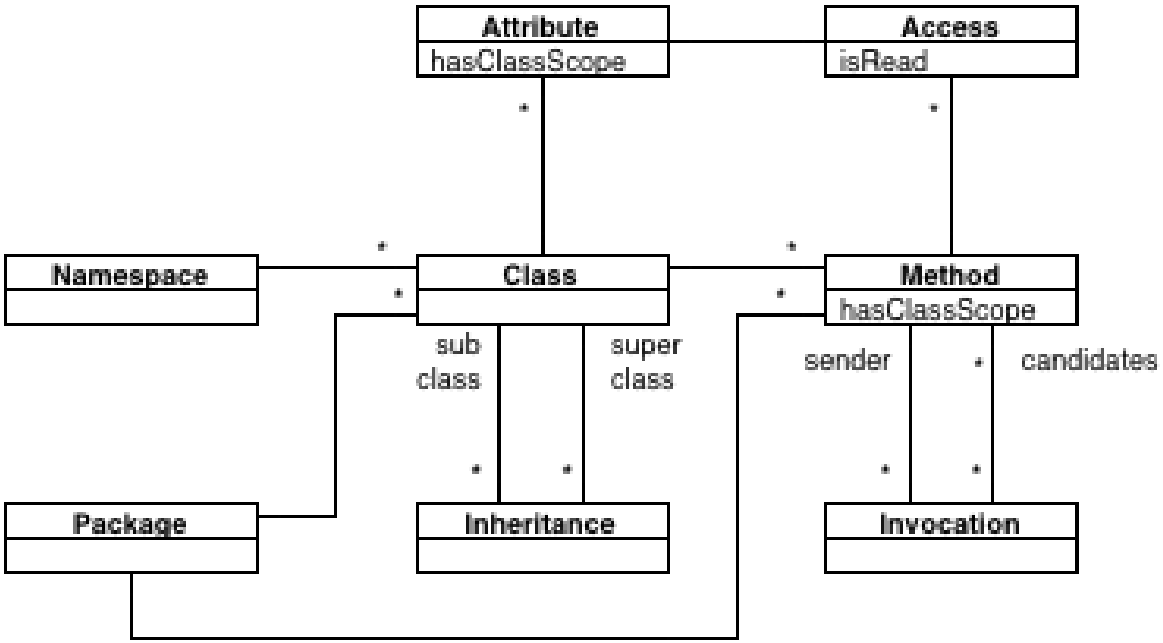
Moose


Moose Panel
Models
ArgoUML-0.34
ArgoUML-0.34
ArgoUML-0.34 (MooseModel)
All accesses - All famixaccesses
All annotation instances - All famixannotationinstances
All annotation types - All famixannotationtypes
All attributes - All famixattributes
All caught exceptions - All famixcaughtexceptions
All classes - All famixclasses
All comments - All famixcomments
All declared exceptions - All famixdeclaredexceptions
All enum values - All famixenumvalues
All enums - All famixenums
All inheritances - All famixinheritances
All invocations - All famixinvocations
All local variables - All famixlocalvariables
All methods - All famixmethods
All model classes - All model classes
All model methods - Group
All model namespaces - All model namespaces
All model types - All model types
All namespaces - All famixnamespaces
All parameterizable classes - All famixparameterizableclasses
All parameterized types - All famixparameterizedtypes
All parameters - All famixparameters
All primitive types - All famixprimitivetypes
All thrown exceptions - All famixthrownexceptions
All types - All famixtypes
All unknown variables -
Source Language - Java
All model classes (FAMIXClassGroup)
self select: [:each | each isAnnotatedWith:
'Deprecated']
Group (FAMIXClassGroup)
org::argouml::model::euml::ChangeableKindEUMLImpl
org::argouml::model::euml::ScopeKindEUMLImpl
org::argouml::ocl::CriticOclEvaluator
org::argouml::ocl::OCLEvaluator
org::argouml::ui::AbstractArgoJPanel
org::argouml::ui::LoadSwingWorker
org::argouml::ui::SaveSwingWorker
org::argouml::uml::diagram::ui::ActionAddStereotype
org::argouml::uml::ui::UMLComboBox2
org::argouml::uml::ui::UMLComboBoxModel2
org::argouml::uml::ui::UMLComboBoxNavigator
org::argouml::uml::ui::UMLModelElementListModel2
org::argouml::uml::ui::behavior::state_machines::ActionNewCor
org::argouml::uml::ui::behavior::state_machines::ActionNewFin
org::argouml::uml::ui::behavior::state_machines::ActionNewGu
org::argouml::uml::ui::behavior::state_machines::ActionNewPse
org::argouml::uml::ui::behavior::state_machines::ActionNewSim
org::argouml::uml::ui::behavior::state_machines::ActionNewStu
org::argouml::uml::ui::behavior::state_machines::ActionNewSub
org::argouml::uml::ui::behavior::state_machines::ActionNewSyn
org::argouml::uml::ui::behavior::state_machines::ActionNewTra
org::argouml::uml::ui::behavior::state_machines::PopupMenuNe
org::argouml::uml::ui::behavior::state_machines::UMLSynchSta
org::argouml::uml::ui::foundation::core::ActionSetStructuralFea
org::argouml::uml::ui::foundation::core::UMLStructuralFeatureT

Visualize code in Moose



Software Representation



Challenges

- Learn about tool
- Know the meta-model
- Learn query language
- Barrier

Solution

- Smalltalk Code

```
self allModelClasses select:
        [ :each | each isAnnotatedWith:
                        'Deprecated' ]
```

- Use Natural Language

What are the classes annotated with “deprecated”?

Generate Code

- Python experiment

A Syntactic Neural Model for General-Purpose Code Generation

Pengcheng Yin
Language Technologies Institute
Carnegie Mellon University
pcyin@cs.cmu.edu

Graham Neubig
Language Technologies Institute
Carnegie Mellon University
gneubig@cs.cmu.edu

Abstract

We consider the problem of parsing natural language descriptions into source code written in a general-purpose programming language like Python. Existing data-driven methods treat this problem as a language generation task without considering the underlying syntax of the target programming language. Informed by previous work in semantic parsing, in this paper we propose a novel neural architecture powered by a grammar model to explicitly capture the target syntax as prior knowledge. Experiments find this an effective way to scale up to generation of complex programs from natural language descriptions, achieving state-of-the-art results that well outperform previous code generation and semantic parsing approaches.

1 Introduction

Every programmer has experienced the situation where they know what they want to do, but do not have the ability to turn it into a concrete implementation. For example, a Python programmer

In parallel, the NLP community has developed methods for data-driven semantic parsing, which attempt to map NL to structured logical forms executable by computers. These logical forms can be general-purpose meaning representations (Clark and Curran, 2007; Banarescu et al., 2013), formalisms for querying knowledge bases (Tang and Mooney, 2001; Zettlemoyer and Collins, 2005; Berant et al., 2013) and instructions for robots or personal assistants (Artzi and Zettlemoyer, 2013; Quirk et al., 2015), among others. While these methods have the advantage of being learnable from data, compared to the programming languages (PLs) in use by programmers, the *domain-specific* languages targeted by these works have a schema and syntax that is relatively simple.

Recently, Ling et al. (2016) have proposed a data-driven code generation method for high-level, *general-purpose* PLs like Python and Java. This work treats code generation as a sequence-to-sequence modeling problem, and introduce methods to generate words from character-level models, and copy variable names from input descriptions. However, unlike most work in semantic parsing, it does not consider the fact that code has to be well-defined programs in the target syntax.

About Paper

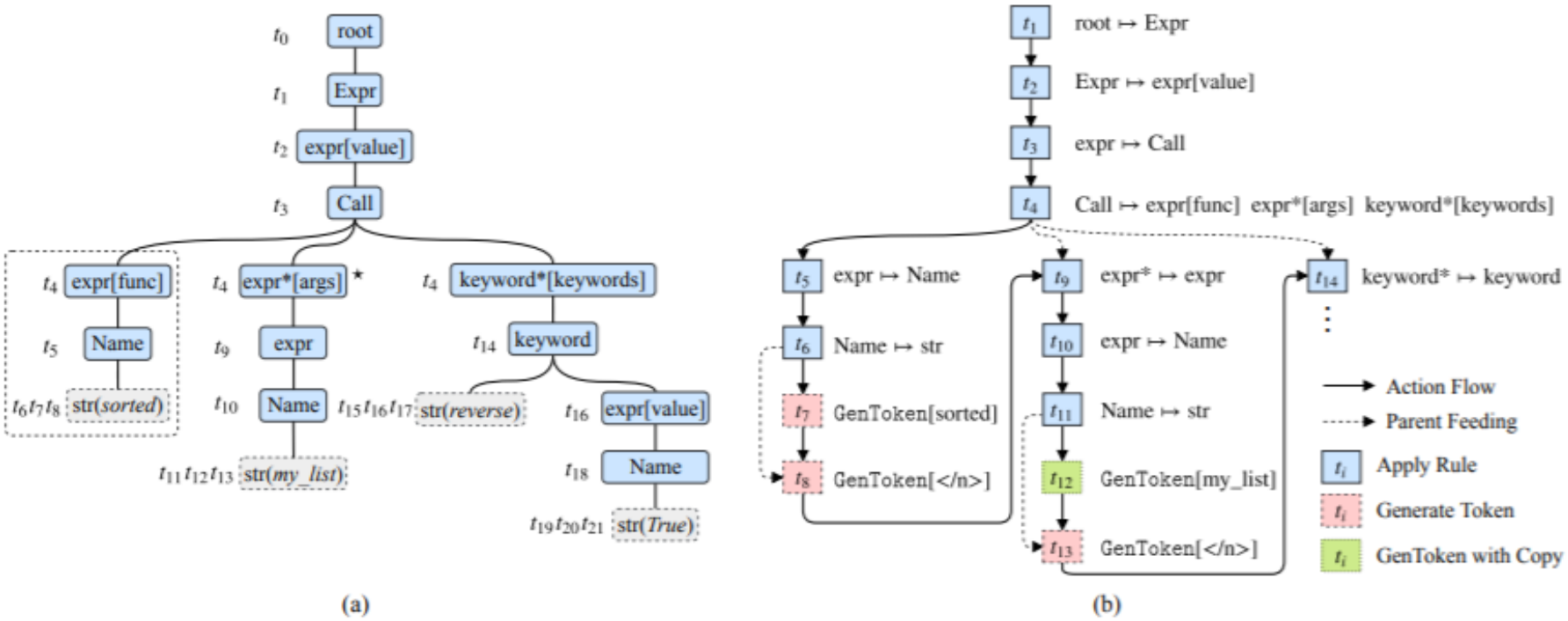
- Translation problem
- Accuracy
- Neural network model
- Model adoption

	HS		DJANGO	
	ACC	BLEU	ACC	BLEU
Retrieval System†	0.0	62.5	14.7	18.6
Phrasal Statistical MT†	0.0	34.1	31.5	47.6
Hierarchical Statistical MT†	0.0	43.2	9.5	35.9
NMT	1.5	60.4	45.1	63.4
SEQ2TREE	1.5	53.4	28.9	44.6
SEQ2TREE–UNK	13.6	62.8	39.4	58.2
LPN†	4.5	65.6	62.3	77.6
Our system	16.2	**75.8**	**71.6**	**84.5**
Ablation Study				
– frontier embed.	**16.7**	**75.8**	70.7	83.8
– parent feed.	10.6	75.7	71.5	84.3
– copy terminals	3.0	65.7	32.3	61.7
+ unary closure	–		70.3	83.3
– unary closure	10.1	74.8	–	

Tools used

- Framework: Theano
- Library: NLTK
- AST: astor
- Language: Python

Paper



Input: `sort my_list in descending order`

Code: `sorted(my_list, reverse=True)`

Challenges in Running Model

- Complex neural network
- Lots of components
- Running Code
- Dataset available for python

Findings

- Difficult to prepare dataset
- Creating neural network
- Neural Network libraries
- NLP library

Learning challenges

- Learning about moose
- Creating a simple query
- Analyzing complex neural network
- Python Dependency

Future work

- Prepare dataset
- Decoupling Code
- Adapt classes for other datasets