I. Approach

Our approach is constructed as follows:

A. Method Declaration

First is method declaration. We used AST Parser Tool to catch each method and get the information, which means, the *startLineNumber, endLineNumber, methodName, methodParameters, methodType, methodBody*. This costs extra time and can be improved in future work.

B. Tokenization

Second is the tokenization of method body and get the frequency of each token, which includes the comment and white space removal process.

C. Categorization

Third, for each method, we have a list of tokens, these tokens fall into 9 categories: *methodParameter, methodType, tokenListNumber, tokenListType, tokenListKeyword, tokenListMarker, tokenListOperator, tokenListOther1, tokenListOther2*. The purpose of doing so is that these categories have different weights, for example, two pieces of code both have 2 type names: "int", that provides no useful information, however, if two pieces of code both have 2 method parameter names: "drawVerticalLine", that means these pair are more likely the same.

A list of categorized tokens are shown in table I:

Token	Category	Freq	Token	Category	Freq
int	Type	1	)	Marker	6
for	Keyword	1	}	Marker	2
i	Other1	8	{	Marker	1
System	Other1	3	+	Operator	6
out	Other1	3	=	Operator	1
println	Other1	3	-	Operator	1
toBinary	Other1	1	¡	Operator	1
Integer	Other1	1	:	Other2	2
toBinaryString	Other1	1	5.0	Num	1
(	Marker	6	33.0	Num	1

TABLE I
A List of Token Frequency

D. Pretreatment of Variable Names

Fourth, similarity of *methodParameter* is caculated with bigram similarity. If "drawVerticalLine" and "VerticalDrawLine" appear in two code fragments respectively, we don't want to take them as different. So two similar variable names will be compared by bigram similarity, if they are similar, they will be taken as the same and their frequencies will be compared later.

E. Similarity Calculation for Each Category

Fifth, after the pretreatment of variable names, a similarity algorithm is applied to each category, and the number of similarity will be caculated, so we have 9 similarity numbers.

F. Final Similarity Calculation and Result

Finally, the 9 similarity numbers are given different weights and the final similarity will be calculated. A threshold is introduced, if the similarity is higher than the threshold, the result will be printed out: clone group number, similarity number, method name 1, start and end line number, method name 2, start and end line number.

A diagram describing the whole process is shown in Fig:1.

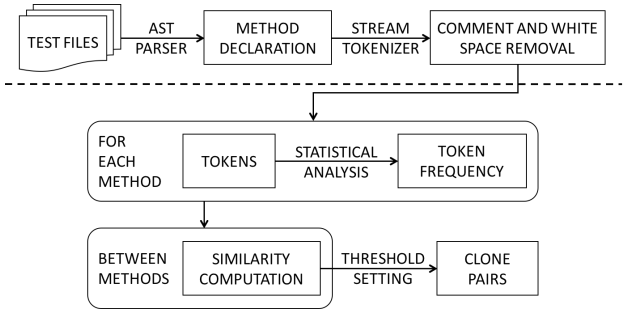

Fig. 1. Overall Project Framework

G. Machine Learning

During the final step, different weights are given to different categories, however, picking 9 weights arbitrarily may not be resonable. So we used Machine Learning method to set the weights. Provided training files with "ground truth", machine learning process will decide the most proper weights, which is much more resonable than setting by hand. After actual testing, it turns out machine learning does provide a better result.

DISCUSS MORE ABOUT MLP HERE

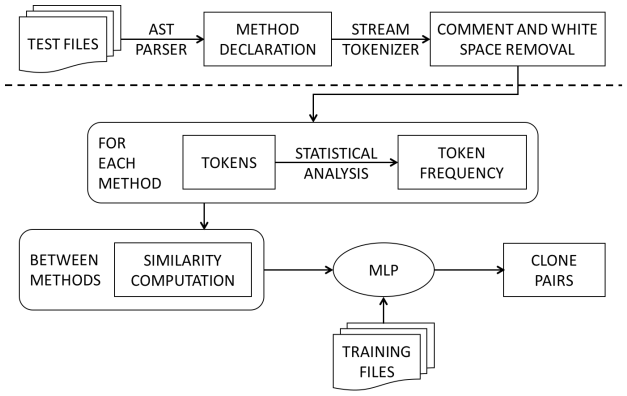

Fig. 2. Project Framework with Machine Learning