Recognizing structural patterns in code
General Idea

Chop Text into interesting pieces (statements)

Create abstract representation of statements

Find dependencies between the representation (k-means)

Ideal Solution: One Parser rule per found dependency
K-Means

Unsupervised learning Algorithm

Clusters Data in k clusters with k centroids

Points of the Cluster have minimal distance to the centroid
Abstract Representation

Example

if ( x < 0 ) return 0 ;
0 2 0 2 1 2 0 1 2 0 1 2 -> (0 2 0 2 1 2 0 1 2)

if ( x = 0 ) return 1 ;
0 2 0 2 1 2 0 1 2 -> (0 2 0 2 1 2 0 1 2)
Problems

Example

if ( x < 0 ) return 0 ;
0 2 0 2 1 2 0 1 2 0 2 1 2 0 1 2 0

if ( x => 6 ) return 1 ;
0 2 0 2 2 1 2 0 1 2 0 2 2 1 2 0 1 2 0

0 for padding

Euclidean distance of v8 ≈ 2,8

comparing different types of data
Results

Change to Notepad++
Future Work

Reversing centroids to text

Find “smarter” way to chop text into pieces

Can this approach distinguish between different languages