

**Table S1.** An example containing scenarios such as functionality type match and method name not mentioned in the context

<b>Context</b>	<pre> substring( i + &lt;num.val&gt; ).toLowerCase (); } return ext ; } &lt;soc&gt; public static void </pre>		
<b>Ground truth (GT)</b>	<pre> &lt;soc&gt; public static void copyFile ( File in , File out ) throws IOException { FileChannel inChannel = new FileInputStream(in).getChannel( ); FileChannel outChannel = new FileOutputStream(out).getChannel( ); try { inChannel.transferTo(&lt;num.val&gt;, inChannel.size(), outChannel); } catch ( IOException e ) { throw e ; } finally { if ( inChannel != null ) inChannel.close ( ); if ( outChannel != null ) outChannel.close ( ); } } } &lt;eoc&gt; </pre>		
	<table border="1"> <tr> <td><b>Perplexity</b></td> <td>1.21</td> </tr> </table>	<b>Perplexity</b>	1.21
<b>Perplexity</b>	1.21		
<b>DeepClone output (DCO)</b>	<pre> &lt;soc&gt; public static void copyStreams ( File source , File dest , File destDir ) throws IOException { BufferedInputStream in = null ; BufferedOutputStream out = null ; try { in = new BufferedInputStream ( new FileInputStream ( source ) ) ; out = new BufferedOutputStream ( new FileOutputStream ( dest ) ) ; out.write(source.getAbsolutePath( )); int len = in.read ( ) ; while (len &gt; &lt;num.val&gt;){ out.write(len); } in.close( ); out.close( ); } catch (Exception ex) { ex.printStackTrace ( ) ; } finally { if ( in!= null ) in.close ( ) ; if ( out!= null ) out.close ( ) ; } } } &lt;eoc&gt; </pre>		
	<table border="1"> <tr> <td><b>Perplexity</b></td> <td>9.246</td> </tr> </table>	<b>Perplexity</b>	9.246
<b>Perplexity</b>	9.246		
<b>Top 1</b>	<pre> &lt;soc&gt; public static void copy ( File source , File dest ) throws Exception { FileInputStream in = new FileInputStream ( source ) ; FileOutputStream out = new FileOutputStream ( dest ) ; int c ; while ((c = in.read()) != -&lt;num.val&gt;){ out.write(c); in.close (); out.close (); } } } &lt;eoc&gt; </pre>		
	<table border="1"> <tr> <td><b>Perplexity</b></td> <td>1.439</td> </tr> </table>	<b>Perplexity</b>	1.439
<b>Perplexity</b>	1.439		
<b>Top 2</b>	<pre> &lt;soc&gt; public static void copyFile ( String source , String dest ) throws IOException { FileChannel in = null , out = null ; try { in = new FileInputStream ( new File ( source ) ) . getChannel ( ) ; out = new FileOutputStream ( new File ( dest ) ) . getChannel ( ) ; in . transferTo ( &lt;num.val&gt; , in . size ( ) , out ) ; } finally { if ( in!=null ) in.close (); if ( out!=null ) out.close (); } } } &lt;eoc&gt; </pre>		
	<table border="1"> <tr> <td><b>Perplexity</b></td> <td>1.321</td> </tr> </table>	<b>Perplexity</b>	1.321
<b>Perplexity</b>	1.321		

**Table S2.** An example of the preprocessing steps applied on a clone method

Original	<pre>// Heuristic: (([0-9,a-z,A-Z,..,\$]+\s*)=\s*(\s*[0-9,a-z,A-Z,..,\$]+\s*)+\s*[0-9,a-z,A-Z,..,\$] // imid = imin + (imax - imin)/2; // imid = (imax + imin)/2; public static int binarySearch1(int arr[], int key, int imin, int imax) { //Implementation: Recursive, primitive type if (imax &lt; imin) return -1; int imid = (imin + imax) / 2; if (arr[imid] &gt; key) return binarySearch1(arr, key, imin, imid - 1); else if (arr[imid] &lt; key) return binarySearch1(arr, key, imid + 1, imax); else return imid; }</pre>
Marking	<pre>// Heuristic: (([0-9,a-z,A-Z,..,\$]+\s*)=\s*(\s*[0-9,a-z,A-Z,..,\$]+\s*)+\s*[0-9,a-z,A-Z,..,\$] // imid = imin + (imax - imin)/2; // imid = (imax + imin)/2; &lt;soc&gt; public static int binarySearch1(int arr[], int key, int imin, int imax) { //Implementation: Recursive, primitive type if (imax &lt; imin) return -1; int imid = (imin + imax) / 2; if (arr[imid] &gt; key) return binarySearch1(arr, key, imin, imid - 1); else if (arr[imid] &lt; key) return binarySearch1(arr, key, imid + 1, imax); else return imid; } &lt;eoc&gt;</pre>
Normalization	<pre>&lt;soc&gt; public static int binarySearch1 ( int arr[], int key, int imin, int imax ) { if ( imax &lt; imin ) return -1 ; int imid = ( imin + imax ) / 2 ; if ( arr [ imid ] &gt; key ) return binarySearch1 ( arr , key , imin , imid - 1 ) ; else if ( arr [ imid ] &lt; key ) return binarySearch1 ( arr , key , imid + 1 , imax ) ; else return imid ; } &lt;eoc&gt;</pre>
Replacement	<pre>&lt;soc&gt; public static int binarySearch1 ( int arr[], int key, int imin, int imax ) { if ( imax &lt; imin ) return &lt;num.val&gt; ; int imid = ( imin + imax ) / &lt;num.val&gt; ; if ( arr [ imid ] &gt; key ) return binarySearch1 ( arr , key , imin , imid - &lt;num.val&gt; ) ; else if ( arr [ imid ] &lt; key ) return binarySearch1 ( arr , key , imid + &lt;num.val&gt; , imax ) ; else return imid ; } &lt;eoc&gt;</pre>

**Table S3.** An example containing scenarios such as exact match, functionality type match and method name mentioned in the context

Context	<pre>v[i]&lt;num.val&gt;; } &lt;soc&gt; public static Rectangle2D [][] transpose (Rectangle2D</pre>		
Ground truth (GT)	<pre>&lt;soc&gt; public static Rectangle2D [][] transpose (Rectangle2D [][] a) { int m=a.length; int n=a[&lt;num.val&gt;].length; Rectangle2D [][] t = new Rectangle2D[n][m]; for(int i=&lt;num.val&gt;; i &lt; m; i++) for (int j = &lt;num.val&gt;; j &lt; n; j++) t[j][i]=a[i][j]; return t; } &lt;eoc&gt;</pre>		
	<table border="1"> <tr> <td>Perplexity</td> <td>1.541</td> </tr> </table>	Perplexity	1.541
Perplexity	1.541		
DeepClone output (DCO)	<pre>&lt;soc&gt; public static Rectangle2D [][] transpose (Rectangle2D [][] a, Rectangle2D [][] b) { int [][] M = new int[a.length][a[&lt;num.val&gt;].length]; for(int i = &lt;num.val&gt;; i &lt; m.length; i++) { for(int j = &lt;num.val&gt;; j &lt; a[i].length; j++){ M[j][i]=b[i][j]; } } return M; } &lt;eoc&gt;</pre>		
	<table border="1"> <tr> <td>Perplexity</td> <td>3.54</td> </tr> </table>	Perplexity	3.54
Perplexity	3.54		
Top 1	<pre>&lt;soc&gt; public static Rectangle2D [][] transpose (Rectangle2D [][] a){ int m = a.length; int n = a[&lt;num.val&gt;].length; Rectangle2D [][] t=new Rectangle2D[n][m]; for (int i = &lt;num.val&gt;;i&lt;m;i++) for (int j=&lt;num.val&gt;;j&lt;n;j++) t[j][i]=a[i][j]; return t; } &lt;eoc&gt;</pre>		
	<table border="1"> <tr> <td>Perplexity</td> <td>1.541</td> </tr> </table>	Perplexity	1.541
Perplexity	1.541		
Top 2	<pre>&lt;soc&gt; public static byte [][] transpose(byte [][] m) { byte [][] n = new byte[m[&lt;num.val&gt;].length][m.length]; for (int j = &lt;num.val&gt;;j&lt;m.length;j++) for (int i = &lt;num.val&gt;;i&lt;n[&lt;num.val&gt;].length;i++) n[i][j]=m[j][i]; return n; } &lt;eoc&gt;</pre>		
	<table border="1"> <tr> <td>Perplexity</td> <td>1.63</td> </tr> </table>	Perplexity	1.63
Perplexity	1.63		

**Table S4.** An example containing scenarios such as functionality type not matched and method name not mentioned in the context

<b>Context</b>	<pre>MAX_IO_CHUNK_SIZE = &lt;num.val&gt; * &lt;num.val&gt; * &lt;num.val&gt; ; private static final long LASTMODIFIED_DIFF_MILLIS = &lt;num.val&gt; ; &lt;soc&gt; public static boolean</pre>	
<b>Ground truth (GT)</b>	<pre>&lt;soc&gt; public static boolean copy(File from, File to, Override override) throws IOException {     FileInputStream in = null;     FileOutputStream out = null;     FileChannel srcChannel = null;     FileChannel destChannel = null;     if (override == null) override = Override.NEWER;     switch (override) {         case NEVER:             if (to.isFile())                 return &lt;num.val&gt; ;             break;         case NEWER:             if (to.isFile() &amp;&amp; (from.lastModified() - LASTMODIFIED_DIFF_MILLIS) &lt; to.lastModified())                 return &lt;num.val&gt; ;             break;     }     to.getParentFile().mkdirs();     try { in = new FileInputStream(from);         out = new FileOutputStream(to);         srcChannel = in.getChannel();         destChannel = out.getChannel();         long position = &lt;num.val&gt; ;         long count = srcChannel.size();         while (position &lt; count) {             long chunk = Math.min(MAX_IO_CHUNK_SIZE, count - position);             position += destChannel.transferFrom(srcChannel, position, chunk);         }         to.setLastModified(from.lastModified());         return &lt;num.val&gt; ;     } finally {         CommonUtils.close(srcChannel);         CommonUtils.close(destChannel);         CommonUtils.close(out);         CommonUtils.close(in);     } } &lt;eoc&gt;</pre>	
	<b>Perplexity</b>	3.109
<b>DeepClone output (DCO)</b>	<pre>&lt;soc&gt; public static boolean deleteDirectory(File f) {     try {         final File[] fFiles = f.listFiles();         for (int i = &lt;num.val&gt; ; i &lt; fFiles.length; i++) {             if (fFiles[i].isDirectory())                 deleteDirectory(fFiles[i]);         }     } catch (Exception e) {         System.out.println("&lt;str.val&gt; + f + &lt;str.val&gt; + i + &lt;str.val&gt; );         return &lt;num.val&gt; ;     }     return &lt;num.val&gt; ; } &lt;eoc&gt;</pre>	
	<b>Perplexity</b>	6.137
<b>Top 1</b>	<pre>&lt;soc&gt; private void deleteDirectory ( File dir ) {     File [ ] a = dir . listFiles ( ) ;     if ( a != null ) {         for ( File f : a ) {             if ( f.isDirectory () ) {                 deleteDirectory ( f ) ;             } else {                 f.delete ( ) ;             }         }     }     dir.delete ( ) ; } &lt;eoc&gt;</pre>	
	<b>Perplexity</b>	1.887
<b>Top 2</b>	<pre>&lt;soc&gt; public static void deleteDirectory ( File dir ) {     File [ ] files = dir . listFiles ( ) ;     for ( File f : files ) {         if ( f . isDirectory ( ) ) {             deleteDirectory ( f ) ;         } else             f.delete ( ) ;     }     dir.delete (); } &lt;eoc&gt;</pre>	
	<b>Perplexity</b>	1.831