Supplementary material for Bimodal Modelling of Source Code and Natural Language

May 29, 2015

1 Datasets Samples

1.1 Synthetic Data - Text

Sample 1

```csharp
string result = String.Join("\n", input_string.Split("\n\n").Select((string x) => x.ToLower()).OrderBy(x => x));
```

- all elements to small letters for each line and sort
- all elements to small letters for each line and order
- all elements to lowercase for each line and sort
- all elements to lowercase for each line and order
- all elements lowercase for each line and sort
- all elements lowercase for each line and order
- all elements lowercase for each line and order
- all elements to lower case for each line and sort
- all elements to lower case for each line and order
- all elements to small letters for each new line and sort

Sample 2

```csharp
var result = input_string.Split(',').Select(
    (string x) => x.Substring(x.Length - 1)).First();
```

- get last character of each element separated by a comma and get first element
- get last character of each element separated by a comma and get 1st element
- each element get last character separated by a comma and get first element
- each element get last character separated by a comma and get 1st element
- each element get last letter separated by a comma and get first element
• each element get last letter separated by a comma and get 1st element
• each element last char separated by a comma and get first element
• each element last char separated by a comma and get 1st element
• each element last letter separated by a comma and get first element
• each element last letter separated by a comma and get 1st element

Sample 3

```csharp
var result = input_string.Split('
').Select({}
    (string x) => Double.Parse(x)).Average();
```

• each element parse double for each line and get mean
• each element convert to double for each new line and get average
• all elements parse to double for each line and get mean
• all elements parse number for each line and find mean
• all elements parse double for each new line and average
• all elements as number for each new line and get mean
• all elements as number for each new line and get average

1.2 StackOverflow

These are three random example from the StackOverflow dataset.

Sample 1

```csharp
Assert.AreEqual(dict, dictClone)
```

• dictionary with equals keys
• dictionaries have same keys and values but aren’t equal

Sample 2

```csharp
int minutes = (int)elapsed.TotalMinutes;
double fsec = 60 * (elapsed.TotalMinutes - minutes);
int sec = (int)fsec;
int ms = 1000 * (fsec - sec);
string tsOut = String.Format("{0}:{1:D2}.{2}", minutes, sec, ms);
```

• stopwatch
• minutes
• how to generate seconds elapsed
• stopwatch get seconds
• string formats time stopwatch
• convert of a second to millisecond
• show stopwatch in seconds
• stopwatch get minute second hours

Sample 3

serializer.Converters.Add(new DTOJsonConverter());
Interfaces.IEntity entity = serializer.Deserialize(jsonReader);

• deserializing using interface
• json serialization down casting

1.3 DotNetPerls

These are three random example from the DotNetPerls dataset.

Sample 1

using System;

class Program
{
    static void Main()
    {
        // Sort range of elements.
        int[] values = { 4, 7, 2, 0 };
        Array.Sort(values, 0, 3);
        foreach (int value in values)
        {
            Console.Write(value);
            Console.Write(' ');
        }
        Console.WriteLine();
    }
}

• sorted int array search
• sort adjust other item sort
• sorting array without using any array
• sort numbers starting with 0
• sorted array of int
• integer array sort ascending
• write a function to sort 2 arrays of numbers
• how to sort the two different array
• order by array values
• sort indexes of an array
• provide sorting function in place
• how to keep the index of sorted int array with string array
• reorder array
• simple string sort

Sample 2

```csharp
using System;
using System.Web.UI;

namespace WebApplication1
{
    public partial class List : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Get response.
            var response = base.Context.Response;
            // Redirect permanently.
            response.RedirectPermanent("http://www.dotnetperls.com/list", false);
        }
    }
}
```

• response . redirect in asp
• asp redirect methods
• asp redirect to website
• response redirect in page_load
• 302 redirect
• asp redirect old links
• asp redirect page
• httpresponse redirect asp
• response asp response transfer
• asp response redirect
• asp response.redirect 301
• redirect to another page by class asp
• asp redirect
• page moved redirect

Sample 3
// Custom method
string s1 = CharTool.CharCombine(a, b, c, d); // [above]
string s2 = CharTool.CharCombine(d, c, b, a);

// Concatenation
string s1 = a.ToString() + b.ToString() + c.ToString() + d.ToString();
string s2 = d.ToString() + c.ToString() + b.ToString() + a.ToString();

- concat char
- char combine
- array concat benchmark
- concatenate two char arrays
- concatenate two char
- combine character
- combining arrays
- can you concat char and string
- combine two arrays in constructor
- char array concat
- combine two arrays in constructor
- concat benchmark
- concatenate two char

2 Code Generation Samples and ground truth

Randomly generated samples, given a test query.

2.1 StackOverflow

2.1.1 Example 1

Input Query: timespan day the week

Generated:
DateTime DateTime = DateTime.Now(0);

Ground-truth:
foreach(string s in Days(new DateTime(2010, 1, 1), new DateTime(2010, 2, 1))) {
    Console.WriteLine(s);
}
2.1.2 Example 2
Input Query: regex remove leading lowercase
Generated:
```csharp
string Regex = str.Replace( , 'char').s( , , str) ;
```
Ground-truth:
```csharp
var input = "Blue Cross Blue Shield 12356";
var sb = new StringBuilder();
foreach (var ch in input) {
    if (char.IsUpper(ch)) { // only keep uppercase
        sb.Append(ch);
    }
}
var output = sb.ToString(); // "BCBS"
```

2.1.3 Example 3
Input Query: file exists on directory
Generated:
```csharp
var path = new File ( Directory )
```
Ground-truth:
```csharp
System.IO.File.Exists(path)
```

2.2 DotNetPerls
Input Query: char ispunctuation
Generated
```csharp
using System;
class Program{
    static void Main () {
        for (int i = 0; i < 128; i++) {
            if (char.IsPunctuation((char)i)){
                Console.WriteLine("case '{0}':", (char)i);
            }
        }
    }
}
```
Ground truth
```csharp
using System;
class Program {
    static char Main() {
        if (value.s2()) {
            _max = StartNew.StartNew('_') && (s1._max*1000000).s1("0.00 ns", '%') ;
            return new Punctuation() ;
        }
    }
}
```