Bimodal Modelling of Source Code & Natural Language

Miltos Allamanis*, Daniel Tarlow, Andrew Gordon, Yi Wei

Microsoft Research

* University of Edinburgh, UK. Work done primarily while author was an intern.
Applications of Joint Models of Code & NL

Code Retrieval

NL Retrieval for Source Code

and eventually code synthesis...
Source Code & Natural Language: the Machine Learning Perspective

Source Code is a highly structured object with well-defined semantics;

Interesting relation with bimodal models of NL and images;
A Conditional Generative Model

"get the first letter of each word in string and uppercase"

string s;
string[] words = s.ToUpper().split(' ');  
string[] firstLetters = new string[words.Length];
for (int i=0; i < words.Length; i++) {
    firstLetters[i] = words.Substring(0,1);
}
Models of Source Code – Design Choices

Token-level Models

```
for (int i = 0; i < 10; i++){
    Console.WriteLine(i);
}
```

Syntactic Models
A Neural Log-Bilinear Bimodal Model of Code

Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal neural language models."
Maddison, Chris and Daniel Tarlow. "Structured generative models of natural source code."
A Neural Log-Bilinear Bimodal Model of Code

\[ s_\theta(v, \mathcal{L}, C_{\leq n}) = (l \otimes c)^T r + b_{n \rightarrow v} \]

Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal neural language models."
Maddison, Chris and Daniel Tarlow. "Structured generative models of natural source code."
A Neural Log-Bilinear Bimodal Model of Code

\[ c = \sum_{j=1}^{J} H_j c_{\phi_j} \]

\[ l = \frac{1}{|\mathcal{L}|} \sum_{w \in \mathcal{L}} l_w \]

\[ s_\theta(v, \mathcal{L}, C_{\leq n}) = (l \odot c)^T r + b_{n \rightarrow v} \]

Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal neural language models."
Maddison, Chris and Daniel Tarlow. "Structured generative models of natural source code."
A Neural Log-Bilinear Bimodal Model of Code

\[ P(v | L, C_{\leq n}) \propto \exp s_\theta(v, L, C_{\leq n}) \]

\[ s_\theta(v, L, C_{\leq n}) = (l \odot c)^T r + b_{n \rightarrow v} \]

Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal neural language models."
Maddison, Chris and Daniel Tarlow. "Structured generative models of natural source code."
Training the Model

Noise Contrastive Estimation (NCE)

Stochastic Gradient Descend and AdaGrad

Bayesian Optimization for Hyperparameters

Dropout
Sanity Check:
String Manipulation Synthetic Data

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

each element parse double separated by a space and get mean
each element parse double separated by a space and get average
each element convert to double separated by a space and get mean
each element convert to double separated by a space and get average
each element parse to double separated by a space and get mean
Additive vs Multiplicative Model

Train Set

“each element parse double separated by a space and get max”

```
var res = input_string.Split(' ').Select((string x) => Double.Parse(x)).Max();
```

Test Set

“each element parse double separated by a tab and get max”

```
var res = input_string.Split('\t').Select((string x) => Double.Parse(x)).Min();
```

Generated:

```
var res = input_string.Split('\t').Select((string x) => Double.Parse(x)).Max();
```
Additive vs Multiplicative Model

Train Set

“each element parse double separated by a space and get max”

```csharp
var res = input_string.Split(' ').Select(
    (string x) => Double.Parse(x)).Max();
```

Test Set

“each element parse double separated by a tab and get min”

```csharp
var res = input_string.Split('	').Select(
    (string x) => Double.Parse(x)).Min();
```

Generated:

```csharp
var res = input_string.Split('	').Select(
    (string x) => Double.Parse(x)).Max();
```
Additive vs Multiplicative Model

```
Train Set
“each element parse double separated by a space and get max”
var res=input_string.Split(' ').Select((string x) => Double.Parse(x)).Max();
```

```
Test Set
“each element parse double separated by a tab and get max”
var res=input_string.Split('	').Select((string x) => Double.Parse(x)).Max();
```
StackOverflow Data & Augmenting Data with Bing Queries
Natural Language “Query”

Make first letter of a string upper case

Code Snippets

```csharp
public static string FirstCharToUpper(string input)
{
    if (String.IsNullOrEmpty(input))
        throw new ArgumentException("ARGH!");
    return input.First().ToString().ToUpper() + String.Join(",", input.Skip(1));
}
```

EDIT: This version is shorter. For a faster solution take a look at Equico’s answer

```csharp
public static string FirstCharToUpper(string input)
{
    if (String.IsNullOrEmpty(input))
        throw new ArgumentException("ARGH!");
    return input.First().ToString().ToUpper() + input.Substring(1);
}
```
foreach (Suit suit in (Suit[])) Enum.GetValues(typeof(Suit))
{
}

C# Configuration Manager - ConnectionStrings

Check your machine.config. If you only want your entry, you can add a <clear /> element to the <connectionStrings> element like so...

```
<connectionStrings>
  <clear />
  <add name="Target" connectionString="server=MYSERVER; Database=MYDB; Integrated Security=SSPI;" />
</connectionStrings>
```
How do I enumerate an enum

```csharp
foreach (Suit suit in (Suit[]) Enum.GetValues(typeof(Suit)))
{
}
```

http://stackoverflow.com/questions/105372/how-do-i-enumerate-an-enum
## Retrieval Evaluation - MRR Performance

<table>
<thead>
<tr>
<th>Code Retrieval</th>
<th>Model</th>
<th>StackOverflow Test 1</th>
<th>StackOverflow Test 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicative</td>
<td>0.18</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>Additive</td>
<td>0.10</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>NL only</td>
<td>0.12</td>
<td>0.13</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Retrieval</th>
<th>Model</th>
<th>StackOverflow Test 1</th>
<th>StackOverflow Test 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicative</td>
<td>0.43</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>Additive</td>
<td>0.22</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>NL only</td>
<td>0.25</td>
<td>0.26</td>
<td></td>
</tr>
</tbody>
</table>

*Test 1*: Code snippets from training set with new natural language queries.  
*Test 2*: New code snippets and new natural language queries.
Synthesis Samples

> timespan day the week
DateTime DateTime=DateTime.Now(0);

> file exists on directory
var path = new File(directory)
> timespan day the week
DateTime DateTime=DateTime.Now(0);

foreach(string s in Days(new DateTime(2010, 1, 1), new DateTime(2010, 2, 1))) {
    Console.WriteLine(s);
}

> file exists on directory
var path = new File(directory)

System.IO.File.Exists(path)
path = Path.GetFullPathInternal(path);
new FileIOPermission(
    FileIOPermissionAccess.Read,
    new string[] { path },
    false, false).Demand();
flag = InternalExists(path);

1. wpf get directory name from path
2. determine a file exist on shared folder
3. open file dialog class
4. create directory pathname
5. load binary file to variable
Conclusions

Promising results on real world data & profitable to model source code and NL together.

Multiplicative combination of modalities is necessary.

Need for better, curated datasets to advance the quality of the bimodal models of source code & NL.