Automatic String Replace by Examples

A. De Lorenzo, E. Medvet, A. Bartoli
University of Trieste, Italy

Genetic and Evolutionary Computation Conference
Text Search-and-Replace

Input Text
- Log
- Email corpus
- Web templates
- Invoices
- ...

Processing Engine
- Java/JavaScript/PHP
- Text Editor (sed,...)
- ...

Modified Text

Search-and-Replace task description

SEARCH pattern (regular expression)
+ REPLACEMENT expression
Simple Example: Date Format Change

Today is 1-23-13
He left on 3-13-12
Great 1-1-13 party!
How are you doing?

SEARCH pattern
(\d+)-(\d+)-(\d+)

REPLACEMENT expression
$2/$1/$3

today is 23/1/13
he left on 13/3/12
great 1/1/13 party!
How are you doing?
In practice

- Search-and-replace tasks must be described by hand
- Requires technically-savvy users
- Often difficult to debug
 Wouldn't it be nice if...

- Search-and-Replace tasks could be described merely by a few examples?
Our work

- Completely AUTOMATIC
- No similar proposals
- Of course, with limitations / constraints...
How it works (in a nutshell)

1. Identify a **context** where changes have to be confined (genetic programming)

   ... Bartoli is ok @Bartoli ... out of context

   ... in context

   ... bartoli is ok @Bartoli ... Bartoli is ok @Baxxxx ...

2. Build the **replacement expression** $r$ (deterministically)

3. Generate the **regular expression** $s$ (genetic programming)
Phase 1: Remark

- We do **not** require that the user specifies a context explicitly

... Bartoli is ok @Bartoli ...

... Bartoli is ok @Baxxxx ...

- We need a way to identify a context **automatically**
1. **Build an ancillary example set** composed of:

<table>
<thead>
<tr>
<th>Input strings</th>
<th>To-be-changed strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>I like @GECCO13 conf</td>
<td>CC013</td>
</tr>
<tr>
<td>RT @MaleLabTs New paper</td>
<td>leLabTs</td>
</tr>
<tr>
<td>Bartoli is ok @Bartoli</td>
<td>rtoli</td>
</tr>
<tr>
<td>nothing new here</td>
<td>(empty)</td>
</tr>
</tbody>
</table>

Constructed automatically from Input Strings + Desired Output Strings

2. **Generate a single regex extracting a superstring**

--->

the context

@GECCO13
@MaleLabTs
@Bartoli
(empty)
Regex Generation from Examples

- Build upon our GECCO 2012 work
  - Text extraction
  - Generate regex automatically from examples (with GP)
  - Available online
    (http://regex.inginf.units.it/)
  - Regex extracts exactly what is specified (i.e., not a superstring)

- Key difference: **fitness**
- Promotes regexes with minimal distance between:
  - string extracted by the **first capturing group** of the candidate
  - to-be-changed string
1. For each example, build a replacement expression deterministically

- Identify:
  - context extracted from input string
  - "corresponding" substring in the output string

- "Play" with them (see the paper for details)

2. Select the replacement expression that occurs most often
Phase 3

1. **Take the examples of the overall task**
   
   ```
   today is 1-23-13
   he left on 3-13-12
   great 1-1-13 party!
   
   today is 23/1/13
   he left on 13/3/12
   great 1/1/13 party!
   ```

2. **Generate** a regex based on these examples and `r`

3. **Generate the regex** `s`
   - Based on examples and `r`

   - Multiobjective fitness (NSGA II)
     - Minimal difference in number of capturing groups
       - in the candidate
       - occurring in `r`
     - Minimal distance between
       - string generated by `< candidate + r >`
       - desired string
Experiments

- Tasks (500 with changes, 500 unchanged)
  - Full tweet anonymization
  - Partial IP anonymization
  - Date format change
  - Phone number change

- Experiments
  - Learning set: 20, 50, 100 (balanced)
  - 5-fold cross-validation

- Performance index: Count error rate
Salient Results

- With only 25 positive examples:
  - perfect result for two tasks (on this dataset)
  - "very good" for the two other tasks

(Ok...of course...it depends on what we mean by "good"...)

![Diagram with table and notes]

**Regex learning for mere extraction [4,11] a "difficult" dataset**
Further experimental remarks

- "Many" solutions with "good" performance (not just lucky individuals)

- Relative performance in validation good predictor for relative performance in testing

- Execution time too high to devise interactive use (at least in the near future)

  But at 1-2$/hour is cheaper than a specialist?

| Task                        | $|T^t| + |T^u|$ | Time (min) |
|-----------------------------|----------------------|------------|
| Twitter anonymization       | 20                   | 1          |
|                             | 25                   | 1          |
|                             | 50                   | 2          |
| IP partial anonymization    | 20                   | 12         |
|                             | 25                   | 33         |
|                             | 50                   | 43         |
| Date format change          | 20                   | 18         |
|                             | 25                   | 36         |
|                             | 50                   | 78         |
| Phone number format change  | 20                   | 35         |
|                             | 25                   | 80         |
|                             | 50                   | 132        |
Remarkable and quite promising exercise (we believe)

Keep in mind:
- Results not human competitive (yet ?)
- Execution time has to be improved

Some key-but-unanswered questions:
- How many examples are both "practical" and "adequate" ?
- How to realize whether the examples are "enough" and "adequate" ?
- How to characterize tasks that are just hopeless ?
Thanks for your attention

http://machinelearning.inginf.units.it