Better Glue for Pipelines

CSE504 Project Proposal
Luheng He
Motivation: Pipelined Software for NLP/ML Tasks

<table>
<thead>
<tr>
<th>Typical subtasks for NLP</th>
<th>Typical subtasks for ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Input Reader</td>
<td>Input Reader</td>
</tr>
<tr>
<td>2 Segmentation/tokenization</td>
<td>Pre-processing/Data filtering</td>
</tr>
<tr>
<td>3 Pos-tagging/Parsing/Named-entity Recognition</td>
<td></td>
</tr>
<tr>
<td>4 Feature Extraction for the target task</td>
<td></td>
</tr>
<tr>
<td>5 Parameter Fitting (Learning)</td>
<td></td>
</tr>
<tr>
<td>6 Evaluation/Cross validation</td>
<td></td>
</tr>
<tr>
<td>7 Model Ensemble</td>
<td></td>
</tr>
<tr>
<td>8 Output/Analysis/Visualization</td>
<td></td>
</tr>
</tbody>
</table>

(Mostly) task-independent, off-the-shelf tools

Task-dependent code

Glue
Code
Can we automatically generate glue code?

What’s wrong with glue code:
● Takes time to write, slows down research progress
● Boring and repetitive
● Error-prone
● …

Automatically generate glue code:
● Focus on NLP/ML pipelines for now
● Focus on the case where we need to transform the output data from an upstream software A to the input of a downstream task B
Code (Data structure, API):
```java
class ParsedSentence {
    int[] tokenIds;
    int[] depParents;
    ....
}
```

Specification/Comments:
```plaintext
/* output format =
word_id \t word \t parent_id \t label */
/* input format =
parent_id,child_id,label_id */
```

Sample input/output:
```
1 the 2 DT ...
2 cat 3 NN ...
3 sits 0 VB ...
```

Formal representation and invariants for the data:
```
tokenIds: List[Int], parseTreeArcs: List[(Int, Int)] ...
\forall t \in \text{tokenIds}: 0 \leq t \leq \text{numWords}, \forall (x,y) \in \text{parseTreeArcs}: 0 \leq x, y \leq |\text{tokenIds}| ...
```

Glue code
that transformat output data from software A to the input data of software B.

Tests
based on the invariants.

Specifications
that explains the input/output format.