Automatic Detection of Pseudocodes in Scholarly Documents Using Machine Learning

Suppawong Tuarob\textsuperscript{a}, Sumit Bhatia\textsuperscript{b}, Prasenjit Mitra\textsuperscript{c}, and C Lee Giles\textsuperscript{d}\textsuperscript{a}

\textsuperscript{a}Computer Science and Engineering, Information Sciences and Technology
The Pennsylvania State University, University Park PA 16802
suppawong@psu.edu, sumit@cse.psu.edu, pmitra@ist.psu.edu, giles@ist.psu.edu

• A significant number of scholarly articles in computer science and other disciplines contain algorithms.
• Automatic identification and extraction of these algorithms from scholarly digital documents would enable automatic algorithm indexing, searching, analysis, and discovery. An algorithm search engine, which identifies pseudocodes in scholarly documents and makes them searchable, has been implemented as a part of the CiteSeer\textsuperscript{x} suite.
• Here, we present three methods for detecting pseudocodes in scholarly documents, including an extension of the existing rule based method proposed by Bhatia et al.\textsuperscript{[1]} (PC-RB), one based on machine learning techniques (PC-ML), and a combination of these two (PC-CB).

Approach 1: PC-RB:
1. Assume that each pseudocode has an accompanying caption.
2. Use regular expression to detect the presence of pseudocode captions.

Approach 2: PC-ML:
1. Detect sparse regions (sparse boxes) in the document. Examples of sparse boxes are given in Figure 2.
2. Extract 47 features from each sparse box. Table 1 provides the complete list of the features.
3. 12 base classifiers (listed in Figure 3) are trained and tested on the held-out data.
4. Each first 2, 3, …, 12 ranked classifiers are used for majority voting and probability averaging ensemble methods.

* Note that we also try other ensemble methods such as Adaboost, Bagging, and Rotation Forest but overall the majority voting and probability averaging methods perform much better.

Approach 3: PC-CB:
1. For a given document, run both PC-RB and PC-ML.
2. For each pseudocode box detected by PC-ML, check whether there is a pseudo-code caption detected by PC-RB nearby. If there is, the pseudocode box and the caption are combined.

Evaluation:
We evaluate the three pseudocode detection algorithms on a dataset of 258 scholarly PDF documents randomly selected from CiteSeer\textsuperscript{x} consisting of 275 pseudocodes, using 10-fold document-wise cross validation.

The evaluation metrics include precision, recall, and F-measure.

Discussion:
Table 2 lists notable results. As expected, the rule-based method (PC-RB) yields high precision with a cost of low recall. Using machine learning techniques (PC-ML), the overall performances (in terms of F1) are improved. The combine method (PC-CB) of PC-RB and a majority voting of LMT, Random Forest, and RIPPER classification models performs the best in terms of F1, improving the performance over the state-of-the-art (the rule based method) by 15% (The recall is improved by 22.6%, while the precisions are on par).

Figure 4 compares the performances (in terms of F1) between the ensemble methods and the best baseline classifiers for PC-ML (MLR) and PC-CB (LMT). The X-axis denotes the first k baseline classifiers, ranked by their F1 scores, used in each ensemble method. We conclude that the ensemble methods are useful when the best baseline classifiers are combined. However, the performance of ensemble methods can decrease as the number of classifiers grows.

Future Works:
- Investigate scalability for large datasets such as the over 3 million documents in CiteSeer\textsuperscript{x} repository.
- Employing the co-training technique to expand the training data with unlabeled data.
- Investigate other types of algorithm representation in scientific papers (flowcharts, step-by-step descriptions, etc.).

References:

Table 1: Features set for pseudo-code box classification can be divided into 4 groups: font style based (FS), context based (CB), content based (CT), and structure based (ST).

FIGURE 4: Comparison of the ensemble methods against the best baseline classifiers in PC-ML and PC-CB.

Table 2: Precision, Recall, and F1 of the PSEUDOCODE DETECTION METHODS USING DIFFERENT CLASSIFICATION METHODS. (*"DENOTES MAJORITY VOTING, **"DENOTES PROBABILITY AVERAGING)

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>P%</th>
<th>R%</th>
<th>F1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC-RB</td>
<td>RuleBased</td>
<td>87.12</td>
<td>44.57</td>
<td>58.97</td>
</tr>
<tr>
<td>PC-ML</td>
<td>MLR\textsuperscript{b}</td>
<td>80.35</td>
<td>56.78</td>
<td>66.54</td>
</tr>
<tr>
<td>PC-ML</td>
<td>LMT-RF-RIPPER-MLR</td>
<td>85.31</td>
<td>57.04</td>
<td>68.37</td>
</tr>
<tr>
<td>PC-ML</td>
<td>NB-RIPPER-LMT-MLR</td>
<td>79.61</td>
<td>59.37</td>
<td>68.02</td>
</tr>
<tr>
<td>PC-ML</td>
<td>LMT-RF-RIPPER</td>
<td>78.26</td>
<td>60.05</td>
<td>67.96</td>
</tr>
<tr>
<td>PC-CB</td>
<td>LMT\textsuperscript{b}</td>
<td>79.64</td>
<td>67.89</td>
<td>73.30</td>
</tr>
<tr>
<td>PC-CB</td>
<td>LMT-RF-RIPPER</td>
<td>83.73</td>
<td>67.17</td>
<td>75.95</td>
</tr>
<tr>
<td>PC-CB</td>
<td>NB-RIPPER-LMT-MLR</td>
<td>78.28</td>
<td>70.72</td>
<td>74.31</td>
</tr>
<tr>
<td>PC-CB</td>
<td>NB</td>
<td>37.86</td>
<td>75.89</td>
<td>50.52</td>
</tr>
</tbody>
</table>