Self-Adaptive Classifier for Efficient Text-stream Processing
Naoki Yoshinaga (Univ. Tokyo) and Masaru Kitsuregawa (NII; Univ. Tokyo)

Introduction
A social text stream (e.g., twitter) mirrors the state of real world, so analyzing a real-time text stream is beneficial for reducing natural disasters, monitoring sentiment, predicting stock market etc.

Challenge: The content and volume of flow changes dramatically in a text stream, reflecting a change in the real world

Proposal
We dynimize a linear classifier based on feature sequence trie [Yoshinaga & Kitsuregawa '09] so that it adaptively speeds up classification while processing a text stream

Classification based on feature sequence trie [Yoshinaga & Kitsuregawa, EMNLP '09]
Use of conjunctive features (e.g., n-grams) improves accuracy but slows down processing time in NLP tasks

Then, solve common classification problems in advance to quickly solve new problems as their instances
- Use global statistics to select common problems
- Store problems in a feature sequence trie for fast retrieval

Problem: it cannot effectively speed up when a burst occurs and the topic (content) shifts in a text stream

Self-adaptive classification for text stream [this paper]
Keep updating a set of common classification problems while processing text:
1. build/enumerate common classification problems by adding frequent features in input one by one
2. get a margin if exists, o/w compute/store a margin

Experiments
- Data: Tweet stream on 3.11 Earthquake (9M posts in Japanese)
- Tasks: base-phrase chunking / dependency parsing
- Models: pointwise chunker / shift-reduce parser [Sassano '04]
- Base classifier: PA-I with 3rd-order poly kernel

Overall classification performance for tweets on 3.11 Earthquake

Impact of the number of common classification problems, k

<table>
<thead>
<tr>
<th>Method</th>
<th>Parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Time [ms/classify]</td>
</tr>
<tr>
<td>[Kudo &amp; Matsumoto '03]</td>
<td>0.0221</td>
</tr>
<tr>
<td>[Yoshinaga &amp; Kitsuregawa '09]</td>
<td>0.0118</td>
</tr>
<tr>
<td>This paper</td>
<td>0.0088</td>
</tr>
<tr>
<td>(LFU, k=2²⁰)</td>
<td>0.0081</td>
</tr>
<tr>
<td>(LRU, k=2²⁰)</td>
<td>0.0077</td>
</tr>
<tr>
<td>(LRU, k=2²⁴)</td>
<td>0.0070</td>
</tr>
</tbody>
</table>

Environment: Intel Core i7-3720QM 2.6GHz CPU server with 16GB RAM

All the codes are available as open-source softwares at http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/