Back to Patterns:
Efficient Morphological Analysis with Feature-Sequence Trie

Naoki Yoshinaga
Institute of Industrial Science, The University of Tokyo
Whereas data is increasing, models become slower

- **Text data has been increasing since the birth of the Web**
  - SNS posts via smartphones
  - Communication via zoom/slack

- **NLP models become slower, focusing on accuracy**
  - *Efficient* neural methods are only *relatively efficient* and are not fast

<table>
<thead>
<tr>
<th>Implementation of Japanese Morphological Analyzer (MA)</th>
<th>Speed [sents/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Juman [Kurohashi+ 1994]</td>
<td>8802</td>
</tr>
<tr>
<td>MeCab [Kudo+ 2004]</td>
<td>52410</td>
</tr>
<tr>
<td>KyTea [Neubig+ 2011]</td>
<td>4892</td>
</tr>
<tr>
<td>Juman++V1 [Morita+ 2015]</td>
<td>16</td>
</tr>
<tr>
<td>Juman++V2 [Tolmachev+ 2018] [Tolmachev+ 2018]</td>
<td>4803</td>
</tr>
</tbody>
</table>

The outdated yet *sota efficient* methods have been used for ages to process the increasing textual data for sociolinguistics and marketing
Proposal: Pattern-based method for Japanese MA

• **Approach**: making pattern-based methods more accurate, instead of making neural methods more efficient

• **Proposal**: Pattern-based Japanese morphological analysis (MA) word segmentation, POS tagging, lemmatization
  
  • Regard segmentation and tagging as multi-class classification problem
  
  • Greedily solve this classification problem from left to right using patterns extracted from the training data and a dictionary

  Avoid expensive argmax operations used in learning-based methods
Running example (stop videos and look)
趣味のない人がいる。

shumi no nai hito ga iru.

Pattern | Word | POS (level 1)
趣味 | 趣味 | NOUN
の | の | ADP
ない | ない | ADJ
人 | 人 | NOUN
が | が | ADP
いる | いる | VERB
。 | 。 | PUNCT
Results (excerpt)

• Compare our method (Jagger) to sota efficient learning-based methods (MeCab, Vibrato, Vaporetto) using the same dictionary

  • Environments: M2 MacBook Air with a 3.5-GHz CPU and 24-GB RAM

<table>
<thead>
<tr>
<th>Method</th>
<th>Kyoto-U. Text Corpus (news)</th>
<th>Kyoto-U. Web Doc. Leads Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>speed [sent/s]</td>
<td>mem [MiB]</td>
</tr>
<tr>
<td>MeCab</td>
<td>66,455</td>
<td>55.81</td>
</tr>
<tr>
<td>Vibrato</td>
<td>142,983</td>
<td>97.75</td>
</tr>
<tr>
<td>Vaporetto</td>
<td>117,767</td>
<td>658.80</td>
</tr>
<tr>
<td>Jagger</td>
<td><strong>1,007,344</strong></td>
<td><strong>26.39</strong></td>
</tr>
</tbody>
</table>

Jagger processes **1M sents/s** with accuracy comparable to baselines
Takeaways

• Since accuracies are becoming saturated on NLP benchmarks, let’s focus more on underrepresented metrics, e.g., efficiency.

• Back to Patterns: Patterns are more powerful than you think
  • It can rival learning-based methods in Japanese MA in terms of accuracy, and is 7-16x faster with 1/2-1/20 memory footprint.

• Take a speed-intensive approach to absolute efficiency in NLP
  • Making very slow neural methods (slightly) fast seems uncompelling
  • Making a fast pattern-based method more accurate is compelling

Code: http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/jagger/