Port mtplz to Joshua

Machine Translation Marathon
8–14 September 2014

Matt Post 
Pēteris Ėnikiforoovs
Motivation

- All decoders are either phrase- or syntax-based: cdec, Moses*, Jane, Joshua, Phrasal, Portage, UCAM
- Sometimes you want speed, sometimes power
- Research potential having both (e.g., deep chart integration)
Joshua

- Goal: deep integration (lattice decoding, multiple language models, multithreading)
- Community desire to distribute it through Apache projects (e.g., Maven)
  - It would be nice to use a faster formalism!
- A permissive license (BSD 2-clause)
Heafield, Kayser, and Manning (2014) wrote a “bare-bones” phrase-based decoder, “mtplz”

- 4.0 to 7.7 times faster than Moses

- 2,830 lines of code (not counting KenLM)

- KenLM himself on hand, offering consultations at basement-bottom rates

- Backup plan: cube pruning
“It shouldn’t take that long to port a bare-bones phrase-based decoder.”

— Matt Post (Sunday, September 7, 2014)
High level steps

- Crash run-through of mtplz ideas and code
- Determine how to translate C++ exclusives
- Port phrase-table reader
- Write tests
- JNI interface with KenLM
- Cube-pruning implementation
mtplz source code — highly optimized

// This part should be self-explanatory — KPU
00000000004096e0 <main>:

4096e0: 41 57           push   %r15
4096e2: 41 56           push   %r14
4096e4: 41 55           push   %r13
4096e6: 41 54           push   %r12
4096e8: 55              push   %rbp
4096e9: 53              push   %rbx
4096ea: 48 81 ec 38 06 00 00  sub   $0x638,%rsp
4096f1: 48 8b 05 50 d6 2a 00  mov    0x2ad650(%rip),%rax
4096f8: 89 7c 24 24     mov    %edi,0x24(%rsp)
4096fc: 48 89 74 24 28  mov    %rsi,0x28(%rsp)

// TODO: optimize
409701: 48 8d 54 24 4f  lea    0x4f(%rsp),%rdx
409706: 48 8d 35 3f f5 08 00  lea    0x8f53f(%rip),%rsi
40970d: 48 8d 7c 24 70  lea    0x70(%rsp),%rdi
409712: 8b 18           mov    (%rax),%ebx
409714: 89 dd           mov    %ebx,%ebp
409716: d1 ed           shr    %ebp

// RDRR, LOL
409718: e8 53 f1 ff ff ff  callq   408870 <__ZNSsC1EPKcRKSaIcE@plt>
40971d: 48 8d 74 24 70  lea    0x70(%rsp),%rsi

(this is a joke)
# Progress Report

<table>
<thead>
<tr>
<th>Item</th>
<th>mtplz</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>compiles?</td>
<td>yes (*)</td>
<td>yes</td>
</tr>
<tr>
<td>lines of code</td>
<td>2,830</td>
<td>3,418</td>
</tr>
<tr>
<td>classes</td>
<td>89</td>
<td>50</td>
</tr>
</tbody>
</table>
High level steps

- Crash run through of mtplz ideas and code
- Determine how to translate C++ exclusives
- Port phrase table reader
- Write tests
- JNI interface with KenLM
- Cube-pruning implementation
This fall

- Finish port
- Release Joshua version 6.0
  - Includes a number of other improvements
  - Will likely be decoder-only (relying on Moses for building models)

joshua-decoder.org
Poster boaster session
MT leaderboard

• Setup and install in seconds
• Hosted on Google App Engine (= free hosting)
• Manage submissions, automatically grade them
• Display leaderboard (coming: coopera-board)
• Easily define new assignments (includes five)

**code**  github.com/mjpost/leaderboard
**class**  mt-class.org/jhu/leaderboard.html
Submit your assignments [here](https://example.com). Results will be updated immediately.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Handle</th>
<th>Setup Number</th>
<th>Align AER</th>
<th>Decode Model score</th>
<th>Evaluate Accuracy</th>
<th>Rerank BLEU</th>
<th>Inflect Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NPS</td>
<td>-inf</td>
<td>0.361</td>
<td>-1273.243</td>
<td>0.733</td>
<td>-inf</td>
<td>0.934</td>
</tr>
<tr>
<td>2</td>
<td>fallingoffthebeam</td>
<td>11.000</td>
<td>0.123</td>
<td>-1339.466</td>
<td>0.842</td>
<td>29.005</td>
<td>0.705</td>
</tr>
<tr>
<td>3</td>
<td>ksk</td>
<td>2.000</td>
<td>0.169</td>
<td>-1254.000</td>
<td>0.785</td>
<td>29.080</td>
<td>0.690</td>
</tr>
<tr>
<td>4</td>
<td>rk</td>
<td>42.700</td>
<td>0.241</td>
<td>-1237.886</td>
<td>0.513</td>
<td>29.200</td>
<td>0.673</td>
</tr>
<tr>
<td>5</td>
<td>Tejasvam.Singh</td>
<td>-inf</td>
<td>0.355</td>
<td>-inf</td>
<td>-inf</td>
<td>-inf</td>
<td>0.670</td>
</tr>
<tr>
<td>6</td>
<td>matt</td>
<td>-inf</td>
<td>inf</td>
<td>-inf</td>
<td>0.522</td>
<td>-inf</td>
<td>0.666</td>
</tr>
<tr>
<td>7</td>
<td>HarryTheHB</td>
<td>5.000</td>
<td>0.352</td>
<td>-1223.265</td>
<td>0.512</td>
<td>29.212</td>
<td>0.643</td>
</tr>
<tr>
<td>8</td>
<td>JoeSachmo</td>
<td>23.000</td>
<td>0.371</td>
<td>-inf</td>
<td>-inf</td>
<td>-inf</td>
<td>0.642</td>
</tr>
<tr>
<td>9</td>
<td>baseline</td>
<td>89.000</td>
<td>0.376</td>
<td>-1316.091</td>
<td>0.504</td>
<td>28.863</td>
<td>0.641</td>
</tr>
<tr>
<td>10</td>
<td>query</td>
<td>77.770</td>
<td>0.182</td>
<td>-1266.898</td>
<td>0.786</td>
<td>28.773</td>
<td>0.620</td>
</tr>
<tr>
<td>11</td>
<td>default</td>
<td>100.000</td>
<td>0.681</td>
<td>-1439.874</td>
<td>0.449</td>
<td>27.351</td>
<td>0.585</td>
</tr>
<tr>
<td>12</td>
<td>oracle</td>
<td>-inf</td>
<td>-inf</td>
<td>-1215.561</td>
<td>-inf</td>
<td>40.057</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>BersaKAIN</td>
<td>56.015</td>
<td>inf</td>
<td>-inf</td>
<td>-inf</td>
<td>27.952</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>jay</td>
<td>-1.000</td>
<td>0.300</td>
<td>-1286.206</td>
<td>0.515</td>
<td>26.585</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>rooms_to_let</td>
<td>-inf</td>
<td>inf</td>
<td>-inf</td>
<td>0.670</td>
<td>-inf</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>JaLozR</td>
<td>7.000</td>
<td>0.331</td>
<td>-inf</td>
<td>0.516</td>
<td>-inf</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>lièvre</td>
<td>6.000</td>
<td>0.177</td>
<td>-1298.857</td>
<td>-inf</td>
<td>-inf</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>aricvelbel</td>
<td>100.000</td>
<td>0.123</td>
<td>-1421.774</td>
<td>-inf</td>
<td>-inf</td>
<td>-inf</td>
</tr>
<tr>
<td></td>
<td>realcaoyuan</td>
<td>-inf</td>
<td>inf</td>
<td>-inf</td>
<td>-inf</td>
<td>-inf</td>
<td>-inf</td>
</tr>
</tbody>
</table>