

C++



Java



Port mtplz to Joshua

Machine Translation Marathon
8–14 September 2014

Matt Post

Pēteris Ņikiforovs



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)

A circular logo featuring a Joshua tree in a desert landscape, positioned behind the title text.

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)

mtplz

- 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
0000000004096e0 <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      callq 408870 <_ZNSsC1EPKcRKSaIcE@plt>
 40971d:    48 8d 74 24 70       lea   0x70(%rsp),%rsi
```

(this is a joke)

Progress Report

Item	mtplz	port
compiles?	yes (*)	yes
lines of code	2,830	3,418
classes	89	50

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

MT leaderboard



Submit your assignments [here](#). Results will be updated immediately.

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