



Left Language Model State for Syntactic Machine Translation



Kenneth Heafield

heafield@cs.cmu.edu

Hieu Hoang

hieu.hoang@asiaonline.net

Philipp Koehn

pkoehn@inf.ed.ac.uk

Tetsuo Kiso

tetsuo-s@is.naist.jp

Marcello Federico

federico@fbk.eu

Concatenation in a 5-Gram Language Model

$$\begin{aligned}
 & p(\text{Australia is one of the few}) \\
 \times & \quad p(\text{countries that maintain diplomatic relations with North}) \\
 \times & \quad \text{adjust}(\text{one of the few, countries that maintain diplomatic}) \\
 = & p(\text{Australia is one of the few countries that maintain diplomatic relations with North})
 \end{aligned}$$

State

Sentence fragments have left state and right state:

Left State

Right State

countries that maintain diplomatic relations with North Korea .

The decoder can recombine fragments with equal state.

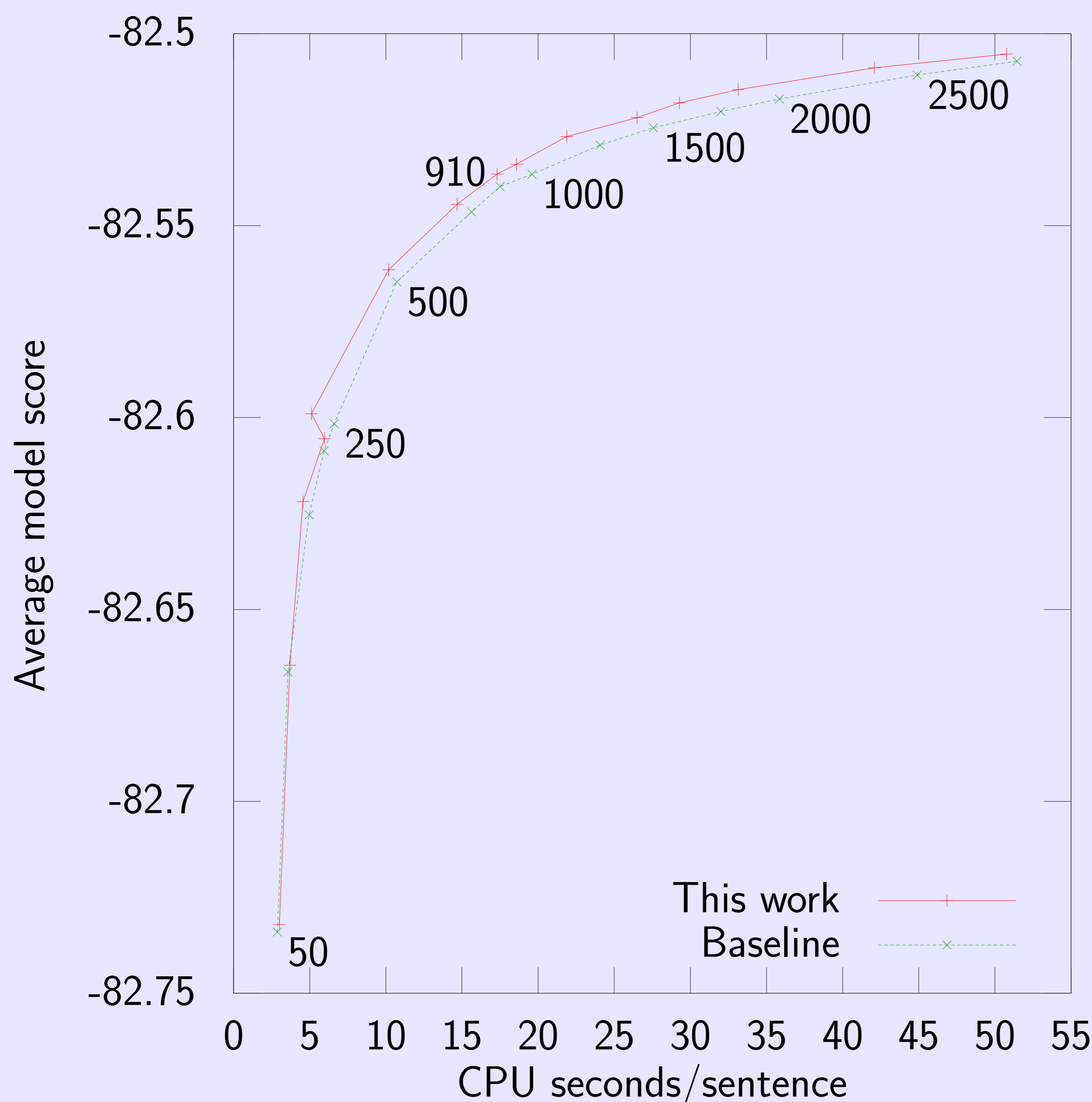
Optimizing Concatenation

Baseline LMs minimize right state length. In addition, we:

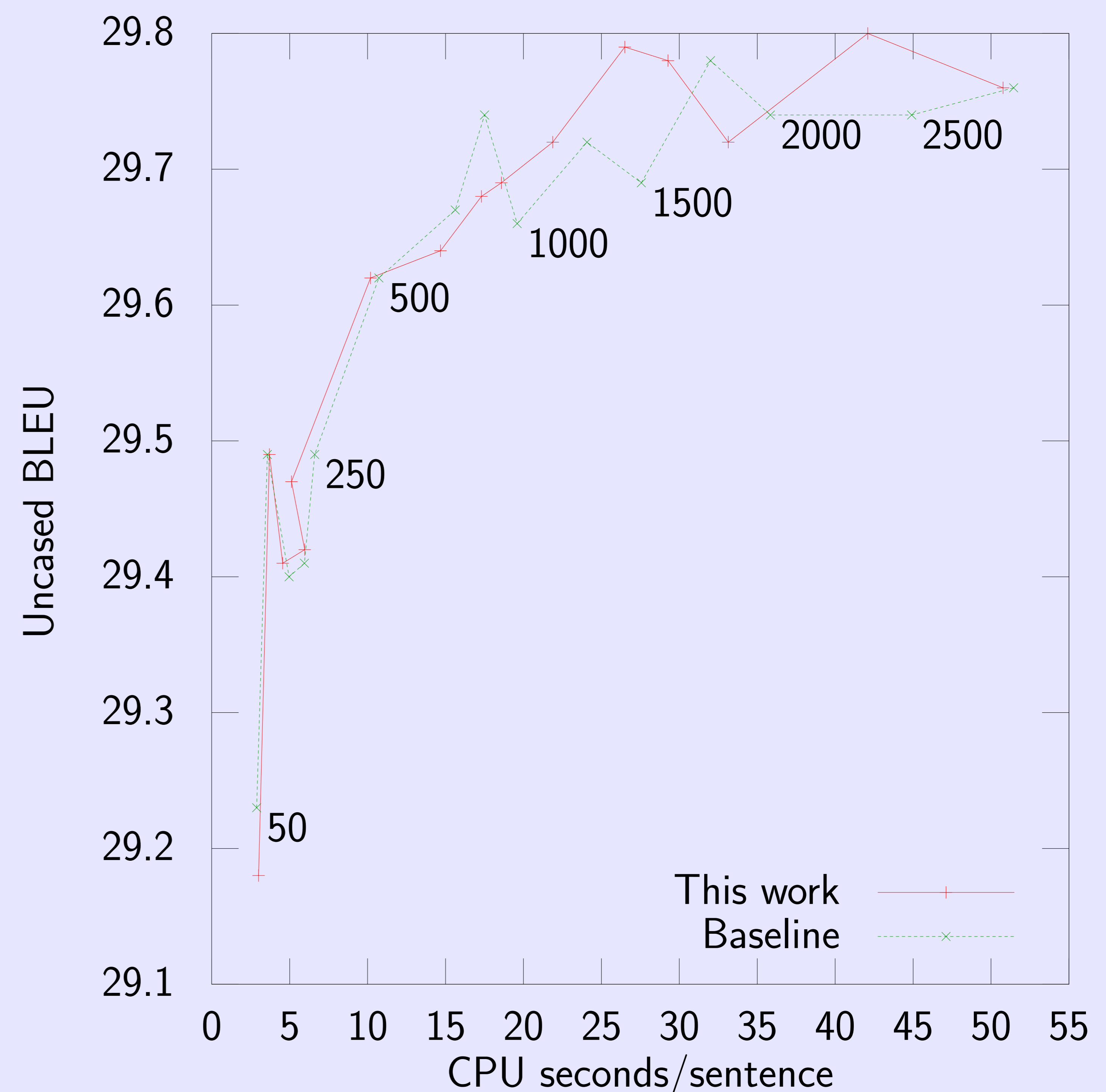
- Minimize left state length, increasing recombination
- Encode left state using pointers, reducing lookup cost
- Exit scoring early when an n -gram is provably not present

Time-Accuracy Tradeoff

Model score versus Time



BLEU versus Time



State Length Predicts Score

Left Length	Right Length			
	1	2	3	4
0	-0.741	-1.062	-1.357	-1.701
1	-0.269	-0.429	-0.588	-0.836
2	-0.129	-0.236	-0.362	-0.567
3	0.007	-0.061	-0.128	-0.314
4	0.220	0.202	0.169	0.037

Short left state predicts poor performance.

Conclusion

- Equivalent quality with 11% net reduction in CPU time.
- Left state minimization combines fragments that perform poorly.
- Right state minimization combines fragments that perform well.
- Future work using state length as a rest cost estimator.
- Clean high-level C++ interface for language models in syntactic decoders.
- Live in Moses and cdec.

<http://kheafield.com/code/kenlm/>