Exact Decoding of Phrase-Based Translation Models through Lagrangian Relaxation: Supplementary Material

Yin-Wen Chang
MIT CSAIL
Cambridge, MA 02139, USA
yinwen@csail.mit.edu

Michael Collins
Department of Computer Science,
Columbia University,
New York, NY 10027, USA
mcollins@cs.columbia.edu

A An Example Run of the Algorithm in Figure 3

Figure 1 gives an example run of the algorithm. After 31 iterations the algorithm detects that the dual is no longer decreasing rapidly enough, and runs for $K = 10$ additional iterations, tracking which constraints are violated. Constraints $y(6) = 1$ and $y(10) = 1$ are each violated 10 times, while other constraints are not violated. A recursive call to the algorithm is made with $\mathcal{C} = \{6, 10\}$, and the algorithm converges in a single iteration, to a solution that is guaranteed to be optimal.

B Speeding up the DP: A* Search

In the algorithm depicted in Figure 3, each time we call $Optimize(\mathcal{C} \cup \mathcal{C}', u)$, we expand the number of states in the dynamic program by adding hard constraints. On the graph level, adding hard constraints can be viewed as expanding an original state in $\mathcal{Y}'$ to $2^{|\mathcal{C}|}$ states in $\mathcal{Y}'_{\mathcal{C}}$, since now we keep a bit-string $b_{\mathcal{C}}$ of length $|\mathcal{C}|$ in the states to record which words in $\mathcal{C}$ have or haven't been translated. We now show how this observation leads to an A* algorithm that can significantly improve efficiency when decoding with $\mathcal{C} \neq \emptyset$.

For any state $s = (w_1, w_2, n, l, m, r, b_{\mathcal{C}})$ and Lagrange multiplier values $u \in \mathbb{R}^N$, define $\beta_{\mathcal{C}}(s, u)$ to be the maximum score for any path from the state s to the end state, under Lagrange multipliers u, in the graph created using constraint set $\mathcal{C}$. Define $\pi(s) = (w_1, w_2, n, l, m, r)$, that is, the corresponding state in the graph with no constraints ($\mathcal{C} = \emptyset$). Then for any values of s and u, we have

$$\beta_{\mathcal{C}}(s, u) \leq \beta_{\emptyset}(\pi(s), u)$$

That is, the maximum score for any path to the end state in the graph with no constraints, forms an upper bound on the value for $\beta_{\mathcal{C}}(s, u)$.

This observation leads directly to an A* algorithm, which is exact in finding the optimum solution, since we can use $\beta_{\emptyset}(\pi(s), u)$ as the admissible estimates for the score from state s to the goal (the end state). The $\beta_{\emptyset}(s', u)$ values for all s' can be calculated by running the Viterbi algorithm using a backwards path. With only $1/2^{|\mathcal{C}|}$ states, calculating $\beta_{\emptyset}(s', u)$ is much cheaper than calculating $\beta_{\mathcal{C}}(s, u)$ directly. Guided by $\beta_{\emptyset}(s', u)$, $\beta_{\mathcal{C}}(s, u)$ can be calculated efficiently by using A* search.

Using the A* algorithm leads to significant improvements in efficiency when constraints are added. Section 6 presents comparison of the running time with and without A* algorithm.

Input German: es bleibt jedoch dabei , dass kolumbien ein land ist , das aufmerksam beobachtet werden muss .

t	$L(u^{t-1})$	$y^t(i)$	derivation y^t												
1	-11.8658	0 0 0 0 1 3 0 3 3 4 1 1 0 0 0 0 1	5, 6 that	10, 10 is	8, 9 a country	6, 6 that	10, 10 is	8, 9 a country	6, 6 that	10, 10 is	8, 8 a	9, 12 country that	17, 17 .		
2	-5.46647	2 2 4 0 2 0 1 0 0 0 1 0 1 1 1 1 1	3, 3 however ,	1, 1 it	2, 3 is , however	5, 5 ,	3, 3 however ,	1, 1 it	2, 3 is , however	5, 5 ,	7, 7 colombia	11, 11 ,	16, 16 must	13, 15 be closely monitored	17, 17 .
	⋮														
32	-17.0203	1 1 1 1 1 0 1 1 1 2 1 1 1 1 1 1 1	1, 5 nonetheless ,	7, 7 colombia	10, 10 is	8, 8 a	9, 12 country that	16, 16 must	13, 15 be closely monitored	17, 17 .					
33	-17.1727	1 1 1 1 1 2 1 1 1 0 1 1 1 1 1 1 1	1, 5 nonetheless ,	6, 6 that	8, 9 a country	6, 6 that	7, 7 colombia	11, 12 , which	16, 16 must	13, 15 be closely monitored	17, 17 .				
34	-17.0203	1 1 1 1 1 0 1 1 1 2 1 1 1 1 1 1 1	1, 5 nonetheless ,	7, 7 colombia	10, 10 is	8, 8 a	9, 12 country that	16, 16 must	13, 15 be closely monitored	17, 17 .					
35	-17.1631	1 1 1 1 1 0 1 1 1 2 1 1 1 1 1 1 1	1, 5 nonetheless ,	7, 7 colombia	10, 10 is	8, 8 a	9, 12 country that	16, 16 must	13, 15 be closely monitored	17, 17 .					
36	-17.0408	1 1 1 1 1 2 1 1 1 0 1 1 1 1 1 1 1	1, 5 nonetheless ,	6, 6 that	8, 9 a country	6, 6 that	7, 7 colombia	11, 12 , which	16, 16 must	13, 15 be closely monitored	17, 17 .				
37	-17.1727	1 1 1 1 1 0 1 1 1 2 1 1 1 1 1 1 1	1, 5 nonetheless ,	7, 7 colombia	10, 10 is	8, 8 a	9, 12 country that	16, 16 must	13, 15 be closely monitored	17, 17 .					
38	-17.0408	1 1 1 1 1 2 1 1 1 0 1 1 1 1 1 1 1	1, 5 nonetheless ,	6, 6 that	8, 9 a country	6, 6 that	7, 7 colombia	11, 12 , which	16, 16 must	13, 15 be closely monitored	17, 17 .				
39	-17.1658	1 1 1 1 1 2 1 1 1 0 1 1 1 1 1 1 1	1, 5 nonetheless ,	6, 6 that	8, 9 a country	6, 6 that	7, 7 colombia	11, 12 , which	16, 16 must	13, 15 be closely monitored	17, 17 .				
40	-17.056	1 1 1 1 1 0 1 1 1 2 1 1 1 1 1 1 1	1, 5 nonetheless ,	7, 7 colombia	10, 10 is	8, 8 a	9, 12 country that	16, 16 must	13, 15 be closely monitored	17, 17 .					
41	-17.1732	1 1 1 1 1 2 1 1 1 0 1 1 1 1 1 1 1	1, 5 nonetheless ,	6, 6 that	8, 9 a country	6, 6 that	7, 7 colombia	11, 12 , which	16, 16 must	13, 15 be closely monitored	17, 17 .				
		0 0 0 0 0 • 0 0 0 • 0 0 0 0 0 0 0	$count(6) = 10; count(10) = 10; count(i) = 0$ for all other i **adding constraints: 6 10**												
42	-17.229	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1, 5 nonetheless ,	7, 7 colombia	6, 6 that	8, 12 a country that	16, 16 must	13, 15 be closely monitored	17, 17 .						

Figure 1: An example run of the algorithm in Figure 3. At iteration 32, we start the $K = 10$ iterations to count which constraints are violated most often. After K iterations, the count for 6 and 10 is 10, and all other constraints have not been violated during the K iterations. Thus, hard constraints for word 6 and 10 are added. After adding the constraints, we have $y^t(i) = 1$ for $i = 1 \ldots N$, and the translation is returned, with a guarantee that it is optimal.