Optimal Beam Search for Machine Translation

Alexander Rush, Yin-Wen Chang, and Michael Collins
EMNLP 2013

October 19, 2013

Beam Search

beam search is the *de facto* method for translation decoding

- very fast even in the worst-case
- accurate in practice
- implemented in many real-world systems

Beam Search

beam search is the *de facto* method for translation decoding

- very fast even in the worst-case
- accurate in practice
- implemented in many real-world systems

however it provides no formal guarantees about search error

Goal

goal: fast, optimal translation decoding

- better understand what makes translation hard
- quantify search error from beam search
- (at some point) improve translation accuracy

Overview

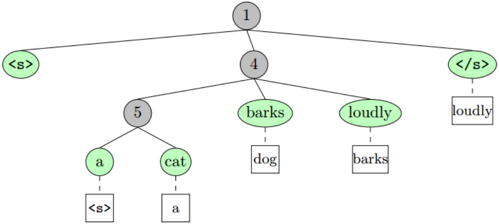
1. translation as constrained graph search
2. certificate properties of beam search
3. relaxation methods for bounding
4. experimental results

Tasks

- phrase-based translation

wir müssen **diese kritik** ernst nehmen
we must take **this criticism** seriously

- syntax-based translation



Overview

1. translation as constrained graph search
2. certificate properties of beam search
3. relaxation methods for bounding
4. experimental results

Phrase-Based Translation Score

- ω; the translation model score
- σ; the language model score

example:

wir müssen diese kritik ernst nehmen

score $=$

Phrase-Based Translation Score

- ω; the translation model score
- σ; the language model score

example:

wir müssen diese kritik ernst nehmen
we must

score = ω(wir müssen, we must) + σ(<s>, we) + σ(we, must) +

Phrase-Based Translation Score

- ω; the translation model score
- σ; the language model score

example:

wir müssen diese kritik ernst **nehmen**
we must **take**

score = ω(wir müssen, we must) + σ(<s>, we) + σ(we, must) +
ω(nehmen, take) + σ(must, take) +

Phrase-Based Translation Score

- ω; the translation model score
- σ; the language model score

example:

wir müssen **diese kritik** ernst nehmen
we must take **this criticism**

score = ω(wir müssen, we must) + σ(<s>, we) + σ(we, must) +
ω(nehmen, take) + σ(must, take) +
ω(diese kritik, this criticism) + σ(take, this) + σ(these, criticism) +

Phrase-Based Translation Score

- ω; the translation model score
- σ; the language model score

example:

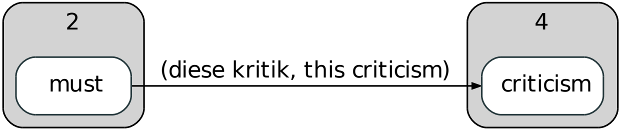
wir müssen diese kritik **ernst** nehmen
we must take this criticism **seriously**

score = ω(wir müssen, we must) + σ(<s>, we) + σ(we, must) +
ω(nehmen, take) + σ(must, take) +
ω(diese kritik, this criticism) + σ(take, this) + σ(these, criticism) +
ω(ernst, seriously) + σ(criticism, seriously) + σ(seriously, </s>)

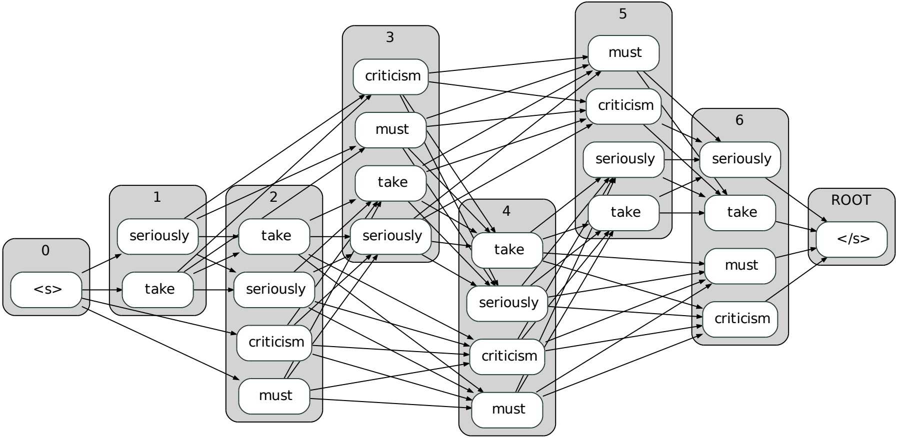
Problem Representation

represent all possible translations in a weighted directed graph

- $\mathcal{V}, \mathcal{E}$; the vertices/edges in the graph.
- $\theta \in \mathbb{R}^{|\mathcal{E}|}$; the weights on edges.
- Vertices are # of source words used and last English word.



$$\begin{aligned}\theta(e) &= \omega(\text{diese kritik}, \text{this criticism}) + \\ &\quad \sigma(\text{must}, \text{this}) + \sigma(\text{this}, \text{criticism})\end{aligned}$$



0
<s>
1
seriously
take
2
take
seriously
criticism
must
3
criticism
must
take
seriously
4
take
seriously
criticism
must
5
must
criticism
seriously
take
6
seriously
take
must
criticism
ROOT
</s>

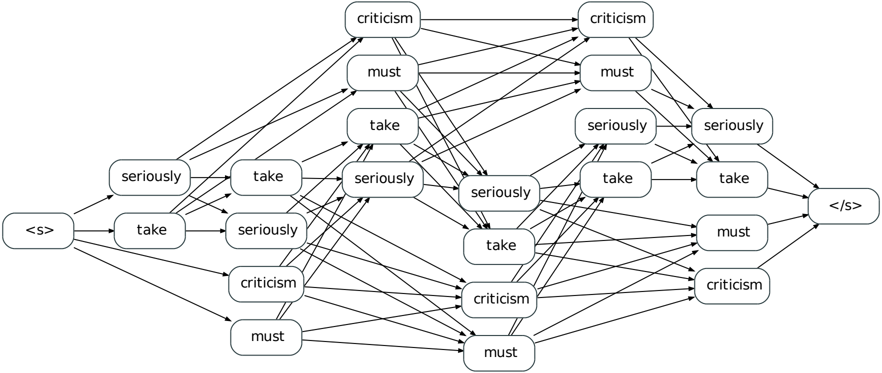
Best Path Problem

- $\mathcal{V}, \mathcal{E}$; the vertices/edges in the graph
- $\theta \in \mathbb{R}^{|\mathcal{E}|}, \tau \in \mathbb{R}$; the weights on edges and an offset
- $\mathcal{X} \subset \{0,1\}^{|\mathcal{E}|}$; the paths in the graph

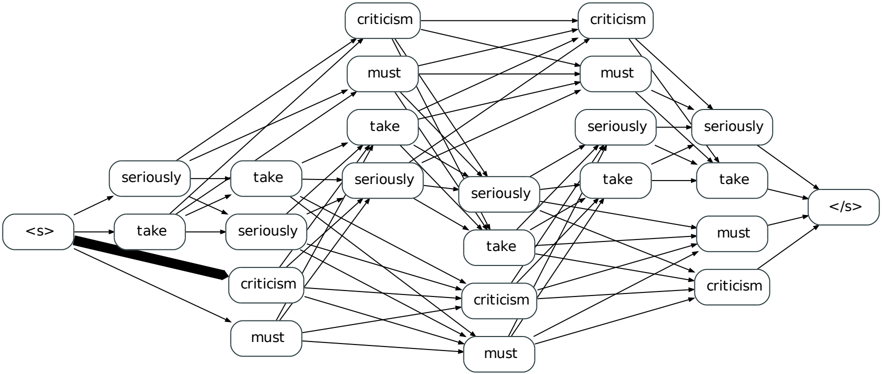
$$\max_{y \in \mathcal{X}} \sum_{e \in \mathcal{E}} \theta(e) y(e) + \tau = \max_{y \in \mathcal{X}} \theta^\top y + \tau$$

- there are an exponential number of paths $|\mathcal{X}|$.

- can be solved in polynomial time, $O(|\mathcal{E}|)$.

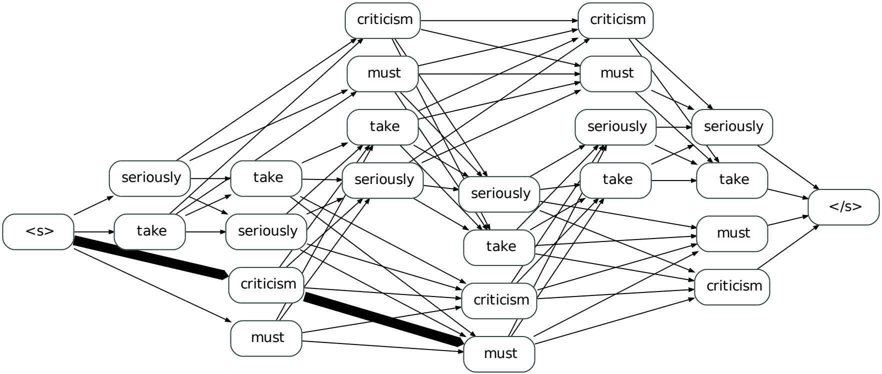


wir müssen diese kritik ernst nehmen

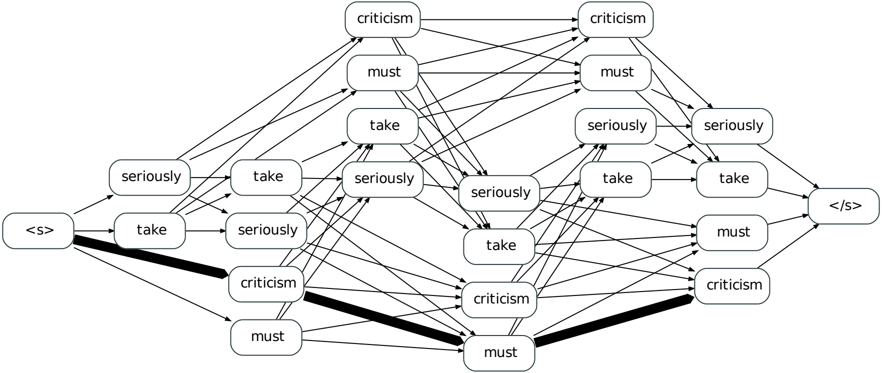


wir müssen **diese kritik** ernst nehmen

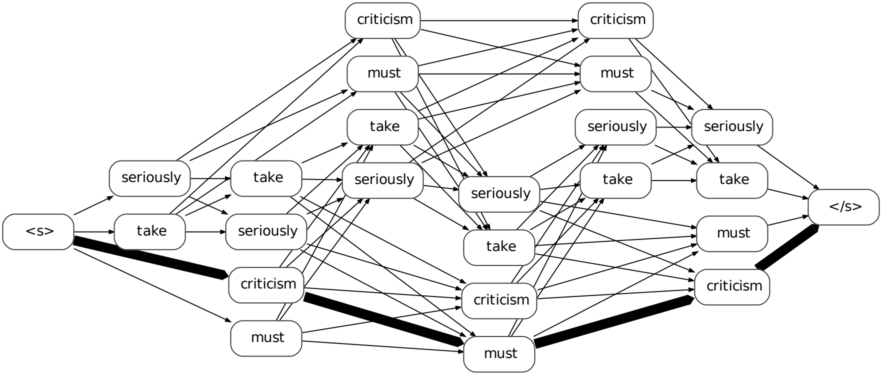
this criticism



wir müssen diese kritik ernst nehmen
this criticism **we must**



wir müssen **diese kritik** ernst nehmen
this criticism we must **this criticism**



wir müssen diese kritik ernst nehmen

this criticism we must this criticism

Constrained Paths

problem: constrain maximization to valid paths

- $A \in \mathbb{R}^{|b| \times |\mathcal{E}|}$; a matrix of linear constraints
- $b \in \mathbb{R}^{|b|}$; a constraint vector
- Ay; a signature.

constrained paths

$$\mathcal{X}' = \{y \in \mathcal{X} : Ay = b\}$$

Example: Source-Language Constraints

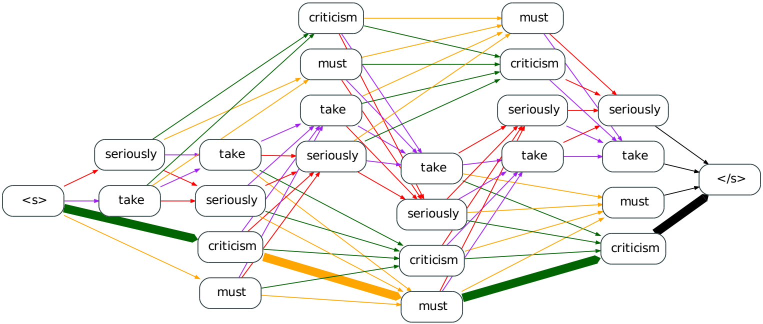
$$A = \begin{array}{l} \text{wir} \\ \text{müssen} \\ \text{diese} \\ \text{kritik} \\ \text{nehmen} \\ \text{ernst} \end{array} \overset{\begin{array}{ccccc} 1 & 2 & 3 & 4 & \dots \end{array}}{\begin{pmatrix} 1 & 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \end{pmatrix}}$$

- each word must be used exactly once for valid $y \in \mathcal{X}'$.

$$Ay = b = \begin{array}{l} \text{wir} \\ \text{müssen} \\ \text{diese} \\ \text{kritik} \\ \text{nehmen} \\ \text{ernst} \end{array} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Violation

$$Ay = \begin{matrix} \text{wir} \\ \text{müssen} \\ \text{diese} \\ \text{kritik} \\ \text{nehmen} \\ \text{ernst} \end{matrix} \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$



wir müssen diese kritik ernst nehmen

this criticism we must this criticism

Overview

1. translation as constrained graph search
2. certificate properties of beam search
3. relaxation methods for bounding
4. experimental results

Beam Search

beam search: explore hypotheses in order.

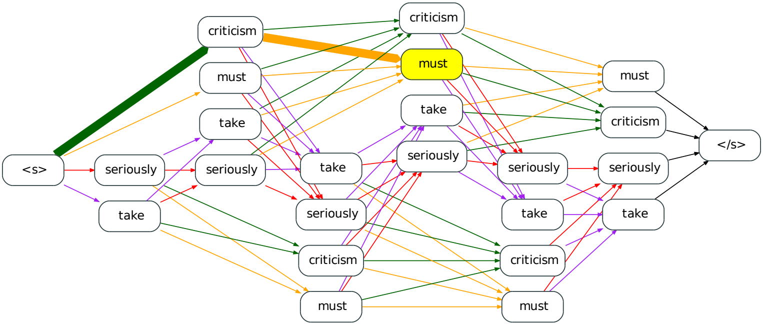
for each hypothesis check:

1. the hypothesis has a valid signature
2. the hypothesis is possibly still optimal
3. the hypothesis fits in the beam

Check 1: Signature Check

is this a valid bitstring?

$$Ay = \begin{matrix} \text{wir} \\ \text{müssen} \\ \text{diese} \\ \text{kritik} \\ \text{nehmen} \\ \text{ernst} \end{matrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

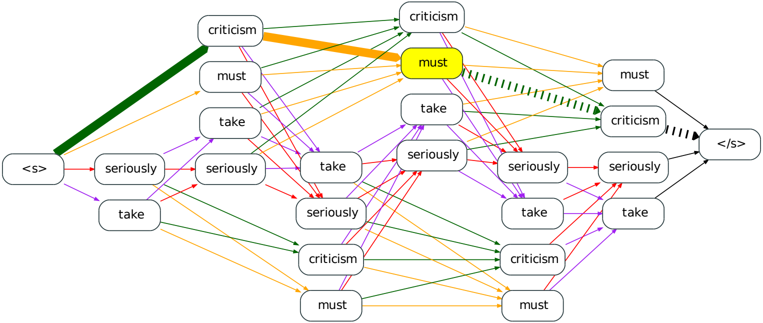


Check 2: Bounding

assume we can compute

- lb $\leq$ opt; a lower-bound on the optimal score
- ubs $\in \mathbb{R}^{|\mathcal{V}|}$; upper bounds on future scores

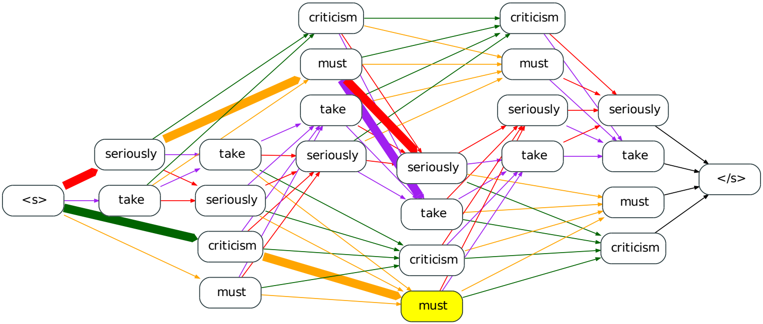
does the inequality $\theta^\top y$ + ubs(v)$\geq$ lb hold?



Check 3: Pruning

are there less than β hypotheses better than y in "beam"?

- β; size of a "hard" beam threshold.



Properties

1. Feasible Path
   the result returned by beam search is a lower bound on the optimal score.

2. Certificate
   if pruning (check 3) is never applied, the result returned by beam search is optimal.

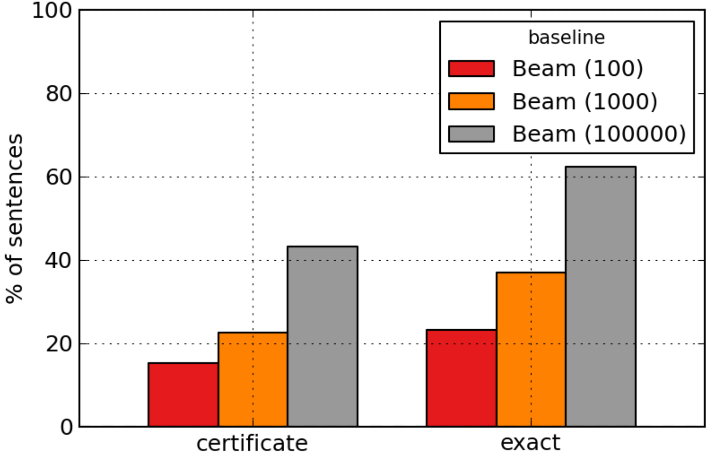
Properties

1. Feasible Path
   the result returned by beam search is a lower bound on the optimal score.

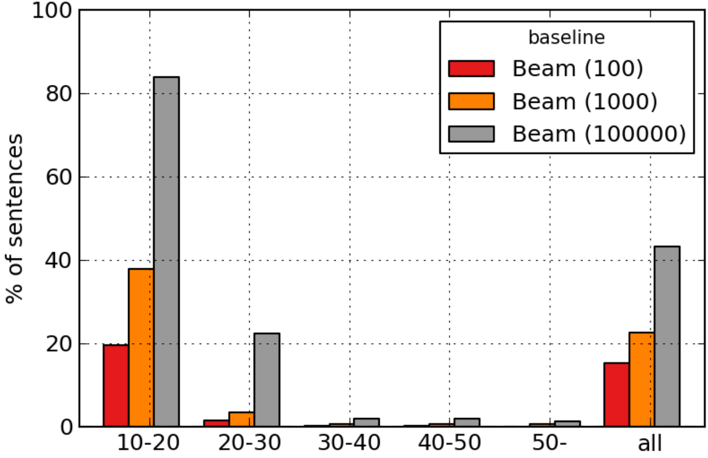
2. Certificate
   if pruning (check 3) is never applied, the result returned by beam search is optimal.

unfortunately in practice certificates are rare.

Preview Results: Beam Search Optimality



Preview Results: Beam Search Certificates



Overview

1. translation as constrained graph search
2. certificate properties of beam search
3. relaxation methods for bounding
4. experimental results

Lagrangian

recall:

$$\max_{y \in \mathcal{X}'} \theta^\top y + \tau$$

$$\mathcal{X}' = \{y \in \mathcal{X} : Ay = b\}$$

define the Lagrangian with multipliers $\lambda \in \mathbb{R}^{|b|}$

$$\begin{aligned} L(\lambda) &= \max_{y \in \mathcal{X}} \theta^\top y + \tau - \lambda^T (Ay - b) \\ &= \max_{y \in \mathcal{X}} (\theta^\top - \lambda^\top A) y + (\tau - \lambda^\top b) \\ &= \max_{y \in \mathcal{X}} \theta'^\top y + \tau' \end{aligned}$$

Properties

Lagrangian

$$L(\lambda) = \max_{y \in \mathcal{X}} \theta^\top y + \tau - \lambda^T (Ay - b) = \max_{y \in \mathcal{X}} \theta'^\top y + \tau'$$

- Preserves Constrained Scores
  if $y \in \mathcal{X}'$ then it has the same objective with modified weights

$$\theta'^\top y + \tau' = \theta^\top y + \tau$$

- Upper Bound
  the best path is an upper bound on the optimal score

$$L(\lambda) = \mathrm{ub} \geq \mathrm{opt}$$

Tighter Upper Bounds

goal: tightest upper bound

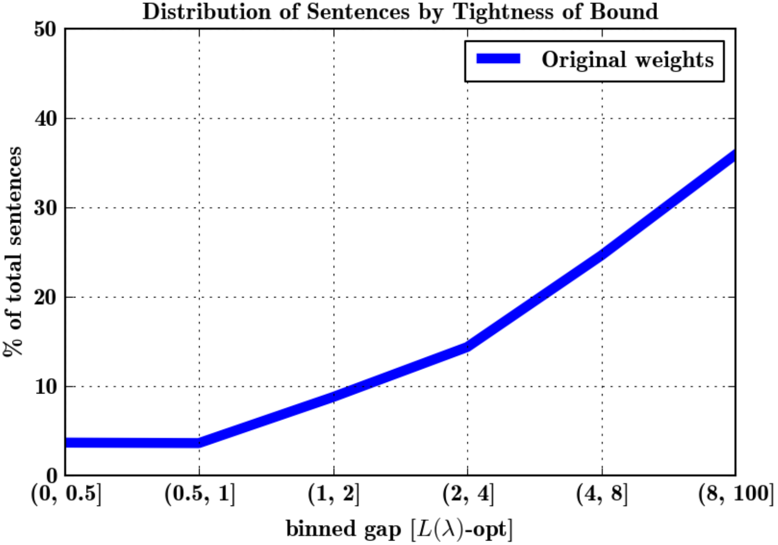
$$\min_{\lambda \in R^{|b|}} L(\lambda)$$

strategy: minimize by subgradient descent ($\alpha_k \in \mathbb{R}$ is a rate)

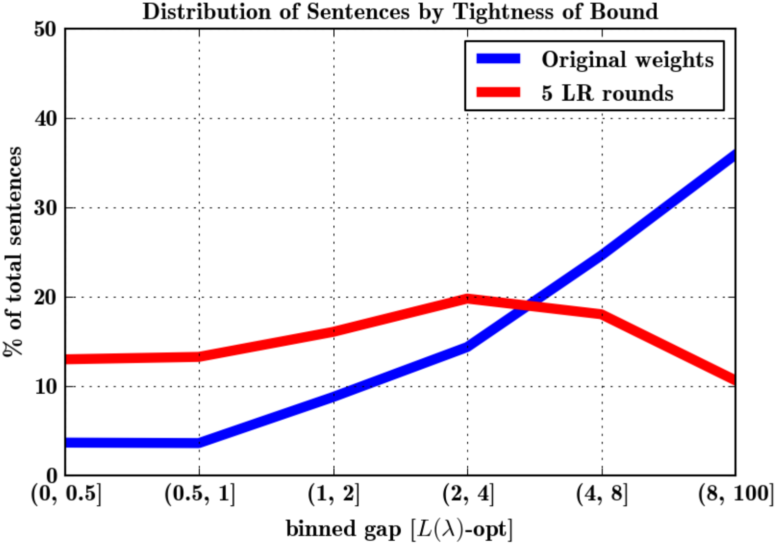
$$y = \arg\max_{y \in \mathcal{X}} \theta'^{\top} y + \tau'$$

$$\lambda^{(k)} \leftarrow \lambda^{(k-1)} - \alpha_k (Ay - b)$$

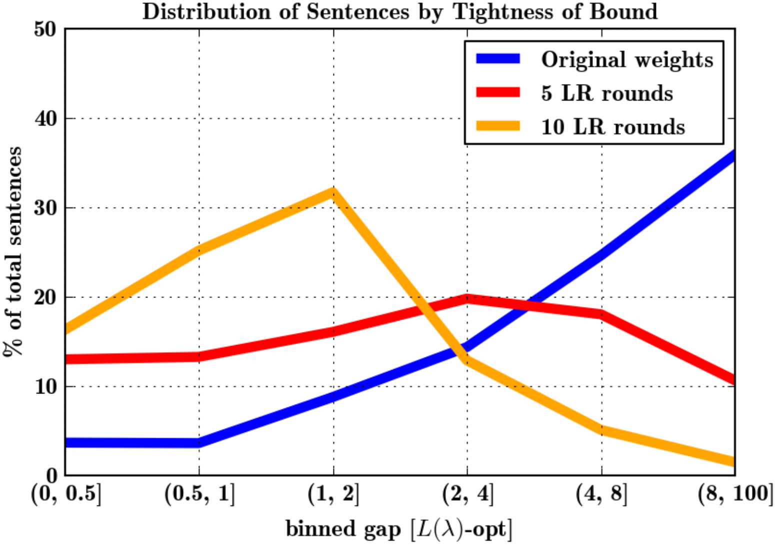
Preview Results: Lagrangian Relaxation and Bounds



Preview Results: Lagrangian Relaxation and Bounds



Preview Results: Lagrangian Relaxation and Bounds



Putting It All Together

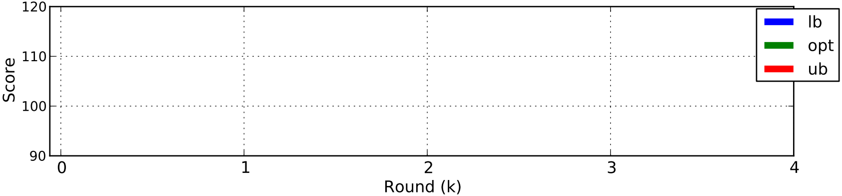
Lagrangian Relaxation

- can decrease the upper bound score
- may not find optimal solution

Beam Search

- can increase the lower bound score
- tighter bounds increase chance of optimal solution

strategy: use modified weights from LR in Beam Search



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

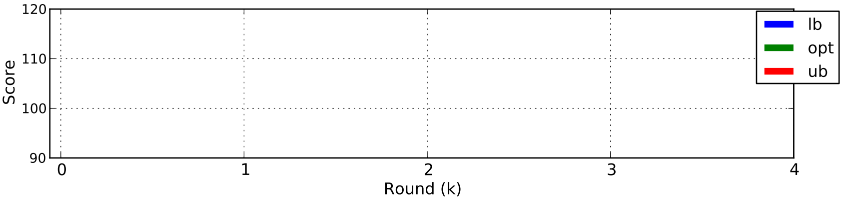
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \dots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

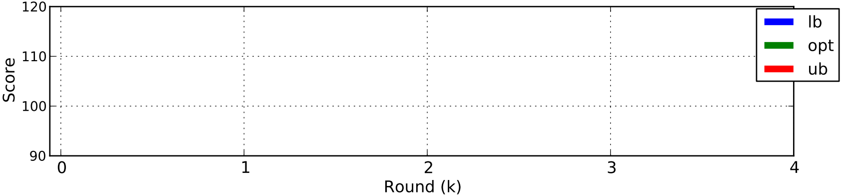
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

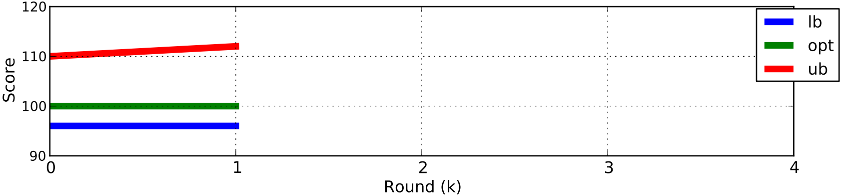
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

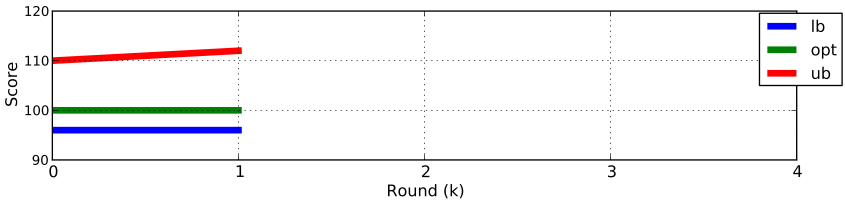
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

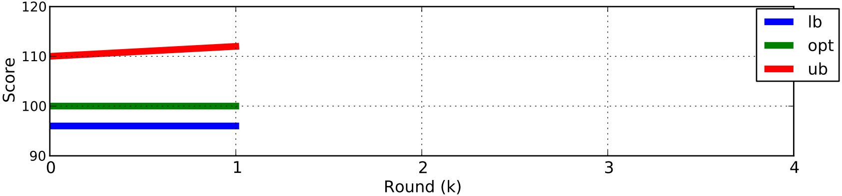
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\text{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

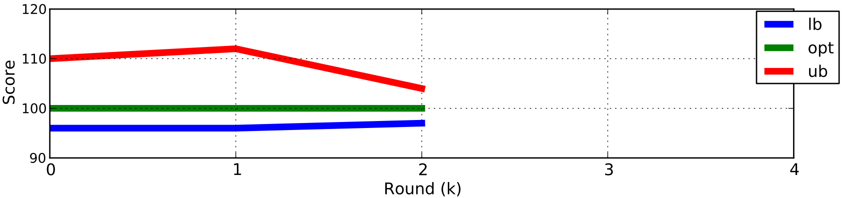
$\text{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\text{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \text{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\text{lb}^{(k)}, \text{cert} \leftarrow \text{BeamSearch}(\theta', \tau', \text{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\text{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

  $\lambda^{(0)} \leftarrow 0$

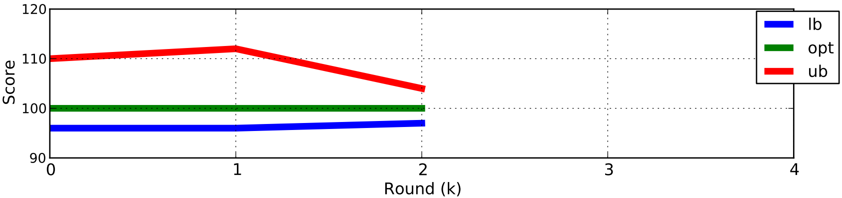
  $\mathrm{lb}^{(0)} \leftarrow -\infty$

  for k in $1 \ldots K$ **do**

    $\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

    $\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

    if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

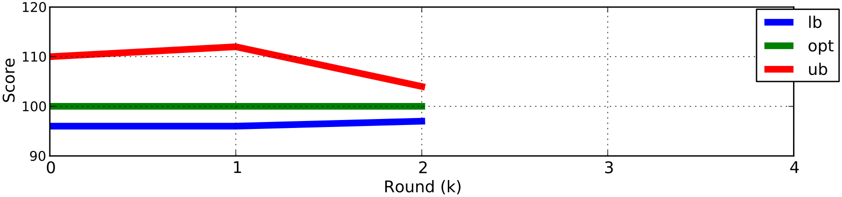
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

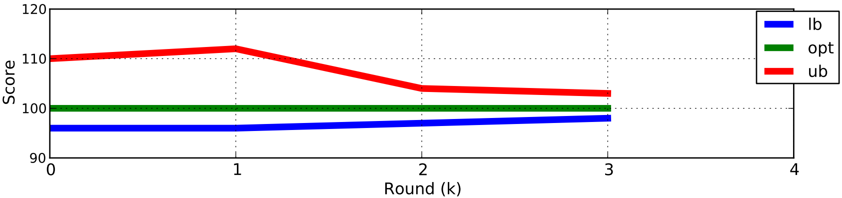
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

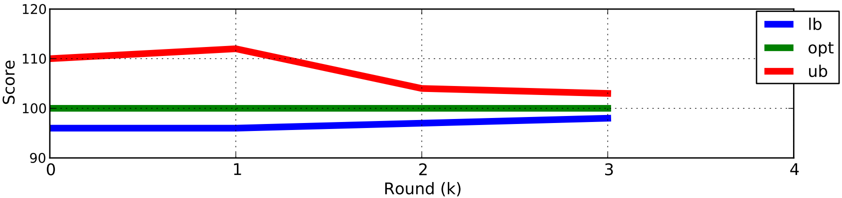
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \dots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

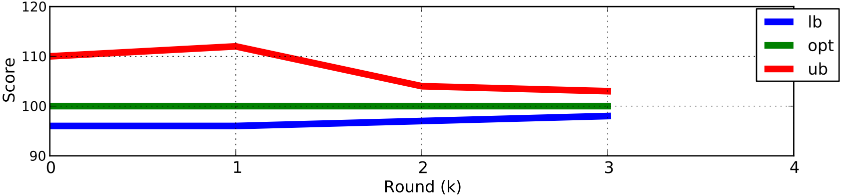
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

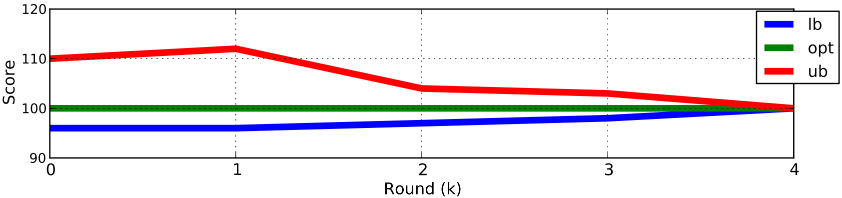
$\mathrm{lb}^{(0)} \leftarrow -\infty$

for k in $1 \ldots K$ **do**

$\mathrm{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\mathrm{lb}^{(k)}, \mathrm{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \mathrm{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\mathrm{lb}^{(k)}$



procedure $\textsc{OptBeam}(\alpha, \beta)$

$\lambda^{(0)} \leftarrow 0$

$\text{lb}^{(0)} \leftarrow -\infty$

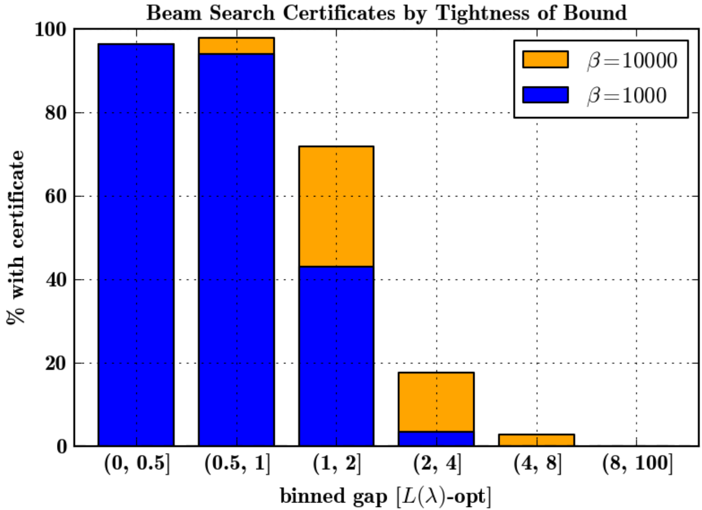
for k in $1 \ldots K$ **do**

$\text{ub}^{(k)}, \lambda^{(k)}, \theta', \tau' \leftarrow \textsc{LRSubgradient}(\alpha_k, \lambda^{(k-1)})$

$\text{lb}^{(k)}, \text{cert} \leftarrow \textsc{BeamSearch}(\theta', \tau', \text{lb}^{(k-1)}, \beta_k)$

if cert **then return** $\text{lb}^{(k)}$

Preview Results: When is Beam Search Optimal



Overview

1. translation as constrained graph search
2. certificate properties of beam search
3. relaxation methods for bounding
4. experimental results

Data Sets

Phrase-Based Translation

- 1,824 sentences German-to-English sentences from Europarl (Koehn, 2005)
- trigram language model and distortion penalties
- full graph structure from Chang and Collins (2011)

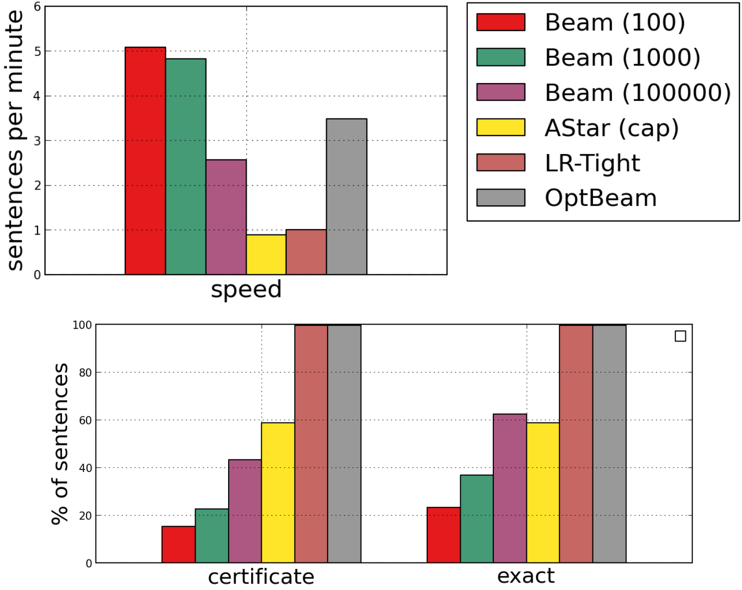
Syntax-Based Translation

- 691 Chinese-to-English sentences from Huang and Mi (2010)
- tree-to-string translation model and trigram language model
- full hypergraph structure from Rush and Collins (2011)

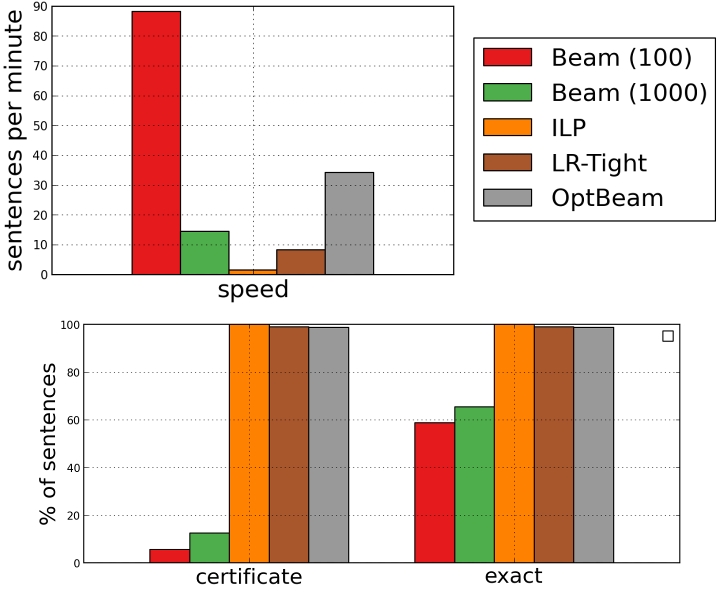
Baselines: Provable Guarantees

- Beam; a beam search decoder based on original weights.
- AStar; A^* search using original weights, θ and τ.
- ILP; a general-purpose integer linear programming solver.
- LR-Tight; LR with incremental constraints.
- OptBeam; this work.

Phrase-Based Optimal Methods



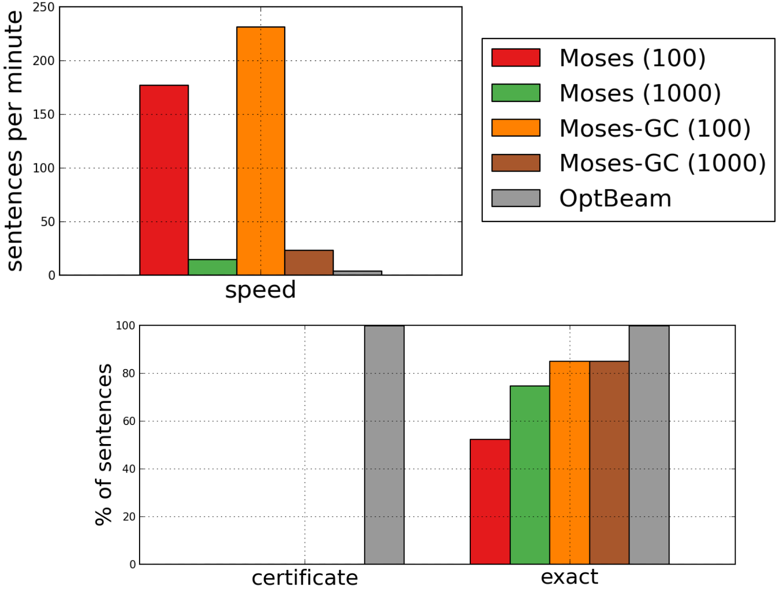
Syntax-Based Optimal Methods



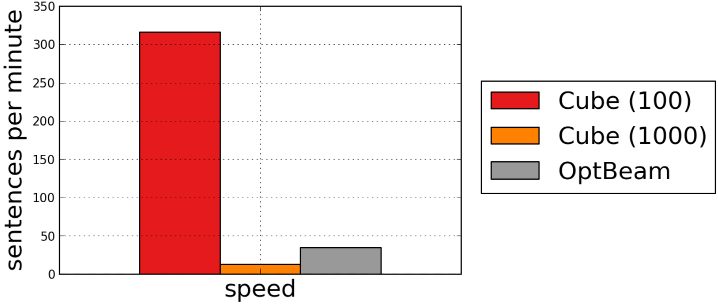
Baselines: Approximate Methods

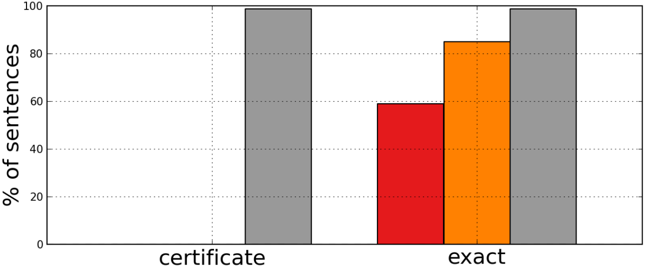
- MOSES-GC; the standard Moses beam search decoder.
- MOSES; Moses without gap constraints (see Chang and Collins (2011)).
- CUBEPRUNING; standard syntax-based decoding algorithm.

Phrase-Based Approximate Methods



Syntax-Based Approximate Methods





Conclusion

summary:

- reviewed conditions for exact beam search for translation.
- used Lagrangian relaxation to improve upper bounds.
- empirically method is effective at finding optimal solutions.

future work:

- training full system using exact decoding.
- bounding technique applied to other approximate algorithms.
- building a toolkit for constrained dynamic programming.

thank you