A Constrained Viterbi Relaxation for Bidirectional Word Alignment

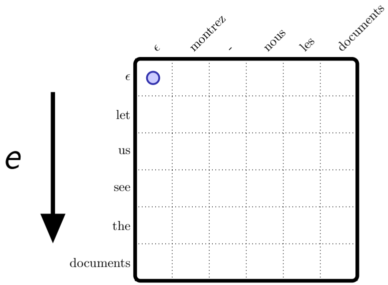
Yin-Wen Chang, Alexander Rush,
John DeNero and Michael Collins
ACL 2014

June 25, 2014

HMM Word Alignment Model

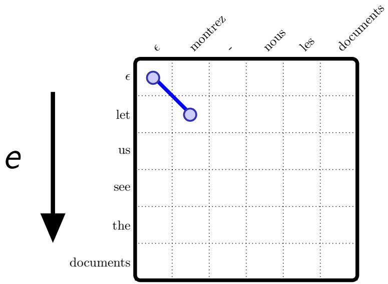
f:	montrez	-	nous	les	documents
j:	1	2	3	4	5

i:	1	2	3	4	5
e:	let	us	see	the	documents

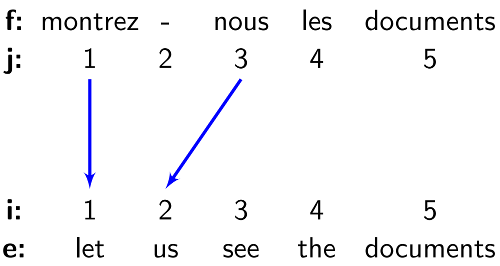


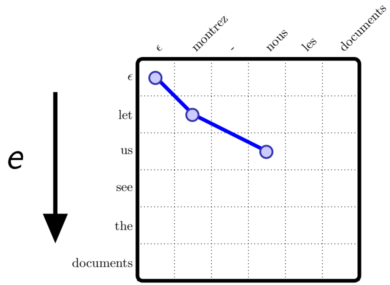
HMM Word Alignment Model

f:	montrez	-	nous	les	documents
j:	1	2	3	4	5
i:	1	2	3	4	5
e:	let	us	see	the	documents

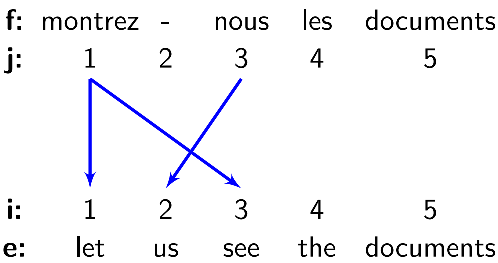


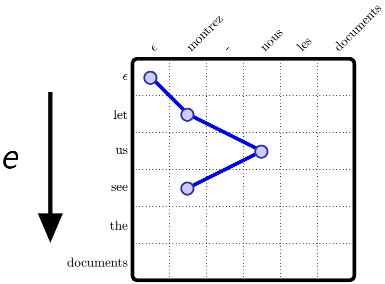
HMM Word Alignment Model



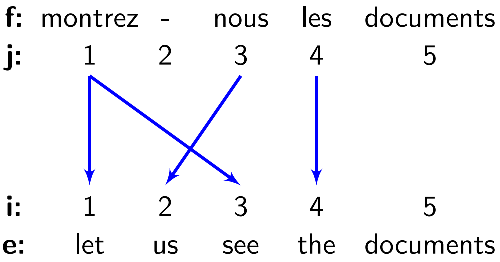


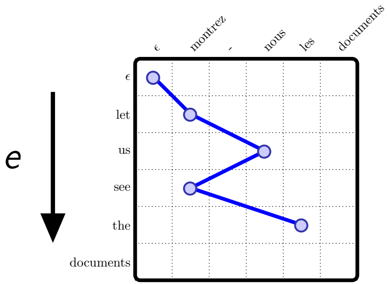
HMM Word Alignment Model



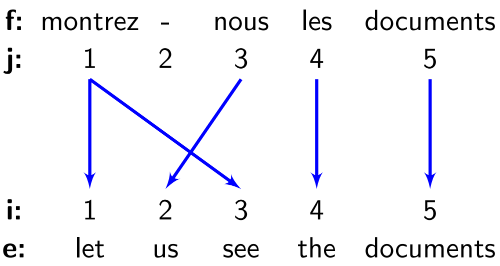


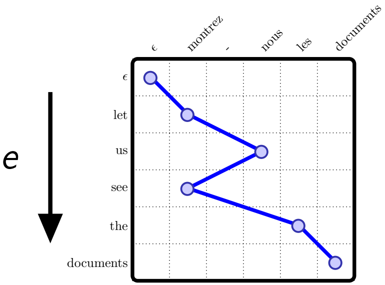
HMM Word Alignment Model



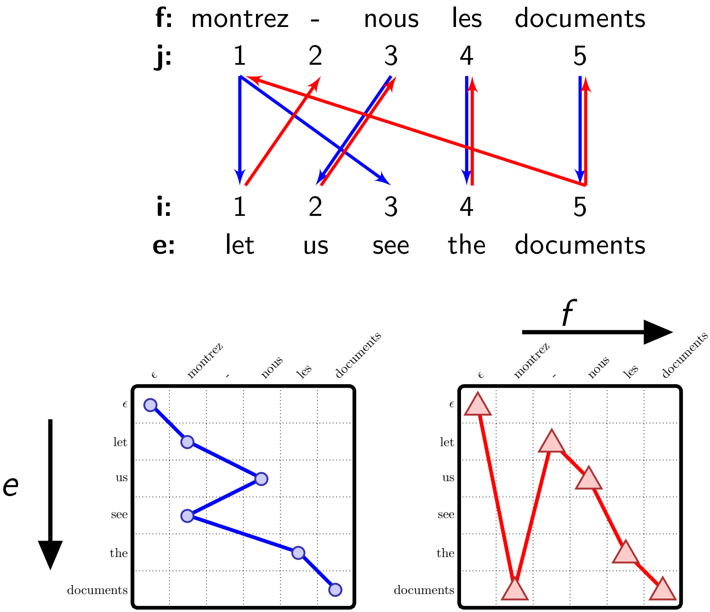


HMM Word Alignment Model

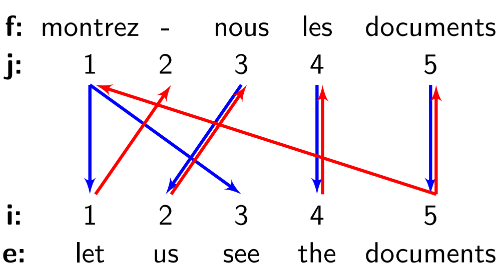


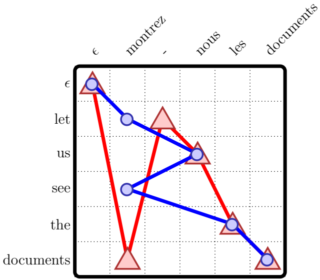


HMM Word Alignment Model



HMM Word Alignment Model





This Work: Bidirectional Alignment

- Most bidirectional formulations are NP-hard to solve.
- Previous attempt used dual decomposition and achieved 6% exact solutions (DeNero and Macherey, 2011).
- **Goal:** increase the number of exact solutions

Contributions

- A new relaxation for decoding the bidirectional model, solvable with a variant of Viterbi algorithm.
- Lagrangian relaxation to enforce the relaxed constraints.
- General techniques for adding constraints and applying pruning.

Outline

Word Alignment: $\mathbf{f} \to \mathbf{e}$

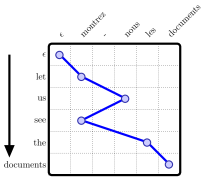
$$f(x;\theta) = \sum_{i,j,j'} \theta(j',i,j)x(j',i,j)$$

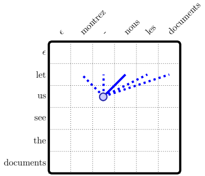
- boundary: $x(0,0) = 1$
- backward consistency:

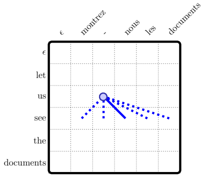
$$x(i,j) = \sum_{j'=0}^{J} x(j',i,j)$$

- forward consistency:

$$x(i,j) = \sum_{j'=0}^{J} x(j,i+1,j')$$

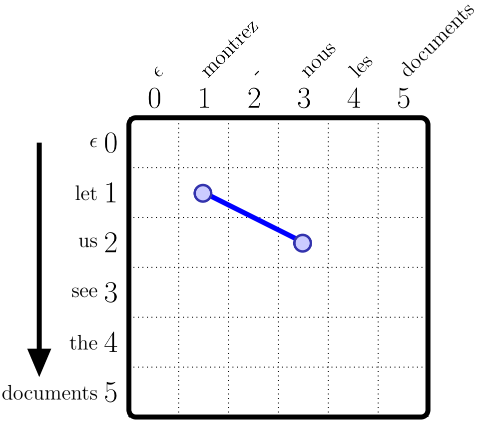






Word Alignment Example: $\mathbf{f} \rightarrow \mathbf{e}$

$$x(1,2,3) = 1$$
$$\theta(1,2,3) = \log(P(\texttt{us}|\texttt{nous})) + \log(P(3|1))$$



Word Alignment: $\mathbf{e}\rightarrow\mathbf{f}$

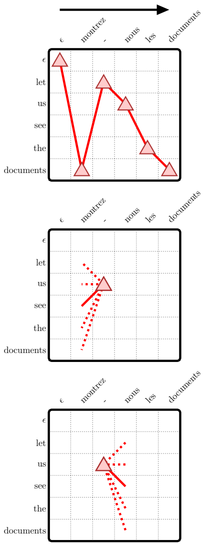
$$g(y;\omega) = \sum_{j,i,i'} \omega(i',i,j) y(i',i,j)$$

- boundary: $y(0,0) = 1$
- backward consistency:

$$y(i,j) = \sum_{i'=0}^{I} y(i',i,j)$$

- forward consistency:

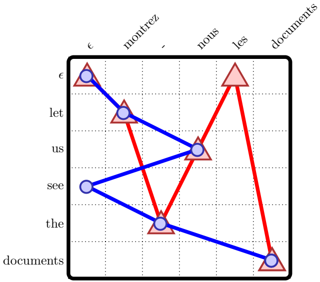
$$y(i,j) = \sum_{i'=0}^{I} y(i,i',j+1)$$



Bidirectional Alignment with Full Agreement

Goal:

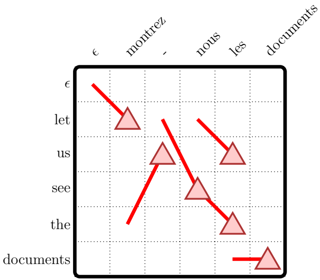
$$x^*, y^* = \arg\max_{x,y} f(x) + g(y) \text{ s.t.}$$
$$x(i,j) = y(i,j) \quad \forall\, i,j \neq 0$$



Outline

The Relaxed Problem

- $\mathcal{Y}$: set of the **e**$\rightarrow$**f** alignment
- $\mathcal{Y}'$: set of the **e**$\rightarrow$**f** alignment without the forward constraints
- $\mathcal{Y} \subset \mathcal{Y}'$



Lagrangian Relaxation

Goal:

$$x^*, y^* = \arg\max_{x \in \mathcal{X}, y \in \mathcal{Y}} f(x) + g(y) \text{ s.t.}$$
$$x(i,j) = y(i,j) \quad \forall\, i,j \neq 0$$

Lagrangian dual:

$$L(\lambda) = \arg\max_{\substack{x \in \mathcal{X}, y \in \mathcal{Y}', \\ x(i,j) = y(i,j)}} f(x) + g'(y; \omega, \lambda)$$

where

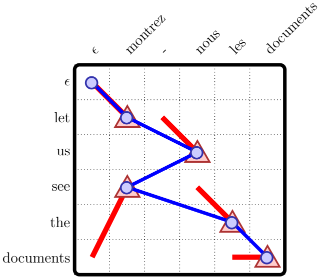
$$g'(y; \omega, \lambda) = g(y; \omega, \lambda) - \sum_{i,j} \underbrace{\lambda(i,j)}_{\text{Lagrange multipliers}} \underbrace{\left(y(i,j) - \sum_{i'} y(i,i',j+1) \right)}_{\text{forward constraints for } y}$$

Viterbi-style Algorithm for computing $L(\lambda)$

$$\max_{\substack{x \in \mathcal{X}, y \in \mathcal{Y}', \\ x(i,j) = y(i,j)}} f(x) + g'(y; \omega, \lambda)$$

$$= \max_{\substack{x \in \mathcal{X}, y \in \mathcal{Y}', \\ x(i,j) = y(i,j)}} f(x) + \sum_{i,j} y(i,j) \max_{i'} \omega'(i', i, j)$$

where $\omega'(i', i, j) = \omega(i', i, j) - \lambda(i, j) + \lambda(i', j-1)$



- Compute the score for each $y(i, j)$
- Standard Viterbi update for computing $x(i, j)$, adding in the score of $y(i, j)$

The Lagrangian Relaxation Algorithm

- Lagrangian dual is the **upper bound**:

$$L(\lambda) \geq f(x^*) + g(y^*)$$

- Find tightest upper bound:

$$\min_{\lambda} L(\lambda)$$

- Minimize by **subgradient**:
  1. Set (x, y) to the arg max of $L(\lambda)$.

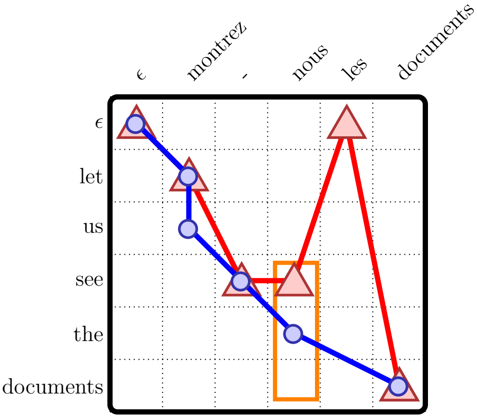
     If (x, y) satisfies the forward constraint, return (x, y)
  2. Else, update $\lambda(i, j)$ for all i, j,

$$\lambda(i,j) \leftarrow \lambda(i,j) - \eta_t \big(y(i,j) - \sum_{i'=0}^{I} y(i,i',j+1)\big)$$

- **Certificate of optimality** upon convergence

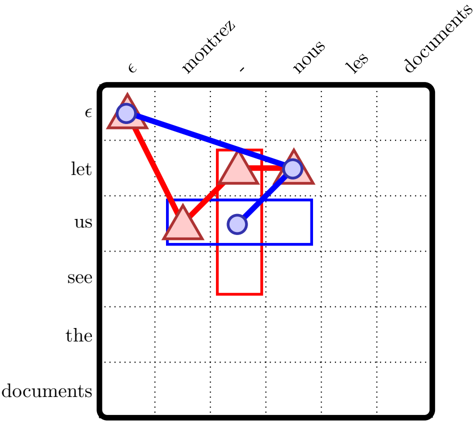
Extension: Adjacent Agreement

- A model that allows adjacent matches
- $x(4,3) = 1$ because $y(3,3) = 1$



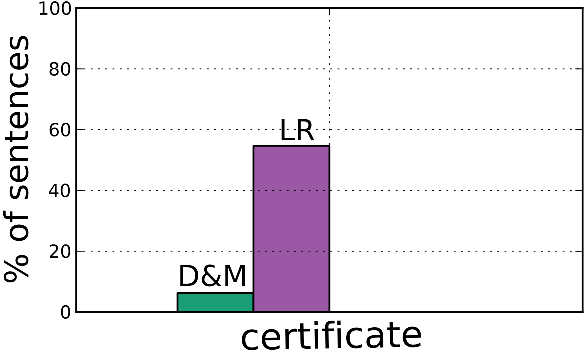
Extension: Adjacent Agreement

- A modified Viterbi algorithm with the same complexity



Preview Results: Lagrangian Relaxation

- Lagrangian relaxation only guarantee to solve the linear programming relaxation
- 54.7 % exact solutions



Outline

Strategy: Adding Constraints

Upper bound:

$$L(\lambda) \geq f(x^*) + g(y^*)$$

Current gap:

$$L(\lambda) - (f(x^*) + g(y^*))$$

Tightening:

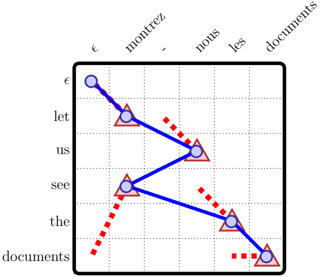
finding a different dual with better gap

$$\text{Find } L'(\lambda) \leq L(\lambda)$$

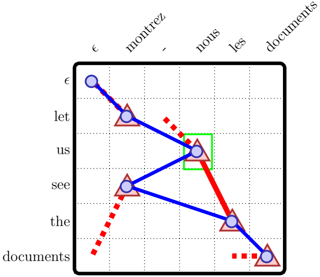
Strategy: Adding Constraints

- Re-introduce a relaxed forward constraint on link $y(i, j)$
- Adding the most often violated constraints
- For example, adding constraint for link $y(2, 3)$:

Feasible under $L(\lambda)$
Not feasible under $L'(\lambda)$



Feasible under both
$L(\lambda)$ and $L'(\lambda)$



Outline

Bidirectional Alignment

- HMM Word Alignment
- Lagrangian Relaxation

Tightening

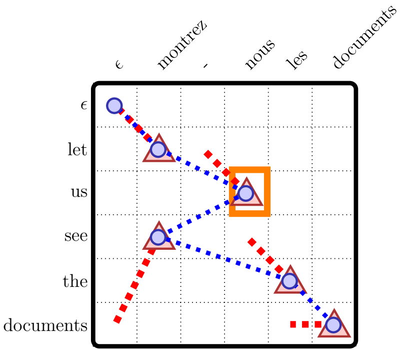
- Adding Constraints
- Pruning

Results

Relaxed Max-marginal

- Improve efficiency while keeping optimality guarantee
- M: Relaxed max-marginal values

  The highest dual value of all alignments using the link $x(i, j)$
- Example: $M(2, 3; \lambda)$



Exact Coarse-to-Fine Pruning

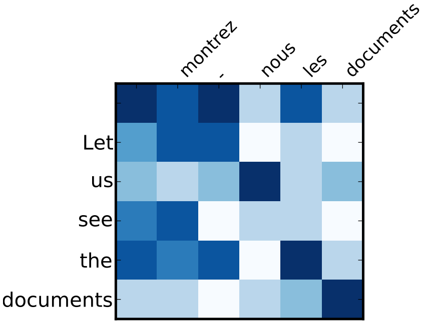
- We can safely remove an alignment link $x(i,j)$ if

$$M(i,j;\lambda) < \text{lb}$$

- **Lower bound:**

$$f(x) + g(y) \leq f(x^*) + g(y^*)$$

for some x, y that is valid



Exact Coarse-to-Fine Pruning

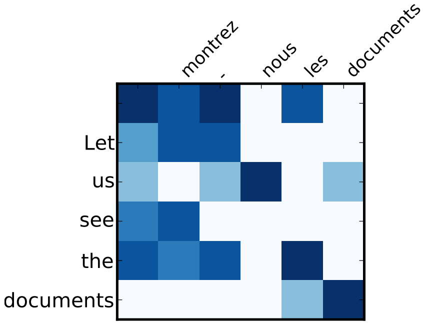
- We can safely remove an alignment link $x(i, j)$ if

$$M(i, j; \lambda) < \text{lb}$$

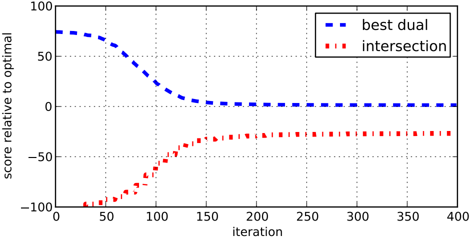
- **Lower bound:**

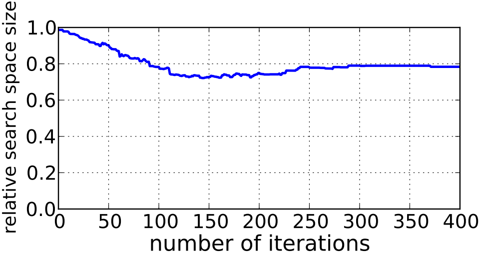
$$f(x) + g(y) \leq f(x^*) + g(y^*)$$

for some x, y that is valid



Preview Results: Pruning

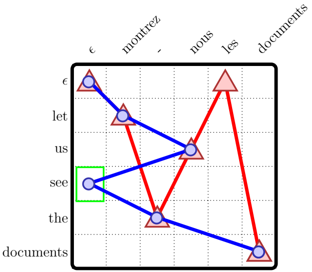




Finding Lower Bounds

A greedy heuristic algorithm:

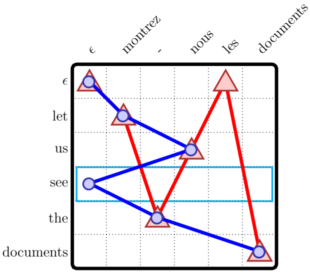
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



Finding Lower Bounds

A greedy heuristic algorithm:

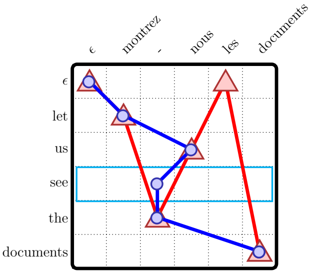
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



Finding Lower Bounds

A greedy heuristic algorithm:

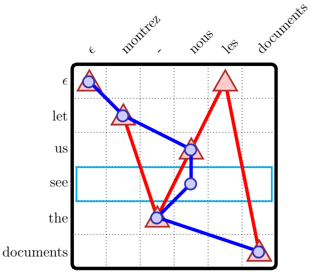
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



Finding Lower Bounds

A greedy heuristic algorithm:

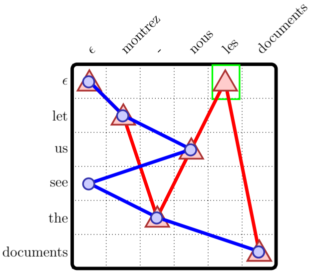
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



Finding Lower Bounds

A greedy heuristic algorithm:

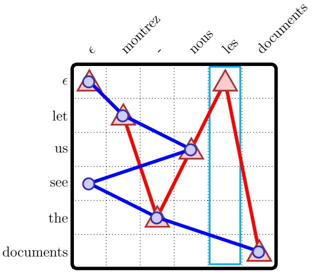
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



Finding Lower Bounds

A greedy heuristic algorithm:

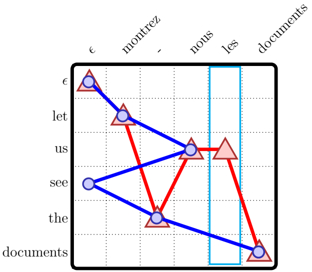
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



Finding Lower Bounds

A greedy heuristic algorithm:

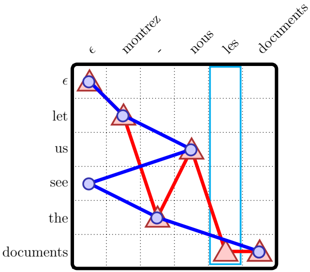
- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



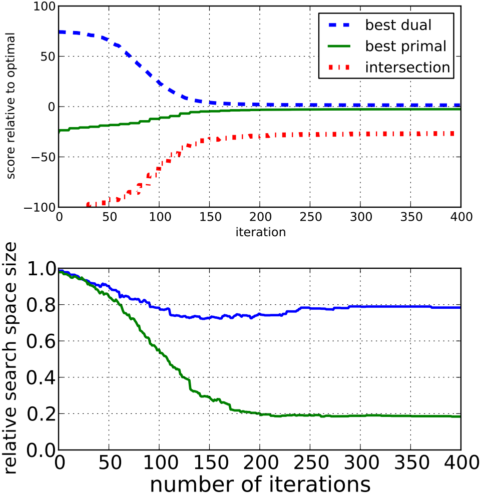
Finding Lower Bounds

A greedy heuristic algorithm:

- Repeat until
  - there exists no null-aligned word, or
  - there is no score increase.



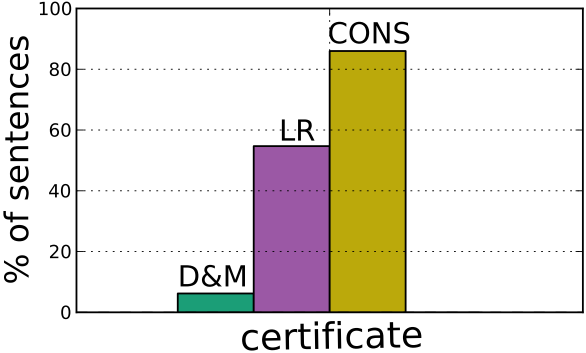
Results: Coarse-to-Fine Pruning with Good Lower Bound



Experiments

- Trained on 6.2 million words of Chinese-English FBIS data
- Evaluated on 150 sentence pairs of NIST 2002 data
- Identical to DeNero and Macherey (2011)

Results: with Adding Constraints and Pruning

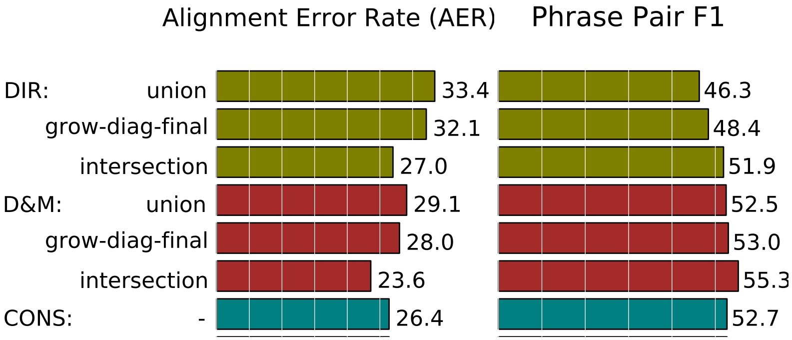


Results: Speed and Optimal Solutions

	time	certificate (%)
ILP	924.24	100.0
LR	6.33	54.7
CONS	21.08	86.0
D&M	-	6.2

- ILP: Integer linear programming
- LR: Our Lagrangian relaxation algorithm
- CONS: LR with adding constraints
- D&M: Dual decomposition algorithm by DeNero and Macherey (2011)

Results: Accuracy



- DIR: directional alignments
- When the algorithm does not converge:
  - D&M uses combination procedures
  - CONS uses the highest scoring feasible solution

Conclusion

- A Lagrangian relaxation algorithm for bidirectional alignment
- Adding constraints incrementally
- Coarse-to-fine pruning
- Convergence on 86% sentences, compared to 6% reported by DeNero and Macherey (2011)
- **Future work:** apply these techniques to a more flexible model with a wider range of directional matches

Conclusion

- A Lagrangian relaxation algorithm for bidirectional alignment
- Adding constraints incrementally
- Coarse-to-fine pruning
- Convergence on 86% sentences, compared to 6% reported by DeNero and Macherey (2011)
- **Future work:** apply these techniques to a more flexible model with a wider range of directional matches

Thank you!