An Exact Dual Decomposition Algorithm for Shallow Semantic Parsing with Constraints

Dipanjan Das
CMU → Google

André F.T. Martins
CMU/IST → Priberam

Noah A. Smith
CMU

*SEM 2012
8 June, 2012
Frame-Semantic Parsing

Given a sentence, its syntactic parse and a predicate:

I want to go to Montreal on Sunday
I want to go to Montreal on Sunday
Frame-Semantic Parsing

I want to go to Montreal on Sunday

Frame: Encodes an event or scenario

TRAVEL
I want to go to Montreal on Sunday.
Role: a participant of the frame

Traveler

T R A V E L

Traveler

I want to go to Montreal on Sunday

Goal

Time

P V ADP V ADP N ADP N
Frame-Semantic Parsing

Multiple frame-semantic structures per sentence:

Traveler

I want to go to Montreal on Sunday

Desiring

Event

Goal

Time
Frame-Semantic Parsing

**Previous work:**
Das, Schneider, Chen and Smith (NAACL 2010)
Das and Smith (ACL 2011)
Das and Smith (NAACL 2012)
Frame-Semantic Parsing

Frame identification

Argument identification
Frame-Semantic Parsing

Frame identification

Argument identification (SRL)

**focus of this talk**
Structure of the FrameNet Lexicon
Structure of the FrameNet Lexicon

PLACING
Agent
Cause
Goal
Theme
Area
Time
Structure of the FrameNet Lexicon

**P**LACING
- Agent
- Cause
- Goal
- Theme

**frame**

- Area
- Time
Structure of the FrameNet Lexicon

- **PLACING**
- Agent
- Cause
- Goal
- Theme
- Area
- Time

Roles

Frame
Structure of the FrameNet Lexicon

- **Core roles**: Agent, Cause, Goal, Theme
- **Non-core roles**: Area, Time

The frame is represented by the PLACING category.
Structure of the FrameNet Lexicon

- **Core roles:** Agent, Cause, Goal, Theme
- **Non-core roles:** Area, Time

Frame excludes relationship
Structure of the FrameNet Lexicon

- Core roles: Agent, Cause, Goal, Theme
- Non-core roles: Area, Time
- Frame excludes relationship
Structure of Lexicon and Data

**TRANSITIVE_ACTION**
- Agent
- Cause
- Patient
- Event
- Place
- Time

**PLACING**
- Agent
- Cause
- Goal
- Theme
- Area
- Time

**DISPERSAL**
- Agent
- Cause
- Individuals
- Distance
- Time

**INSTALLING**
- Agent
- Component
- Fixed_location
- Area
- Time

**STORING**
- Agent
- Location
- Theme
- Area
- Time

**STORE**
- Possessor
- Resource
- Supply
- Descriptor
Structure of Lexicon and Data

- **TRANSITIVE_ACTION**
  - Agent
  - Cause
  - Patient
  - Event
  - Place
  - Time

- **PLACING**
  - Agent
  - Cause
  - Goal
  - Theme
  - Area
  - Time

- **DISPERSAL**
  - Agent
  - Cause
  - Individuals
  - Distance
  - Time

- **INSTALLING**
  - Agent
  - Component
  - Fixed_location
  - Area
  - Time

- **STORING**
  - Agent
  - Location
  - Theme
  - Area
  - Time

- **STORE**
  - Possessor
  - Resource
  - Supply
  - Descriptor

Inheritance and used by relationships are indicated by arrows connecting the concepts.
Frame-Semantic Parsing

Frame identification

Argument identification (SRL)

focus of this talk
Bengal’s massive stock of food was reduced to nothing
Bengal’s massive stock of food was reduced to nothing.
Bengal’s massive *stock* of food was reduced to nothing.
Bengal’s massive stock of food was reduced to nothing
Argument Identification

**STORE**

- **Possessor**: Bengal’s
- **Resource**: massive stock of food
- **Descriptor**: massive
- **Use**: Bengal’s massive stock of food
- **Supply**: Ø
Ideal mapping!
Argument Identification

**STORE**

**Possessor**

**Resource**

**Descriptor**

**Use**

**Supply**

**STORE**

stock

Possessor

Resource

Descriptor

Use

Supply

Bengal’s massive stock of food

Bengal’s massive stock of food

massive

massive

Ø

massive stock of food
Argument Identification

Violates overlap constraints

Bengal’s massive stock of food

Bengal’s stock of food

Bengal’s massive stock of food

massive stock

massive

stock

Possessor

Resource

Descriptor

Use

Supply

STORE

Argument Identification

stock

Violates overlap constraints
Argument Identification

Other types of structural constraints

Mutual exclusion constraint

PLACING
Agent
Cause
Goal
Theme
Area
Time

Other types of structural constraints

The waiter placed food on the table.

In Kabul, hauling water put food on the table.

Mutual exclusion constraint


The waiter placed food on the table.

Agent

In Kabul, hauling water put food on the table.

Cause
If an agent places something, there cannot be a cause role in the sentence.
Argument Identification

Other types of structural constraints

- Similarity
  - Dimension
  - Differentiating_fact
  - Entity_1
  - Entity_2
  - Degree

Requires constraint:

- difference.N
- resemble.V
- unlike.A
- vary.V
Argument Identification

Other types of structural constraints

A mulberry resembles a loganberry.

First entity

Second entity

Requires constraint

SIMILARITY
Dimension
Differentiating_fact
Entity_1
Entity_2
Degree

Argument Identification

Other types of structural constraints

A mulberry resembles.

*°A mulberry° resembles.
Argument Identification

**Possessor**: Bengal’s massive stock of food

**Resource**: food

**Descriptor**: massive

**Use**: Bengal’s massive stock of food

**Supply**: Ø

**Store**: stock

A constrained optimization problem
Argument Identification

STORE

stock

Possessor

Resource

Descriptor

Use

Supply

Bengal’s massive stock of food

score(role ↔ span)

Bengal

massive stock

food

massive

Bengal’s massive stock of food

∅
Argument Identification

**Stock**

**Possessor**

**Resource**

**Descriptor**

**Use**

**Supply**

\[
\text{score}(\text{role} \leftrightarrow \text{span}) = w \cdot g(\text{role, span, frame})
\]
Argument Identification

A constrained optimization problem

\[ \mathcal{Z}_{\text{role} \leftrightarrow \text{span}} \]
Argument Identification

A constrained optimization problem

$Z_{\text{role} \leftrightarrow \text{span}}$

a binary variable for each role, span tuple
Argument Identification

A constrained optimization problem

\[ z_{\text{role} \leftrightarrow \text{span}} \]

\[ z = \langle z_{\text{role} \leftrightarrow \text{span}} \rangle \]

a binary vector for all role, span tuples
Argument Identification

A constrained optimization problem

maximize \( \sum_{\text{roles,spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score(\text{role} \leftrightarrow \text{span})} \)

w.r.t. \( z \)
A constrained optimization problem

$$\text{maximize} \sum_{\text{roles,spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score}(\text{role} \leftrightarrow \text{span})$$

w.r.t. $z$

s.t. $\forall$ roles, $\sum_{\text{spans}} z_{\text{role} \leftrightarrow \text{span}} = 1$

Uniqueness
Argument Identification

A constrained optimization problem

\[
\text{maximize} \quad \sum_{\text{roles, spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score}(\text{role} \leftrightarrow \text{span})
\]

w.r.t. \( z \)

s.t. \( \forall \) roles, \( \sum_{\text{spans}} z_{\text{role} \leftrightarrow \text{span}} = 1 \)

\( \forall \) sentence positions, \( \sum_{\text{span covers position}} \sum_{\text{roles}} z_{\text{role} \leftrightarrow \text{span}} \leq 1 \)

Prevents overlap
A constrained optimization problem

\[
\text{maximize } \sum_{\text{roles,spans}} z_{\text{role}\leftrightarrow\text{span}} \cdot \text{score}(\text{role }\leftrightarrow\text{span})
\]

w.r.t. \( z \)

s.t. \( \forall \text{ roles, } \sum_{\text{spans}} z_{\text{role}\leftrightarrow\text{span}} = 1 \)

\( \forall \text{ sentence positions, } \sum_{\text{span covers position}} \sum_{\text{roles}} z_{\text{role}\leftrightarrow\text{span}} \leq 1 \)

\( \forall \text{ role}_1 \text{ excludes } \text{role}_2, z_{\text{role}_1\leftrightarrow\emptyset} + z_{\text{role}_2\leftrightarrow\emptyset} \geq 1 \)

Exclusion constraints
Argument Identification

A constrained optimization problem

\[
\text{maximize} \quad \sum_{\text{roles,spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score}(\text{role} \leftrightarrow \text{span})
\]

w.r.t. \( z \)

s.t \( \forall \text{ roles, } \sum_{\text{spans}} z_{\text{role} \leftrightarrow \text{span}} = 1 \)

\( \forall \text{ sentence positions, } \sum_{\text{span covers position}} \sum_{\text{roles}} z_{\text{role} \leftrightarrow \text{span}} \leq 1 \)

\( \forall \text{ role}_1 \text{ excludes } \text{ role}_2, z_{\text{role}_1 \leftrightarrow \varnothing} + z_{\text{role}_2 \leftrightarrow \varnothing} \geq 1 \)

\( \forall \text{ role}_1 \text{ requires } \text{ role}_2, z_{\text{role}_1 \leftrightarrow \varnothing} - z_{\text{role}_2 \leftrightarrow \varnothing} = 0 \)

Requires constraints
Argument Identification

An integer linear program (ILP)

\[
\text{maximize } \sum_{\text{roles, spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score}(\text{role} \leftrightarrow \text{span}) \\
\text{w.r.t. } z \\
\text{s.t } \forall \text{ roles}, \sum_{\text{spans}} z_{\text{role} \leftrightarrow \text{span}} = 1 \\
\forall \text{ sentence positions}, \sum_{\text{span covers position}} \sum_{\text{roles}} z_{\text{role} \leftrightarrow \text{span}} \leq 1 \\
\forall \text{ role}_1 \text{ excludes role}_2, z_{\text{role}_1 \leftrightarrow \emptyset} + z_{\text{role}_2 \leftrightarrow \emptyset} \geq 1 \\
\forall \text{ role}_1 \text{ requires role}_2, z_{\text{role}_1 \leftrightarrow \emptyset} - z_{\text{role}_2 \leftrightarrow \emptyset} = 0
\]
Argument Identification

An integer linear program (ILP)

maximize \( \sum_{\text{roles, spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score}(\text{role} \leftrightarrow \text{span}) \)

w.r.t. \( z \)

s.t. \( \forall \text{ roles}, \sum_{\text{spans}} z_{\text{role} \leftrightarrow \text{span}} = 1 \)

\( \forall \text{ sentence positions}, \sum_{\text{span}} \sum_{\text{roes}} z_{\text{role} \leftrightarrow \text{span}} \leq 1 \)

\( \forall \text{ role}_1 \text{ excludes role}_2, z_{\text{role}_1 \leftrightarrow \emptyset} + z_{\text{role}_2 \leftrightarrow \emptyset} \geq 1 \)

\( \forall \text{ role}_1 \text{ requires role}_2, z_{\text{role}_1 \leftrightarrow \emptyset} - z_{\text{role}_2 \leftrightarrow \emptyset} = 0 \)

Similar to Punyakanok et al. (2004)
An integer linear program (ILP)

\[
\text{maximize } \sum_{\text{roles, spans}} z_{\text{role}\leftrightarrow\text{span}} \cdot \text{score(role } \leftrightarrow \text{ span)}
\]

w.r.t. \(z\)

s.t \(\forall \) roles, \(\sum_{\text{spans}} z_{\text{role}\leftrightarrow\text{span}} = 1\)

\(\forall\) sentence positions, \(\sum_{\text{span covers position}} \sum_{\text{roles}} z_{\text{role}\leftrightarrow\text{span}} \leq 1\)

\(\forall\) role\(_1\) excludes role\(_2\), \(z_{\text{role}_1\leftrightarrow\emptyset} + z_{\text{role}_2\leftrightarrow\emptyset} \geq 1\)

\(\forall\) role\(_1\) requires role\(_2\), \(z_{\text{role}_1\leftrightarrow\emptyset} - z_{\text{role}_2\leftrightarrow\emptyset} = 0\)

Often, very slow solutions
Argument Identification

An integer linear program (ILP)

maximize \[ \sum_{\text{roles,spans}} z_{\text{role} \leftrightarrow \text{span}} \cdot \text{score}(\text{role} \leftrightarrow \text{span}) \]

w.r.t. \( z \)

s.t \( \forall \text{ roles}, \sum_{\text{spans}} z_{\text{role} \leftrightarrow \text{span}} = 1 \)

\( \forall \text{ sentence positions}, \sum_{\text{span covers position}} \sum_{\text{roles}} z_{\text{role} \leftrightarrow \text{span}} \leq 1 \)

\( \forall \text{ role}_1 \text{ excludes role}_2, z_{\text{role}_1 \leftrightarrow \emptyset} + z_{\text{role}_2 \leftrightarrow \emptyset} \geq 1 \)

\( \forall \text{ role}_1 \text{ requires role}_2, z_{\text{role}_1 \leftrightarrow \emptyset} - z_{\text{role}_2 \leftrightarrow \emptyset} = 0 \)

Fast ILP solvers proprietary
Argument Identification

An alternate approach

Alternating Directions Dual Decomposition or $\text{AD}^3$

(Martins et al., ICML 2011, EMNLP 2011)
Argument Identification

An alternate approach

Alternating Directions Dual Decomposition or \( \text{AD}^3 \)

(Martins et al., ICML 2011, EMNLP 2011)

Different from subgradient-based dual decomposition

(Rush et al., 2010)
Argument Identification

An alternate approach

basic part:  role $\leftrightarrow$ span

entire space:  all role $\leftrightarrow$ span tuples
Argument Identification

An alternate approach

basic part: role ↔ span

entire space: all role ↔ span tuples

Break down the problem into many small components

e.g.
– find the best span for a role
– for a sentence position, find the best role ↔ span tuple
– for a pair of mutually exclusive roles, find the best role ↔ span tuple
Argument Identification

An alternate approach

basic part: role $\leftrightarrow$ span

entire space: all role $\leftrightarrow$ span tuples

Break down the problem into many small components

e.g.

-- find the best span for a role
-- for a sentence position, find the best role $\leftrightarrow$ span tuple
-- for a pair of mutually exclusive roles, find the best role $\leftrightarrow$ span tuple

impose agreement between components
Argument Identification

An alternate approach

For each component, a binary vector:

\[\mathbf{z}^{\text{comp.}} = \langle \mathbf{z}^{\text{comp.}}_{\text{role} \leftrightarrow \text{span}} \rangle\]
Argument Identification

An alternate approach

For each component, a binary vector:

$$z^{\text{comp.}} = \langle z^{\text{comp.}}_{\text{role} \leftrightarrow \text{span}} \rangle$$

Total score assigned to $z^{\text{comp.}}$:  

$$\text{score}(z^{\text{comp.}})$$
Argument Identification

An alternate approach

For each component, a binary vector:

\[ \mathbf{z}_{\text{comp.}} = \langle \mathbf{z}_{\text{role} \leftrightarrow \text{span}} \rangle \]

Total score assigned to \( \mathbf{z}_{\text{comp.}} \):

\[ \text{score}(\mathbf{z}_{\text{comp.}}) \]

uses

\[ \text{score}(\text{role} \leftrightarrow \text{span}) \]
Argument Identification

An alternate approach

$$\max \sum_{\text{all components}} \text{score}(z^{comp.})$$

w.r.t. $z^{comp.} \in Z^{comp.}, \forall \text{components}$

s.t. $z^{\text{role}\leftrightarrow\text{span}} = u_{\text{role}\leftrightarrow\text{span}} \forall \text{ components, roles and spans}$
Argument Identification

An alternate approach

$$\max \sum_{\text{all components}} \text{score}(z^{\text{comp.}})$$

w.r.t. $z^{\text{comp.}} \in Z^{\text{comp.}}, \forall \text{components}$

s.t. $z^{\text{comp.}}_{\text{role\leftrightarrow span}} = u_{\text{role\leftrightarrow span}} \forall \text{ components, roles and spans}$

witness vector $u$

for consensus
Argument Identification

An alternate approach

\[
\text{Primal:} \quad \max \sum_{\text{all components}} \text{score}(z^{\text{comp.}})
\]

w.r.t. \(z^{\text{comp.}} \in Z^{\text{comp.}}, \forall \text{components}\)

s.t. \(z^{\text{comp.}}_{\text{role} \leftrightarrow \text{span}} = u_{\text{role} \leftrightarrow \text{span}} \forall \text{ components, roles and spans}\)
Argument Identification

An alternate approach

Primal':

\[
\max \sum \text{score}(z^{comp.}) \\
\text{all components}
\]

w.r.t. \( z^{comp.} \in \mathbb{Z}^{comp.}, \forall \text{components} \)

s.t. \( z_{role\leftrightarrow span}^{comp.} = u_{role\leftrightarrow span}^{role\leftrightarrow span} \forall \text{ components, roles and spans} \)

Integer constraints relaxed
Argument Identification

An alternate approach

**Primal':**

\[
\max \sum \text{score}(z^{\text{comp.}}) \quad \text{all components}
\]

w.r.t. \(z^{\text{comp.}} \in Z^{\text{comp.}}, \forall \text{components}\)

s.t. \(z^{\text{comp.}}_{\text{role} \leftrightarrow \text{span}} = u_{\text{role} \leftrightarrow \text{span}} \forall \text{ components, roles and spans}\)

An **augmented Lagrangian** function

\[
\sum \text{score}(z^{\text{comp.}}) + \lambda^{\text{comp.}} \cdot z^{\text{comp.}} - \lambda^{\text{comp.}} \cdot u - \frac{\rho}{2} \|z^{\text{comp.}} - u\|^2
\]
An alternate approach

An augmented Lagrangian function

\[ \sum_{\text{all components}} \text{score}(z^{\text{comp.}}) + \lambda^{\text{comp.}} \cdot z^{\text{comp.}} - \lambda^{\text{comp.}} \cdot u - \frac{\rho}{2} \| z^{\text{comp.}} - u \|^2 \]

Saddle point can be found using several decoupled worker problems for each component
An alternate approach

An **augmented Lagrangian** function

\[
\sum_{\text{all components}} \text{score}(z^{\text{comp.}}) + \lambda^{\text{comp.}} \cdot z^{\text{comp.}} - \lambda^{\text{comp.}} \cdot u
- \frac{\rho}{2} \| z^{\text{comp.}} - u \|_2^2
\]

Three types of iterative updates:

1. Lagrange multiplier updates \((\lambda)\)
2. Consensus variable updates \((u)\)
3. \(z\) updates
Argument Identification

An alternate approach

An **augmented Lagrangian** function

\[
\sum_{\text{all components}} \text{score}(z^{\text{comp.}}) + \lambda^{\text{comp.}} \cdot z^{\text{comp.}} - \lambda^{\text{comp.}} \cdot u - \frac{\rho}{2} \|z^{\text{comp.}} - u\|^2
\]

Three types of iterative updates:

1. Lagrange multiplier updates \((\lambda)\)
2. Consensus variable updates \((u)\)
3. \(z\) updates  \(\text{At decoupled workers}\)
e.g. for each role, we have a worker that imposes a XOR/uniqueness constraint

$$\text{minimize } \frac{1}{2} \sum_{\text{spans}} (\hat{z}_{\text{role} \leftrightarrow \text{span}}^{\text{comp.}} - a_{\text{role} \leftrightarrow \text{span}})^2$$

w.r.t. every $\hat{z}_{\text{role} \leftrightarrow \text{span}}^{\text{comp.}} \in [0, 1]$

s.t. $\sum_{\text{spans}} \hat{z}_{\text{role} \leftrightarrow \text{span}}^{\text{comp.}} = 1$

3. $\mathcal{Z}$ updates At decoupled workers
Argument Identification

e.g. for each role, we have a worker that imposes a XOR/uniqueness constraint

\[
\begin{align*}
\text{minimize } & \frac{1}{2} \sum_{\text{spans}} (\hat{z}_{\text{role}\leftrightarrow\text{span}}^{\text{comp.}} - a_{\text{role}\leftrightarrow\text{span}})^2 \\
\text{w.r.t. every } & \hat{z}_{\text{role}\leftrightarrow\text{span}}^{\text{comp.}} \in [0, 1] \\
\text{s.t. } & \sum_{\text{spans}} \hat{z}_{\text{role}\leftrightarrow\text{span}}^{\text{comp.}} = 1
\end{align*}
\]

Projection onto a simplex
a simple sort operation

3. \(\hat{z}\) updates

At decoupled workers
e.g. for each role, we have a worker that imposes a XOR/uniqueness constraint

$$\min \frac{1}{2} \sum_{\text{spans}} (\hat{z}_{\text{role} \leftrightarrow \text{span}}^{\text{comp.}} - a_{\text{role} \leftrightarrow \text{span}})^2$$

w.r.t. every $\hat{z}_{\text{role} \leftrightarrow \text{span}}^{\text{comp.}} \in [0, 1]$

s.t. $\sum_{\text{spans}} \hat{z}_{\text{role} \leftrightarrow \text{span}}^{\text{comp.}} = 1$

3. $\hat{z}$ updates

At decoupled workers

Challenge:
define fast, simple workers

Projection onto a simplex
a simple sort operation
The three other constraints map to variants of the sort operation.

3. $\zeta$ updates at decoupled workers
An Exact Algorithm

**Branch and Bound**

Let $L = \text{augmented Lagrangian function}$

1. Initialize $L = -\infty$
2. Run $\text{AD}^3$.
   - If solution integer, we are done.
   - Else, if along the execution of $\text{AD}^3$, an upper bound less than $L$ is obtained, return *infeasible*. (Bound)
   - Otherwise, go to 3.
3. Find the most fractional component of the solution.
   - **Branch**: Constrain it to 0 and go to 2, eventually finding a solution or infeasibility. Do the same by constraining to 1.
   - Return the solution vector that gave the larger value.
Argument Identification

score(\text{role} \leftrightarrow \text{span}) = w \cdot g(\text{role, span, frame})

Maximum conditional log-likelihood of local role span pairs using L-BFGS
(Das, Schneider, Chen and Smith, 2010)
Results
Dataset

*FrameNet 1.5, 2010*

- 877 frames
- 1068 role labels
- 9.3K unique predicate types

**Training set:**
- 3.3K sentences
- 19.6K predicate tokens

**Test set:**
- 2420 sentences
- 4.5K predicate tokens
## Results

**Local inference scheme**

<table>
<thead>
<tr>
<th>Method</th>
<th>$P$</th>
<th>$R$</th>
<th>$F_1$</th>
<th>Overlap</th>
<th>Requires</th>
<th>Excludes</th>
<th>Time in Secs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>67.69</td>
<td>59.76</td>
<td>63.48</td>
<td>441</td>
<td>45</td>
<td>15</td>
<td>1.26 ± 0.01</td>
</tr>
<tr>
<td>SEMAFOR (beam = 2)</td>
<td>70.18</td>
<td>59.54</td>
<td>64.42</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>2.74 ± 0.10</td>
</tr>
<tr>
<td>SEMAFOR (beam = 100)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>29.00 ± 0.25</td>
</tr>
<tr>
<td>SEMAFOR (beam = 10000)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>440.67 ± 5.53</td>
</tr>
<tr>
<td><strong>CPLEX, $LP$</strong></td>
<td>70.34</td>
<td>59.43</td>
<td>64.43</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>32.67 ± 1.29</td>
</tr>
<tr>
<td><strong>CPLEX, exact</strong></td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>43.12 ± 1.26</td>
</tr>
<tr>
<td><strong>AD$^3$, $LP$</strong></td>
<td>70.30</td>
<td>59.45</td>
<td>64.42</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4.17 ± 0.01</td>
</tr>
<tr>
<td><strong>AD$^3$, exact</strong></td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.78 ± 0.04</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
<th>Overlap</th>
<th>Requires</th>
<th>Excludes</th>
<th>Time in Secs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>67.69</td>
<td>59.76</td>
<td>63.48</td>
<td>441</td>
<td>45</td>
<td>15</td>
<td>1.26 ± 0.01</td>
</tr>
<tr>
<td>SEMAFOR (beam = 2)</td>
<td>70.18</td>
<td>59.54</td>
<td>64.42</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>2.74 ± 0.10</td>
</tr>
<tr>
<td>SEMAFOR (beam = 100)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>29.00 ± 0.25</td>
</tr>
<tr>
<td>SEMAFOR (beam = 10000)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>440.67 ± 5.53</td>
</tr>
<tr>
<td>CPLEX, LP</td>
<td>70.34</td>
<td>59.43</td>
<td>64.43</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>32.67 ± 1.29</td>
</tr>
<tr>
<td>CPLEX, exact</td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>43.12 ± 1.26</td>
</tr>
<tr>
<td>AD$^3$, LP</td>
<td>70.30</td>
<td>59.45</td>
<td>64.42</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4.17 ± 0.01</td>
</tr>
<tr>
<td>AD$^3$, exact</td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.78 ± 0.04</td>
</tr>
</tbody>
</table>

Performance within the same range
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>$P$</th>
<th>$R$</th>
<th>$F_1$</th>
<th>Overlap</th>
<th>Requires</th>
<th>Excludes</th>
<th>Time in Secs.</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>67.69</td>
<td>59.76</td>
<td>63.48</td>
<td>441</td>
<td>45</td>
<td>15</td>
<td>1.26 ± 0.01</td>
<td>No violations</td>
</tr>
<tr>
<td>SEMAFOR (beam = 2)</td>
<td>70.18</td>
<td>59.54</td>
<td>64.42</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>2.74 ± 0.10</td>
<td>Performance within the same range</td>
</tr>
<tr>
<td>SEMAFOR (beam = 100)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>29.00 ± 0.25</td>
<td>No violations</td>
</tr>
<tr>
<td>SEMAFOR (beam = 10000)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>440.67 ± 5.53</td>
<td>Performance within the same range</td>
</tr>
<tr>
<td>CPLEX, $LP$</td>
<td>70.34</td>
<td>59.43</td>
<td>64.43</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>32.67 ± 1.29</td>
<td>No violations</td>
</tr>
<tr>
<td>CPLEX, $exact$</td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>43.12 ± 1.26</td>
<td>No violations</td>
</tr>
<tr>
<td>$\text{AD}^3$, $LP$</td>
<td>70.30</td>
<td>59.45</td>
<td>64.42</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4.17 ± 0.01</td>
<td>No violations</td>
</tr>
<tr>
<td>$\text{AD}^3$, $exact$</td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.78 ± 0.04</td>
<td>No violations</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>$P$</th>
<th>$R$</th>
<th>$F_1$</th>
<th>Overlap</th>
<th>Requires</th>
<th>Excludes</th>
<th>Time in Secs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>67.69</td>
<td>59.76</td>
<td>63.48</td>
<td>441</td>
<td>45</td>
<td>15</td>
<td>1.26 ± 0.01</td>
</tr>
<tr>
<td>SEMAFOR (beam = 2)</td>
<td>70.18</td>
<td>59.54</td>
<td>64.42</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>2.74 ± 0.10</td>
</tr>
<tr>
<td>SEMAFOR (beam = 100)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>29.00 ± 0.25</td>
</tr>
<tr>
<td>SEMAFOR (beam = 10000)</td>
<td>70.43</td>
<td>59.64</td>
<td>64.59</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>440.67 ± 5.53</td>
</tr>
<tr>
<td><strong>CPLEX, LP</strong></td>
<td>70.34</td>
<td>59.43</td>
<td>64.43</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td><strong>32.67 ± 1.29</strong></td>
</tr>
<tr>
<td><strong>CPLEX, exact</strong></td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td><strong>43.12 ± 1.26</strong></td>
</tr>
<tr>
<td><strong>AD$^3$, LP</strong></td>
<td>70.30</td>
<td>59.45</td>
<td>64.42</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td><strong>4.17 ± 0.01</strong></td>
</tr>
<tr>
<td><strong>AD$^3$, exact</strong></td>
<td>70.31</td>
<td>59.45</td>
<td>64.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td><strong>4.78 ± 0.04</strong></td>
</tr>
</tbody>
</table>

Significantly faster than CPLEX
Example

Austria, once expected to waltz smoothly into the European Union, is elbowing its partners, treading on toes and pogo-dancing in a most un-Viennese manner.
Example

Austria, once expected to waltz smoothly into the European Union, is elbowing its partners, treading on toes and pogo-dancing in a most un-Viennese manner.
Austria, once expected to waltz smoothly into the European Union, is elbowing its partners, treading on toes and pogo-dancing in a most un-Viennese manner.
Austria, once expected to waltz smoothly into the European Union, is elbowing its partners, treading on toes and pogo-dancing in a most un-Viennese manner. *Violates a requires constraint!*
Austria, once expected to waltz smoothly into the European Union, is elbowing its partners, treading on toes and pogo-dancing in a most un-Viennese manner.
Conclusions

• Application of a dual decomposition technique that can handle several overlapping components

• An exact branch-and-bound extension

• Faster inference than a proprietary ILP solver

• Slower than beam search, but avoids linguistic violations
Conclusions

• A frame-semantic parser that can handle linguistic constraints present in FrameNet

• Easy to include more constraints that be extracted from other resources like VerbNet
  —(e.g. verb subcategorization information)

• Future work includes learning “softness” of various constraints, which can be easily encoded using AD$^3$
Last Word

Parser available at:

http://www.ark.cs.cmu.edu/SEMAFOR

AD³ Code available at:

http://www.ark.cs.cmu.edu/AD3
Thank You
Thank You