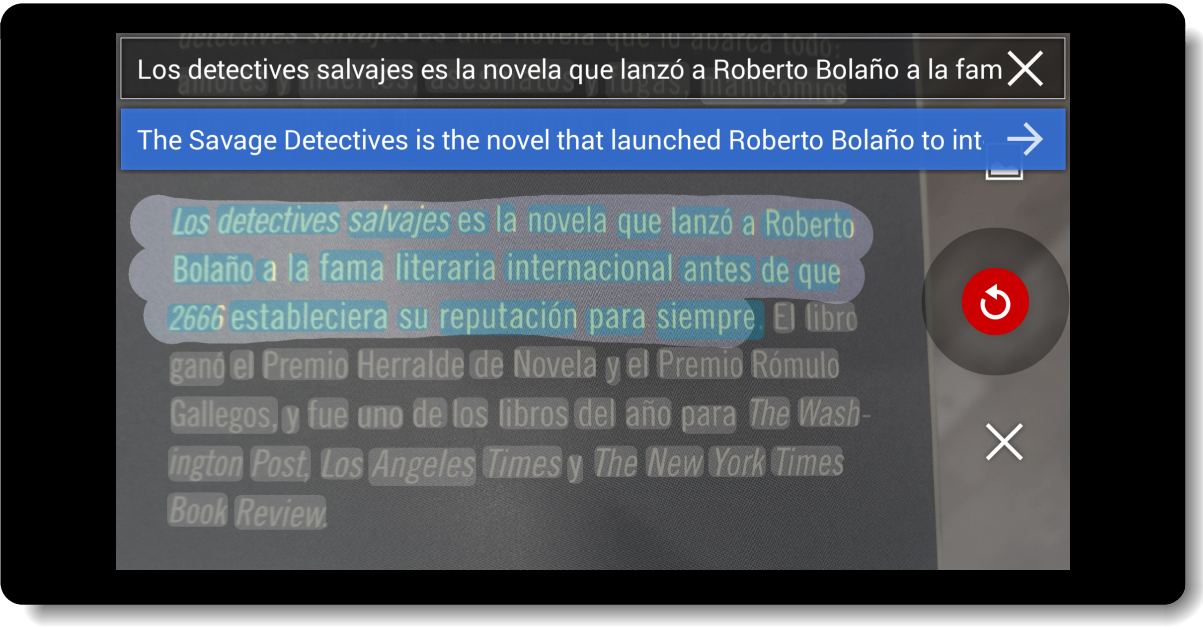
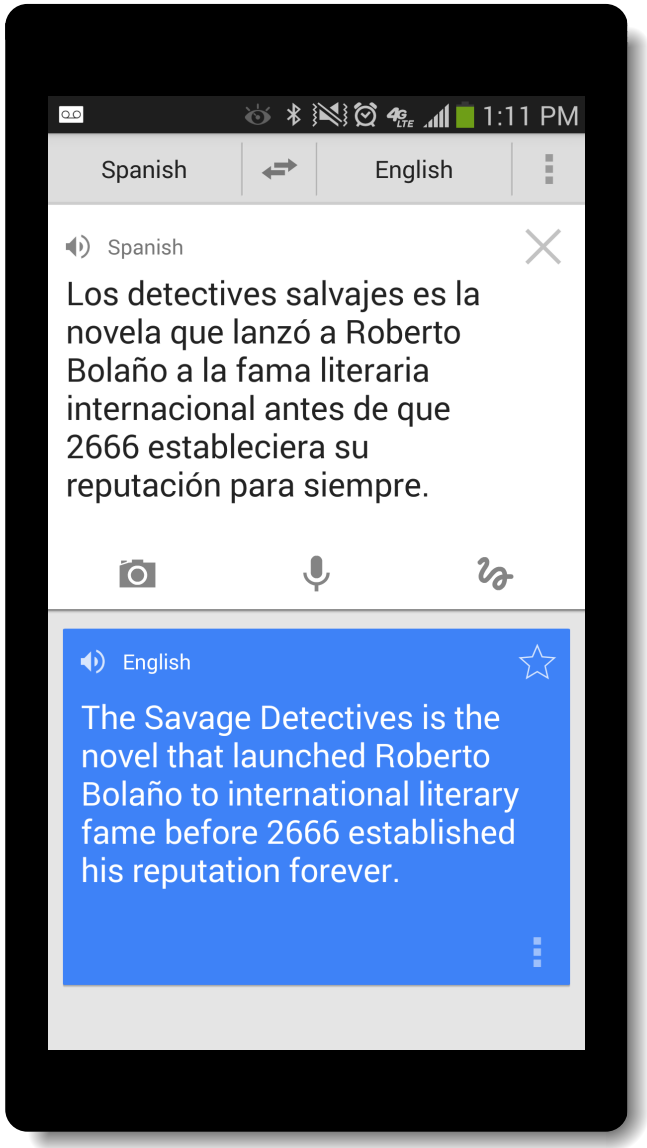
Lagrangian Relaxation for Natural Language Processing

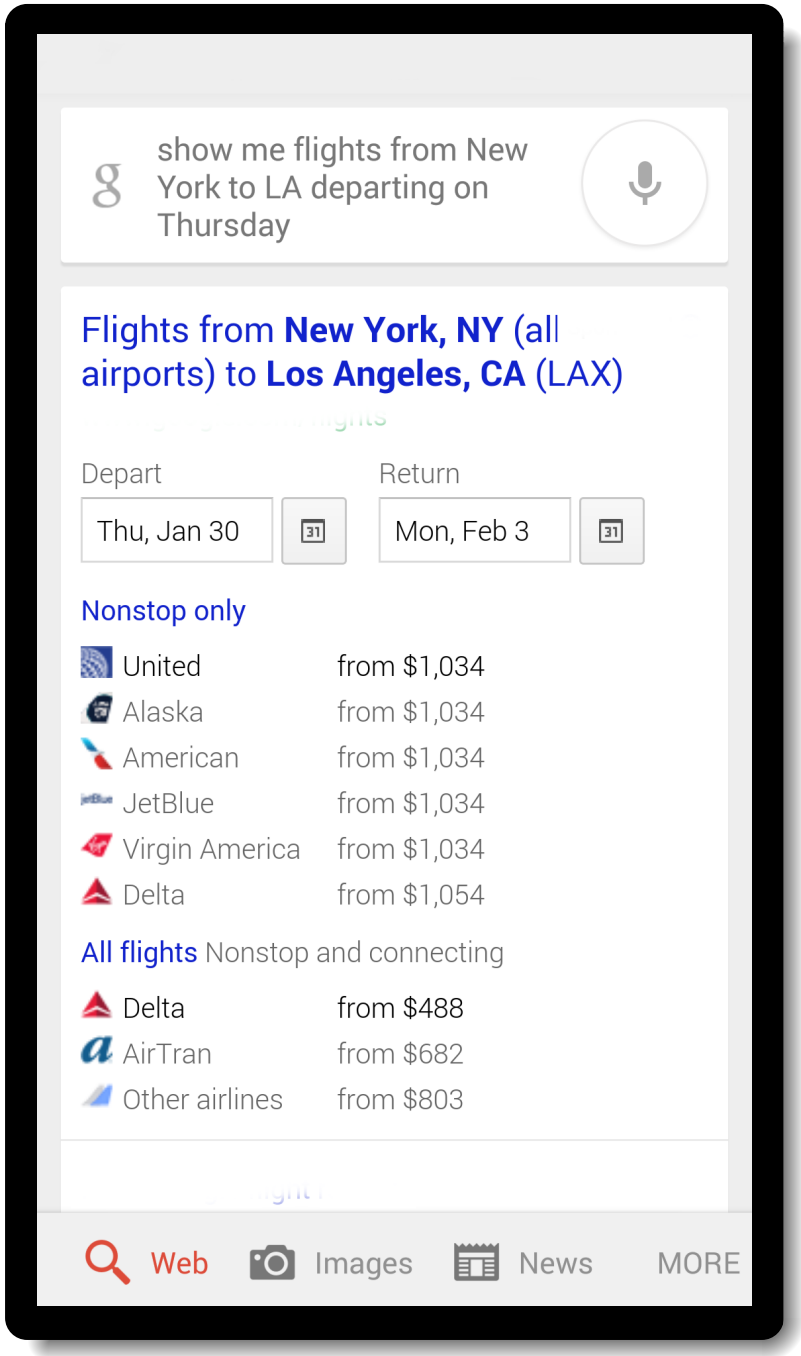
Alexander Rush

MIT CSAIL

Statistical Natural Language Processing


Los detectives salvajes es la novela que lanzó a Roberto Bolaño a la fam
The Savage Detectives is the novel that launched Roberto Bolaño to int
Los detectives salvajes es la novela que lanzó a Roberto Bolaño a la fama literaria internacional antes de que 2666 estableciera su reputación para siempre. El libro ganó el Premio Herralde de Novela y el Premio Rómulo Gallegos, y fue uno de los libros del año para The Washington Post, Los Angeles Times y The New York Times Book Review.


1:11 PM
Spanish
English
Spanish
Los detectives salvajes es la novela que lanzó a Roberto Bolaño a la fama literaria internacional antes de que 2666 estableciera su reputación para siempre.
English
The Savage Detectives is the novel that launched Roberto Bolaño to international literary fame before 2666 established his reputation forever.


show me flights from New York to LA departing on Thursday
Flights from New York, NY (all airports) to Los Angeles, CA (LAX)
Depart
Thu, Jan 30
Return
Mon, Feb 3
Nonstop only
United from $1,034
Alaska from $1,034
American from $1,034
JetBlue from $1,034
Virgin America from $1,034
Delta from $1,054
All flights Nonstop and connecting
Delta from $488
AirTran from $682
Other airlines from $803
Web
Images
News
MORE

Statistical Inference in Natural Language Processing

Goal: Predict best output under a statistical model.

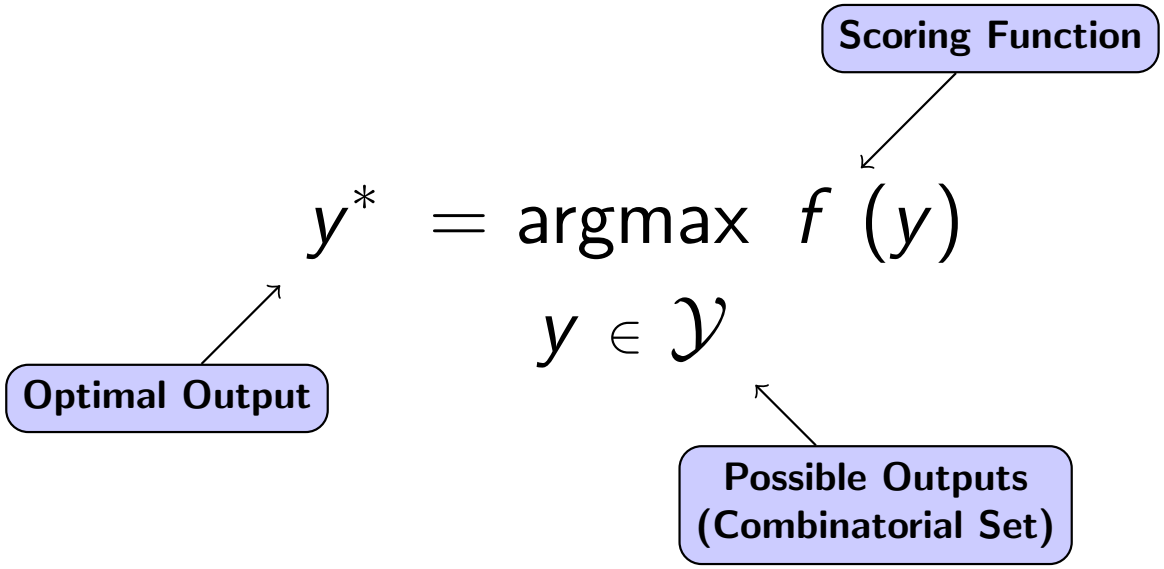
$$y^* = \underset{y \in \mathcal{Y}}{\text{argmax}} \; f(y)$$

Example: Spanish $\rightarrow$ English Translation

$\mathcal{Y}$	Possible English translations
$f(y)$	Probability of translation y
y^*	Optimal translation under this model

Statistical Inference in Natural Language Processing

Goal: Predict best output under a statistical model.

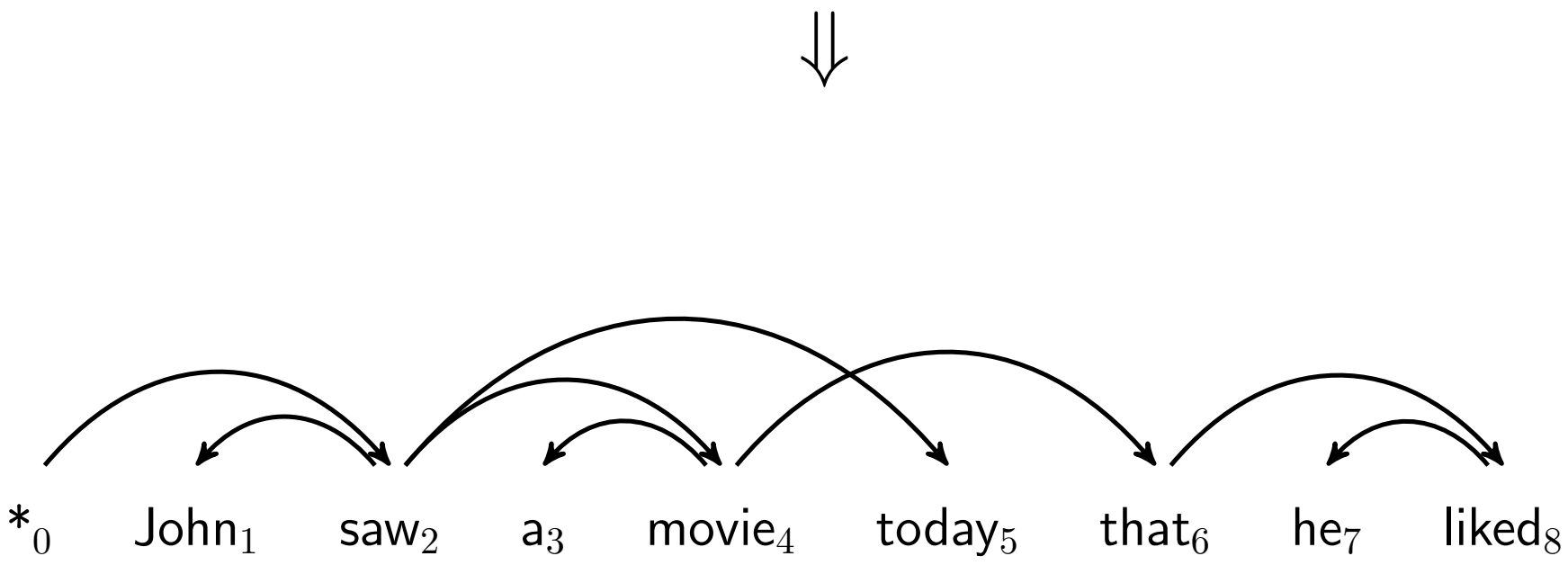


Example: Spanish $\rightarrow$ English Translation

$\mathcal{Y}$	Possible English translations
$f(y)$	Probability of translation y
y^*	Optimal translation under this model

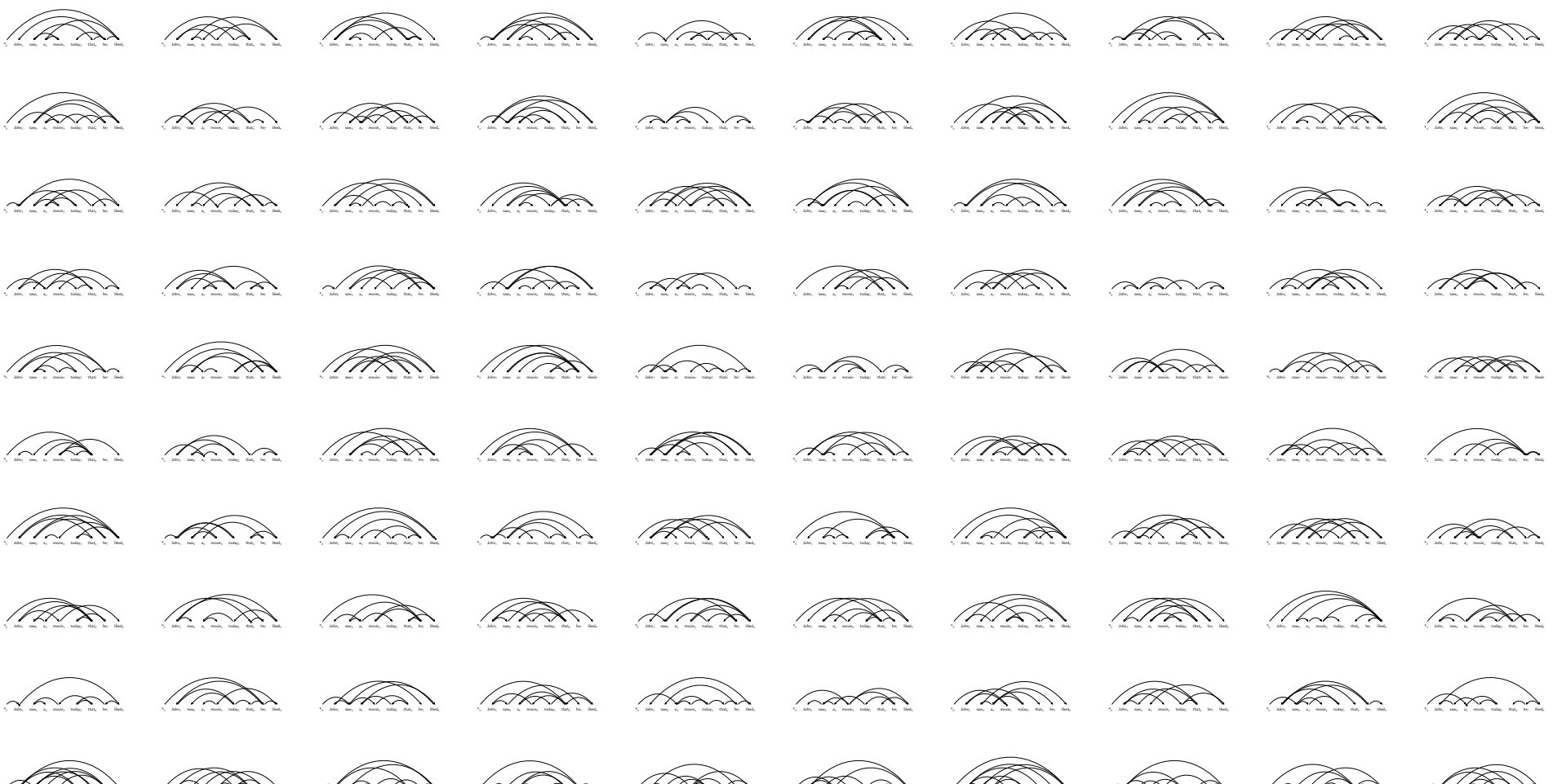
Statistical Model of Syntax

John saw a movie today that he liked.

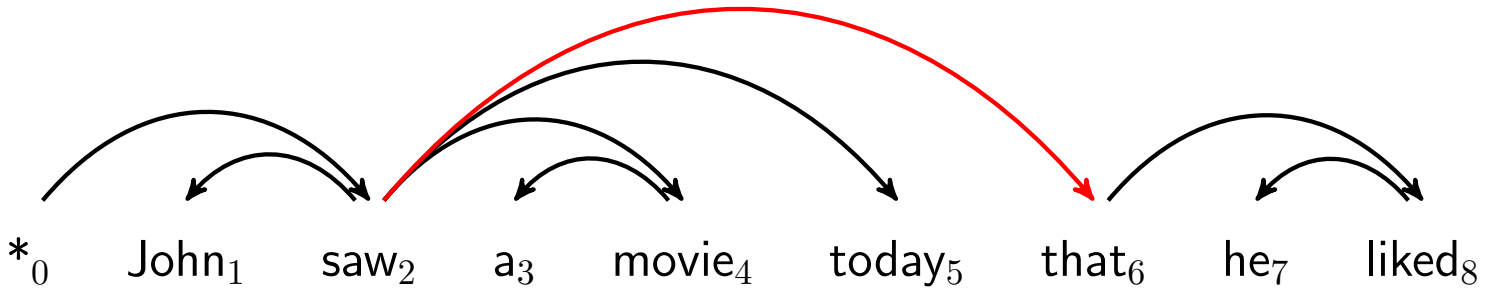


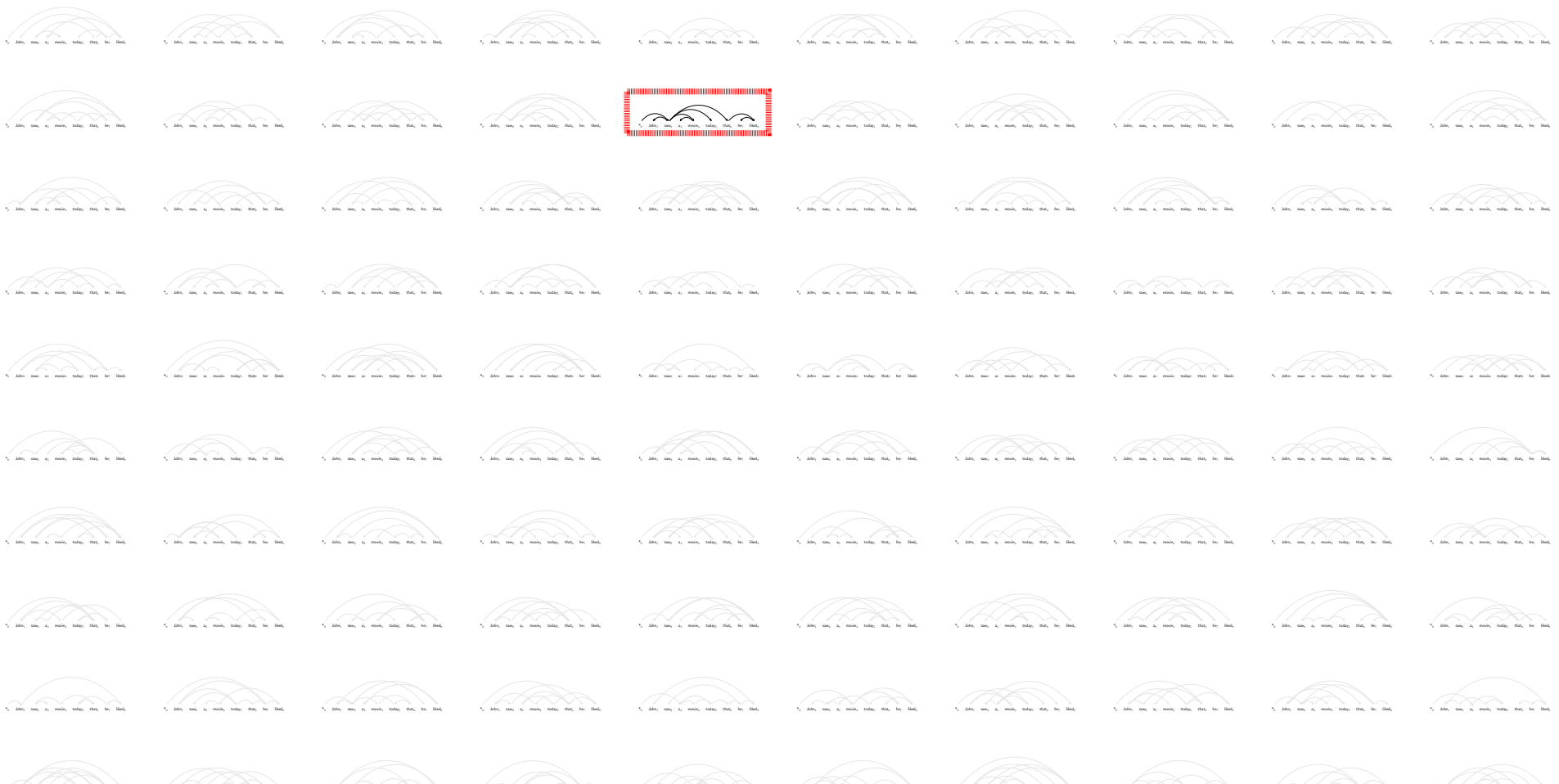
Statistical Model of Syntax

$*_0$ John$_1$ saw$_2$ a$_3$ movie$_4$ today$_5$ that$_6$ he$_7$ liked$_8$

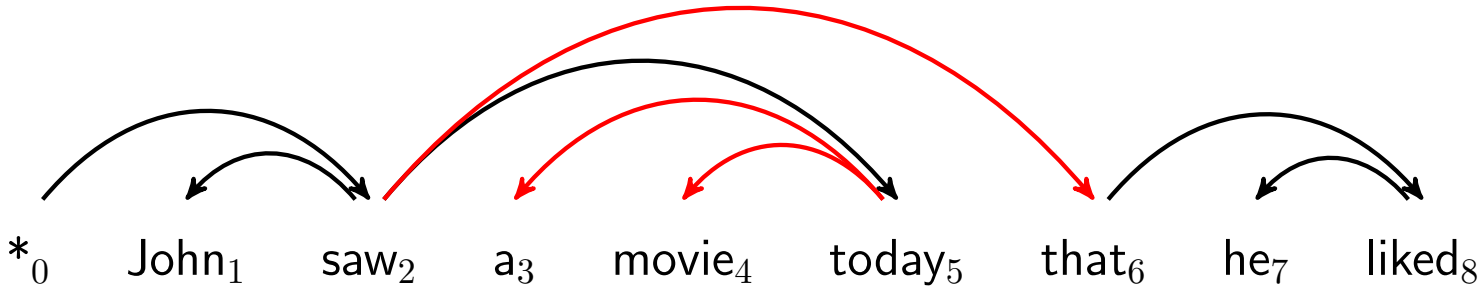


Statistical Model of Syntax

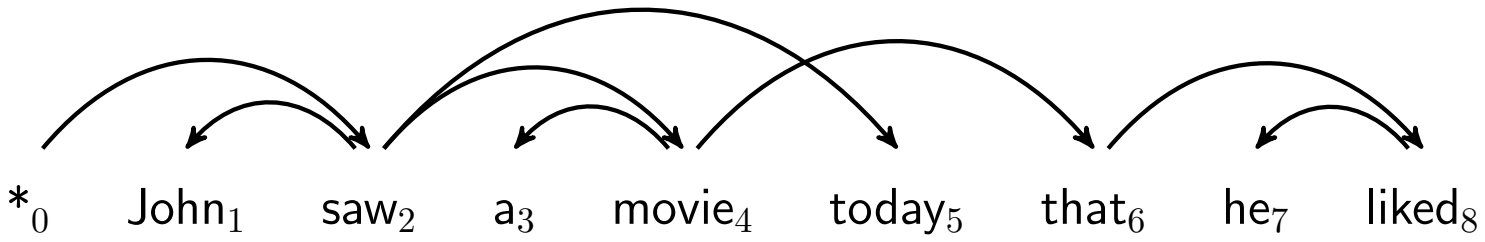


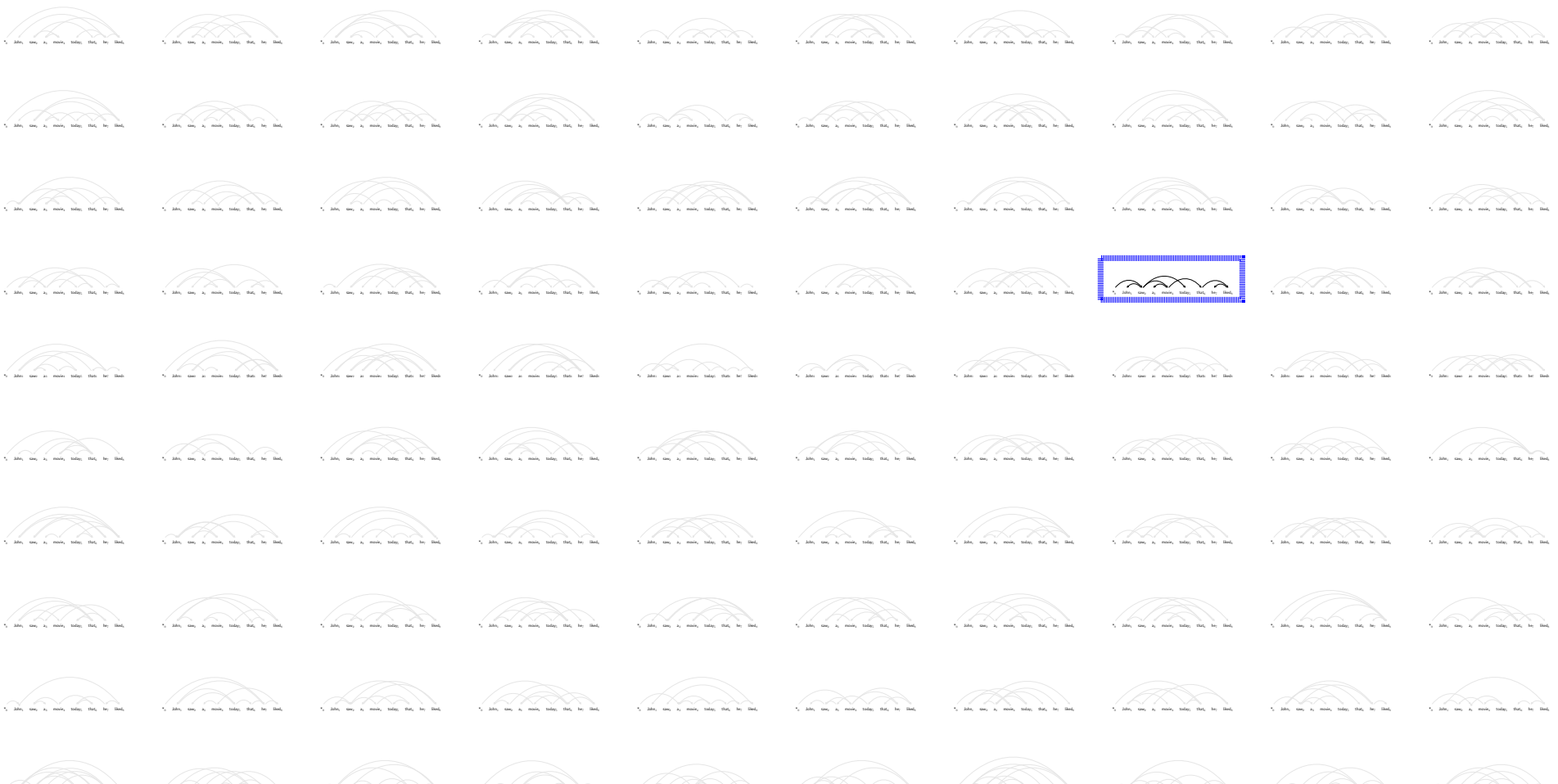


Statistical Model of Syntax

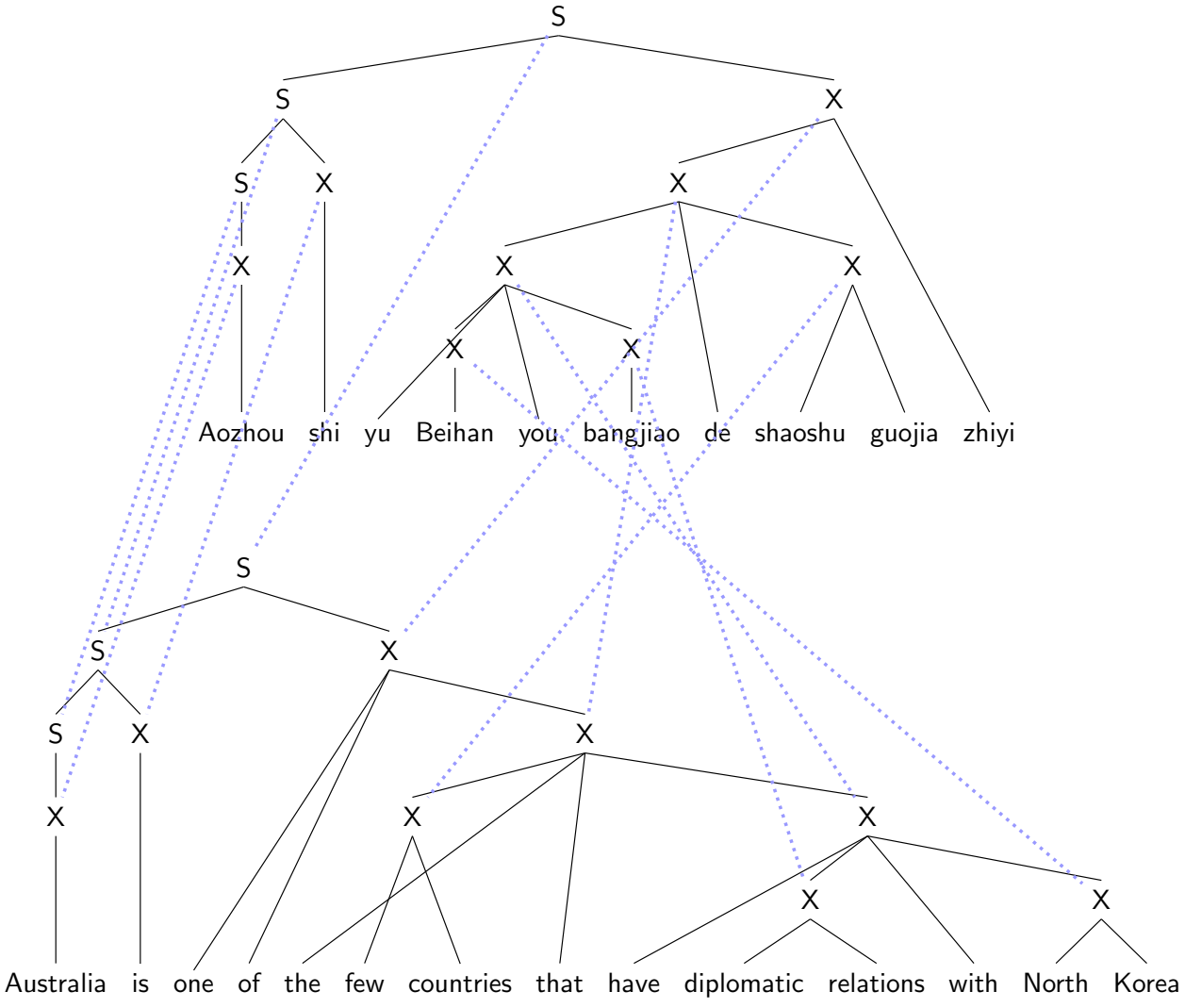


Statistical Model of Syntax



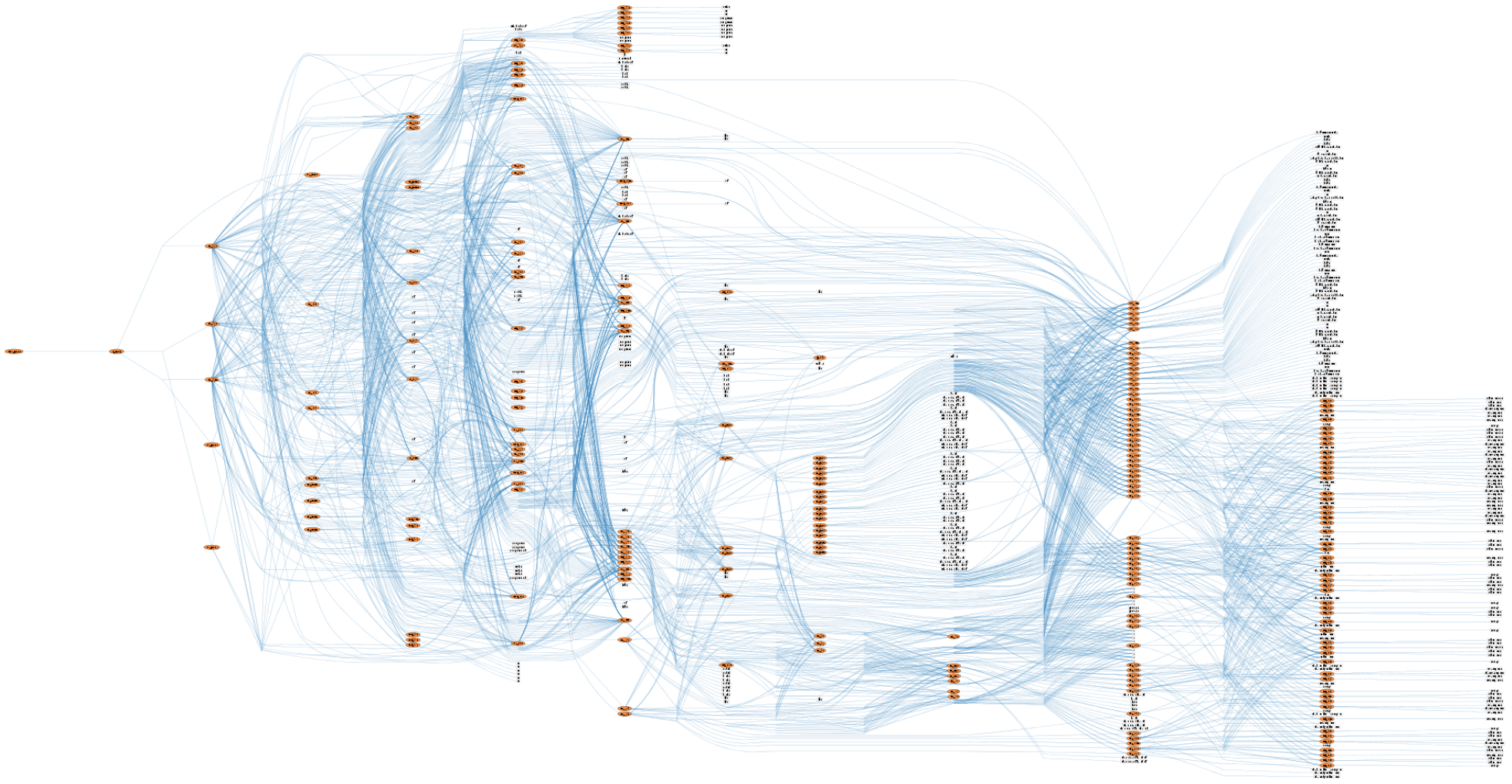


Statistical Model of Translation



Statistical Model of Translation

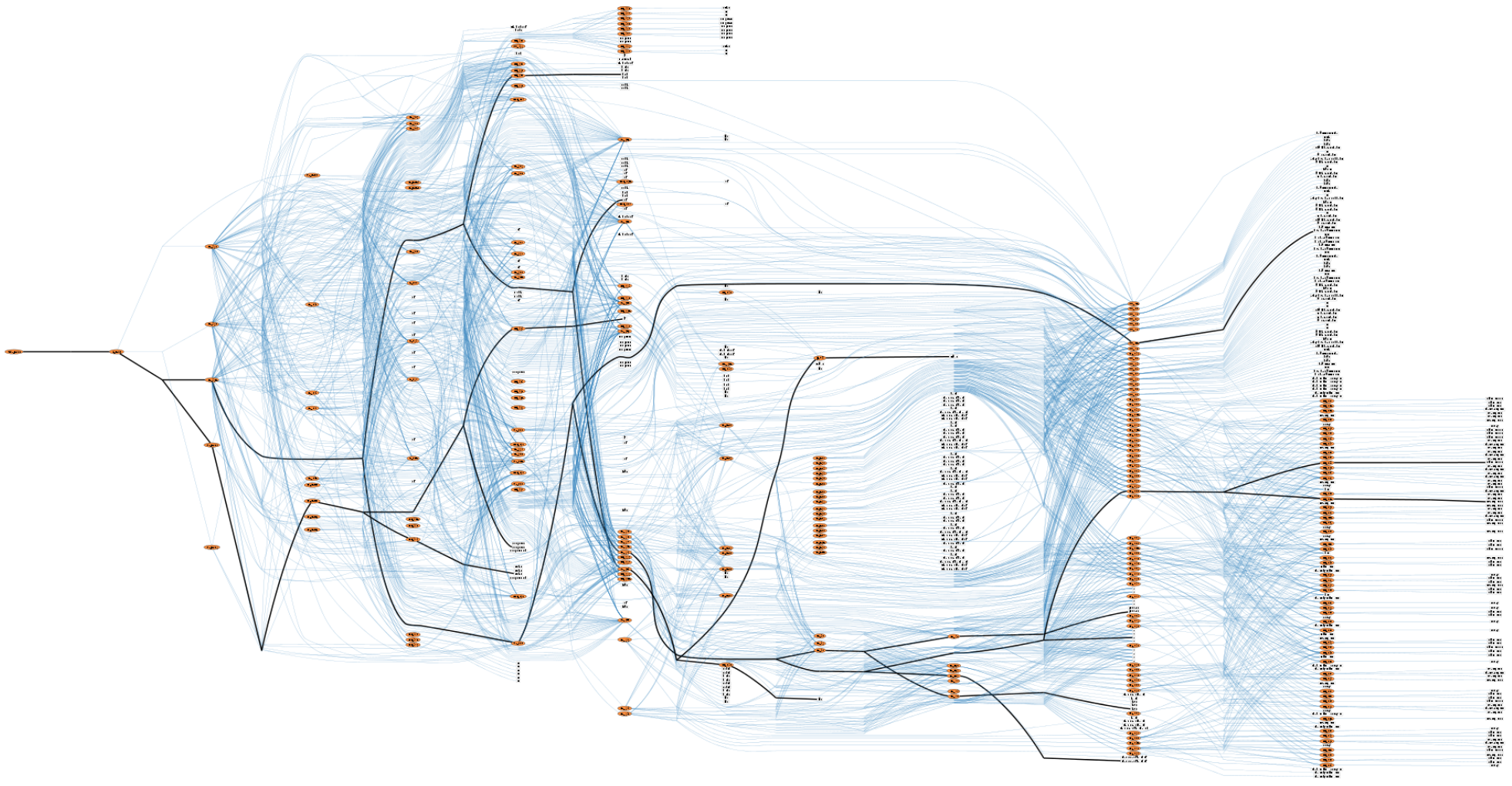
载入 黛 妃 死 因 调查 资料 的 两 台 手 提 电脑 遭 窃



Statistical Model of Translation

载入 黛 妃 死 因 调查 资料 的 两 台 手 提 电脑 遭 窃

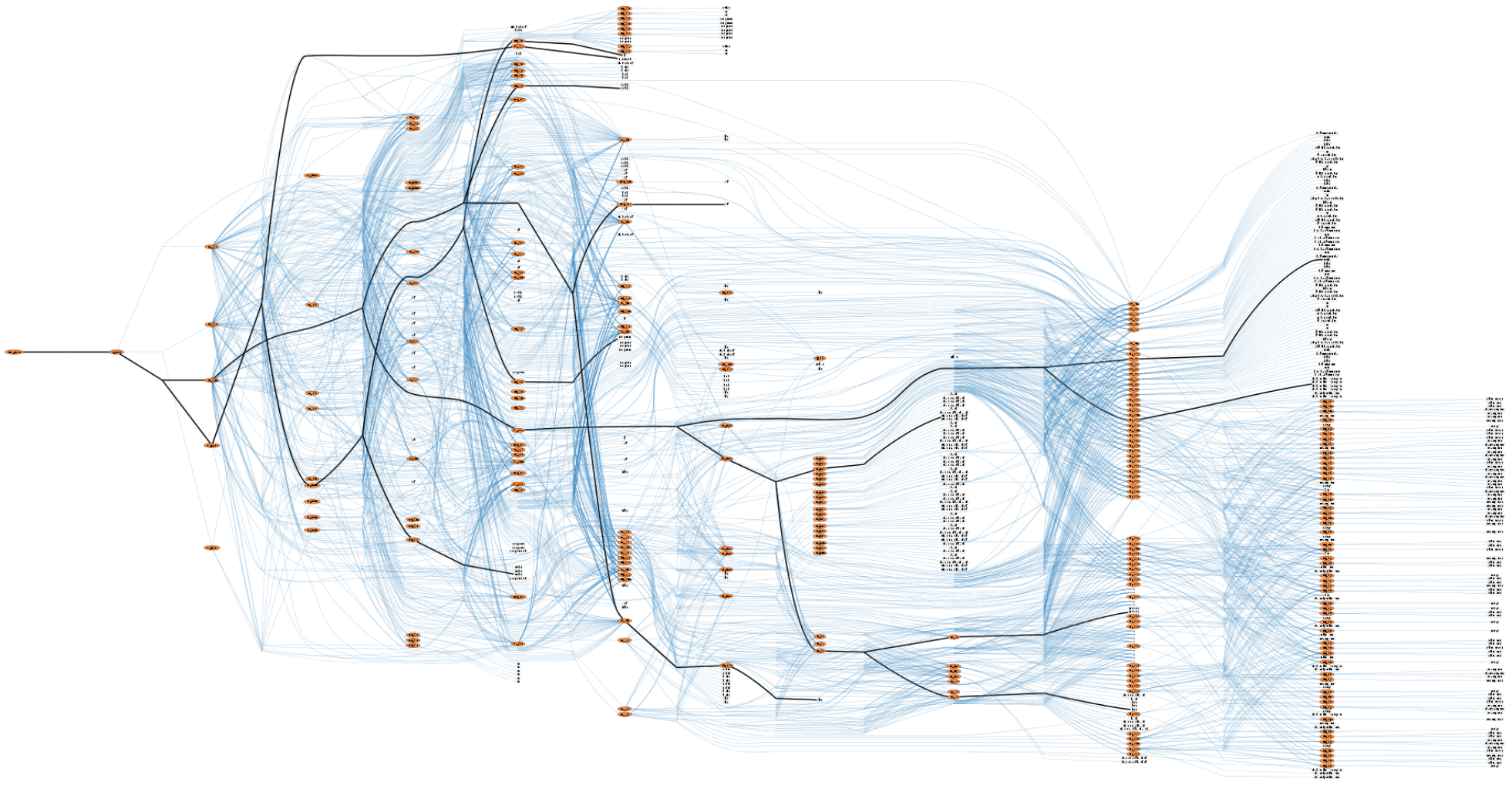
Into devi princess death investigation information of two hands mentioned computers in



Statistical Model of Translation

载入 黛 妃 死 因 调查 资料 的 两 台 手 提 电脑 遭 窃

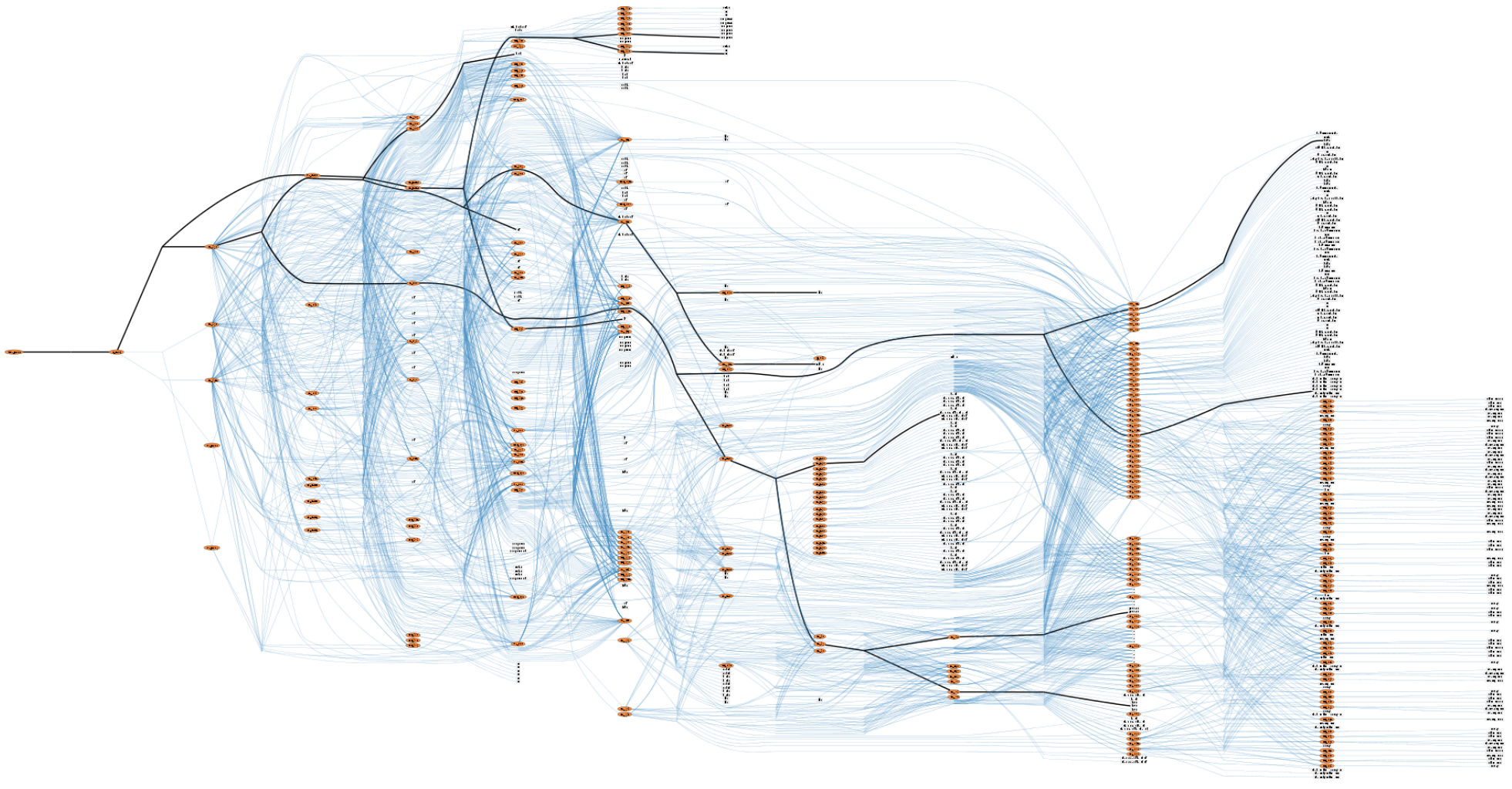
Diana will be recorded down in the death investigation



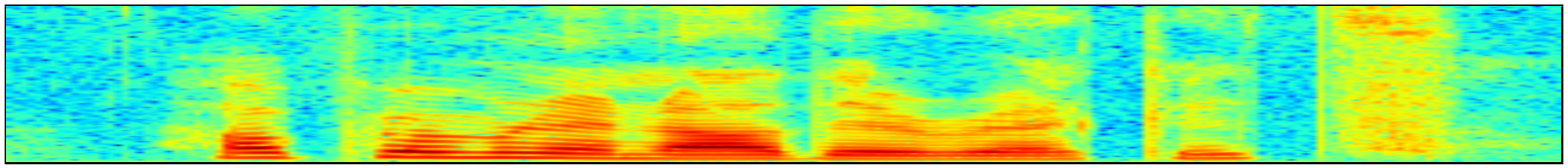
Statistical Model of Translation

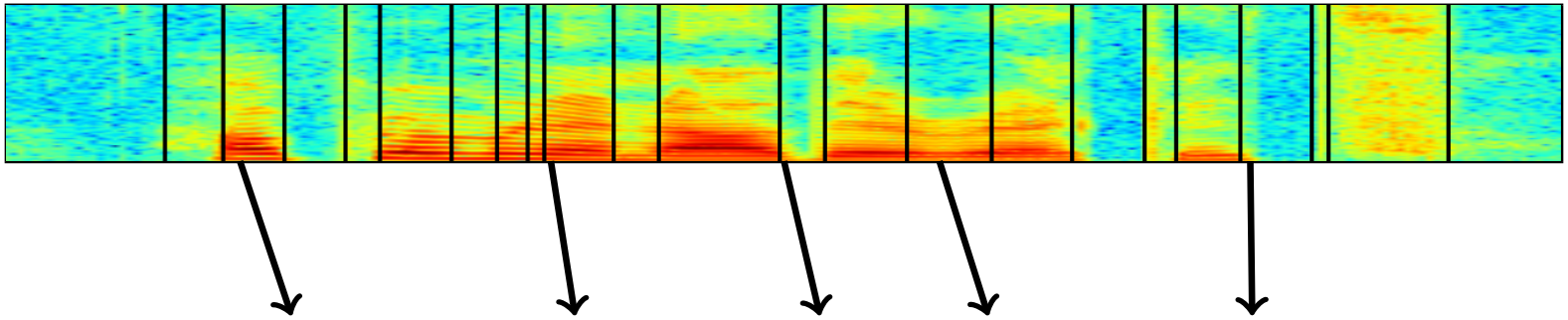
载入 黛 妃 死 因 调查 资料 的 两 台 手 提 电脑 遭 窃

Two laptops with information on the cause of Princess Diana's death were stolen

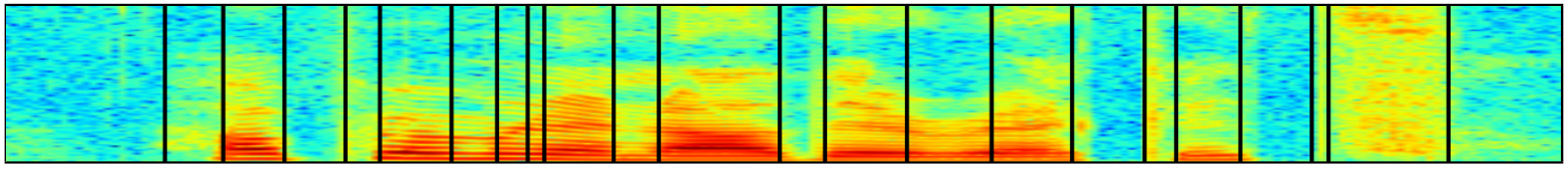


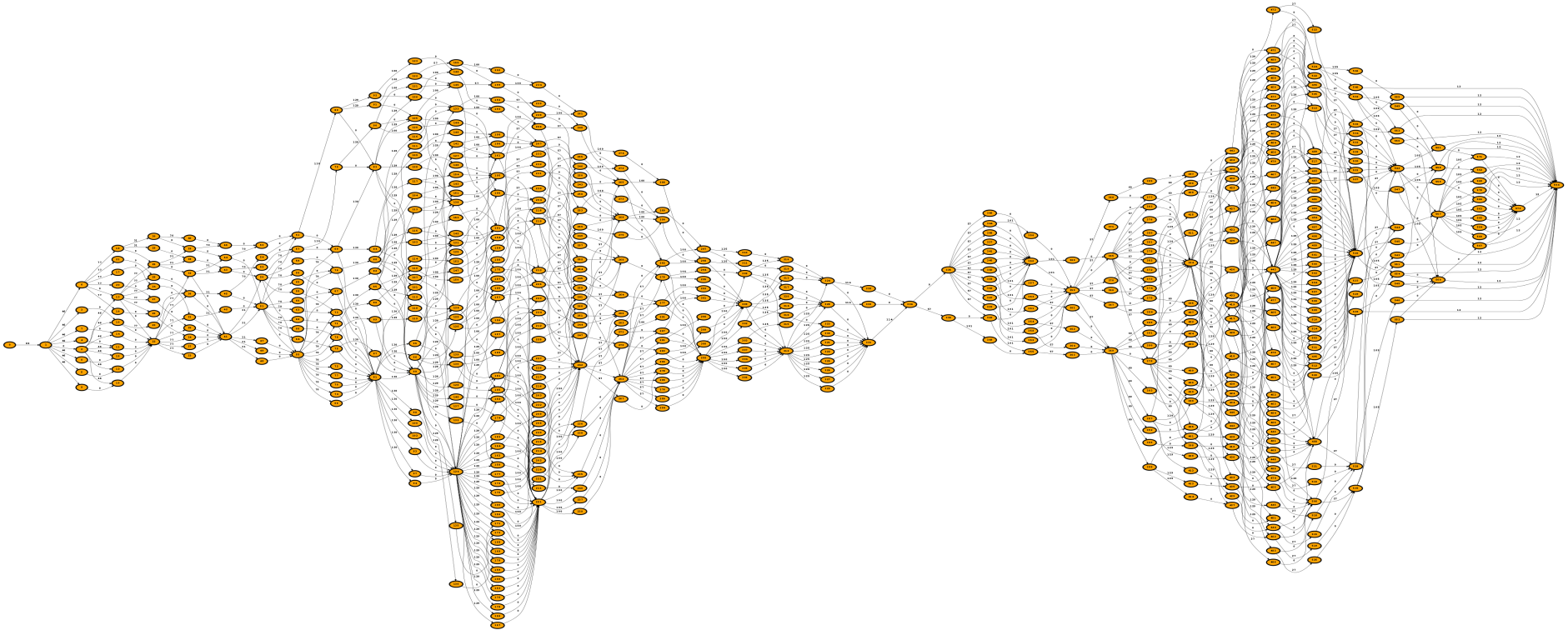
Statistical Model of Speech



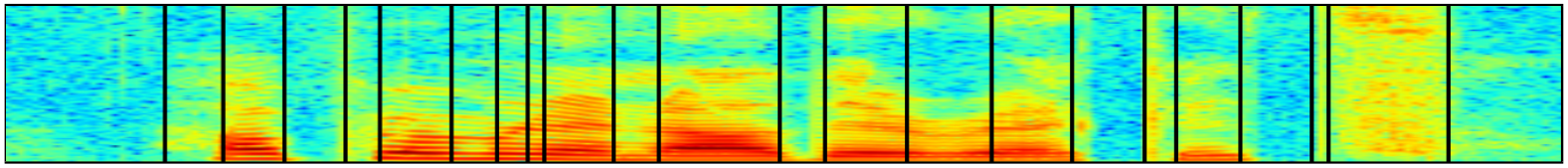


Statistical Model of Speech

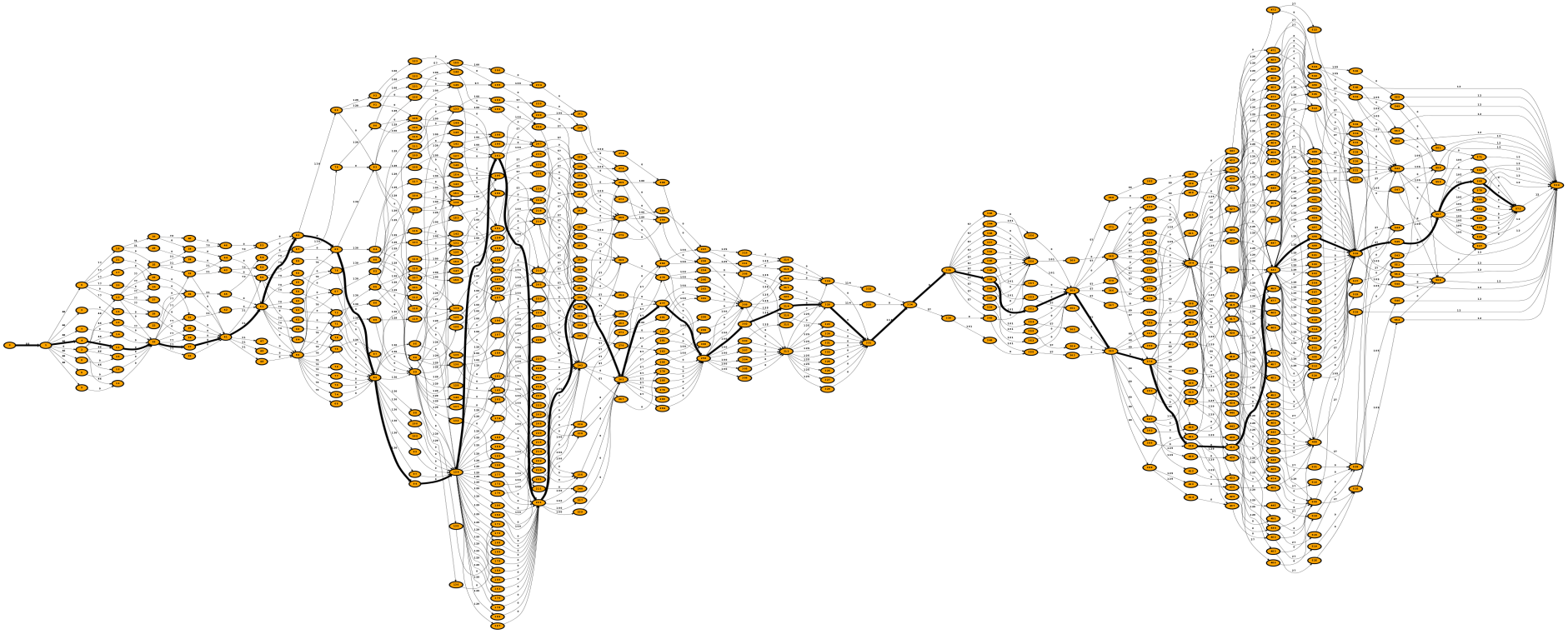




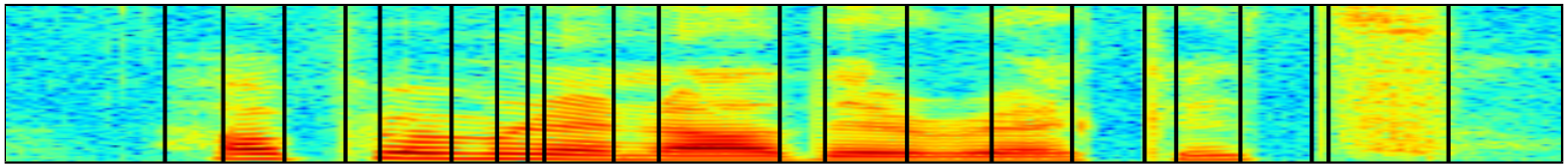
Statistical Model of Speech



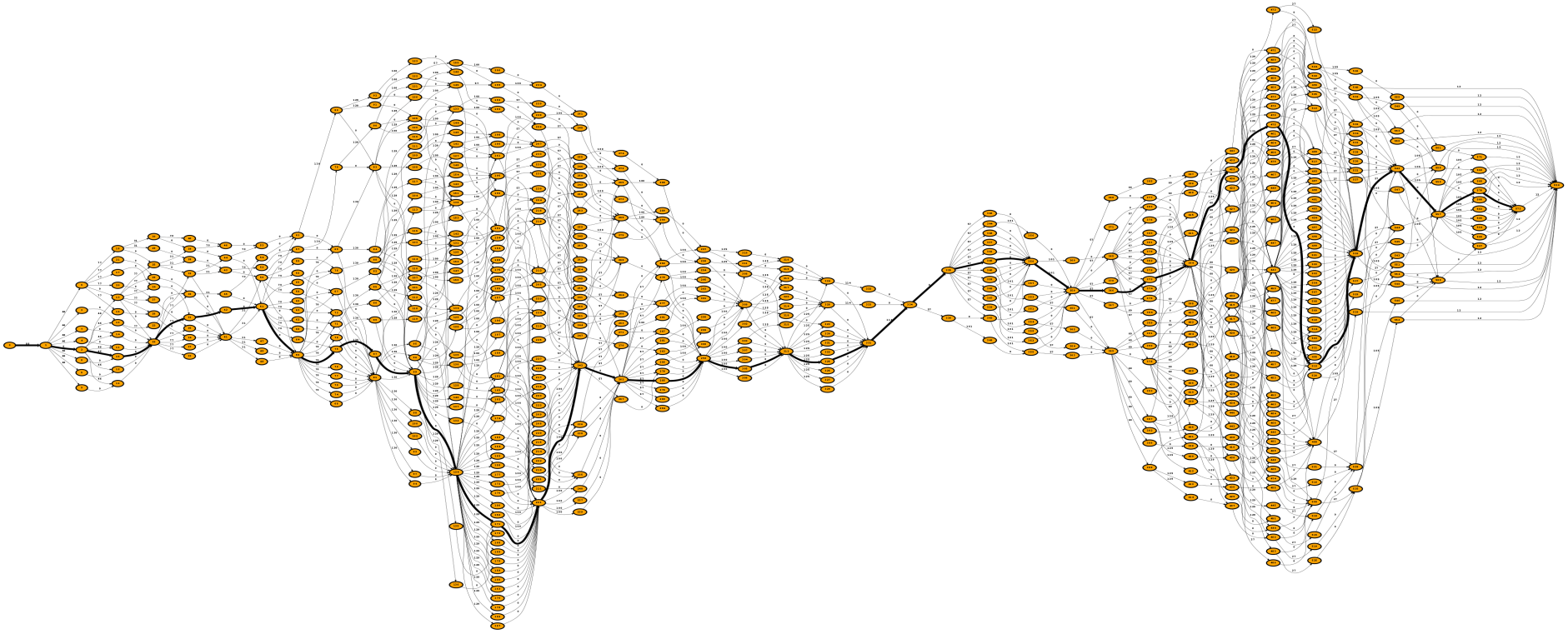
Transcription: Help peppermint on their records



Statistical Model of Speech



Transcription: How permanent are their records



Natural Language Systems

Aim:

- Rich models of language
- Optimal predictions
- Efficient inference

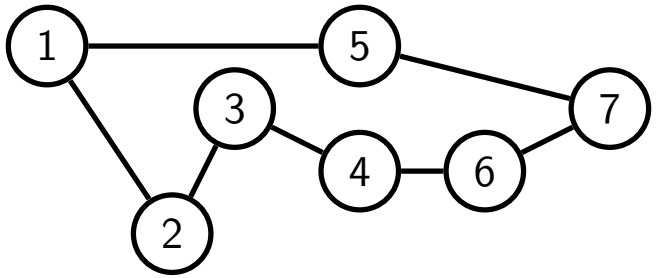
$$\text{Better Model} \Rightarrow \text{Harder Inference}$$

Approach: Lagrangian Relaxation for Natural Language Inference

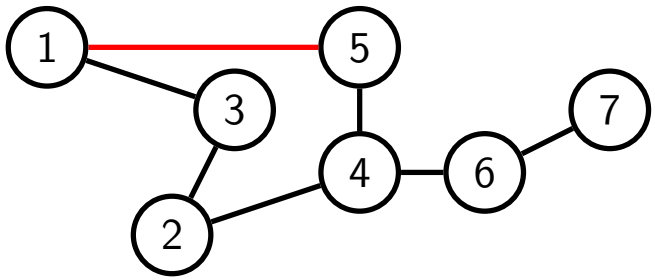
Held and Karp's Algorithm for TSP (Held and Karp, 1971)

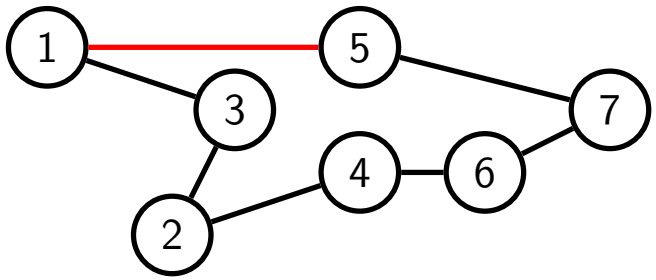
Traveling Salesman Problem: Find the highest-weight tour,

TOUR



TOUR $\subset$ *1-TREE*





Idea: Use Lagrange multipliers to encourage best 1-tree to be a tour.

Lagrangian Relaxation for Natural Language Inference

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

Lagrangian Relaxation allows us to derive novel inference algorithms.

Method:

- Define a larger, relaxed set, $\mathcal{Y} \subset \mathcal{Z}$, to optimize over.
- Introduce Lagrange multipliers to encourage original constraints.

Challenge: Design a relaxed set that produces optimal solutions efficiently.

Benefits of Lagrangian Relaxation

Simple - Uses standard combinatorial algorithms.

- Dynamic Programming
- Shortest Path
- Spanning Tree Algorithms

Efficient - Comparable to heuristic inference algorithms.

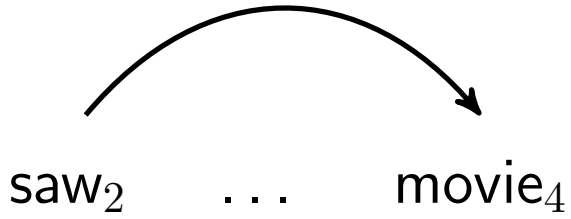
Strong Guarantees - Gives a certificate of optimality when exact.

Overview

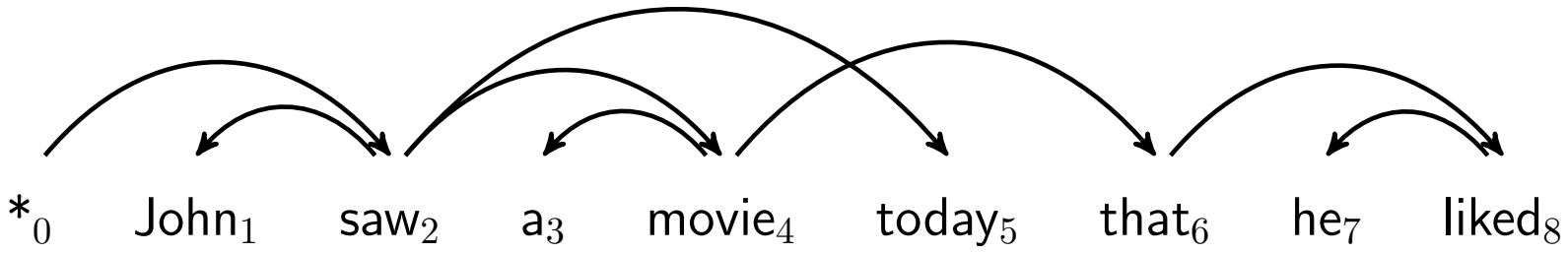
1. Lagrangian Relaxation for Dependency Parsing
2. Lagrangian Relaxation for Syntax-Based Translation
3. Future Work

Dependency Parsing

A dependency arc indicates a *head-modifier* relationship.



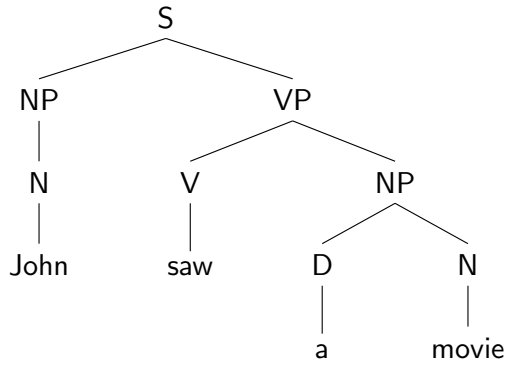
A dependency parse is a directed spanning tree over a sentence.



Statistical Models of Syntactic Parsing

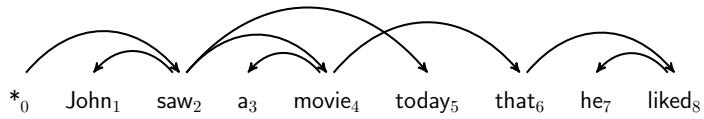
Constituency Parsers (1994 -)

- High accuracy model of syntax
- Predict context-free derivations
- Around 20 words per second



Dependency Parsers (2003 -)

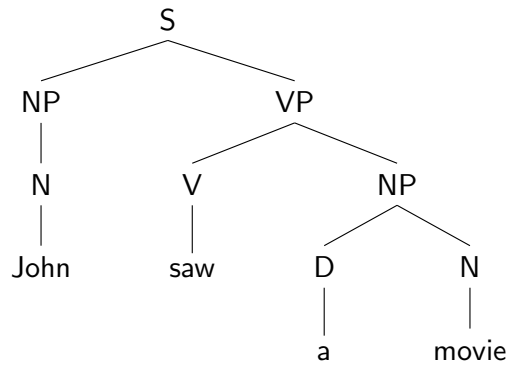
- Comparably accurate model
- Predict dependency structures
- Around 20,000 words per second



Statistical Models of Syntactic Parsing

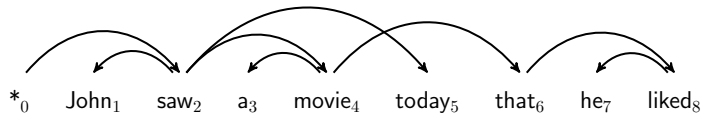
Constituency Parsers (1994 -)

- High accuracy model of syntax
- Predict context-free derivations
- Around 20 words per second



Dependency Parsers (2003 -)

- Comparably accurate model
- Predict dependency structures
- Around 20,000 words per second



Scale of Dependency Parsers

5.2 million books, 500 billion words (Google Books Corpus)

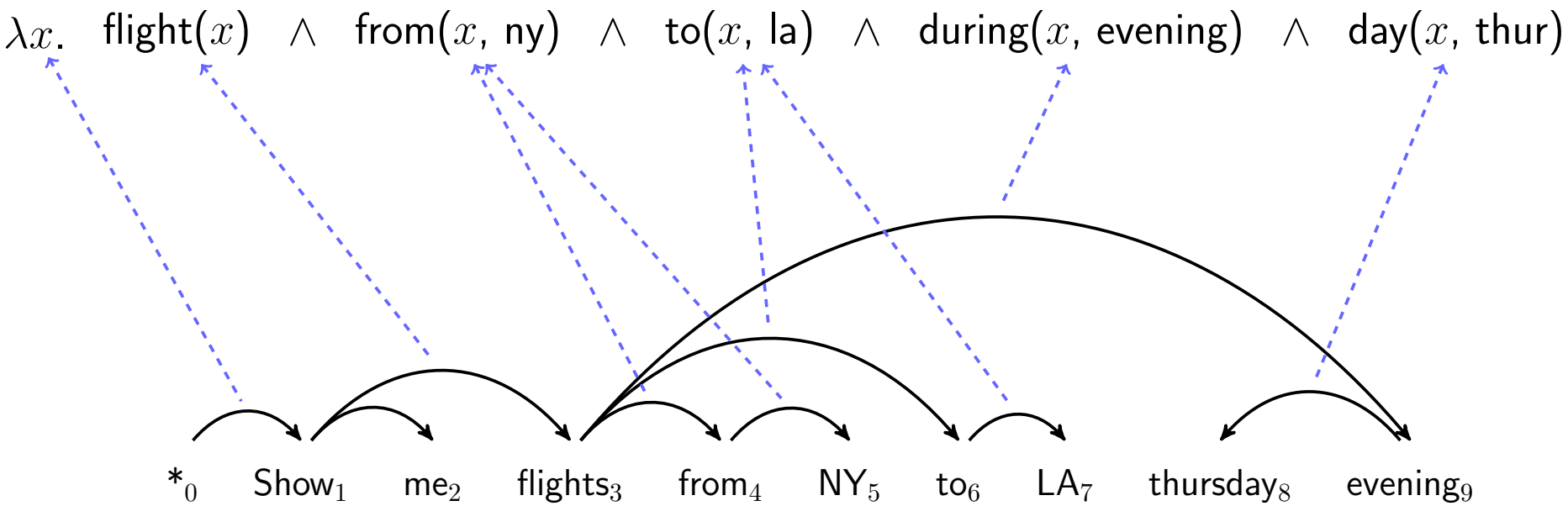
Applications of Dependency Parsing

Many applications rely on fast and accurate dependency parsing:

- Question Answering
- Information Extraction
- Biological Text Processing
- Dialog Systems
- Summarization
- Statistical Machine Translation
- Sentiment Analysis
- Hedge and Negation Detection

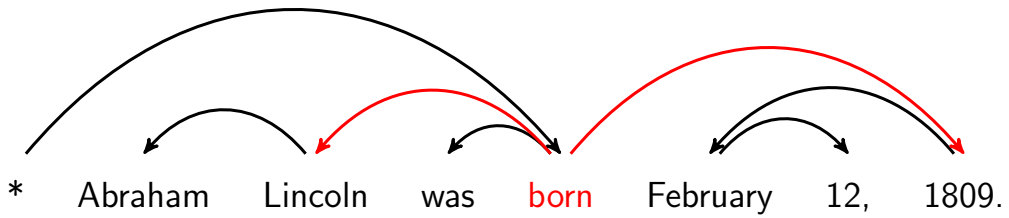
Application: Question Answering

Show me flights from NY to LA Thursday evening.

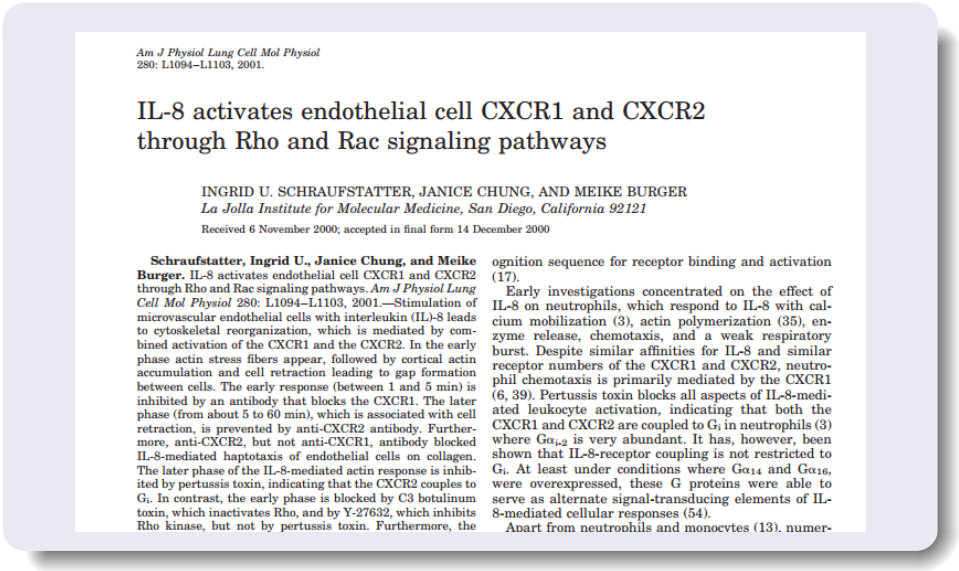


Application: Information Extraction

Abraham Lincoln was born February 12, 1809 in a one-room log cabin on the Sinking Spring Farm in Hardin County, Kentucky (now LaRue County). He is descended from Samuel Lincoln, who arrived in Hingham, Massachusetts, from Norfolk, England, in the 17th century...



Application: Protein Interaction Detection



Am J Physiol Lung Cell Mol Physiol
280: L1094–L1103, 2001.

IL-8 activates endothelial cell CXCR1 and CXCR2 through Rho and Rac signaling pathways

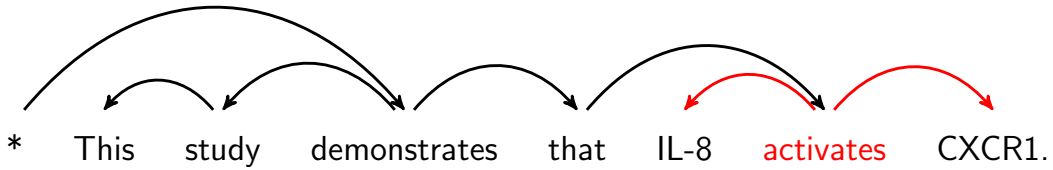
INGRID U. SCHRAUFSTATTER, JANICE CHUNG, AND MEIKE BURGER
La Jolla Institute for Molecular Medicine, San Diego, California 92121

Received 6 November 2000; accepted in final form 14 December 2000

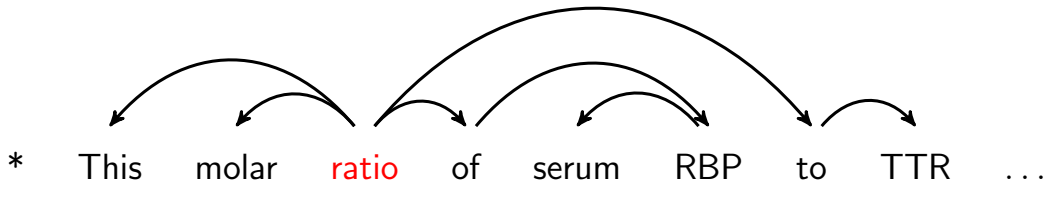
Schraufstatter, Ingrid U., Janice Chung, and Meike Burger. IL-8 activates endothelial cell CXCR1 and CXCR2 through Rho and Rac signaling pathways. *Am J Physiol Lung Cell Mol Physiol* 280: L1094–L1103, 2001.—Stimulation of microvascular endothelial cells with interleukin (IL)-8 leads to cytoskeletal reorganization, which is mediated by combined activation of the CXCR1 and the CXCR2. In the early phase actin stress fibers appear, followed by cortical actin accumulation and cell retraction leading to gap formation between cells. The early response (between 1 and 5 min) is inhibited by an antibody that blocks the CXCR1. The later phase (from about 5 to 60 min), which is associated with cell retraction, is prevented by anti-CXCR2 antibody. Furthermore, anti-CXCR2, but not anti-CXCR1, antibody blocked IL-8-mediated haptotaxis of endothelial cells on collagen. The later phase of the IL-8-mediated actin response is inhibited by pertussis toxin, indicating that the CXCR2 couples to G_i. In contrast, the early phase is blocked by C3 botulinum toxin, which inactivates Rho, and by Y-27632, which inhibits Rho kinase, but not by pertussis toxin. Furthermore, the ognition sequence for receptor binding and activation (17).

Early investigations concentrated on the effect of IL-8 on neutrophils, which respond to IL-8 with calcium mobilization (3), actin polymerization (35), enzyme release, chemotaxis, and a weak respiratory burst. Despite similar affinities for IL-8 and similar receptor numbers of the CXCR1 and CXCR2, neutrophil chemotaxis is primarily mediated by the CXCR1 (6, 39). Pertussis toxin blocks all aspects of IL-8-mediated leukocyte activation, indicating that both the CXCR1 and CXCR2 are coupled to G_i in neutrophils (3) where $G\alpha_{i-2}$ is very abundant. It has, however, been shown that IL-8-receptor coupling is not restricted to G_i. At least under conditions where $G\alpha_{14}$ and $G\alpha_{16}$, were overexpressed, these G proteins were able to serve as alternate signal-transducing elements of IL-8-mediated cellular responses (54).

Apart from neutrophils and monocytes (13), numer-



IL-8 $\Rightarrow$ CXCR1

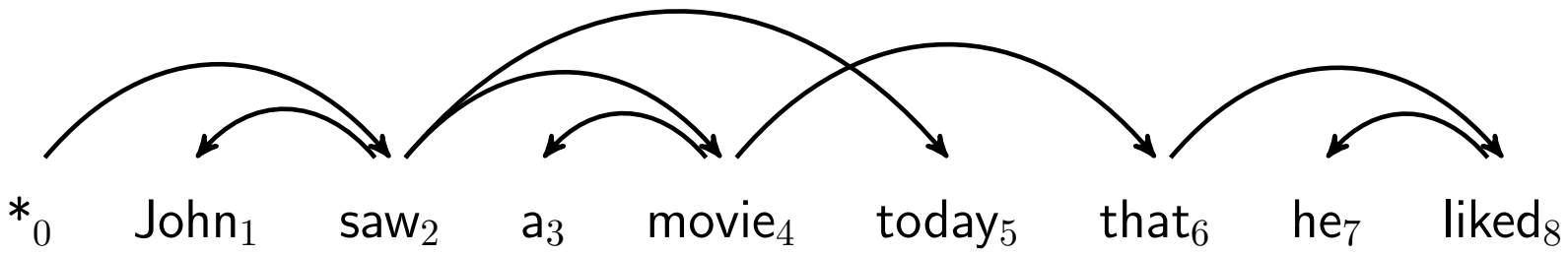


No Interaction

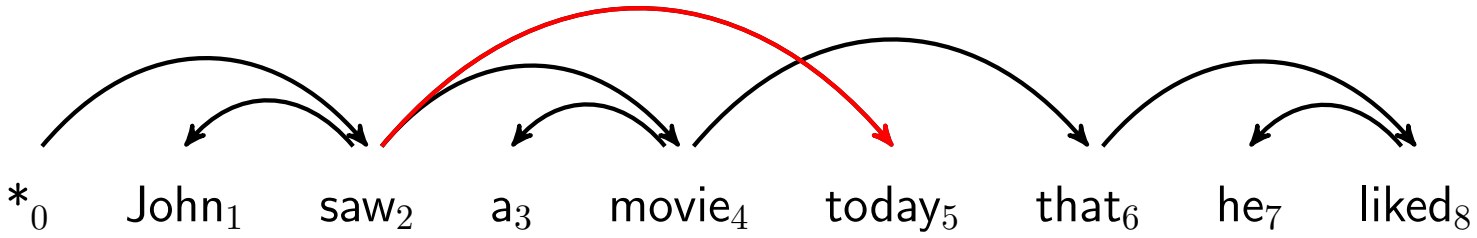
Inference in Dependency Parsing

Let $\mathcal{Y}$ be all possible directed spanning trees.

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$



First-Order Model of Dependency Parsing (McDonald et al., 2005)

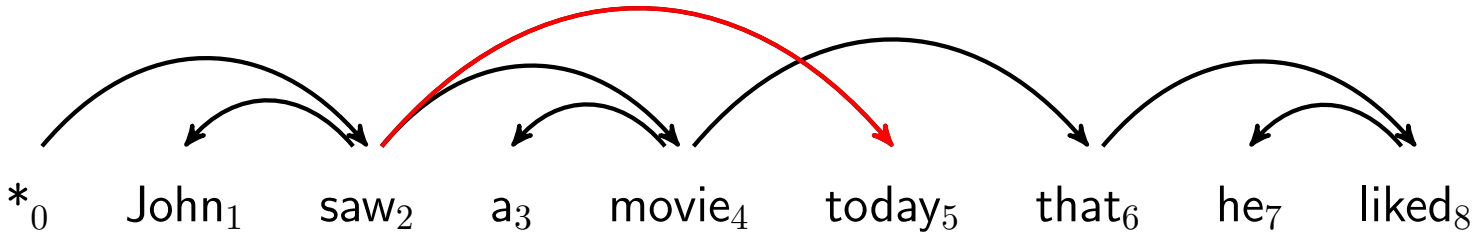


$$f(y) = score(*_0 \rightarrow \texttt{saw}_2) + score(\texttt{saw}_2 \rightarrow \texttt{movie}_4) + score(\texttt{saw}_4 \rightarrow \texttt{today}_5) \ldots$$

where $score(\texttt{saw}_2 \rightarrow \texttt{today}_5) = \log p(\texttt{today}_5 | \texttt{saw}_2)$

or $score(\texttt{saw}_2 \rightarrow \texttt{today}_5) = w \cdot \phi(\texttt{saw}_2, \texttt{today}_5)$

First-Order Model of Dependency Parsing (McDonald et al., 2005)



$$f(y) = score(*_0 \rightarrow \texttt{saw}_2) + score(\texttt{saw}_2 \rightarrow \texttt{movie}_4) + score(\texttt{saw}_4 \rightarrow \texttt{today}_5) \ldots$$

where $score(\texttt{saw}_2 \rightarrow \texttt{today}_5) = \log p(\texttt{today}_5 | \texttt{saw}_2)$

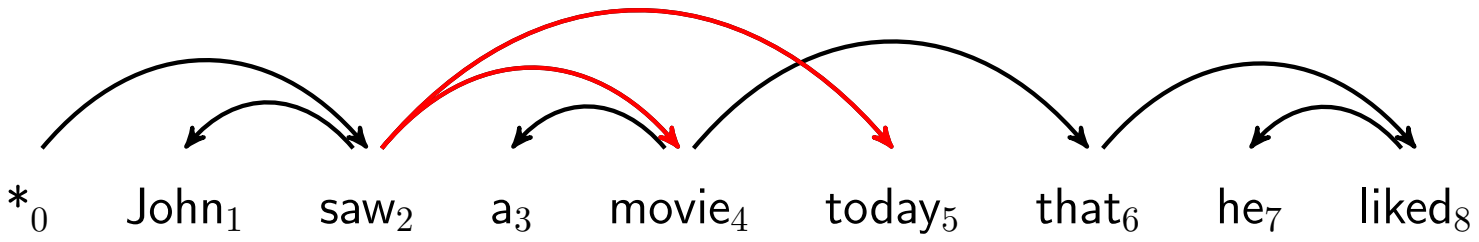
or $score(\texttt{saw}_2 \rightarrow \texttt{today}_5) = w \cdot \phi(\texttt{saw}_2, \texttt{today}_5)$

Inference Problem: $\mathcal{Y}$ - all directed spanning trees.

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

Directed Spanning Tree

Second-Order Model (McDonald et al., 2005)

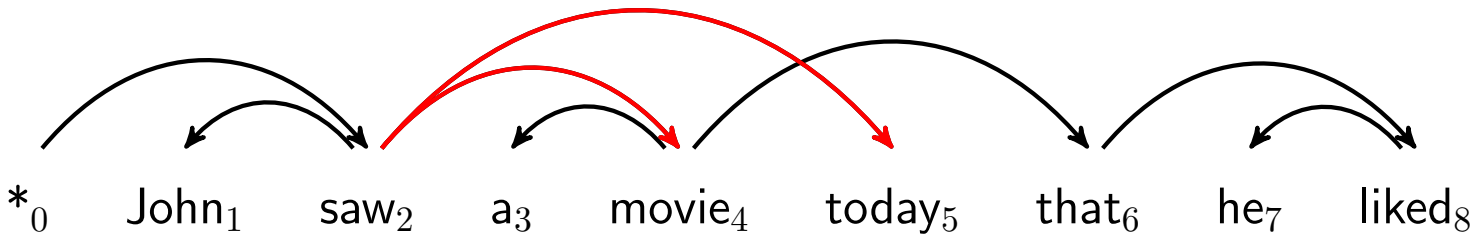


$$f(y) = \mathit{score}(*_0 \rightarrow [\texttt{NULL}]\ \texttt{saw}_2) + \mathit{score}(\texttt{saw}_2 \rightarrow [\texttt{NULL}]\ \texttt{movie}_4) + \mathit{score}(\texttt{saw}_4 \rightarrow [\texttt{movie}_4]\ \texttt{today}_5) + \ldots$$

where $\mathit{score}(\texttt{saw}_4 \rightarrow [\texttt{movie}_4]\ \texttt{today}_5) = \log p(\texttt{today}_5 | \texttt{saw}_2, \texttt{movie}_4)$

or $\mathit{score}(\texttt{saw}_4 \rightarrow [\texttt{movie}_4]\ \texttt{today}_5) = w \cdot \phi(\texttt{saw}_2, \texttt{movie}_4, \texttt{today}_5)$

Second-Order Model (McDonald et al., 2005)



$$f(y) = \textit{score}(*_0 \rightarrow [\texttt{NULL}]\ \texttt{saw}_2) + \textit{score}(\texttt{saw}_2 \rightarrow [\texttt{NULL}]\ \texttt{movie}_4) + \textit{score}(\texttt{saw}_4 \rightarrow [\texttt{movie}_4]\ \texttt{today}_5) + \ldots$$

where $\textit{score}(\texttt{saw}_4 \rightarrow [\texttt{movie}_4]\ \texttt{today}_5) = \log p(\texttt{today}_5 | \texttt{saw}_2, \texttt{movie}_4)$

or $\textit{score}(\texttt{saw}_4 \rightarrow [\texttt{movie}_4]\ \texttt{today}_5) = w \cdot \phi(\texttt{saw}_2, \texttt{movie}_4, \texttt{today}_5)$

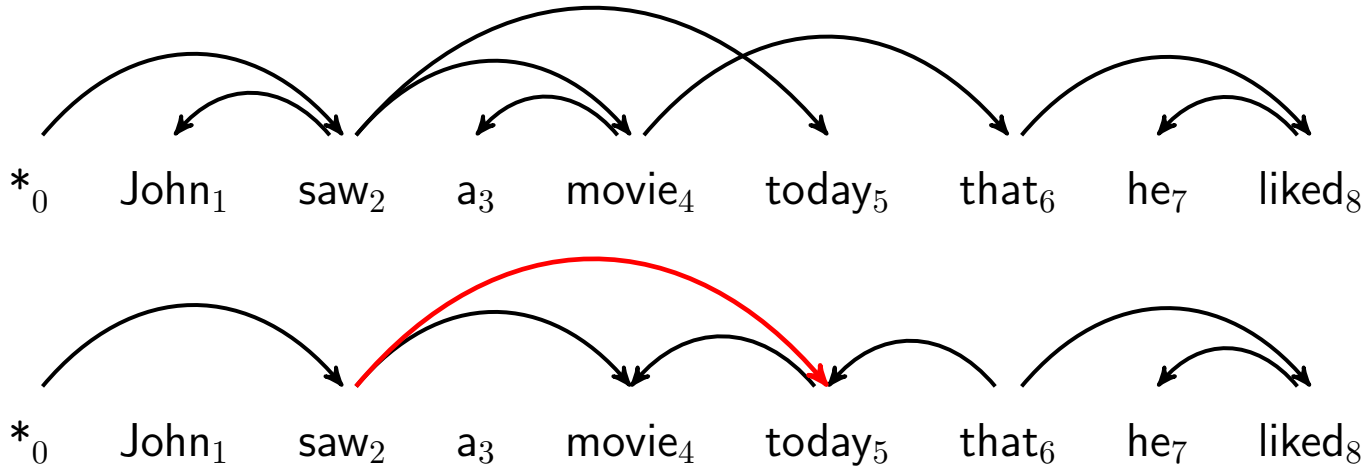
Inference Problem: $\mathcal{Y}$ - all directed spanning trees.

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

NP-Hard

Relaxed Set: Directed Subgraphs

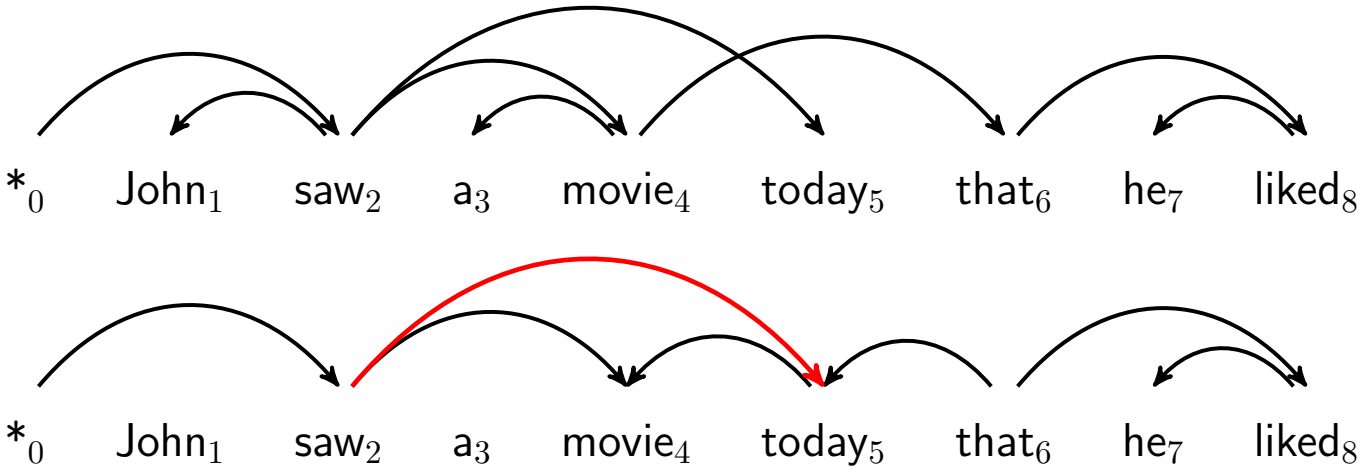
Define $\mathcal{Z}$ as all directed subgraphs, such as



$$\mathcal{Y} \subset \mathcal{Z}$$

Relaxed Set: Directed Subgraphs

Define $\mathcal{Z}$ as all directed subgraphs, such as



$$\mathcal{Y} \subset \mathcal{Z}$$

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

NP-Hard

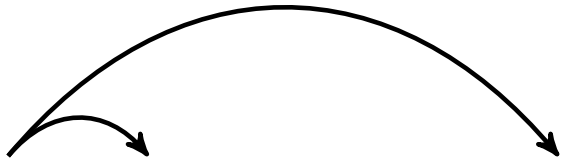
$$\operatorname*{argmax}_{z \in \mathcal{Z}} f(z)$$

Easy to compute

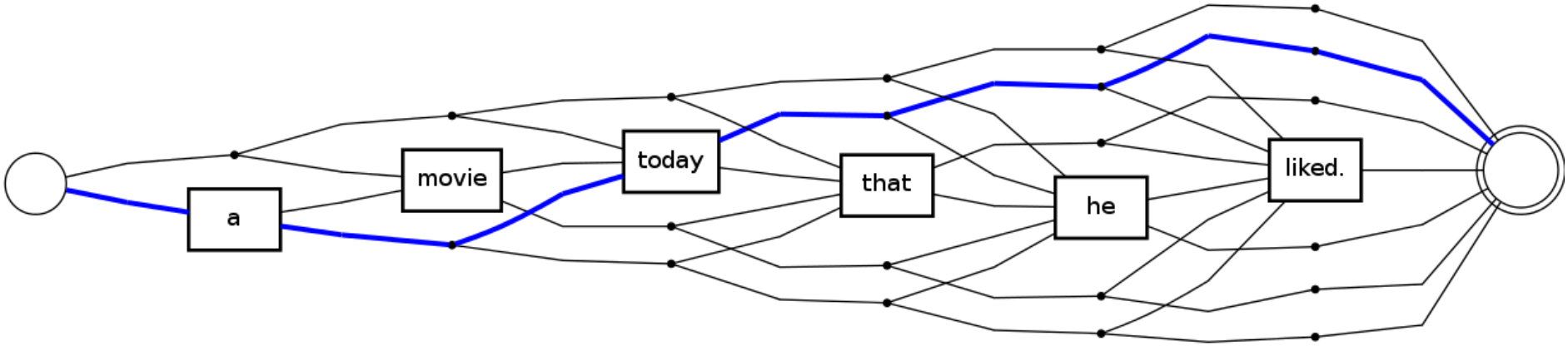
Individual Inference Algorithm

$$\operatorname*{argmax}_{z \in \mathcal{Z}} f(z)$$

Use dynamic programming to find arcs on each side of each word.



saw$_2$ a$_3$ movie$_4$ today$_5$ that$_6$ he$_7$ liked$_8$

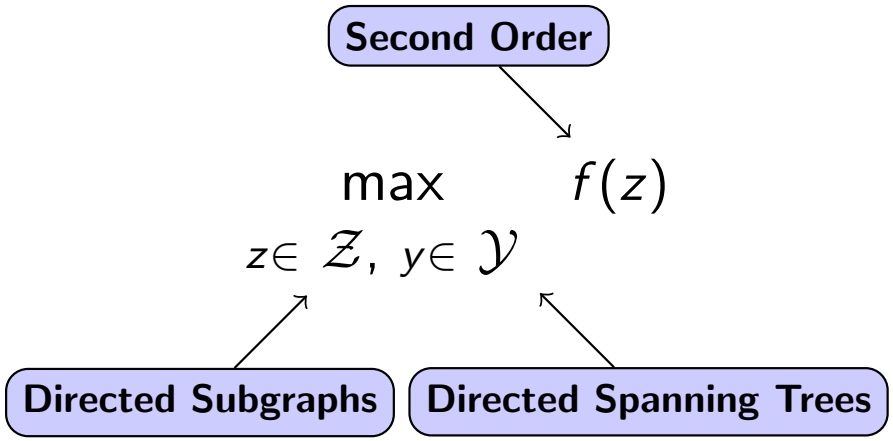


$$\textit{score}(\texttt{saw}_2 \rightarrow \texttt{[NULL]}\ \texttt{a}_3) + \textit{score}(\texttt{saw}_2 \rightarrow \texttt{[a}_3\texttt{]}\ \texttt{today}_5)$$

Reformulation of the Inference Problem

$$\max_{y \in \mathcal{Y}} f(y)$$

Equivalent to



such that for all arcs i, j,

$$z_{ij} = y_{ij}$$

where $y_{ij} = 1$ if arc (i, j) appears in parse y.

Deriving the Algorithm

Inference Problem:

$$\max_{z \in \mathcal{Z}, y \in \mathcal{Y}} f(z) \text{ such that } z_{ij} = y_{ij} \text{ for all arcs } i, j$$

Relaxed Problem:

$$L(\lambda) = \max_{z \in \mathcal{Z}, y \in \mathcal{Y}} f(z) - \sum_{i,j} \lambda_{ij} (z_{ij} - y_{ij})$$

$$= \max_{z \in \mathcal{Z}} \left(f(z) - \sum_{i,j} \lambda_{ij} z_{ij} \right) + \max_{y \in \mathcal{Y}} \left(\sum_{i,j} \lambda_{ij} y_{ij} \right)$$

Individual Inference **Max Directed Spanning Tree**

Deriving the Algorithm

$$L(\lambda) = \max_{z \in \mathcal{Z}} \left(f(z) - \sum_{i,j} \lambda_{ij} z_{ij} \right) + \max_{y \in \mathcal{Y}} \left(\sum_{i,j} \lambda_{ij} y_{ij} \right)$$

Individual Inference **Max Directed Spanning Tree**

Weak Duality: For any λ, $L(\lambda)$ upper bounds the optimal score, i.e.

$$L(\lambda) \geq \max_{y \in \mathcal{Y}} f(y)$$

.

Furthermore if structures are the same, then this upper bound will be tight.

Idea: Iteratively minimize the upper bound $L(\lambda)$.

$$L(\lambda^{(t)}) = \max_{z \in \mathcal{Z}} \left(f(z) - \sum_{i,j} \lambda_{ij}^{(t)} z_{ij} \right) + \max_{y \in \mathcal{Y}} \left(\sum_{i,j} \lambda_{ij}^{(t)} y_{ij} \right)$$

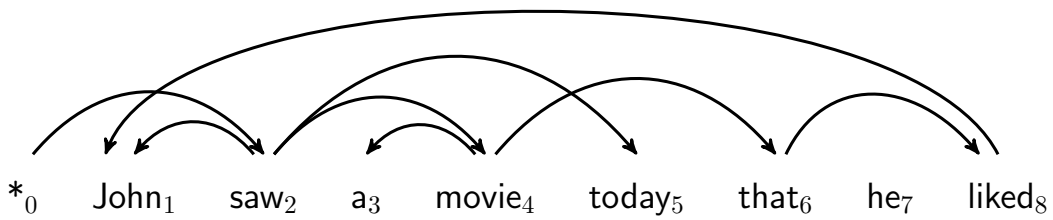
For t from 1 to T,

1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t (z_{ij}^{(t)} - y_{ij}^{(t)})$.

For t from 1 to T,

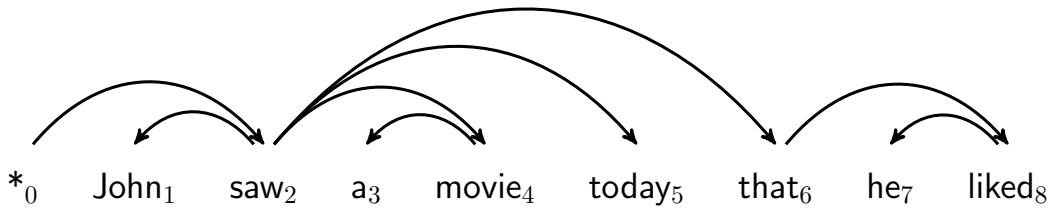
1. **Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.**
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



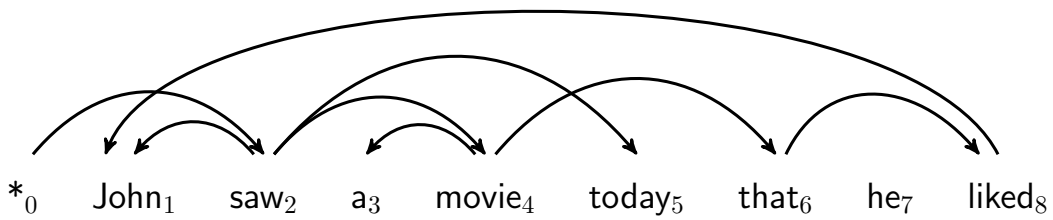
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(1)}$
$\texttt{saw}_2 \rightarrow \texttt{that}_6$	0
$\texttt{liked}_8 \rightarrow \texttt{he}_7$	0
$\texttt{movie}_4 \rightarrow \texttt{that}_6$	0
$\texttt{liked}_8 \rightarrow \texttt{John}_1$	0
...	0

For t from 1 to T,

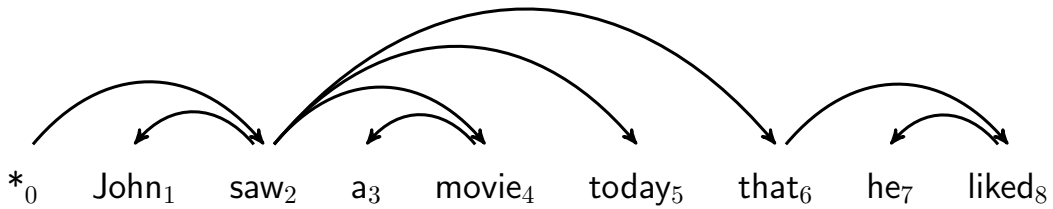
1. **Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.**
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



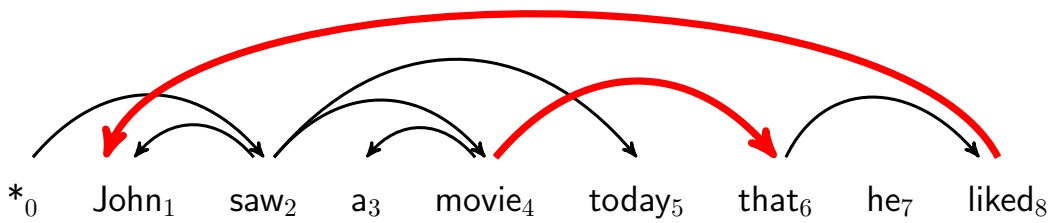
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(1)}$
$\texttt{saw}_2 \rightarrow \texttt{that}_6$	0
$\texttt{liked}_8 \rightarrow \texttt{he}_7$	0
$\texttt{movie}_4 \rightarrow \texttt{that}_6$	0
$\texttt{liked}_8 \rightarrow \texttt{John}_1$	0
...	0

For t from 1 to T,

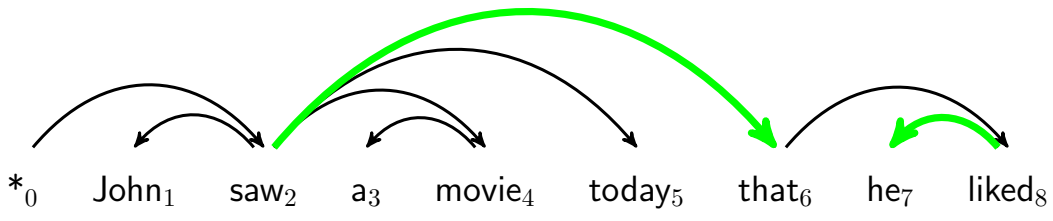
1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. **If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.**
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



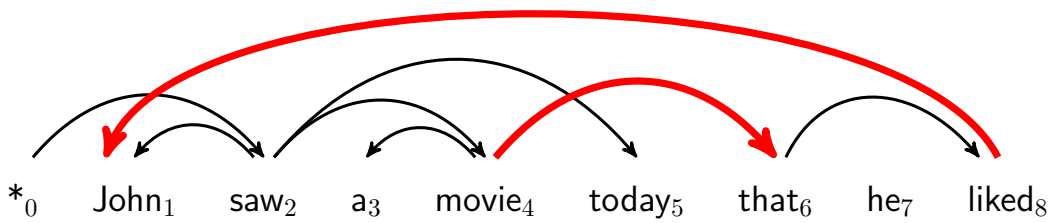
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(1)}$
saw$_2$ $\rightarrow$ that$_6$	0
liked$_8$ $\rightarrow$ he$_7$	0
movie$_4$ $\rightarrow$ that$_6$	0
liked$_8$ $\rightarrow$ John$_1$	0
...	0

For t from 1 to T,

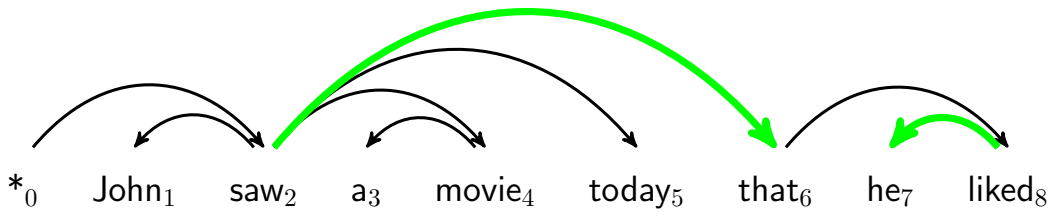
1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. **Update multipliers** $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



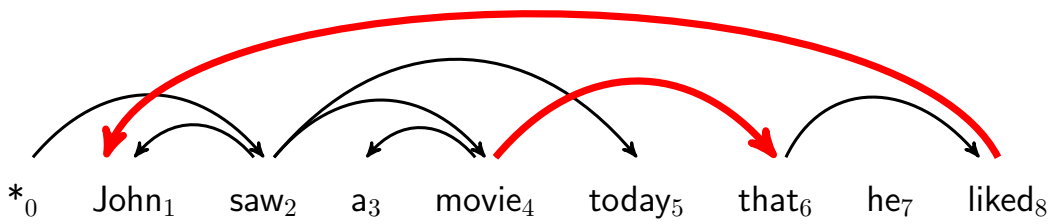
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(2)}$
$\texttt{saw}_2 \rightarrow \texttt{that}_6$	1
$\texttt{liked}_8 \rightarrow \texttt{he}_7$	1
$\texttt{movie}_4 \rightarrow \texttt{that}_6$	-1
$\texttt{liked}_8 \rightarrow \texttt{John}_1$	-1
...	0

For t from 1 to T,

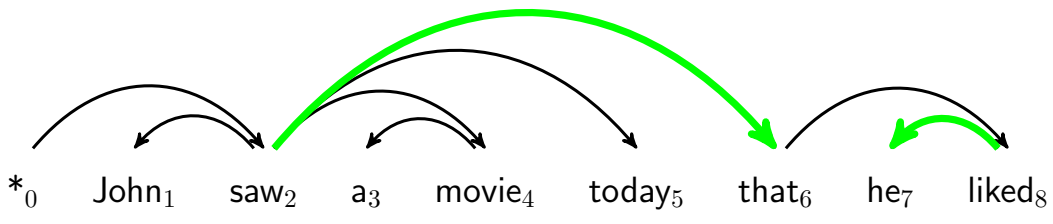
1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



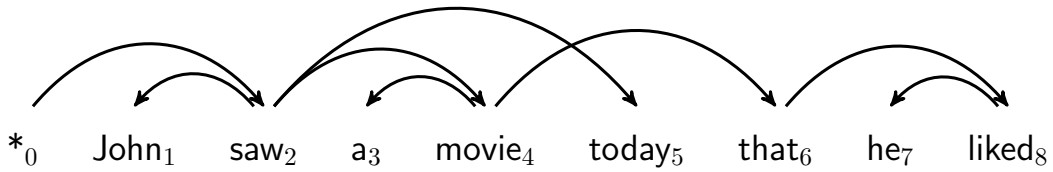
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(2)}$
saw$_2$ $\rightarrow$ that$_6$	1
liked$_8$ $\rightarrow$ he$_7$	1
movie$_4$ $\rightarrow$ that$_6$	-1
liked$_8$ $\rightarrow$ John$_1$	-1
...	0

For t from 1 to T,

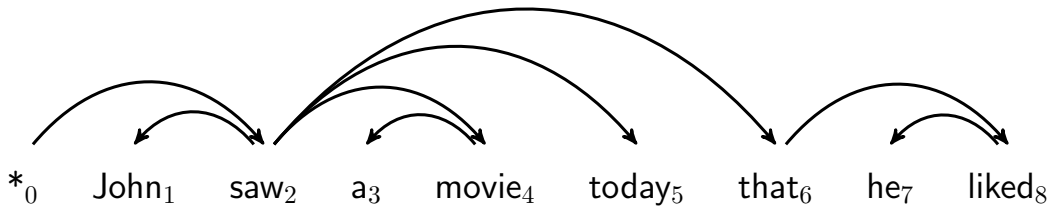
1. **Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.**
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



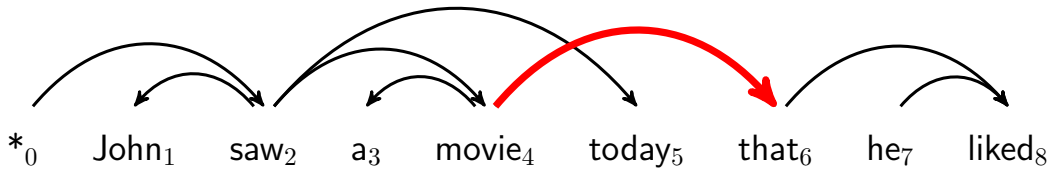
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(2)}$
saw$_2$ $\rightarrow$ that$_6$	1
liked$_8$ $\rightarrow$ he$_7$	1
movie$_4$ $\rightarrow$ that$_6$	-1
liked$_8$ $\rightarrow$ John$_1$	-1
...	0

For t from 1 to T,

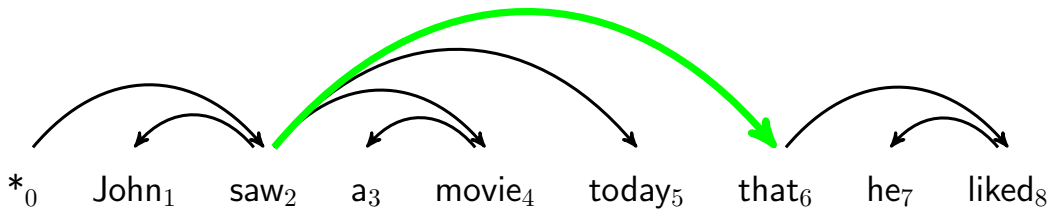
1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. **If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.**
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



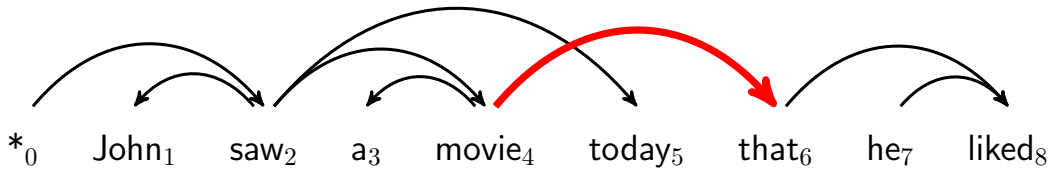
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(2)}$
saw$_2$ $\rightarrow$ that$_6$	1
liked$_8$ $\rightarrow$ he$_7$	1
movie$_4$ $\rightarrow$ that$_6$	-1
liked$_8$ $\rightarrow$ John$_1$	-1
...	0

For t from 1 to T,

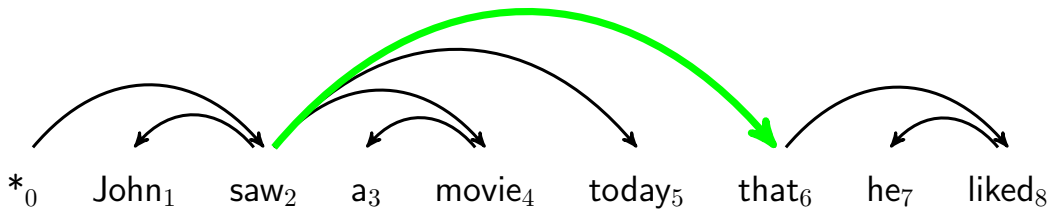
1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. **Update multipliers** $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



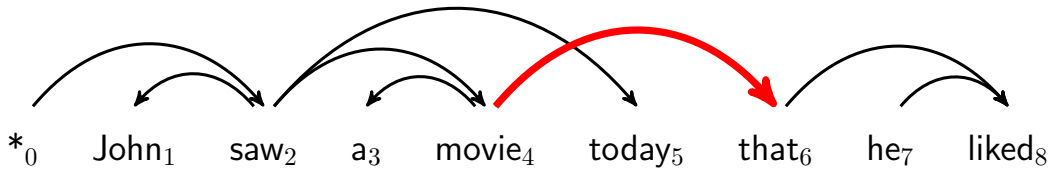
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(3)}$
$\texttt{saw}_2 \rightarrow \texttt{that}_6$	2
$\texttt{liked}_8 \rightarrow \texttt{he}_7$	1
$\texttt{movie}_4 \rightarrow \texttt{that}_6$	-2
$\texttt{liked}_8 \rightarrow \texttt{John}_1$	-1
...	0

For t from 1 to T,

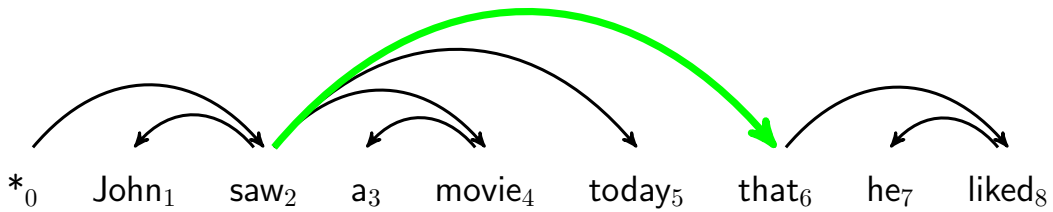
1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



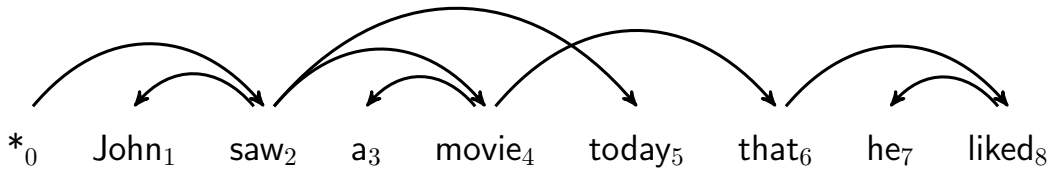
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(3)}$
$\texttt{saw}_2 \rightarrow \texttt{that}_6$	2
$\texttt{liked}_8 \rightarrow \texttt{he}_7$	1
$\texttt{movie}_4 \rightarrow \texttt{that}_6$	-2
$\texttt{liked}_8 \rightarrow \texttt{John}_1$	-1
...	0

For t from 1 to T,

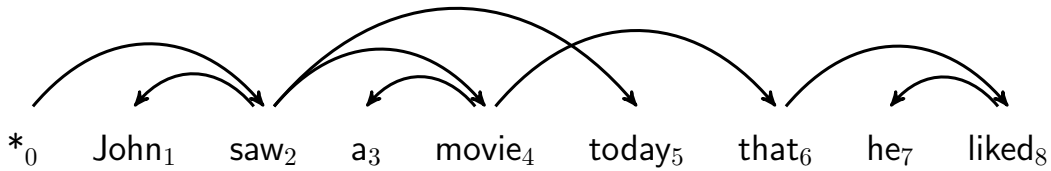
1. **Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.**
2. If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Individual Inference



$$z^{(t)} \leftarrow \text{argmax}_{z \in \mathcal{Z}} f(z) - \sum_{i,j} \lambda_{ij} z_{ij}$$

Max Directed Spanning Tree



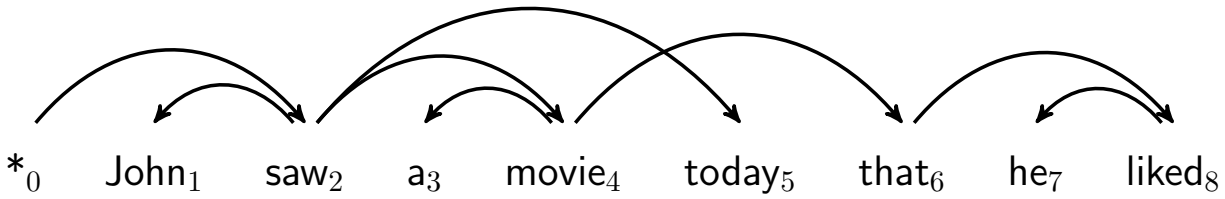
$$y^{(t)} \leftarrow \text{argmax}_{y \in \mathcal{Y}} \sum_{i,j} \lambda_{ij} y_{ij}$$

Arcs	$\lambda_{ij}^{(3)}$
saw$_2$ $\rightarrow$ that$_6$	2
liked$_8$ $\rightarrow$ he$_7$	1
movie$_4$ $\rightarrow$ that$_6$	-2
liked$_8$ $\rightarrow$ John$_1$	-1
...	0

For t from 1 to T,

1. Compute $z^{(t)}$ and $y^{(t)}$ for $\lambda^{(t)}$.
2. **If $z^{(t)}$ and $y^{(t)}$ are the same, return $z^{(t)}$.**
3. Update multipliers $\lambda_{ij}^{(t+1)} \leftarrow \lambda_{ij}^{(t)} - \alpha_t(z_{ij}^{(t)} - y_{ij}^{(t)})$.

Optimal Parse



$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

Arcs	$\lambda_{ij}^{(3)}$
saw$_2$ $\rightarrow$ that$_6$	2
liked$_8$ $\rightarrow$ he$_7$	1
movie$_4$ $\rightarrow$ that$_6$	-2
liked$_8$ $\rightarrow$ John$_1$	-1
...	0

Formal Guarantees of Lagrangian Relaxation

Theorem (Subgradient Descent)

With an appropriate rate sequence $\alpha_1, \alpha_2, \alpha_3, \ldots,$

$$\lim_{t \to \infty} L(\lambda^{(t)}) = \min_{\lambda} L(\lambda)$$

Formal Guarantees of Lagrangian Relaxation

Theorem (Subgradient Descent)

With an appropriate rate sequence $\alpha_1, \alpha_2, \alpha_3, \ldots,$

$$\lim_{t \to \infty} L(\lambda^{(t)}) = \min_{\lambda} L(\lambda)$$

Theorem (Certificate of Optimality)

If the algorithm finds $(y^{(t)}, z^{(t)})$ *with the same tree then,*

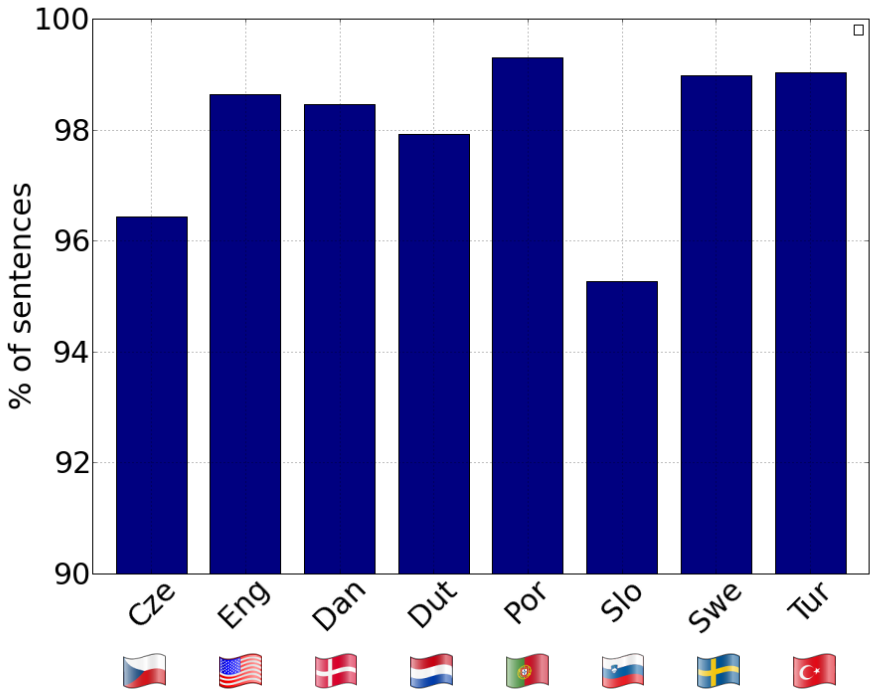
$$f(z^{(t)}) = \max_{y \in \mathcal{Y}} f(y)$$

Otherwise return best dependency parse seen,

$$\operatorname*{argmax}_{y \in y^{(1)}, y^{(2)}, \ldots, y^{(T)}} f(y)$$

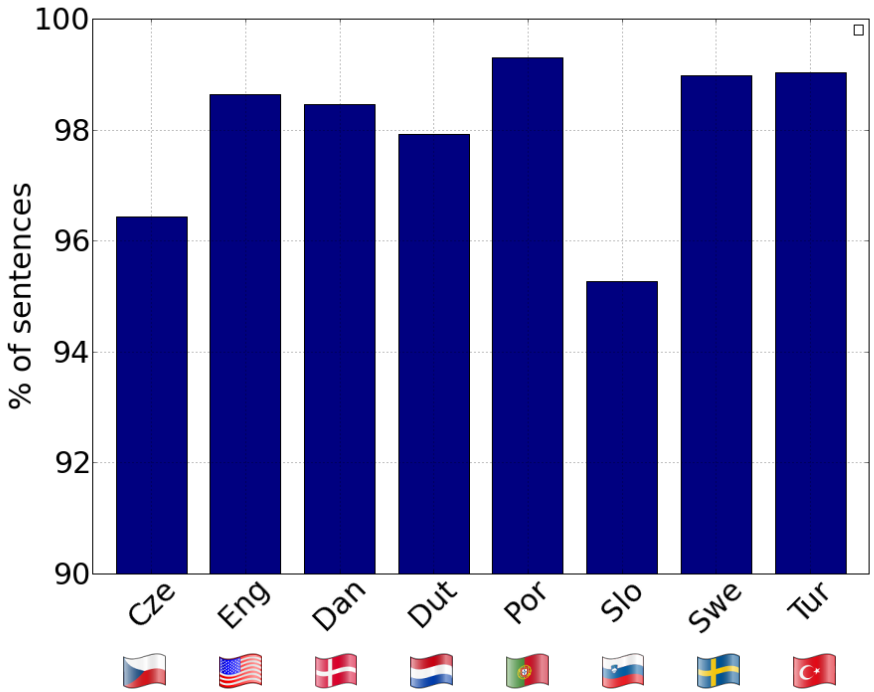
Dependency Parsing Results

Optimal Solutions (with Certificate)



Dependency Parsing Results

Optimal Solutions (with Certificate)



Parsing Accuracy

Results in terms of UAS (% arcs correct)

	First Order	Prev Best	Model
Dan	89.7	91.5	**91.8**
Dut	82.3	85.6	**85.8**
Por	90.7	92.1	**93.0**
Slo	82.4	85.6	**86.2**
Swe	88.9	90.6	**91.4**
Tur	75.7	76.4	**77.6**
Cze	84.4	—	**87.3**
Eng	90.1	—	**92.5**

Prev Best includes:

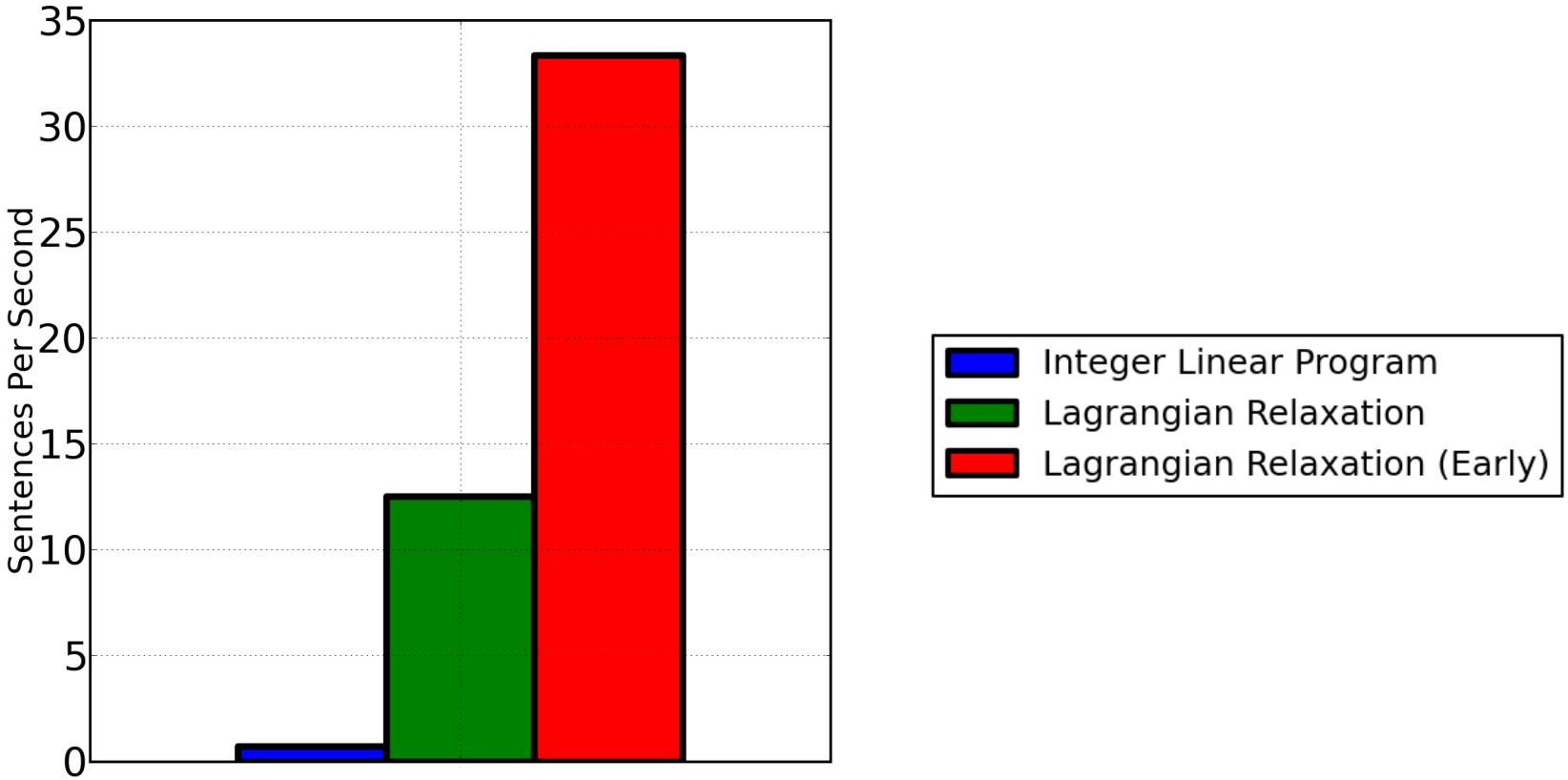
Local Search (McDonald and Pereira, 2006)

Belief Propagation (Smith and Eisner, 2008)

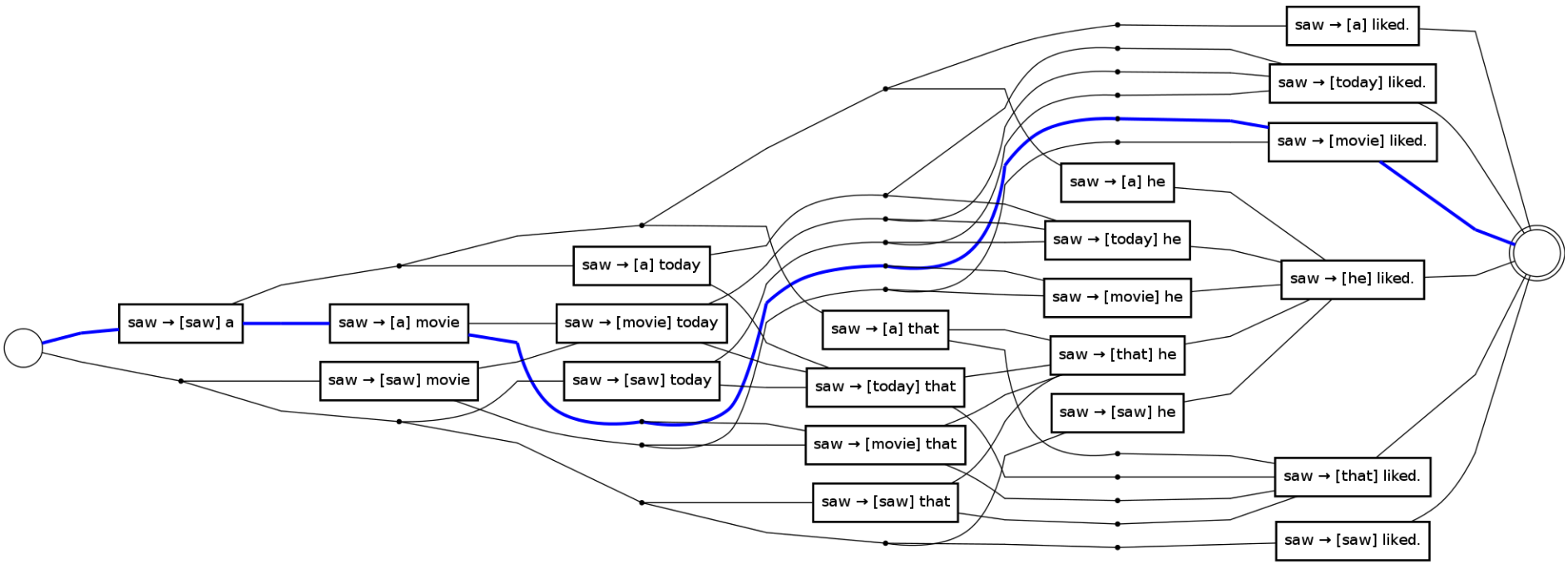
Linear Programming (Martins et al., 2009)

Speed Results for English

	ILP	LR	LR-Early
Accuracy	**92.7**	**92.7**	**92.7**
Speed	0.69	12.5 (18x)	33.3 (48x)



Extension to Head Automata Models (Alshawi, 1996)



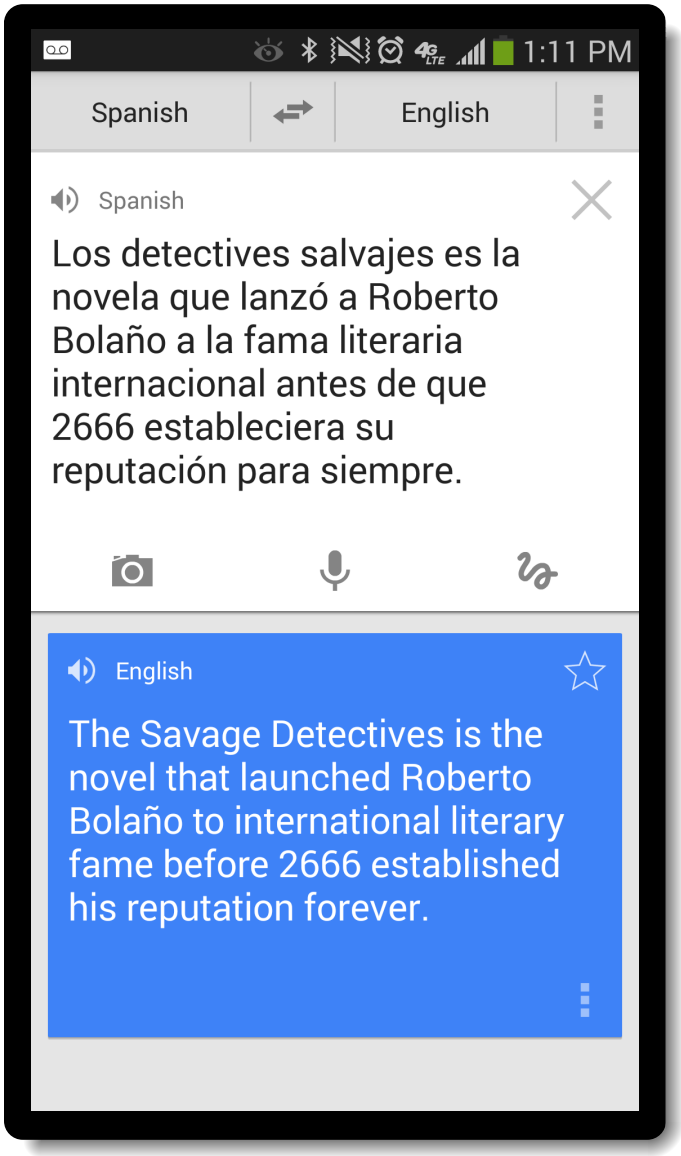
An important and widely-used generalization of dependency parsing:

- Latent-Variable Parsing Models (Balle et al., 2013)
- Parsing over Word Lattices (Alshawi, 1996)
- Tree-Adjoining Grammars (Carreras and Collins, 2009)

Overview

1. Lagrangian Relaxation for Dependency Parsing
2. Lagrangian Relaxation for Syntax-Based Translation
3. Future Work

Machine Translation



Origins of Statistical Machine Translation

. . . one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

Letter from Warren Weaver to Norbert Weiner, 1947

$$\underset{y \in \mathcal{Y}}{\text{argmax}}\, f(y)$$

Decoding

Modern Statistical Translation

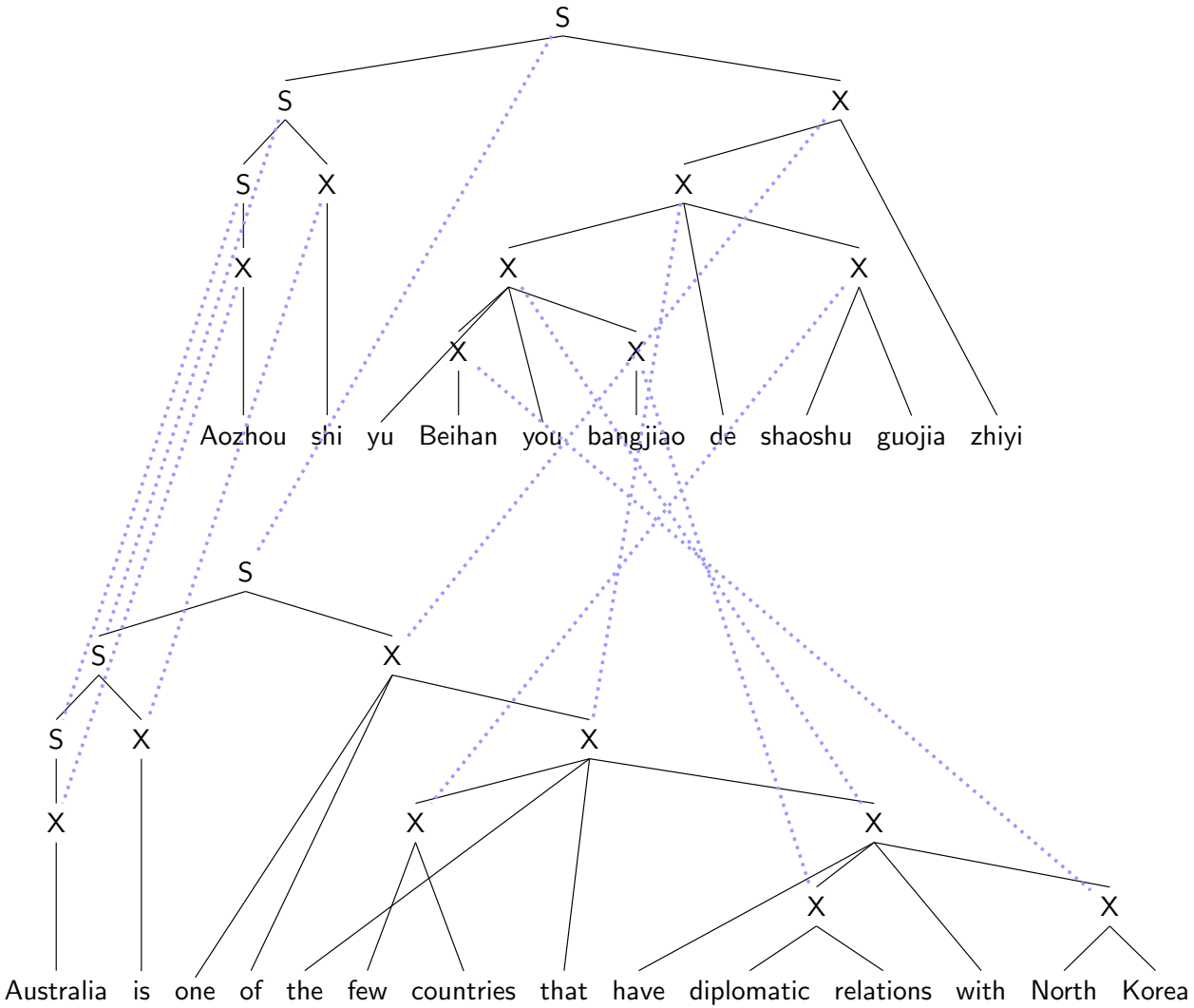
Translation systems are trained with a vast amount of data,

- Training uses 2.5 billion parallel documents.
- Language model trained with 500 billion English words.

Google Translate has used a statistical system since 2006

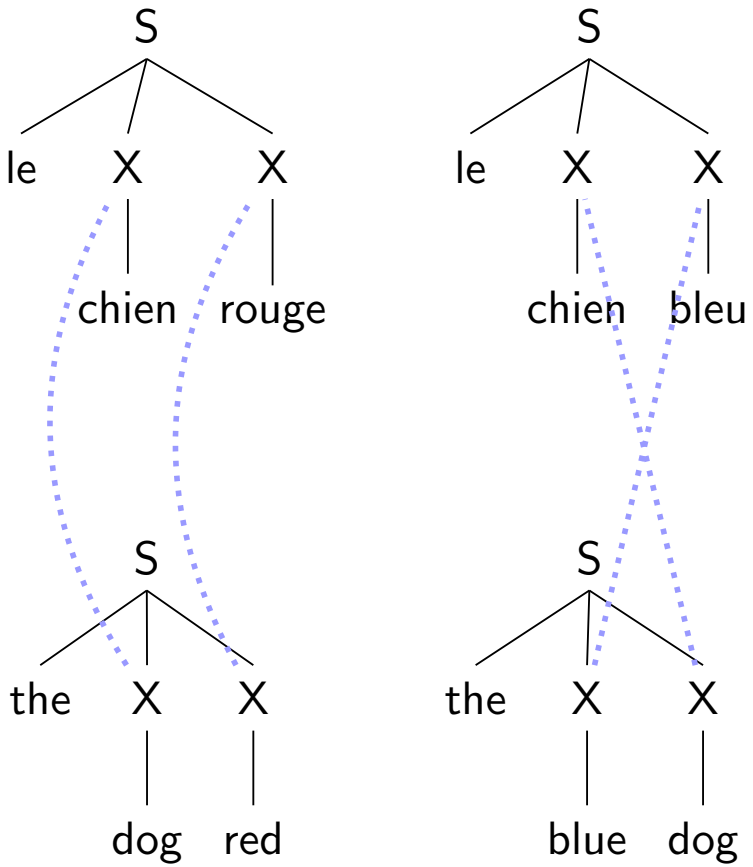
- 80 languages
- 200 million users a month
- Over 10 billion words translated a day

Syntax-Based Machine Translation



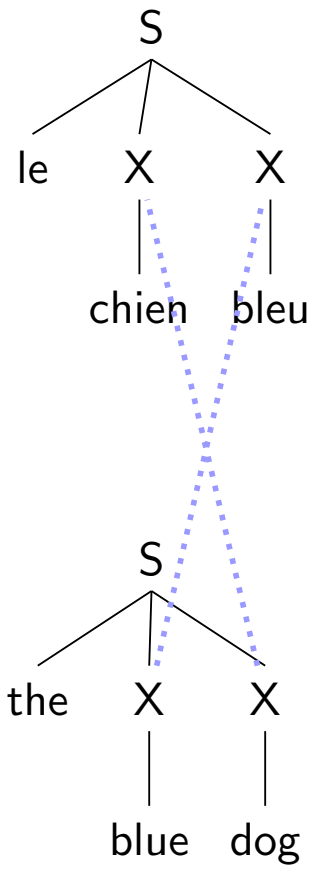
Synchronous Context-Free Grammar (SCFG) (Chiang, 2005)

#	Synchronous Rule
1	S $\rightarrow$ le $X_{\boxed{1}}$ $X_{\boxed{2}}$, the $X_{\boxed{1}}$ $X_{\boxed{2}}$
2	S $\rightarrow$ le $X_{\boxed{1}}$ $X_{\boxed{2}}$, the $X_{\boxed{2}}$ $X_{\boxed{1}}$
3	X $\rightarrow$ chien, dog
4	X $\rightarrow$ bleu, blue
5	X $\rightarrow$ rouge, red



Weighted Synchronous CFG

#	Synchronous Rule	w
1	S $\rightarrow$ le $X_{\boxed{1}}$ $X_{\boxed{2}}$, the $X_{\boxed{1}}$ $X_{\boxed{2}}$	3.0
2	S $\rightarrow$ le $X_{\boxed{1}}$ $X_{\boxed{2}}$, the $X_{\boxed{2}}$ $X_{\boxed{1}}$	1.0
3	X $\rightarrow$ chien, dog	2.0
4	X $\rightarrow$ bleu, blue	4.0
5	X $\rightarrow$ rouge, red	3.0



$$w(\text{S} \rightarrow \text{le } X_{\boxed{1}} \; X_{\boxed{2}}, \text{ the } X_{\boxed{2}} \; X_{\boxed{1}}) +$$
$$w(\text{X} \rightarrow \texttt{chien}, \texttt{dog}) + w(\text{X} \rightarrow \texttt{bleu}, \texttt{blue})$$

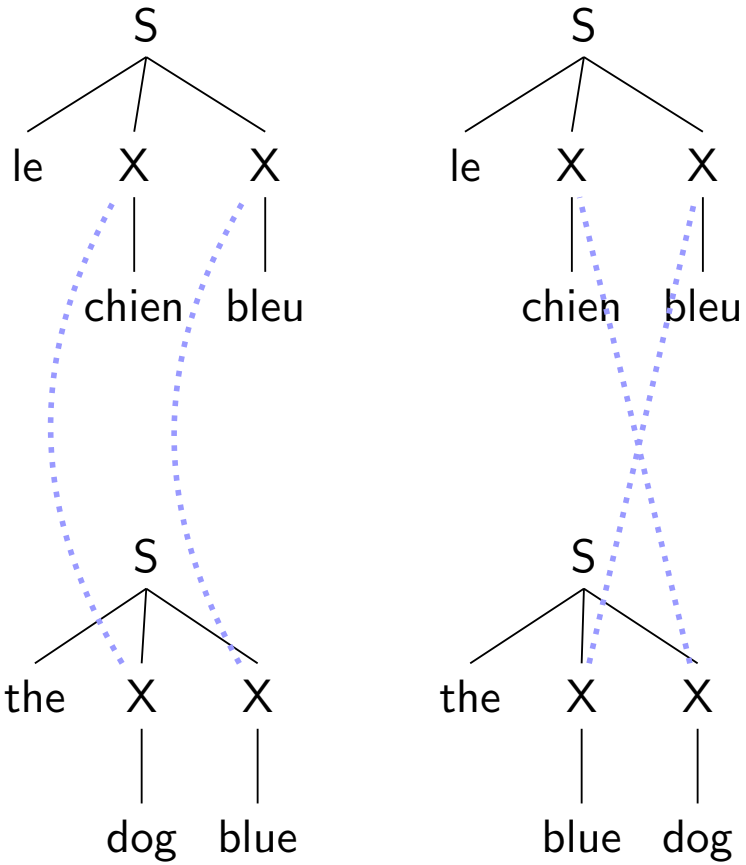
A Realistic Synchronous Grammar

Synchronous Rules				w
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ of researchers , $X_{[2]}$	-4.26
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ $X_{[2]}$ by a scientist	-4.84
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ $X_{[2]}$ of scientists	-1.10
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ scientists in $X_{[2]}$	-1.71
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ scientists $X_{[2]}$	-0.21
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[2]}$ of $X_{[1]}$ scientists	-1.10
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	scientists of $X_{[1]}$ generation $X_{[2]}$	-5.17
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	scientists $X_{[2]}$ $X_{[1]}$	-0.21
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ plant scientist $X_{[2]}$	-3.43
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	$X_{[1]}$ scientists , $X_{[2]}$	-1.69
X	→	$X_{[1]}$ 科学家 $X_{[2]}$,	scientists at the $X_{[1]}$ $X_{[2]}$	-3.48
X	→	$X_{[1]}$ 初期 $X_{[2]}$,	early period after the $X_{[1]}$ $X_{[2]}$	-4.01
X	→	$X_{[1]}$ 初期 $X_{[2]}$,	$X_{[2]}$ during the early days of the $X_{[1]}$	-7.01
X	→	$X_{[1]}$ 初期 $X_{[2]}$,	initial stage of $X_{[1]}$ $X_{[2]}$	-3.21
X	→	$X_{[1]}$ 初期 $X_{[2]}$,	first few days after the $X_{[1]}$ $X_{[2]}$	-7.42
X	→	$X_{[1]}$ 染色体 $X_{[2]}$,	$X_{[2]}$ $X_{[1]}$ chromosomes	-0.39
X	→	$X_{[1]}$ 染色体 $X_{[2]}$,	$X_{[1]}$ of chromosomes $X_{[2]}$	-1.28
X	→	$X_{[1]}$ 染色体 $X_{[2]}$,	$X_{[1]}$ chromosome $X_{[2]}$	-0.53
		⋮		

Translation with a Synchronous Grammar

Consider all synchronous derivations that produce an input sentence.

Input: `le chien bleu`



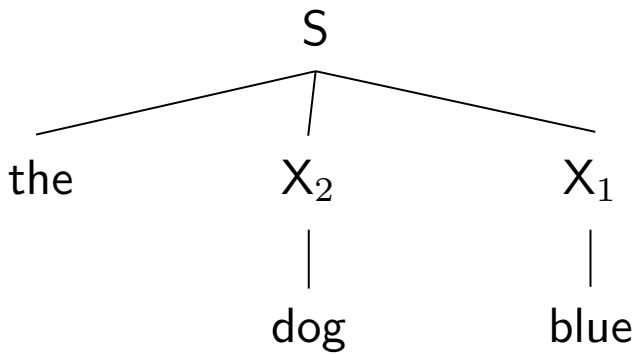
Translation with a Synchronous Grammar

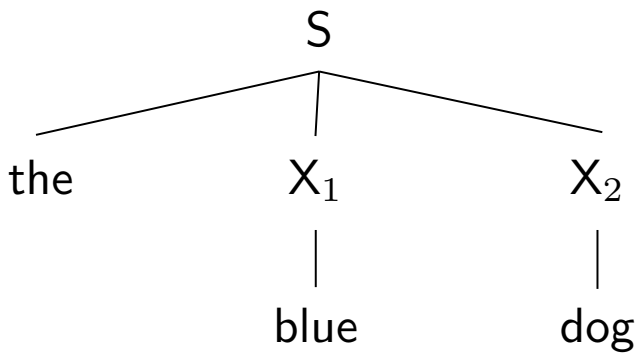
Consider all synchronous derivations that produce an input sentence.

Input: `le chien bleu`

Translation Forest

Forest Rule	w
$S \rightarrow$ the X_2 X_1	3.0
$S \rightarrow$ the X_1 X_2	1.0
$X_1 \rightarrow$ blue	4.0
$X_2 \rightarrow$ dog	2.0

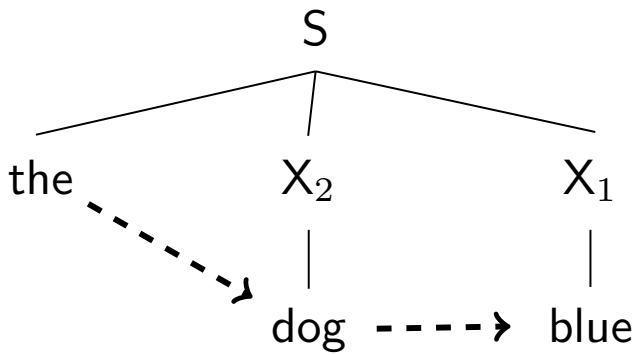




Incorporating a Language Model

Bigram	w_{lm}
the blue	-1.0
the dog	-2.0
dog blue	-5.0
blue dog	-1.0

$$w_{lm}(the, dog) = \log p(dog|the)$$



$$f(y) = w(\mathtt{S} \rightarrow \mathtt{the\ X_2\ X_1}) + w(\mathtt{X_2} \rightarrow \mathtt{dog}) + w(\mathtt{X_1} \rightarrow \mathtt{blue}) + w_{lm}(\mathtt{the,\ dog}) + w_{lm}(\mathtt{dog,\ blue})$$

Translation Inference Problem

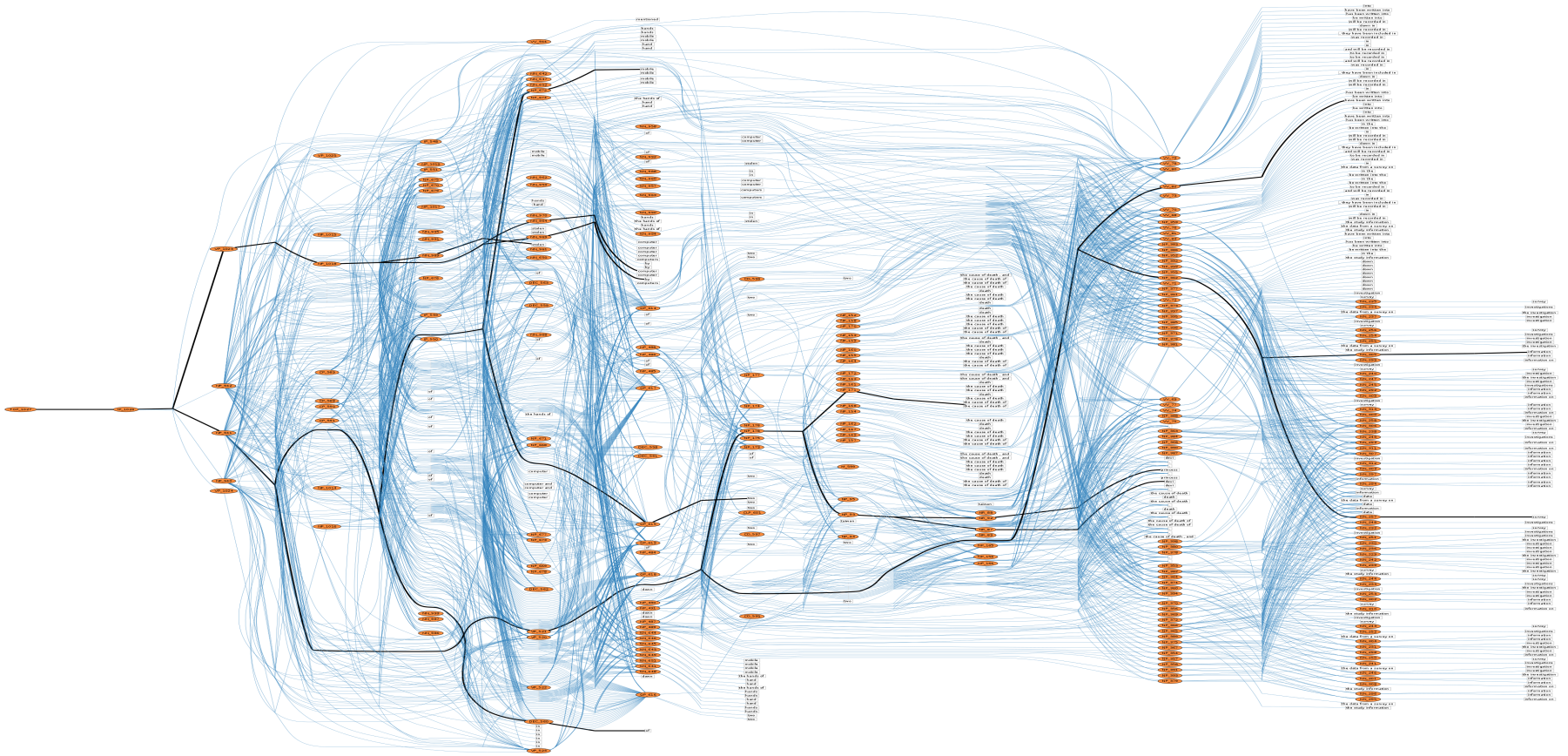
Define $\mathcal{Y}$ as all derivations in the translation forest.

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

Translation Inference Problem

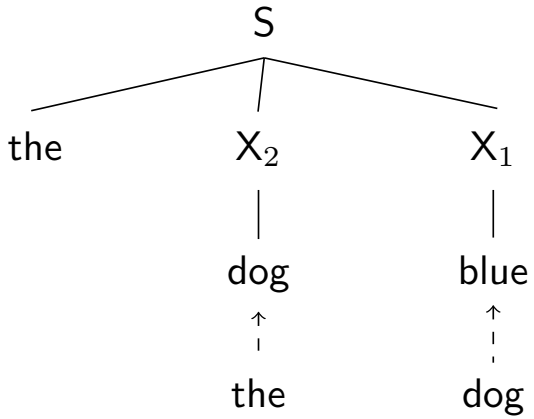
Define $\mathcal{Y}$ as all derivations in the translation forest.

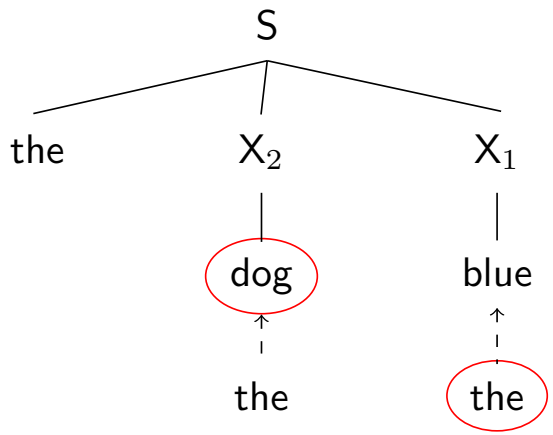
$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$



Relaxed Set: Augmented Derivations

Let $\mathcal{Z}$ be derivations augmented with any previous word, $\mathcal{Y} \subset \mathcal{Z}$



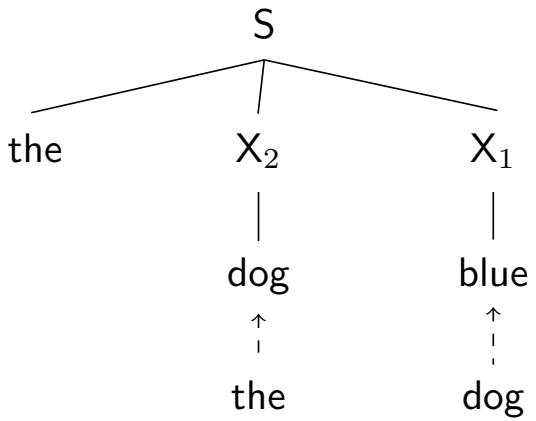


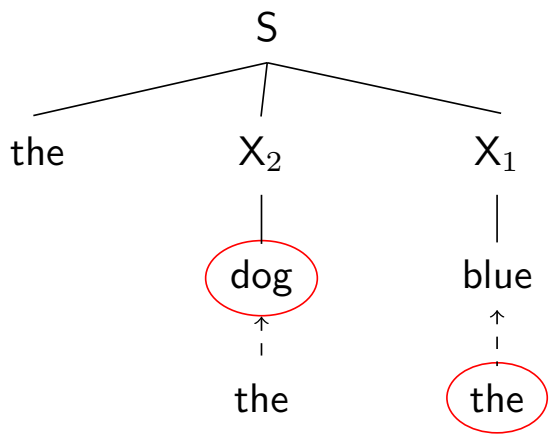
$$f(z) = w(\mathtt{S} \rightarrow \mathtt{the\ X_2\ X_1}) + w(\mathtt{X_2} \rightarrow \mathtt{dog}) + w(\mathtt{X_1} \rightarrow \mathtt{blue}) + w_{lm}(\mathtt{the}, \mathtt{dog}) + w_{lm}(\mathtt{dog}, \mathtt{blue})$$

$$f(z) = w(\mathtt{S} \rightarrow \mathtt{the\ X_2\ X_1}) + w(\mathtt{X_2} \rightarrow \mathtt{dog}) + w(\mathtt{X_1} \rightarrow \mathtt{blue}) + w_{lm}(\mathtt{the}, \mathtt{dog}) + w_{lm}(\mathtt{the}, \mathtt{blue})$$

Relaxed Set: Augmented Derivations

Let $\mathcal{Z}$ be derivations augmented with any previous word, $\mathcal{Y} \subset \mathcal{Z}$





$$f(z) = w(\mathtt{S} \rightarrow \mathtt{the\ X_2\ X_1}) + w(\mathtt{X_2} \rightarrow \mathtt{dog}) + w(\mathtt{X_1} \rightarrow \mathtt{blue}) + w_{lm}(\mathtt{the}, \mathtt{dog}) + w_{lm}(\mathtt{dog}, \mathtt{blue})$$

$$\operatorname*{argmax}_{y \in \mathcal{Y}} f(y)$$

Challenging

$$f(z) = w(\mathtt{S} \rightarrow \mathtt{the\ X_2\ X_1}) + w(\mathtt{X_2} \rightarrow \mathtt{dog}) + w(\mathtt{X_1} \rightarrow \mathtt{blue}) + w_{lm}(\mathtt{the}, \mathtt{dog}) + w_{lm}(\mathtt{dog}, \mathtt{blue})$$

$$\operatorname*{argmax}_{z \in \mathcal{Z}} f(z)$$

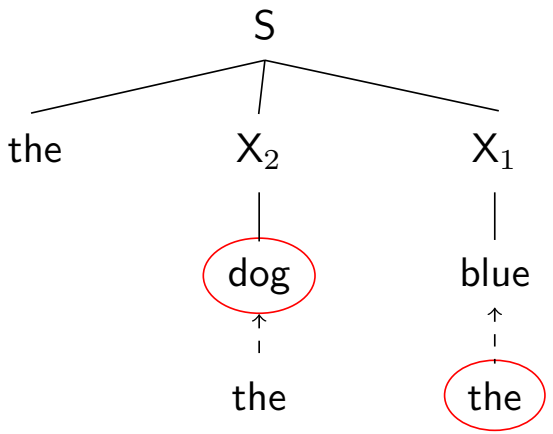
Easy to compute

Greedy Augmenting Algorithm

$$\underset{z \in \mathcal{Z}}{\operatorname{argmax}}\, f(z)$$

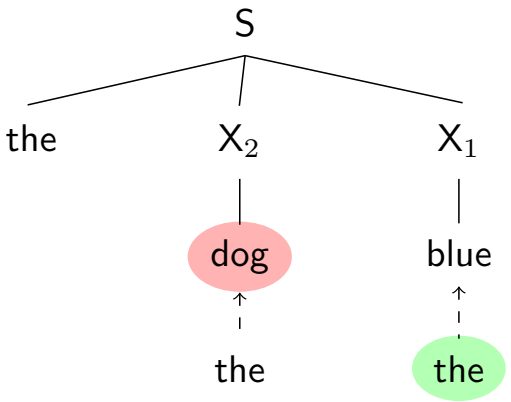
1. For each terminal select the best previous word in the language model.

2. Find highest-weight derivation using dynamic programming (CKY).



Lagrangian Relaxation Algorithm

Constraint: Augmented words must be consistent with the translation.



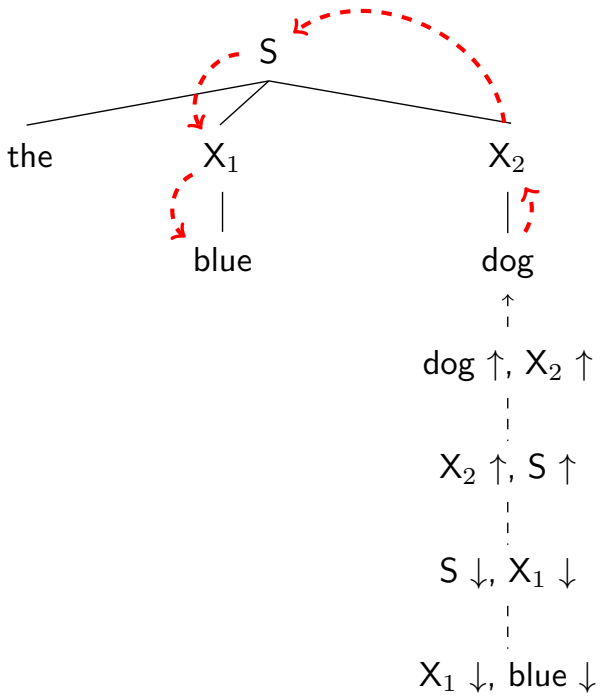
$$L(\lambda) = \max_{z \in \mathcal{Z}} \left(f(z) - \sum_u \lambda_u (z_u - \sum_v z_{u \to v}) \right)$$

For t from 1 to T,

1. Find the current highest-scoring augmented derivation.
2. If it is a consistent, return as optimal translation.
3. Otherwise, update multipliers based on inconsistencies.

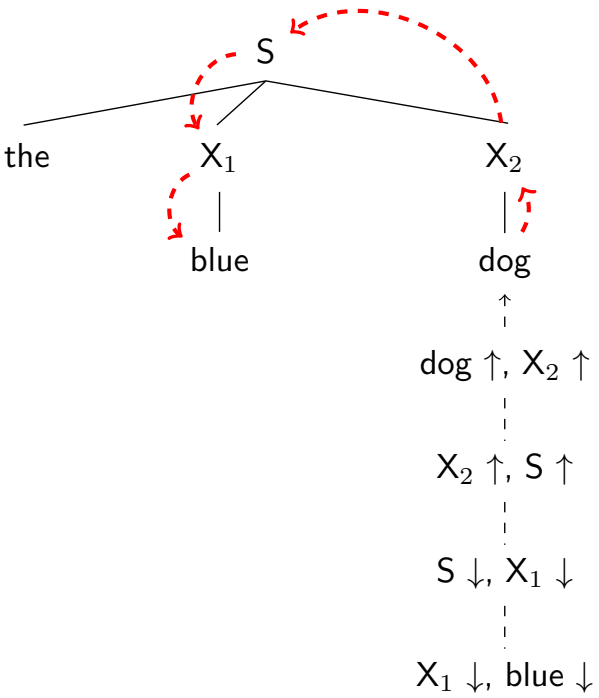
Extended Set and Inference Algorithm

Full Set $\mathcal{Z}$

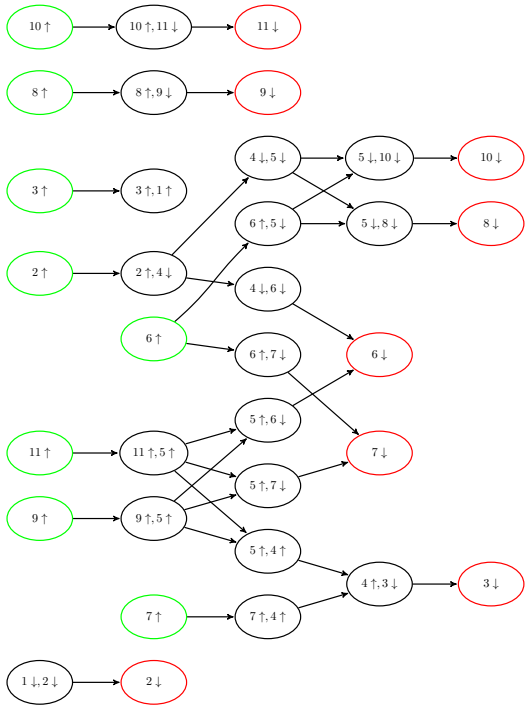


Extended Set and Inference Algorithm

Full Set $\mathcal{Z}$



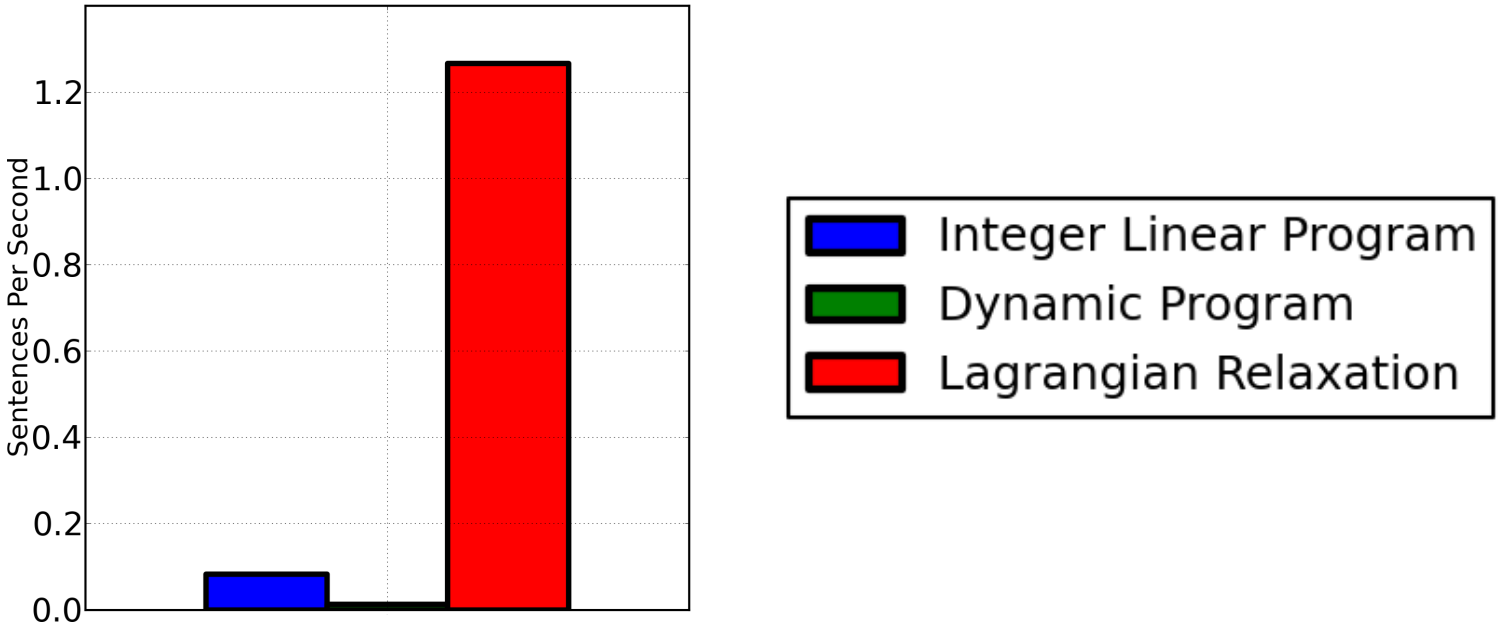
1. All-pairs shortest path.



2. Find highest-scoring derivation with dynamic programming.

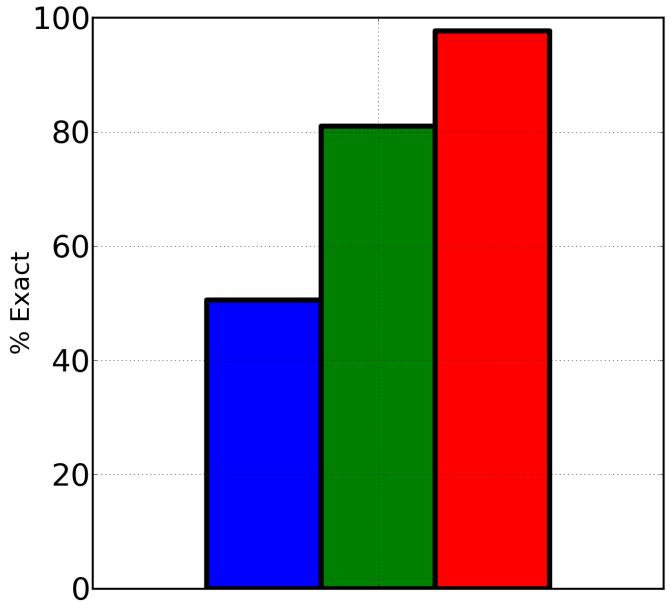
Syntax-Based Translation Results

Method	ILP	DP	LR
Optimal	100.0%	100.0%	97.8%
Speed	0.082	0.013	1.266 (15x, 98x)

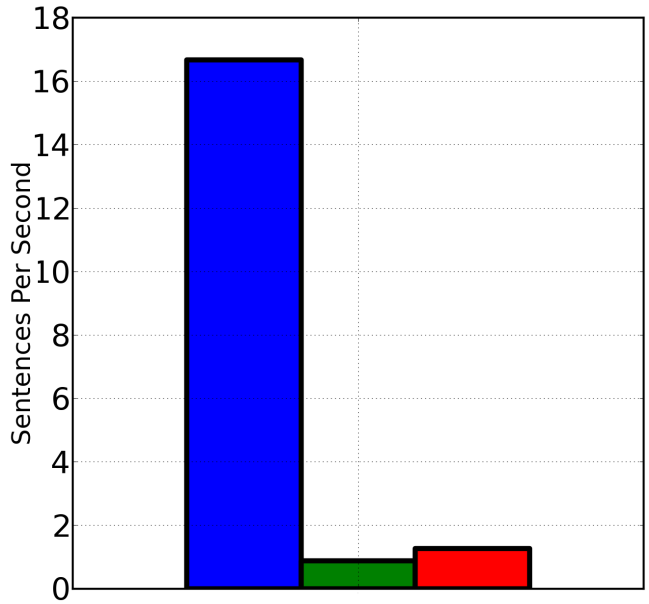


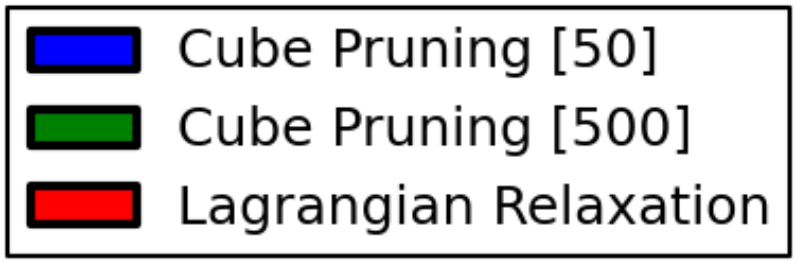
Syntax-Based Translation Results

Exact Solutions



Speed





Lagrangian Relaxation for Natural Language Inference

Simple

- Uses basic combinatorial algorithms.
- Can often utilize standard natural language solvers.

Efficient

- Faster than general-purpose solvers.
- Comparable to heuristic inference algorithms.

Strong Guarantees

- Gives a certificate of optimality when exact.
- Find a certificate on the vast majority of examples.

Impact

Many other published applications in NLP using Lagrangian Relaxation,

- Joint Parsing and Tagging (Rush et al., 2010)
- Bilingual Word Alignment (Denero and Macherey, 2010)
- Further Work on Dependency Parsing (Martins et al., 2011)
- Phrase-Based Machine Translation (Chang and Collins, 2011)
- CCG Supertagging (Auli and Lopez, 2011)
- Biomedical Event Extraction (Riedel and McCallum, 2011)
- Markov Logic Decomposition (Niu et al., 2011)
- Tagging with Global Constraints (Rush et al., 2012)
- Semantic Parsing (Das et al., 2012)
- Weighted Automata Problems (Paul and Eisner, 2012)
- Coordination Structure (Hanamoto et al., 2012)
- Latent-Variable Constituency Parsing (Le Roux et al., 2013)
- Faster Algorithms for Machine Translation (Rush et al., 2013)

Overview

1. Lagrangian Relaxation for Dependency Parsing
2. Lagrangian Relaxation for Syntax-Based Translation
3. Future Work

Research Interest: Information Extraction

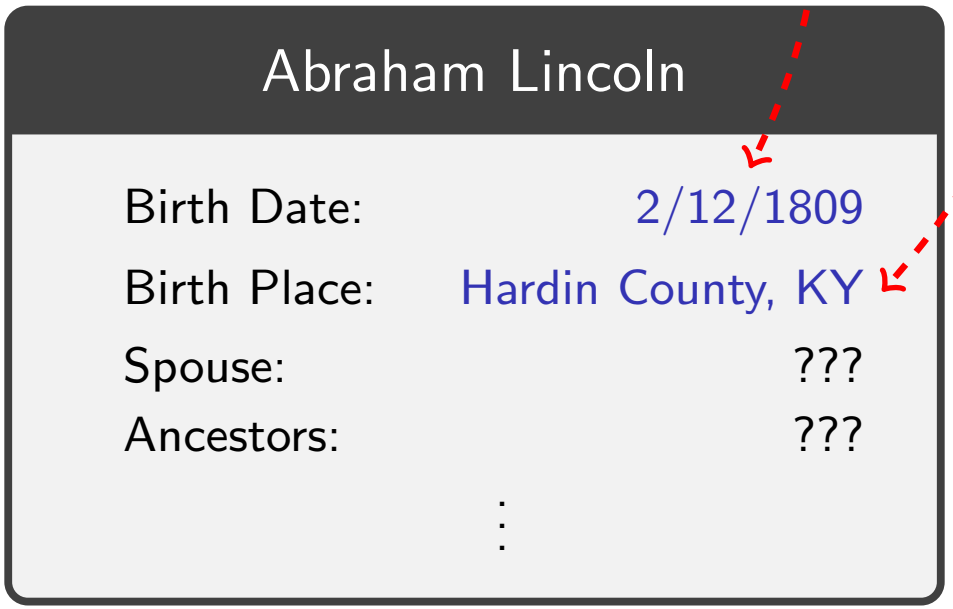
Abraham Lincoln was born February 12, 1809 in a one-room log cabin on the Sinking Spring Farm in Hardin County, Kentucky (now LaRue County). He is descended from Samuel Lincoln , who arrived in Hingham, Massachusetts, from Norfolk, England, in the 17th century...

Abraham Lincoln

Birth Date:	???
Birth Place:	???
Spouse:	???
Ancestors:	???

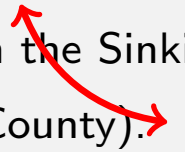
Research Interest: Information Extraction

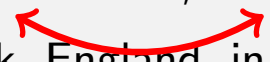
Abraham Lincoln was born February 12, 1809 in a one-room log cabin on the Sinking Spring Farm in Hardin County, Kentucky (now LaRue County). He is descended from Samuel Lincoln , who arrived in Hingham, Massachusetts, from Norfolk, England, in the 17th century...



Research Interest: Information Extraction

Abraham Lincoln was born February 12, 1809 in a one-room log cabin on the Sinking Spring Farm in Hardin County, Kentucky (now LaRue County). He is descended from Samuel Lincoln , who arrived in Hingham, Massachusetts, from Norfolk, England, in the 17th century...





Abraham Lincoln

Birth Date:	2/12/1809
Birth Place:	Hardin County, KY
Spouse:	???
Ancestors:	Samuel Lincoln
⋮	

Samuel Lincoln

Birth Date:	???
Birth Place:	???
⋮	

Research Interest: Information Extraction

That Lincoln, after winning the presidency, made the unprecedented decision to incorporate his eminent rivals into his political family, the cabinet, was evidence of a profound self-confidence and a first indication of what would prove to other a most unexpected greatness. Seward became secretary of state, Chase secretary of the treasury and Bates attorney general. ...

Abraham Lincoln	
Birth Date:	2/12/1809
Birth Place:	Hardin County, KY
Spouse:	???
Ancestors:	Samuel Lincoln
	⋮

Samuel Lincoln

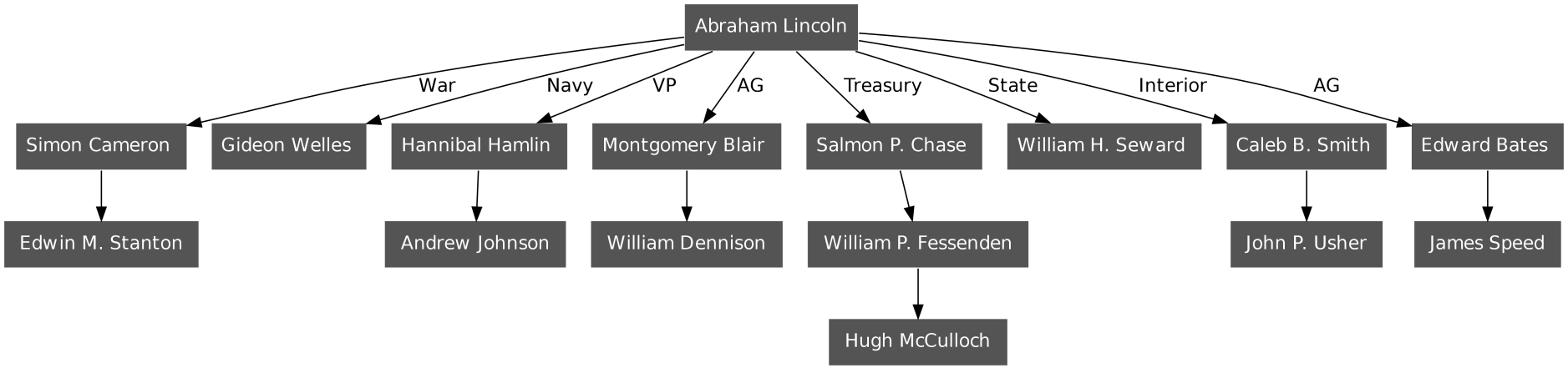
William Seward

Salmon Chase

Edward Bates

Research Interest: Information Extraction

That Lincoln, after winning the presidency, made the unprecedented decision to incorporate his eminent rivals into his political family, the cabinet, was evidence of a profound self-confidence and a first indication of what would prove to other a most unexpected greatness. Seward became secretary of state, Chase secretary of the treasury and Bates attorney general. . . .



Research Interest: Dialogue Systems

``Is the G train running today?''

$$q = \lambda x. \quad \texttt{alert}(x) \wedge \texttt{day}(x, \texttt{thur}) \wedge \texttt{train}(x, \texttt{G})$$

Research Interest: Dialogue Systems

``Is the G train running today?''

$$q = \lambda x. \quad \texttt{alert}(x) \wedge \texttt{day}(x, \texttt{thur}) \wedge \texttt{train}(x, \texttt{G})$$

$$\Downarrow$$

$$q \ y \Rightarrow \top, \quad \texttt{warning}(\texttt{y}, \text{track\_maintenence})$$

``No, it is closed for maintenance.''

Research Interest: Dialogue Systems

Is the G train running today?

No, it is closed for maintenance.

``What line should I take instead?''

$$q = \lambda x. \quad \texttt{route}(x) \wedge \texttt{day}(x, \texttt{thur}) \wedge \neg \texttt{train}(x, \texttt{L})$$

$$\Downarrow$$

$$q\ y \Rightarrow \top\ , \texttt{train}(y, \texttt{F\_Train}) \wedge \texttt{train}(y, \texttt{L\_Train})$$

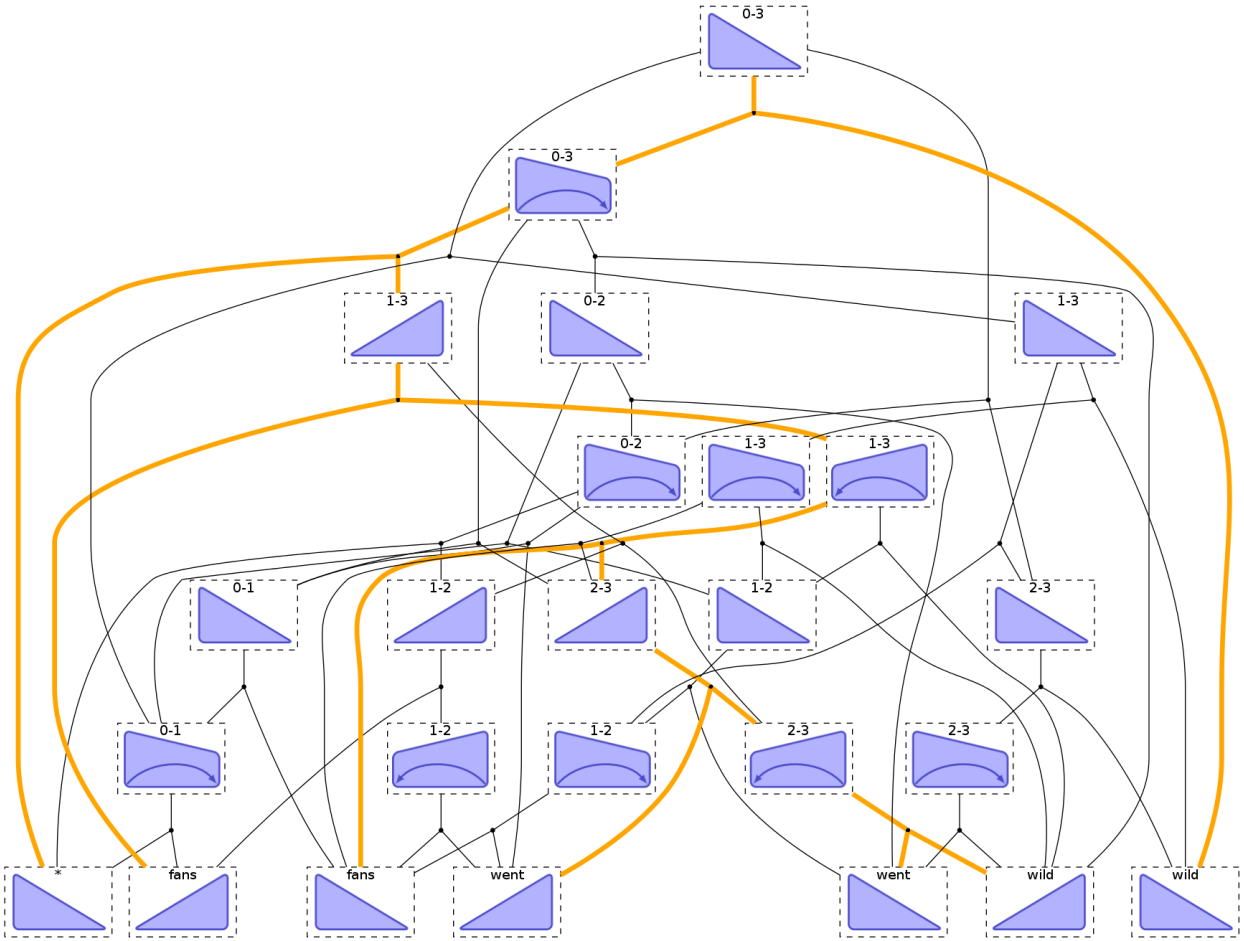
``Take the F train and switch to the L.''

Thank You

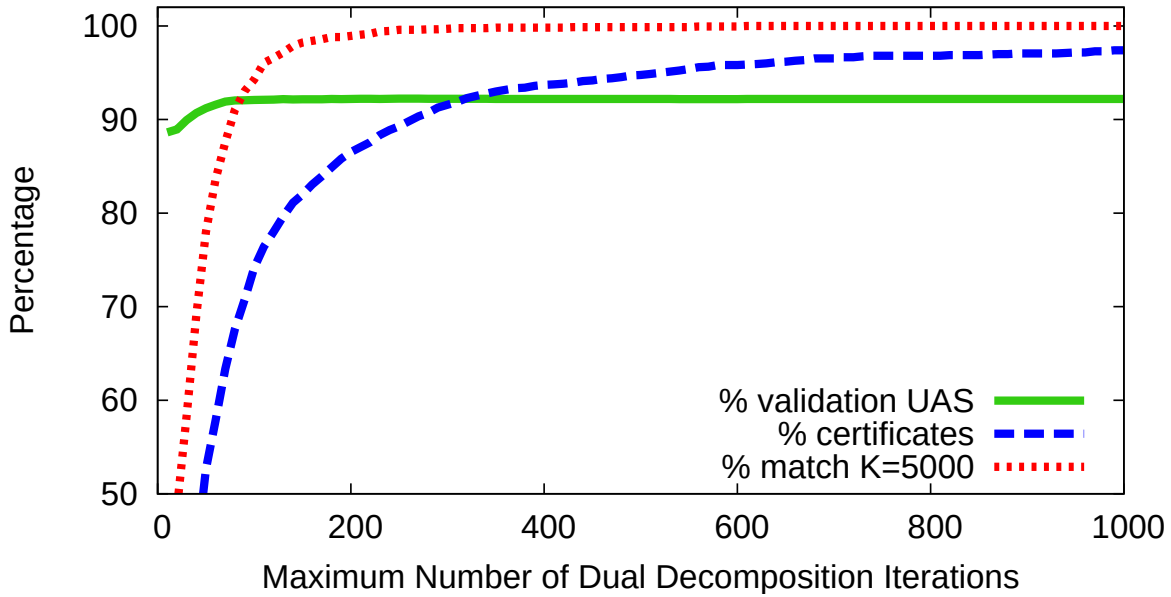
Future Work: Software Toolkits for Inference

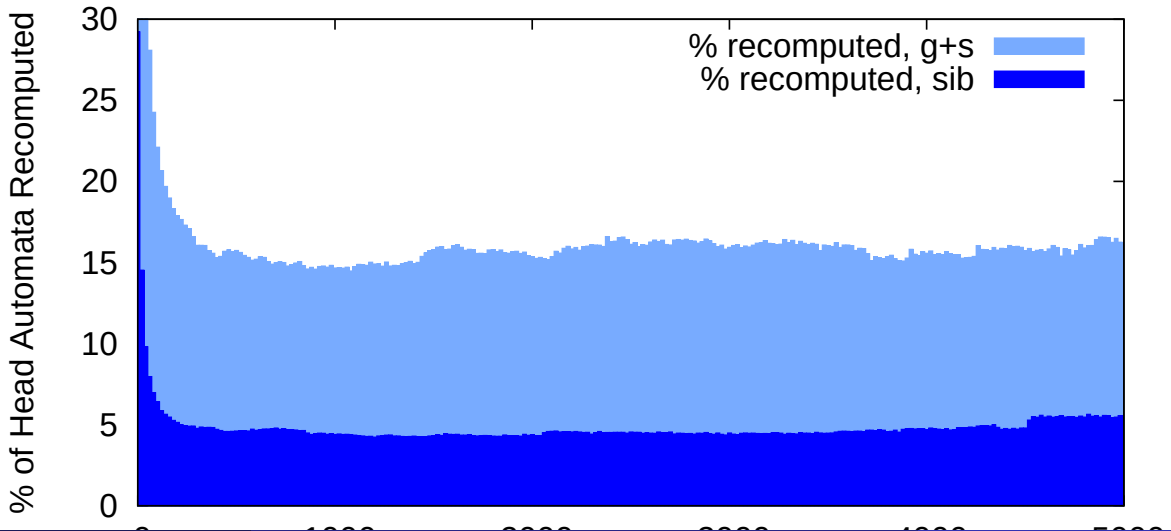
`www.pydecode.org`

Optimized tools for inference and visualization.



Dependency Results





Translation Intersection

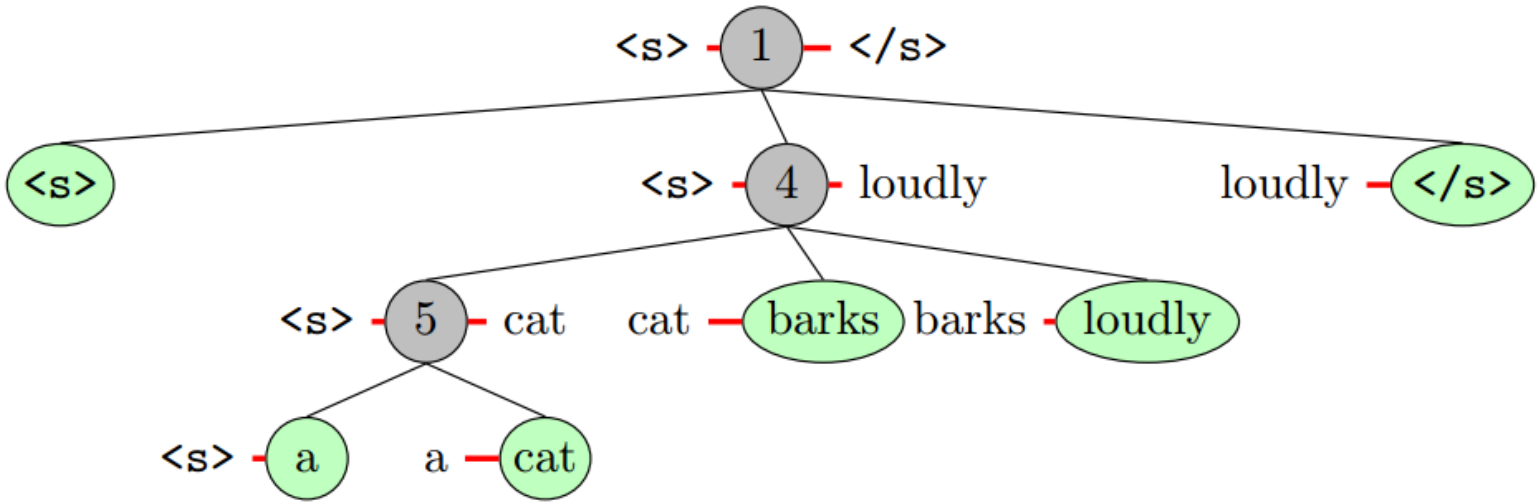
FSA and Context-Free Grammar intersection (Bar-Hillel, 1961)

- Bigram -

$$O(|\mathcal{N}|^3|\mathcal{U}|^3)$$

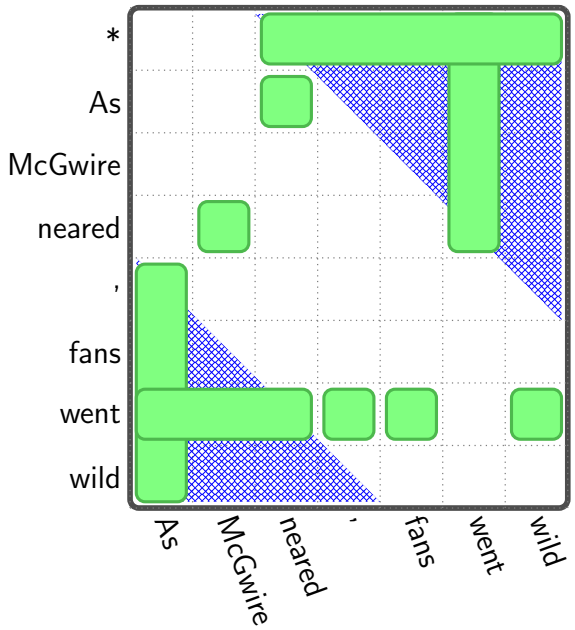
- Trigram -

$$O(|\mathcal{N}|^3|\mathcal{U}|^6)$$



Approximate Dependency Parsing: Vine Pruning

Idea: Predict within a fixed band of matrix.

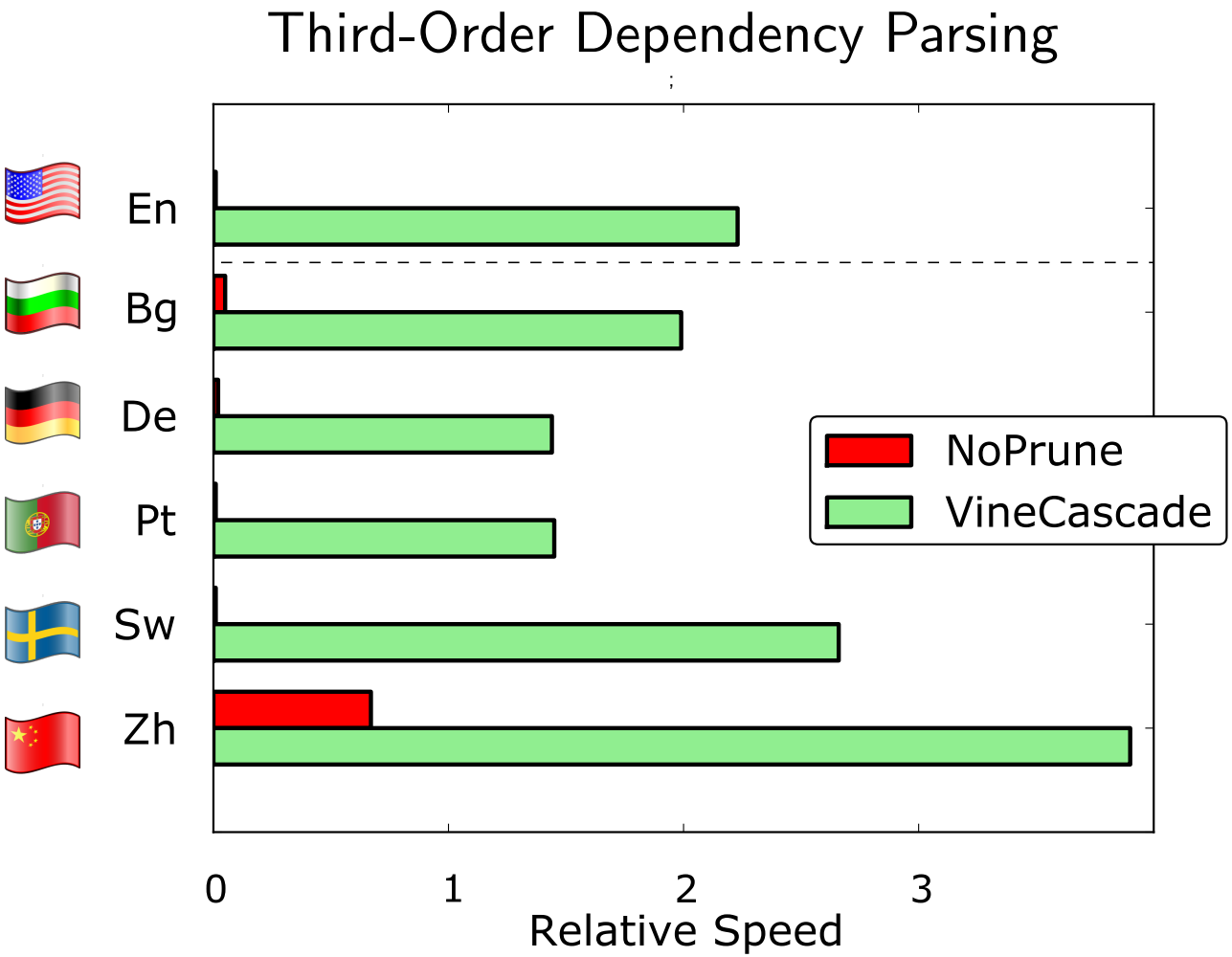


First-Order Parsing
$O(n^3)$

Vine Parsing
$O(n)$

Extension: Use vine parsing to prune possible arcs.

Vine Pruning: Dependency Parsing Speed



Tagging with Inter-Sentence Constraints