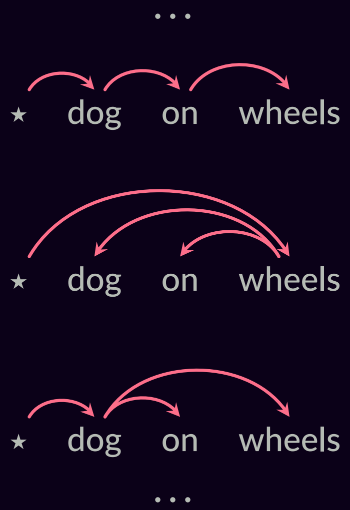
Learning with Sparse Latent Structure

Vlad Niculae
Instituto de Telecomunicações

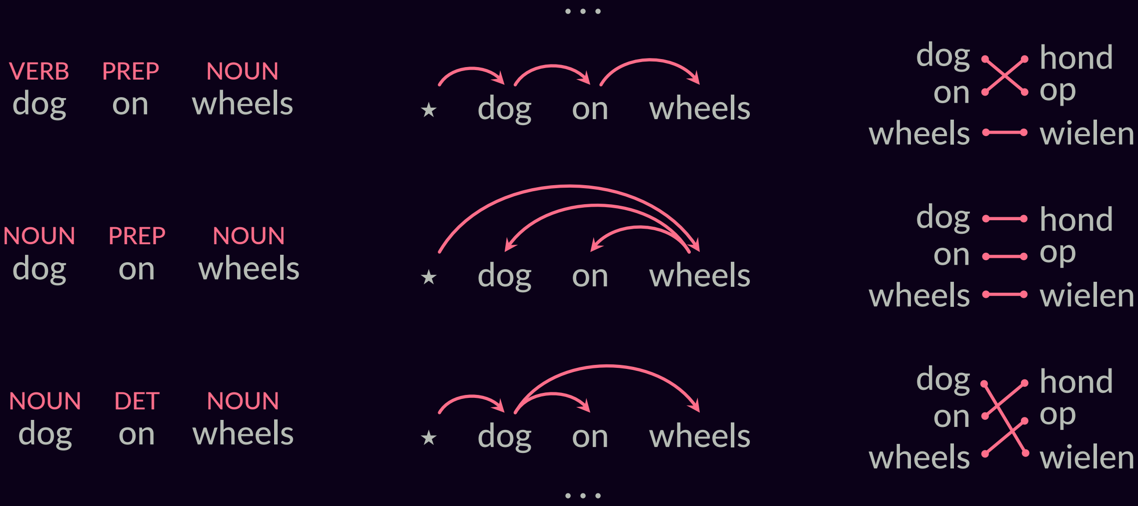
Work with: André Martins, Claire Cardie, Mathieu Blondel

github.com/vene/sparsemap   @vnfrombucharest

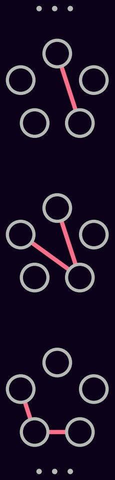
Structured Inference



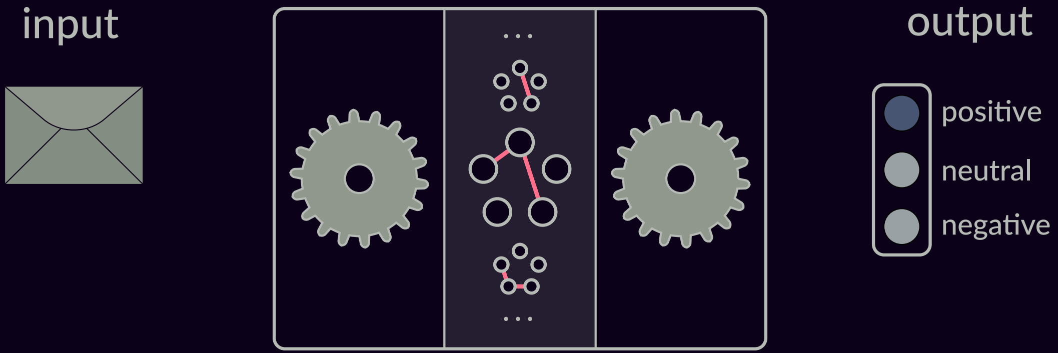
Structured Inference



Structured Inference



Latent Structured Inference

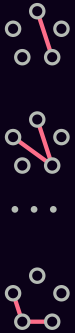


record scratch

freeze frame

How to select an item from a set?

How to select an item from a set?



How to select an item from a set?

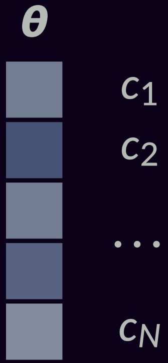
$$c_1$$

$$c_2$$

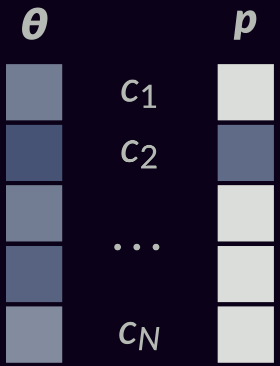
$$\cdots$$

$$c_N$$

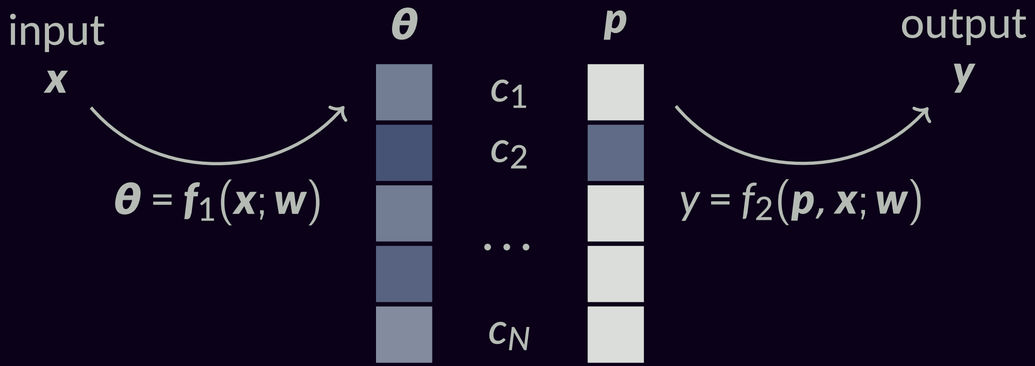
How to select an item from a set?



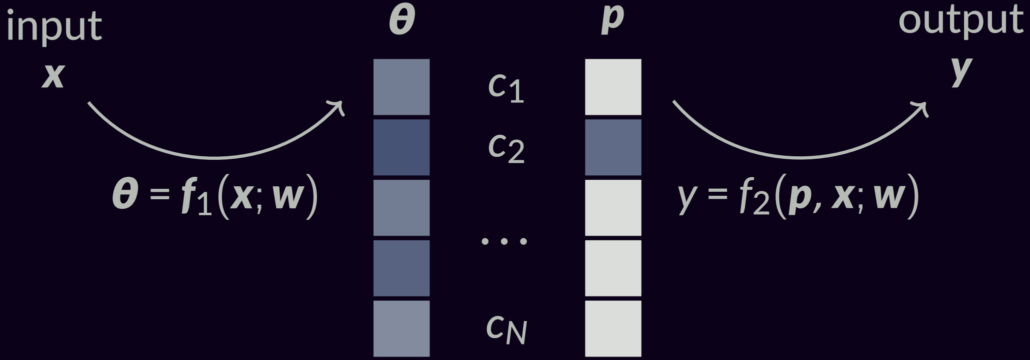
How to select an item from a set?



How to select an item from a set?

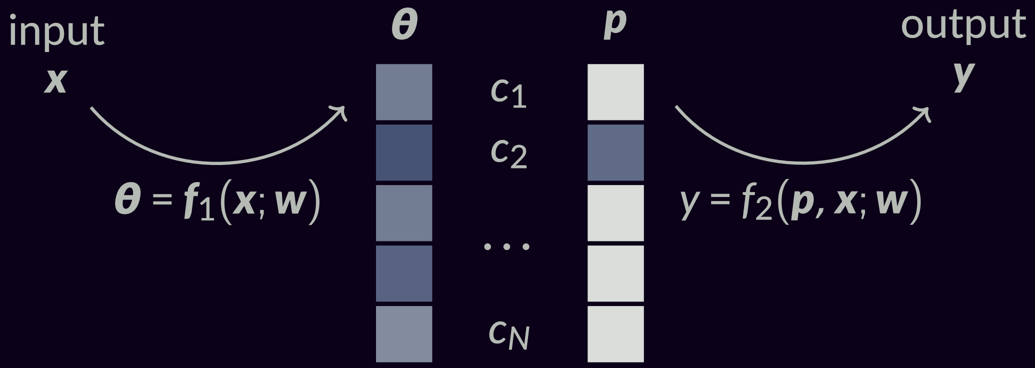


How to select an item from a set?



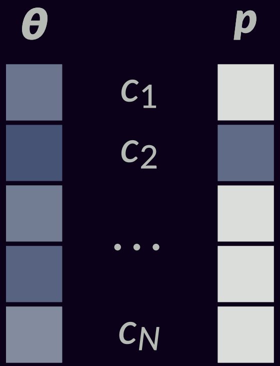
$$\frac{\partial y}{\partial \boldsymbol{w}} = ?$$

How to select an item from a set?



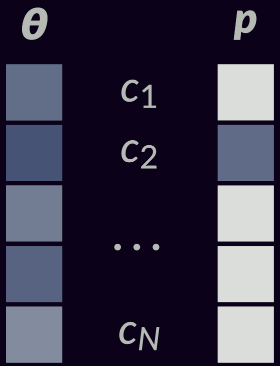
$\frac{\partial y}{\partial \boldsymbol{w}} = ?$ or, essentially, $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$

Argmax



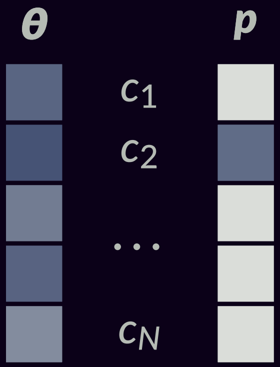
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



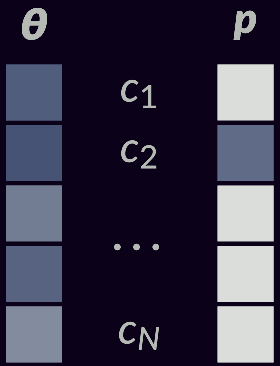
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



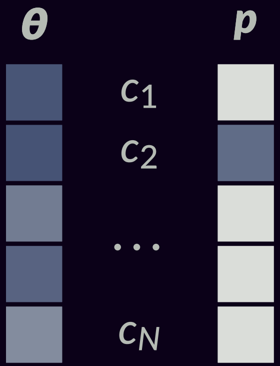
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



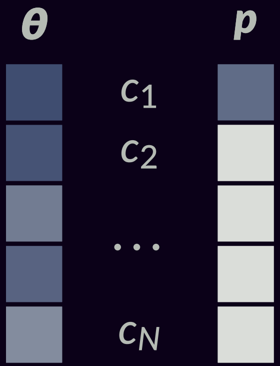
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



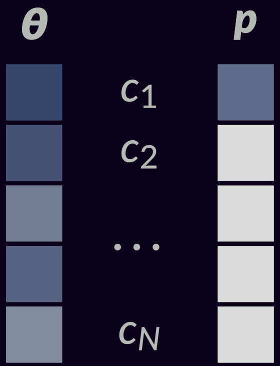
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



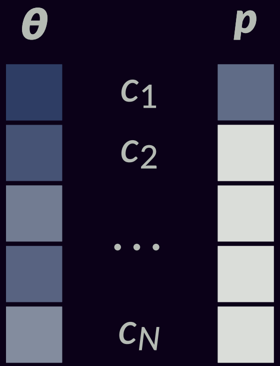
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



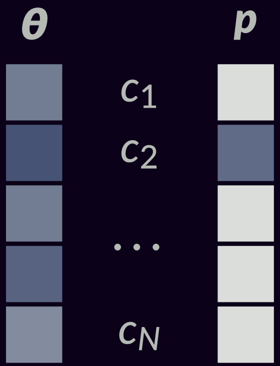
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

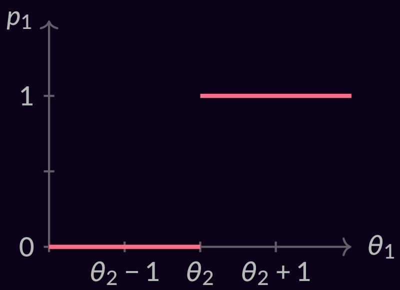
Argmax



$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax

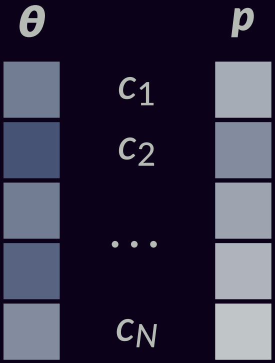


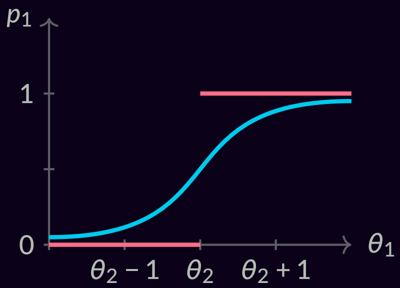


$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \mathbf{0}$$

Argmax vs. Softmax

$$p_j = \exp(\theta_j)/Z$$





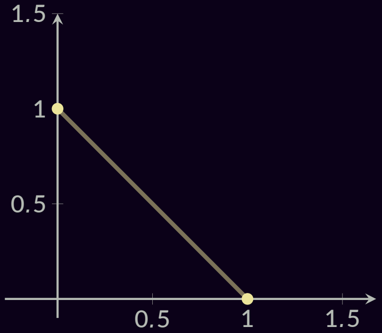
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \operatorname{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$$

Variational Form of Argmax

$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \ \boldsymbol{p} \geq \mathbf{0}, \ \mathbf{1}^\top \boldsymbol{p} = 1\}$$

Variational Form of Argmax

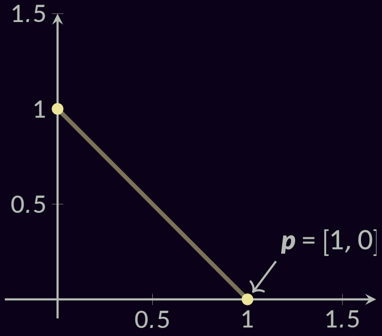
$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \ \boldsymbol{p} \geq \mathbf{0}, \ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

Variational Form of Argmax

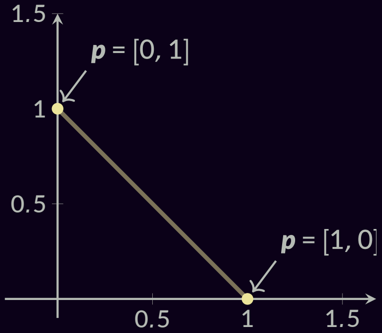
$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \ \boldsymbol{p} \geq \mathbf{0}, \ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

Variational Form of Argmax

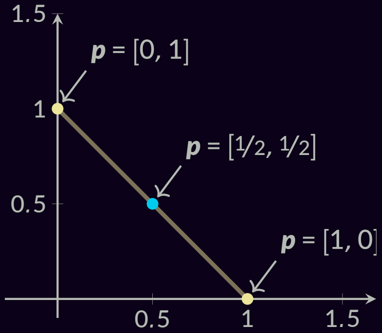
$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \ \boldsymbol{p} \geq \mathbf{0}, \ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

Variational Form of Argmax

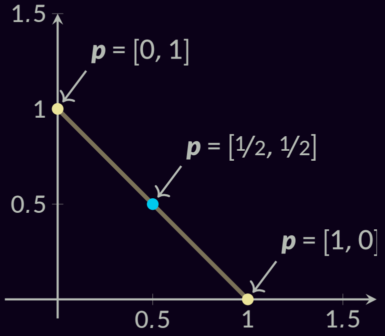
$$\Delta = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \mathbf{0},\ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



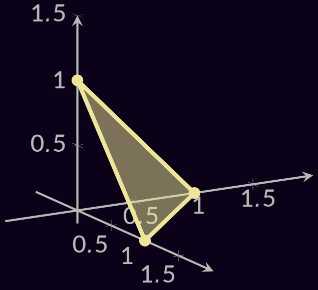
$N = 2$

Variational Form of Argmax

$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \mathbf{0},\ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



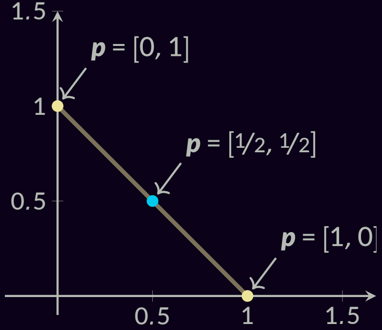
$N = 2$



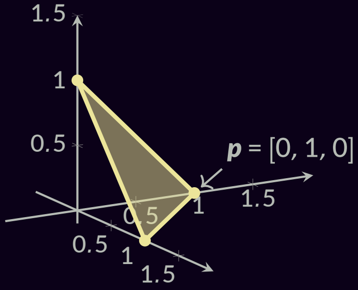
$N = 3$

Variational Form of Argmax

$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \mathbf{0},\ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



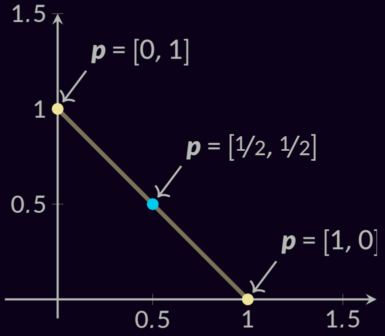
$N = 2$



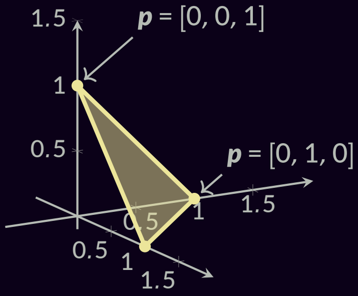
$N = 3$

Variational Form of Argmax

$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \mathbf{0},\ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



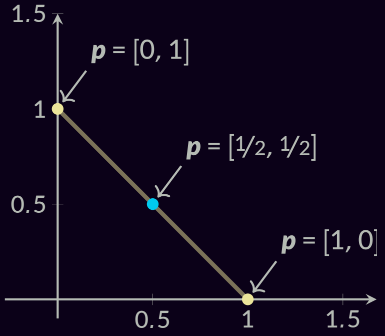
$N = 2$



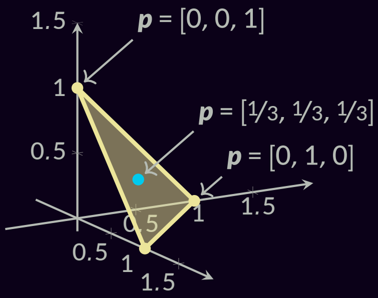
$N = 3$

Variational Form of Argmax

$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \boldsymbol{0},\ \boldsymbol{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

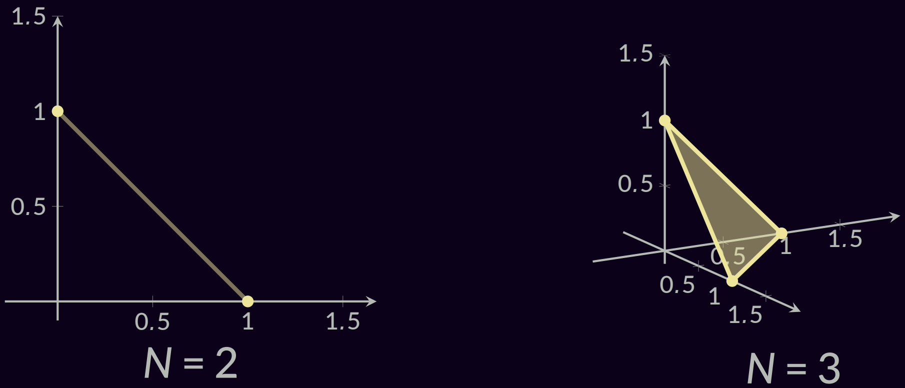


$N = 3$

Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

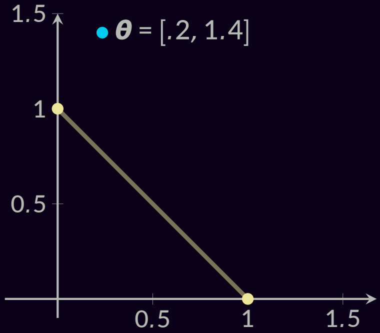
Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



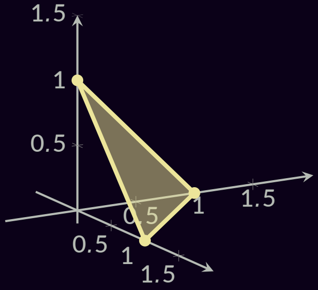
Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



$N = 2$

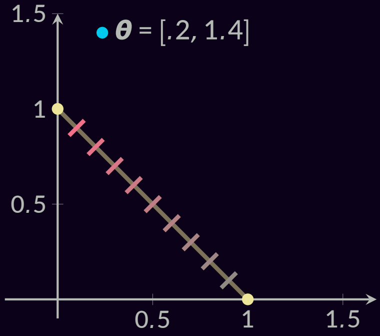


$N = 3$

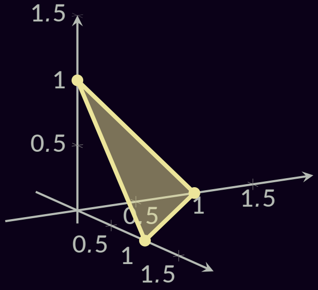
Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



$N = 2$

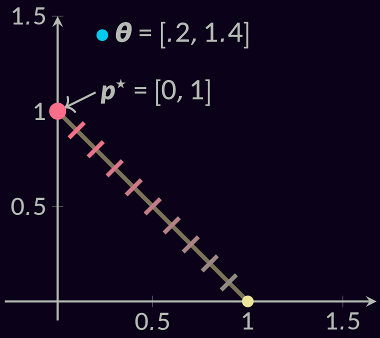


$N = 3$

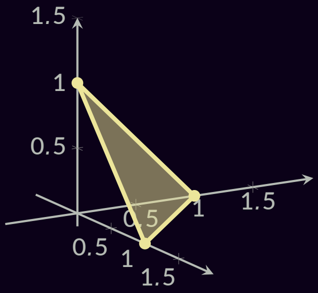
Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



$N = 2$

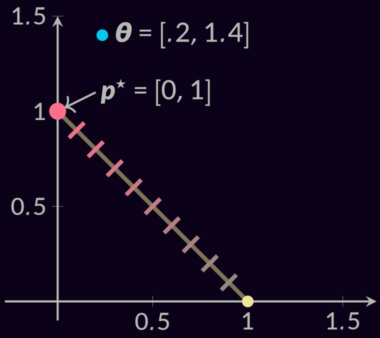


$N = 3$

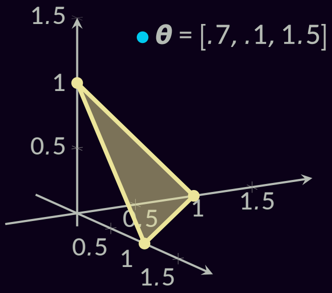
Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



$N = 2$

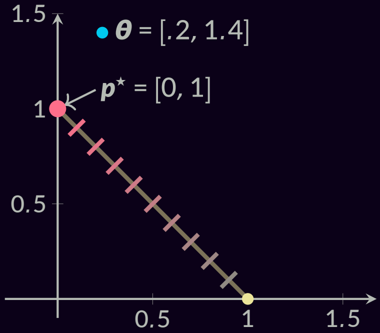


$N = 3$

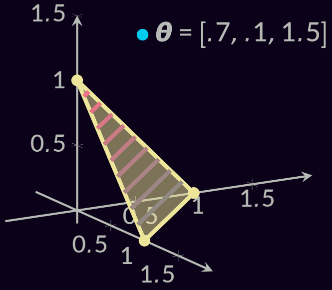
Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



$N = 2$

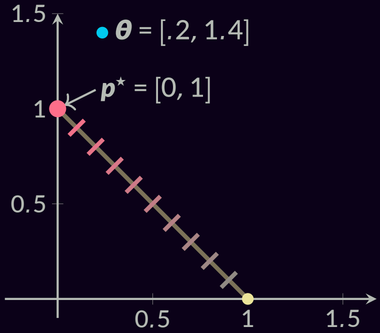


$N = 3$

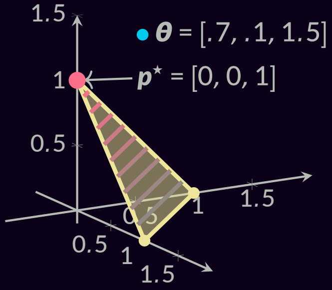
Variational Form of Argmax

$$\max_j \theta_j = \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al, 55)



$N = 2$

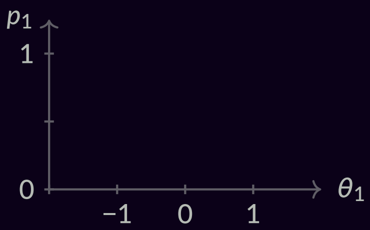


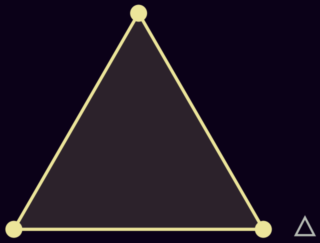
$N = 3$

(Niculae & Blondel, 17)

Smoothed Max Operators

$$\max_{\Omega}(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - \Omega(\boldsymbol{p})$$



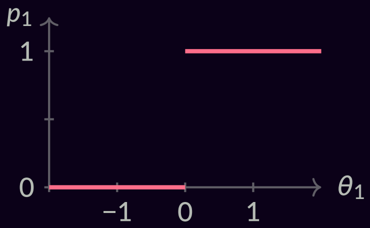


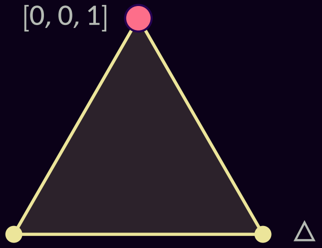
(Niculae & Blondel, 17)

Smoothed Max Operators

$$\max_{\Omega}(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^{\top}\boldsymbol{\theta} - \Omega(\boldsymbol{p})$$

- argmax: $\Omega(\boldsymbol{p}) = 0$



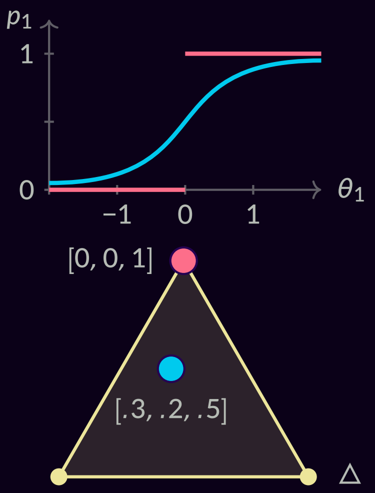


(Niculae & Blondel, 17)

Smoothed Max Operators

$$\max_{\Omega}(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} - \Omega(\boldsymbol{p})$$

- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = \sum_j p_j \log p_j$

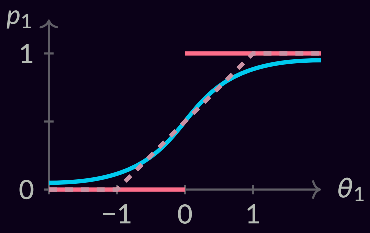


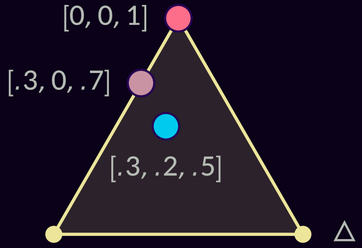
(Niculae & Blondel, 17)

Smoothed Max Operators

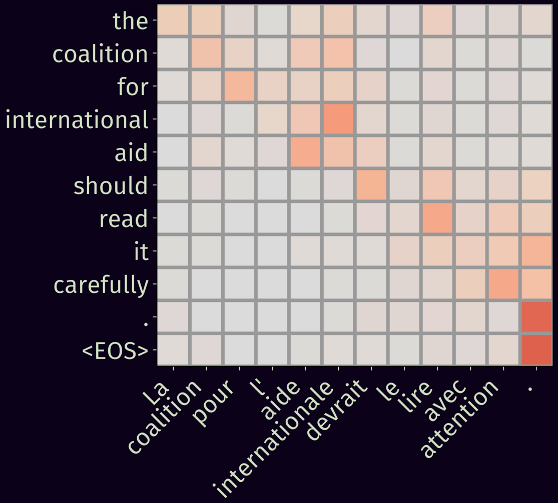
$$\max_\Omega(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - \Omega(\boldsymbol{p})$$

- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2$

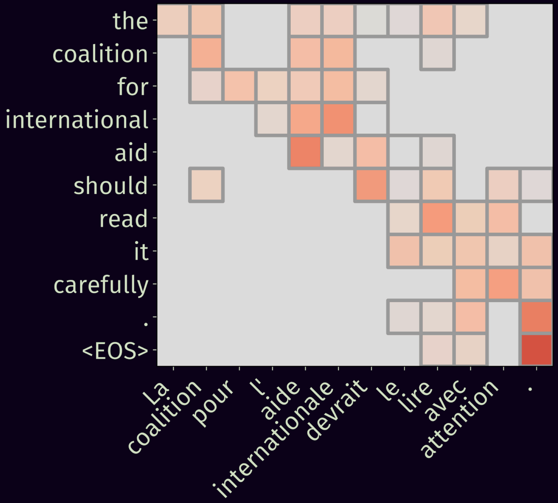




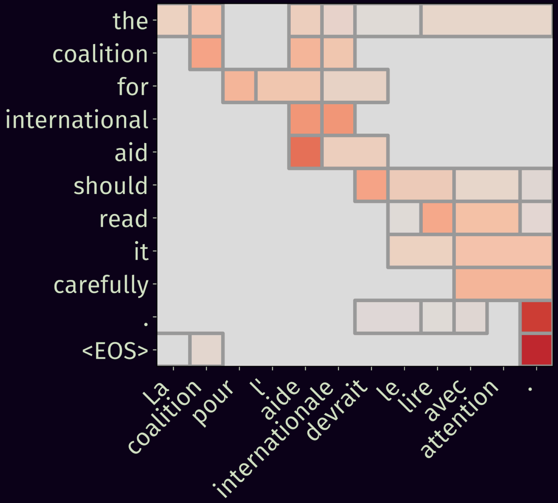
(Martins & Astudillo, 16)


the
coalition
for
international
aid
should
read
it
carefully
.
<EOS>
La
coalition
pour
l'
aide
internationale
devrait
le
lire
avec
attention
.

softmax


the
coalition
for
international
aid
should
read
it
carefully
.
<EOS>
La
coalition
pour
l'
aide
internationale
devrait
le
lire
avec
attention
.

sparsemax


the
coalition
for
international
aid
should
read
it
carefully
.
<EOS>
La
coalition
pour
l'
aide
internationale
devrait
le
lire
avec
attention
.

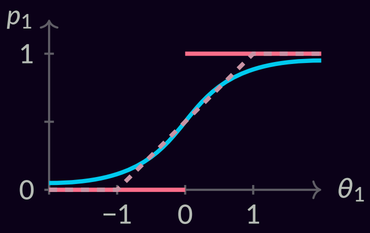
fusedmax ?!

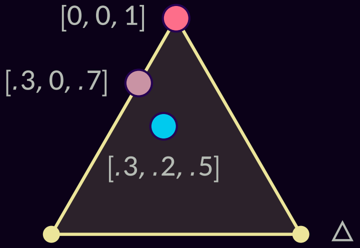
Smoothed Max Operators

(Niculae & Blondel, 17)

$$\max_{\Omega}(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - \Omega(\boldsymbol{p})$$

- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2$





(Niculae & Blondel, 17)

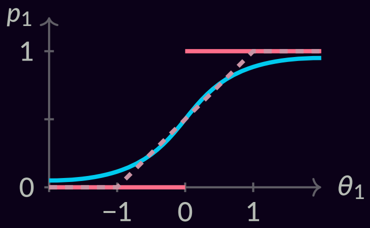
Smoothed Max Operators

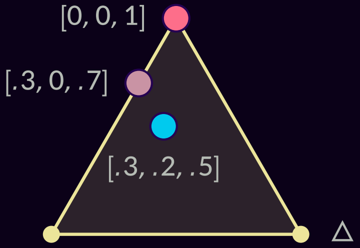
$$\max_{\Omega}(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - \Omega(\boldsymbol{p})$$

- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2$

fusedmax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2 + \sum_j |p_j - p_{j-1}|$

oscarmax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2 + \sum_{i,j} \max(p_i, p_j)$



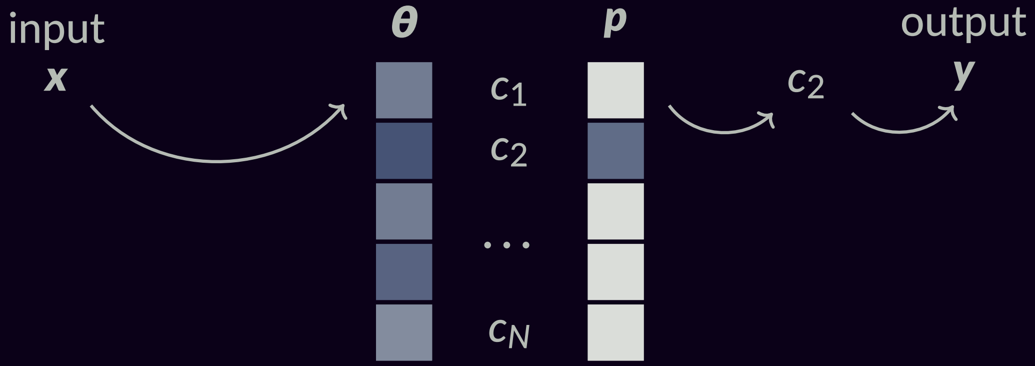


Structured Inference

finally

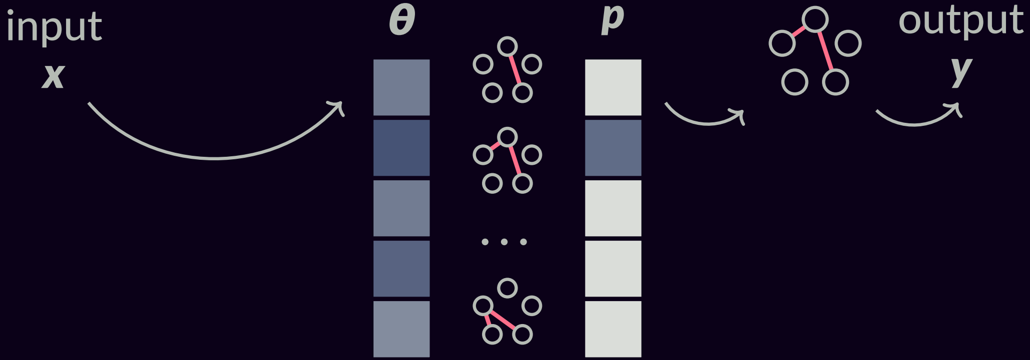
Structured Inference

is essentially a (very high-dimensional) argmax



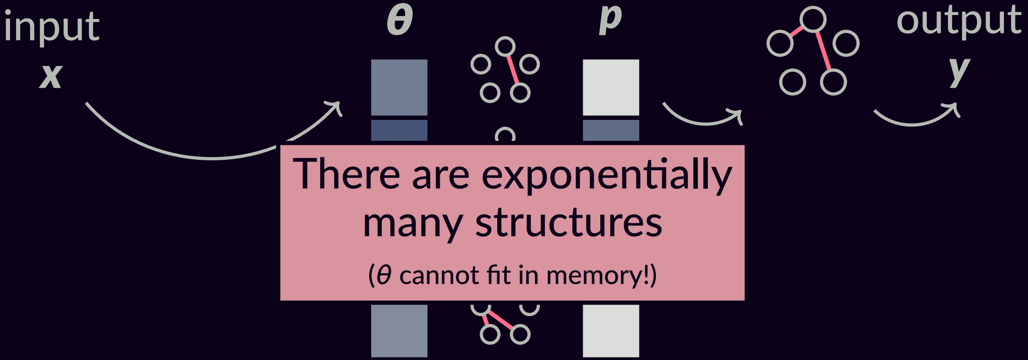
Structured Inference

is essentially a (very high-dimensional) argmax



Structured Inference

is essentially a (very high-dimensional) argmax



There are exponentially many structures

(θ cannot fit in memory!)

Factorization Into Parts

$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

Factorization Into Parts

$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

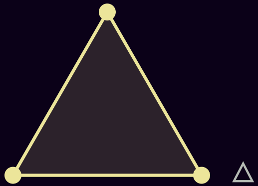
$$\boldsymbol{A} = \begin{array}{r} \star\to\text{dog} \\ \text{on}\to\text{dog} \\ \text{wheels}\to\text{dog} \\ \hline \star\to\text{on} \\ \text{dog}\to\text{on} \\ \text{wheels}\to\text{on} \\ \hline \star\to\text{wheels} \\ \text{dog}\to\text{wheels} \\ \text{on}\to\text{wheels} \end{array} \left[\begin{array}{ccccc} 1 & & 0 & 0 & \\ 0 & & 1 & 1 & \\ 0 & & 0 & 0 & \\ \hline 0 & & 1 & 1 & \\ 1 & \dots & 0 & 0 & \dots \\ 0 & & 0 & 0 & \\ \hline 0 & & 0 & 0 & \\ 0 & & 1 & 0 & \\ 1 & & 0 & 1 & \end{array}\right] \qquad \boldsymbol{\eta} = \left[\begin{array}{r} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{array}\right]$$

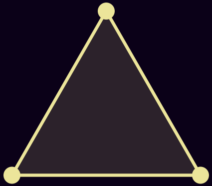
Factorization Into Parts

$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

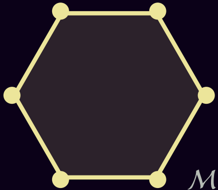
$$\boldsymbol{A} = \begin{array}{r} \star\to\text{dog} \\ \text{on}\to\text{dog} \\ \text{wheels}\to\text{dog} \\ \star\to\text{on} \\ \text{dog}\to\text{on} \\ \text{wheels}\to\text{on} \\ \star\to\text{wheels} \\ \text{dog}\to\text{wheels} \\ \text{on}\to\text{wheels} \end{array} \left[\begin{array}{ccccc|c} 1 & & 0 & 0 & & \\ 0 & & 1 & 1 & & \\ 0 & & 0 & 0 & & \\ \hline 0 & & 1 & 1 & & \\ 1 & \dots & 0 & 0 & \dots & \\ 0 & & 0 & 0 & & \\ \hline 0 & & 0 & 0 & & \\ 0 & & 1 & 0 & & \\ 1 & & 0 & 1 & & \end{array}\right] \quad \boldsymbol{\eta} = \begin{bmatrix} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{array}{r} \text{dog—hond} \\ \text{dog—op} \\ \text{dog—wielen} \\ \text{on—hond} \\ \text{on—op} \\ \text{on—wielen} \\ \text{wheels—hond} \\ \text{wheels—op} \\ \text{wheels—wielen} \end{array} \left[\begin{array}{ccccc} 1 & & 0 & 0 & \\ 0 & & 1 & 1 & \\ 0 & & 0 & 0 & \\ \hline 0 & & 0 & 0 & \\ 1 & \dots & 0 & 0 & \dots \\ 0 & & 1 & 1 & \\ \hline 0 & & 1 & 0 & \\ 0 & & 0 & 0 & \\ 1 & & 0 & 1 & \end{array}\right] \quad \boldsymbol{\eta} = \begin{bmatrix} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{bmatrix}$$



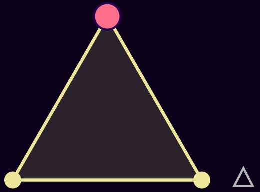


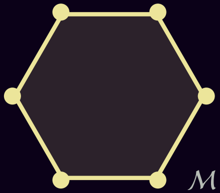
Δ



$\mathcal{M}$

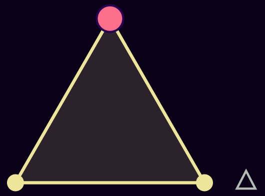
- **argmax** $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

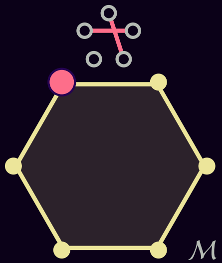




argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta}$

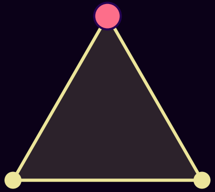
MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\boldsymbol{\eta}$



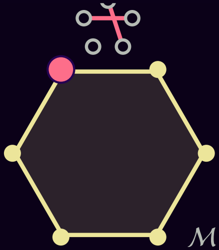


- **argmax** $\displaystyle\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$
- **MAP** $\displaystyle\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

e.g. dependency parsing ⟶ **max. spanning tree**
matching ⟶ **the Hungarian algorithm**



$\triangle$

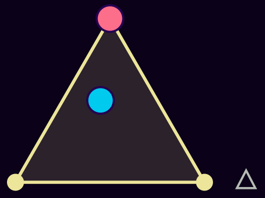


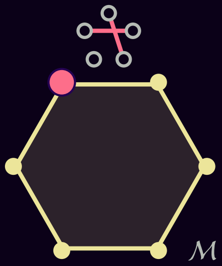
$\mathcal{M}$

● **argmax** $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

● **softmax** $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$ ●



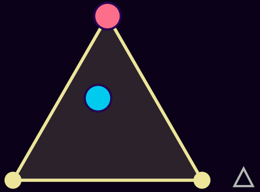


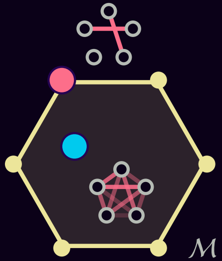
argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$





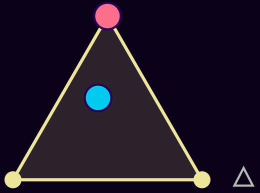
argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

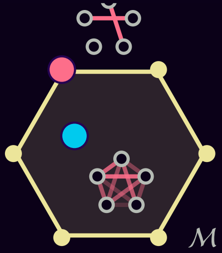
MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

e.g. dependency parsing → **the Matrix-Tree theorem**
matching → **#P-complete!** (Valiant, 79)





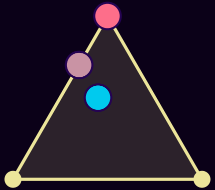
argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta}$

softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

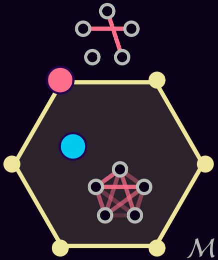
sparsemax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|^2$

MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\boldsymbol{\eta}$

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$



$\triangle$



$\mathcal{M}$

(Niculae, Martins, Blondel, Cardie, 18)

argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

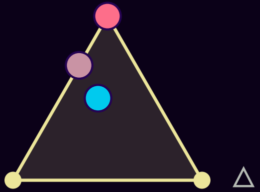
softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

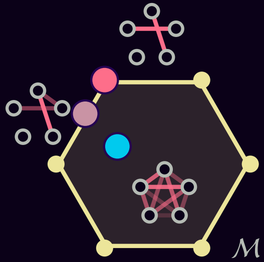
sparsemax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|^2$

MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

SparseMAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$





SparseMAP Inference Solution

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu}\in\mathcal{M}}{\arg\max}\, \boldsymbol{\mu}^\top \boldsymbol{\eta} - \tfrac{1}{2}\|\boldsymbol{\mu}\|^2$$

$=$  $= .6$  $+ .4$ 

$= \boldsymbol{A}\boldsymbol{p}^\star$ with very sparse $\boldsymbol{p}^\star \in \triangle^N$

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu} \in \mathcal{M}}{\arg\max}\, \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective:
    Active Set (Min-Norm Point)

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe)

- select a new corne
- update the (sparse)
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set** (Min-Norm Point)

Active Set achieves **finite** & **linear** convergence!

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set** (Min-Norm Point)

Backward pass

$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}$ is sparse

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - \tfrac{1}{2}\|\boldsymbol{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set** (Min-Norm Point)

Backward pass

$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}$ is sparse

computing $\left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\right)^\top \boldsymbol{dy}$

takes $\mathcal{O}(\dim(\boldsymbol{\mu})\,\mathrm{nnz}(\boldsymbol{p}^\star))$

Algorithms for SparseMAP

$$\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

Completely modular: just add MAP

Greedy Con **pass**

(Frank-W

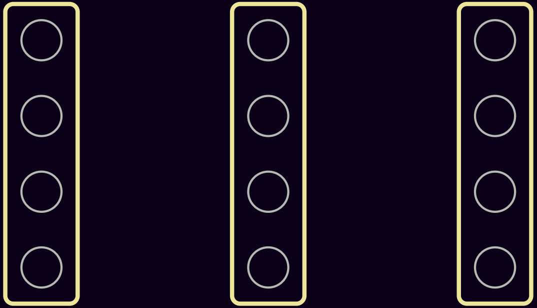
- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set** (Min-Norm Point)

$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}$ is sparse

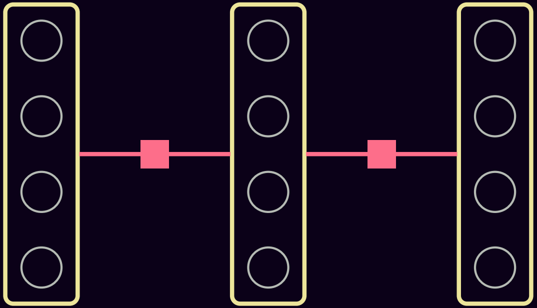
computing $\left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\right)^\top \boldsymbol{dy}$

takes $\mathcal{O}(\dim(\boldsymbol{\mu})\,\mathrm{nnz}(\boldsymbol{p}^\star))$

Structured Attention & Graphical Models



Structured Attention & Graphical Models



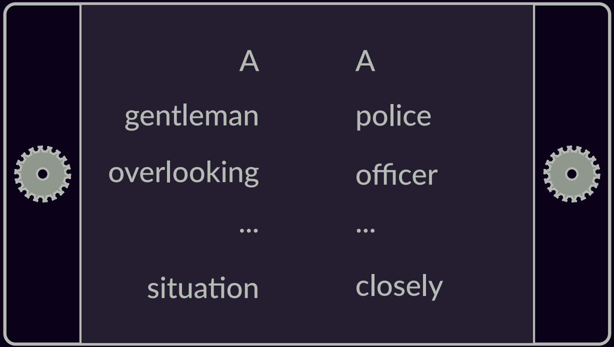
Structured Attention for Alignments

NLI

premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

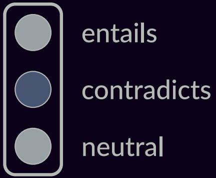
input

(P, H)



(Model: ESIM)

output



Structured Attention for Alignments

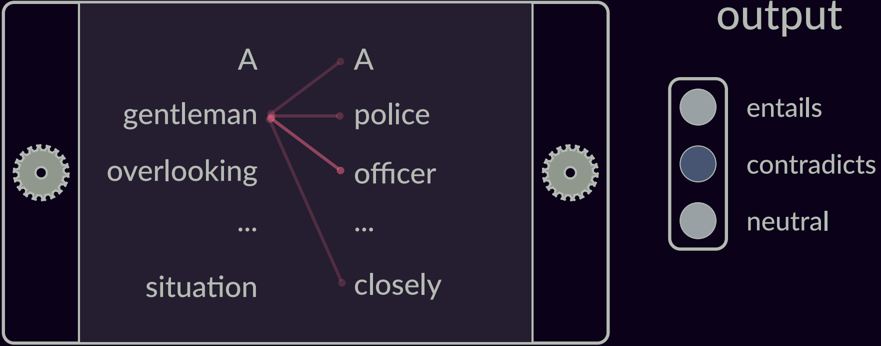
NLI

premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)

output



(Model: ESIM)

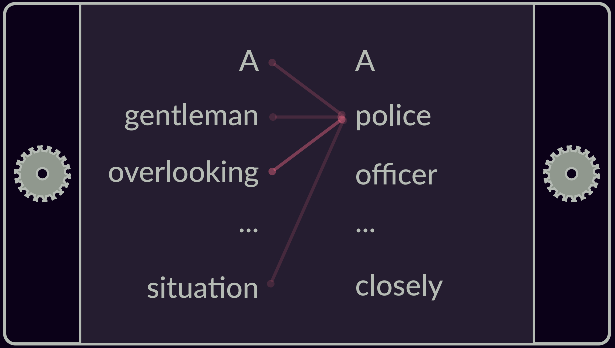
Structured Attention for Alignments

NLI

premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



(Model: ESIM)

output

entails

contradicts

neutral

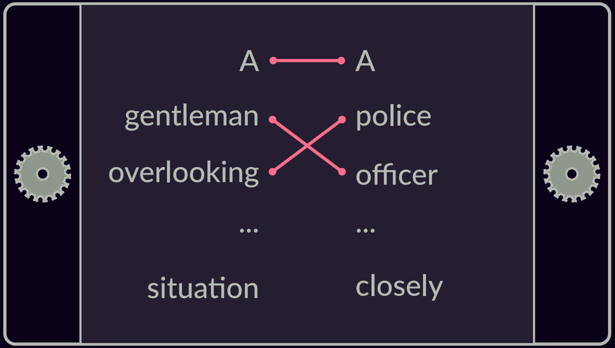
Structured Attention for Alignments

NLI

premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



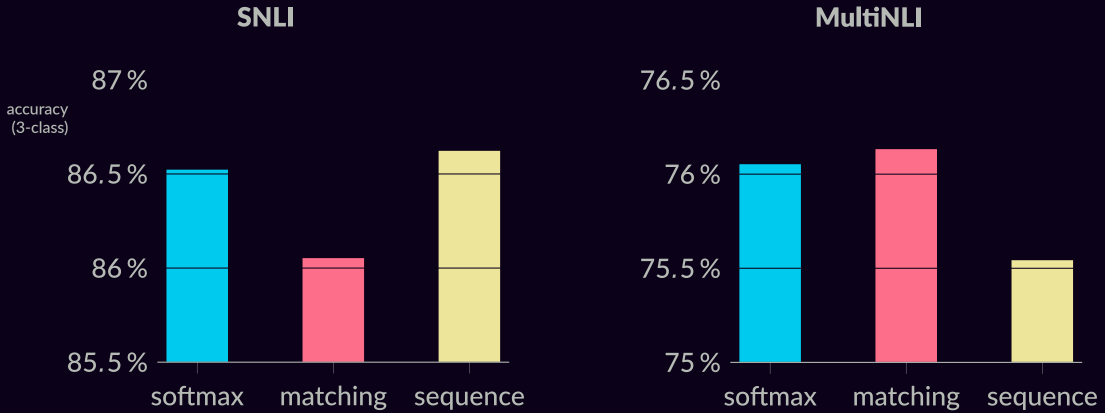
(Proposed model: global matching)

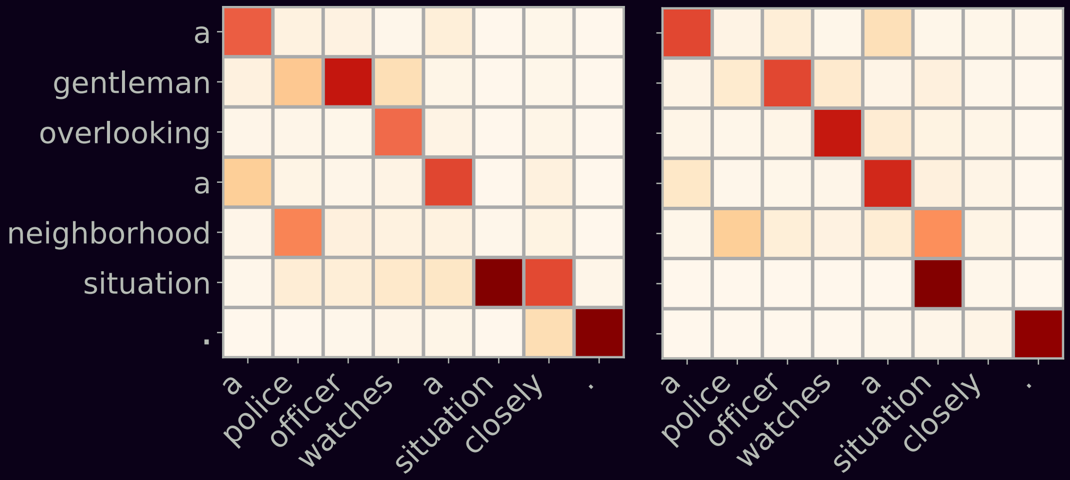
output

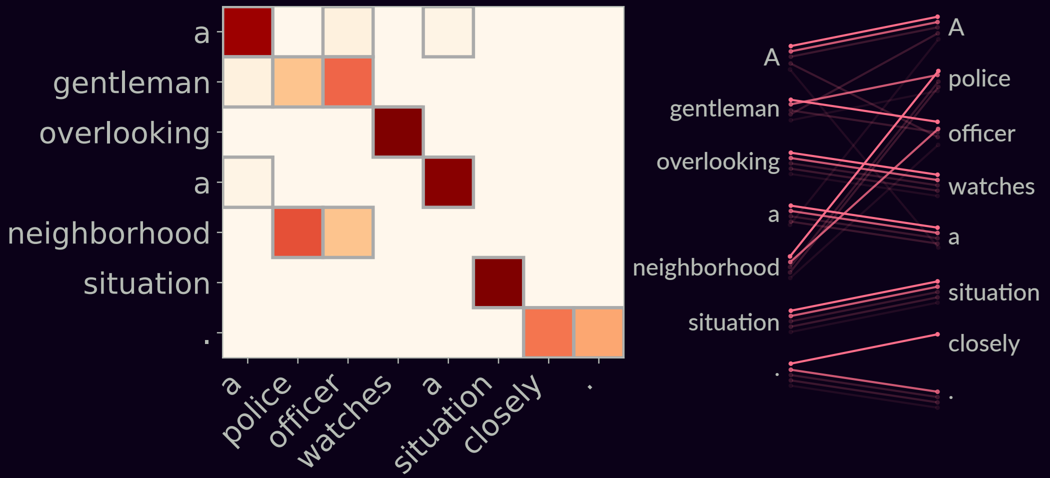
entails

contradicts

neutral


SNLI
MultiNLI
87 %
accuracy
(3-class)
86.5 %
86 %
85.5 %
softmax
matching
sequence
76.5 %
76 %
75.5 %
75 %
softmax
matching
sequence


a
gentleman
overlooking
a
neighborhood
situation
.
a
police
officer
watches
a
situation
closely
.
a
police
officer
watches
a
situation
closely
.


a
gentleman
overlooking
a
neighborhood
situation
.
a
police
officer
watches
a
situation
closely
.
A
gentleman
overlooking
a
neighborhood
situation
.
A
police
officer
watches
a
situation
closely
.

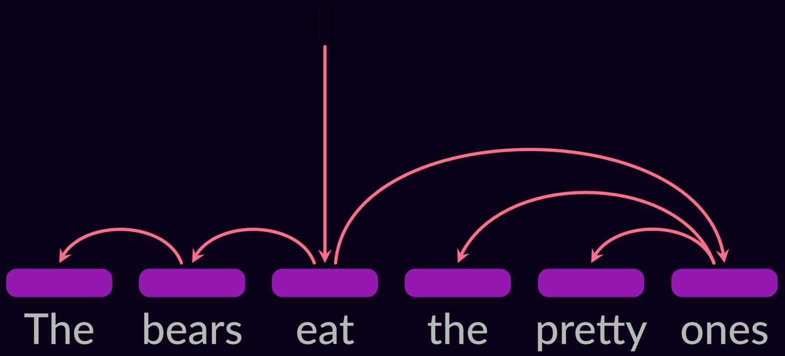
Dynamically inferring the computation graph

Dependency TreeLSTM

(Tai & al, 15)

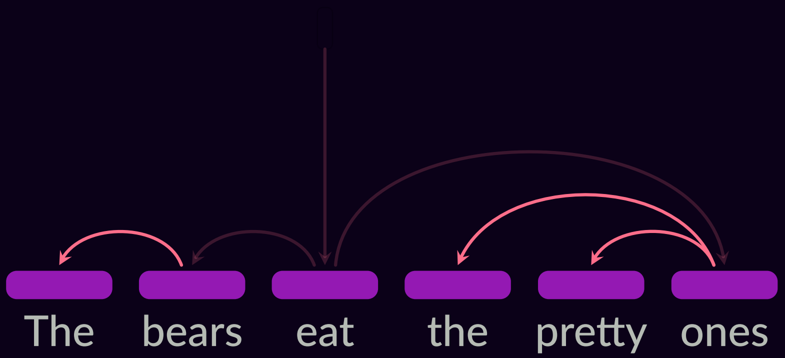
Dependency TreeLSTM

(Tai & al, 15)



Dependency TreeLSTM

(Tai & al, 15)

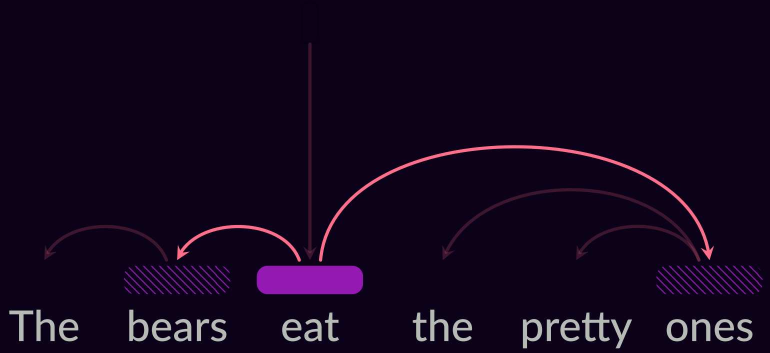


Dependency TreeLSTM

(Tai & al, 15)

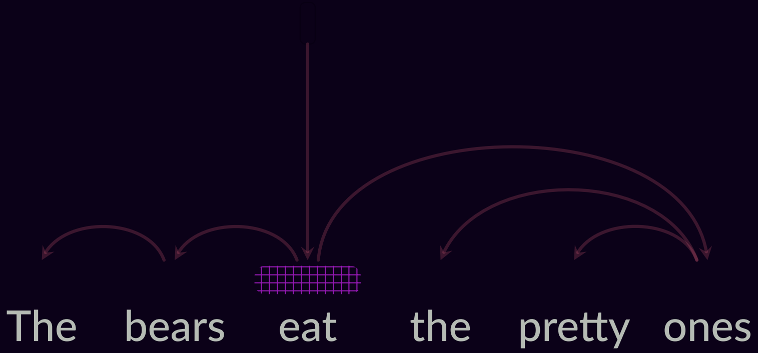
Dependency TreeLSTM

(Tai & al, 15)



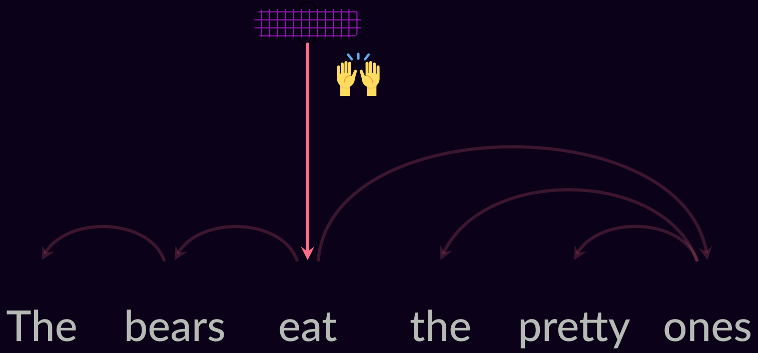
Dependency TreeLSTM

(Tai & al, 15)



Dependency TreeLSTM

(Tai & al, 15)

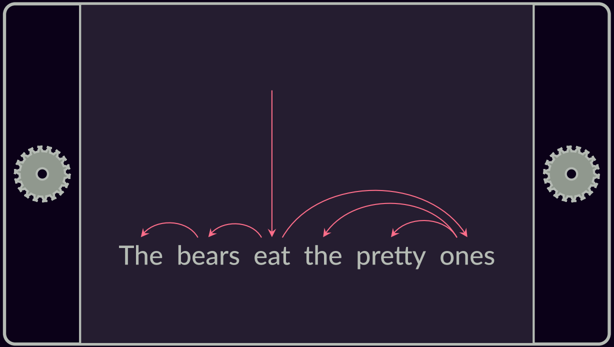


Latent Dependency TreeLSTM

(Niculae, Martins, Cardie, 18)

input

x



output

y

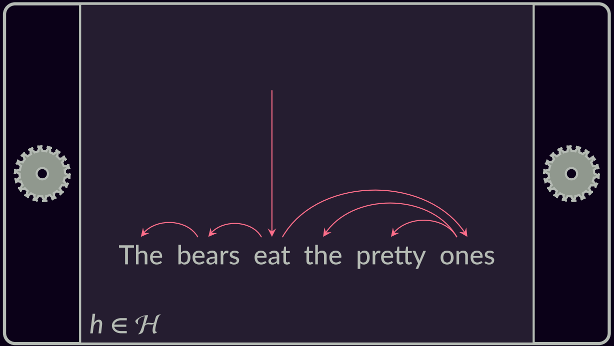
Latent Dependency TreeLSTM

(Niculae, Martins, Cardie, 18)

$$p(y|x) = \sum_{h \in \mathcal{H}} p(y \mid h, x)\, p(h \mid x)$$

input

x



output

y

Structured Latent Variable Models

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p \quad (y \mid h, x)\, p \quad (h \mid x)$$

Structured Latent Variable Models

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

Structured Latent Variable Models

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

Structured Latent Variable Models

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

Exponentially large sum!

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $\boldsymbol{p_{\pi}}$?

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

$$\sum_{h \in \mathcal{H}}$$

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$$

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$$

idea 1 $p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0 argmax

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\mathrm{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

$$\sum_{h \in \mathcal{H}} \quad \frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$$

idea 1 $p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0 argmax 😊

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

$$\sum_{h \in \mathcal{H}} \qquad \frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$$

idea 1 $\quad p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0 $\qquad$ argmax 😊 😟

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

$$\sum_{h \in \mathcal{H}} \qquad \frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$$

😊 😟

idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1 \text{ if } h = h^\star \text{ else } 0$	argmax
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\left(\text{score}_{\boldsymbol{\pi}}(h; x)\right)$	softmax
idea 3		

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

			$\sum_{h \in \mathcal{H}}$	$\dfrac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$
idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0	argmax	😊	😟
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\left(\text{score}_{\boldsymbol{\pi}}(h; x)\right)$	softmax		😊
idea 3				

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by *h*

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $\boldsymbol{p_{\pi}}$?

			$\sum_{h \in \mathcal{H}}$	$\frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$
idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^{\star}$ else 0	argmax	😊	😟
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\big(\text{score}_{\boldsymbol{\pi}}(h; x)\big)$	softmax	😱	😊
idea 3				

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\boldsymbol{\phi}}(y \mid h, x)\, p_{\boldsymbol{\pi}}(h \mid x)$$

parsing model,
using some $\text{score}_{\boldsymbol{\pi}}(h; x)$

How to define $p_{\boldsymbol{\pi}}$?

			$\sum_{h \in \mathcal{H}}$	$\dfrac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$
idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0	argmax	😊	😟
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\big(\text{score}_{\boldsymbol{\pi}}(h; x)\big)$	softmax	😱	😊
idea 3		SparseMAP	😊	😊

SparseMAP Inference

 = .7  + .3 

SparseMAP Inference

 = .7  + .3  + 0  + ...

SparseMAP Inference

 $= .7$  $+ .3$  $+ 0$  $+ \ldots$

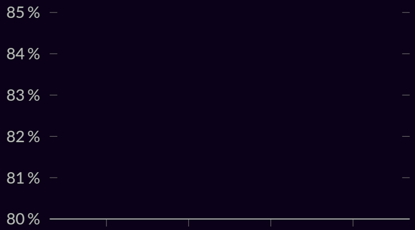
$p(y \mid x) = .7\, p_{\boldsymbol{\phi}}(y \mid$  $) + .3\, p_{\boldsymbol{\phi}}(y \mid$  $)$

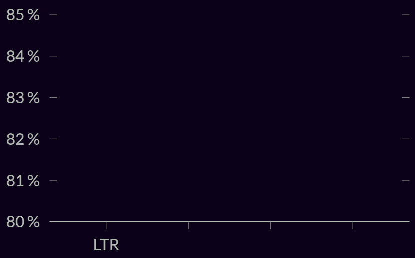
SparseMAP Inference

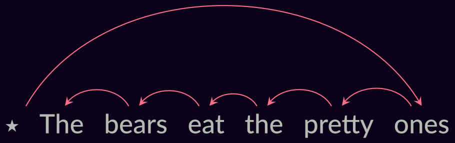
 $= \;.7$  $+\;.3$  $+\;0$  $+\;\ldots$

$p(y \mid x) = \;.7\, p_{\boldsymbol{\phi}}(y \mid$  $) + .3\, p_{\boldsymbol{\phi}}(y \mid$  $)$

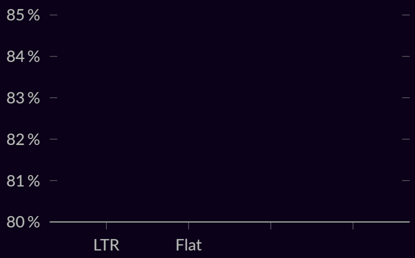
 is not a tree itself: $p(y \mid x) \neq p_{\boldsymbol{\phi}}(y \mid$  $)$!

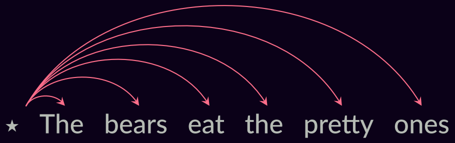

85 %
84 %
83 %
82 %
81 %
80 %



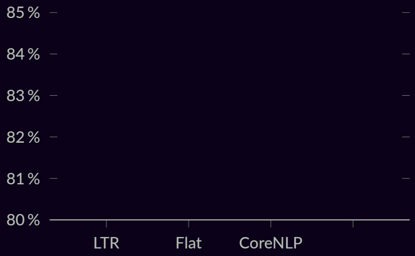


Left-to-right: regular LSTM

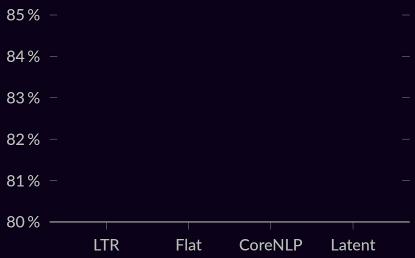


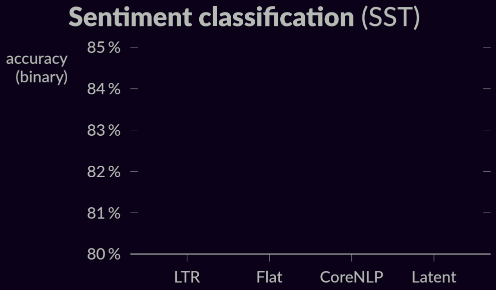


Flat: bag-of-words–like

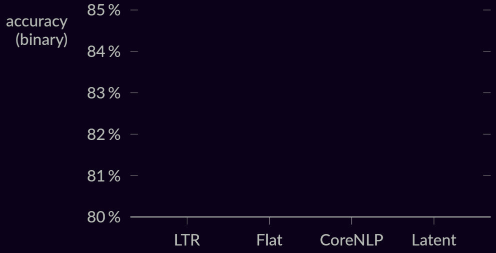


CoreNLP: off-line parser

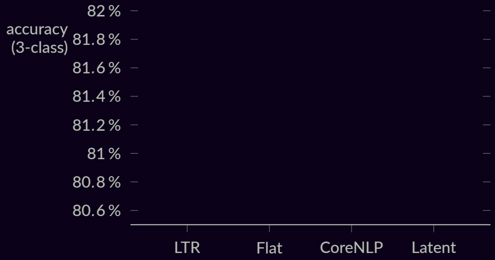

85 %
84 %
83 %
82 %
81 %
80 %
LTR
Flat
CoreNLP
Latent


Sentiment classification (SST)
accuracy (binary)
85 %
84 %
83 %
82 %
81 %
80 %
LTR
Flat
CoreNLP
Latent

Sentiment classification (SST)



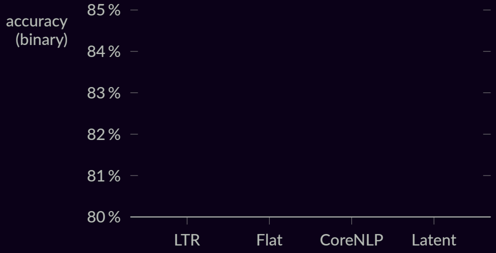
Natural Language Inference (SNLI)



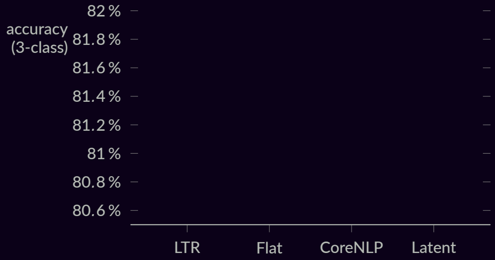
Sentence pair classification (P, H)

$$p(y \mid P, H) = \sum_{h_P \in \mathcal{H}(P)} \sum_{h_H \in \mathcal{H}(H)} p_{\boldsymbol{\phi}}(y \mid h_P, h_H)\, p_{\boldsymbol{\pi}}(h_P \mid P)\, p_{\boldsymbol{\pi}}(h_H \mid H)$$

Sentiment classification (SST)



Natural Language Inference (SNLI)

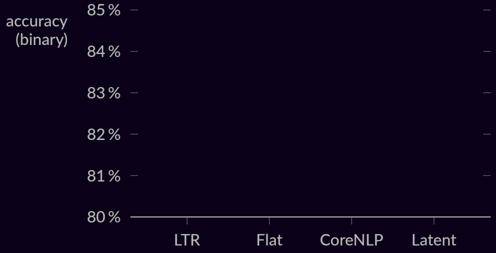


Reverse dictionary lookup

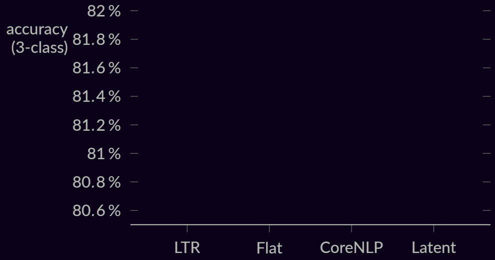
given word description, predict word embedding (Hill et al, 17)

instead of $p(y \mid x)$, we model $\mathbb{E}_{p_{\boldsymbol{\pi}}} \boldsymbol{g}(x) = \sum_{h \in \mathcal{H}} \boldsymbol{g}(x; h)\, p_{\boldsymbol{\pi}}(h \mid x)$

Sentiment classification (SST)

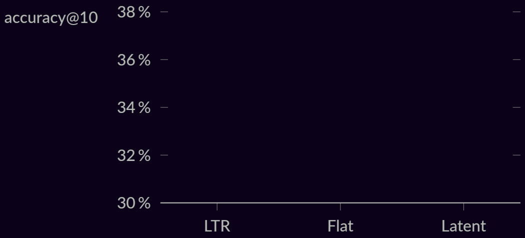


Natural Language Inference (SNLI)

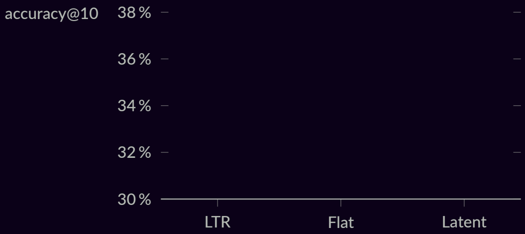


Reverse dictionary lookup

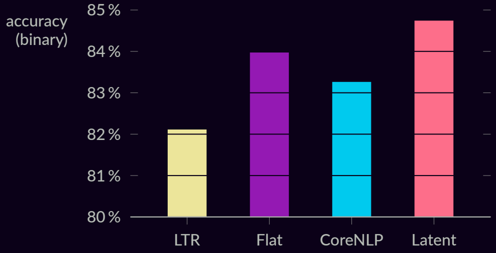
(definitions)



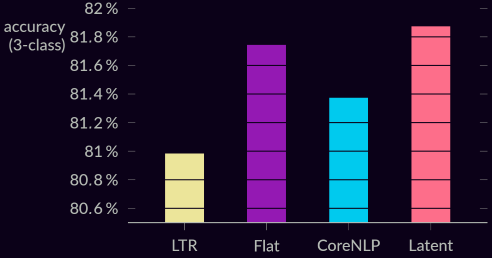
(concepts)



Sentiment classification (SST)

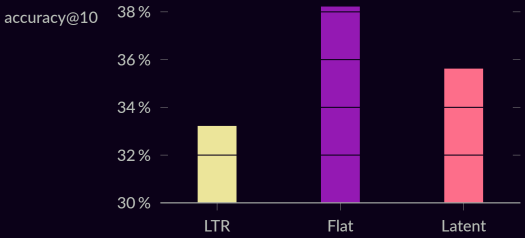


Natural Language Inference (SNLI)

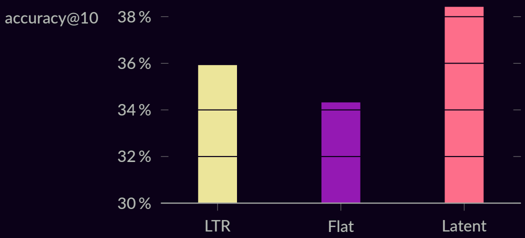


Reverse dictionary lookup

(definitions)

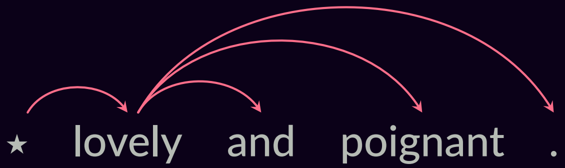


(concepts)



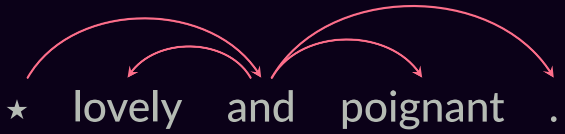
Syntax vs. Composition Order

CoreNLP parse, $p = 21.4\%$



Syntax vs. Composition Order

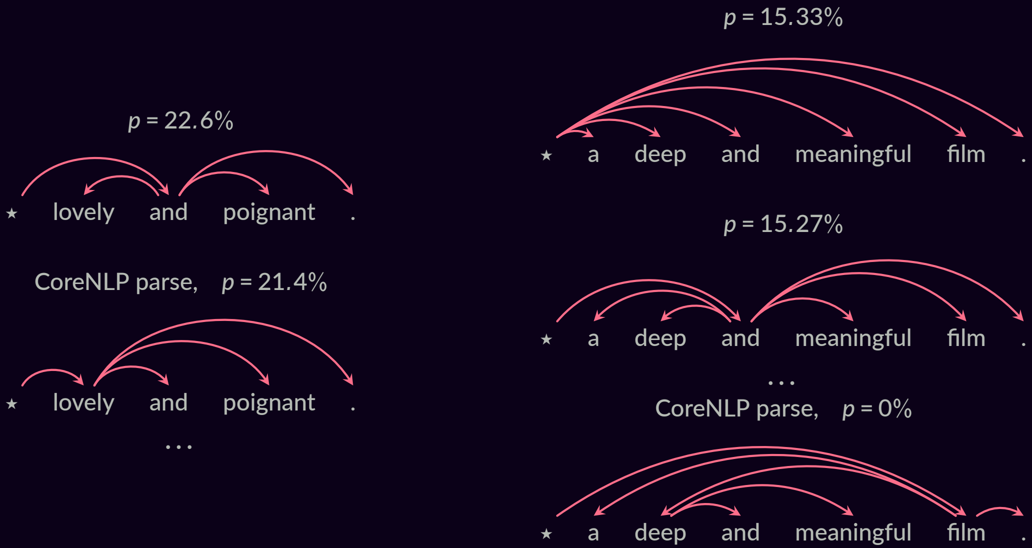
$p = 22.6\%$



CoreNLP parse, $p = 21.4\%$

. . .

Syntax vs. Composition Order



Conclusions

Differentiable & sparse
structured inference

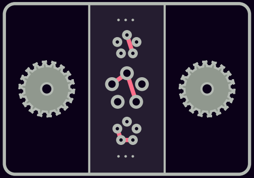
Generic, extensible algorithms

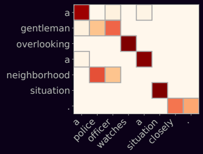
Interpretable structured attention

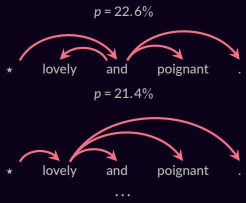
Dynamically-inferred
computation graphs

Catch us at EMNLP:
BlackboxNLP, Thursday 11:00 & EMNLP, Friday 15:36 (3B)

vlad@vene.ro
https://vene.ro
github.com/vene/sparsemap
@vnfrombucharest

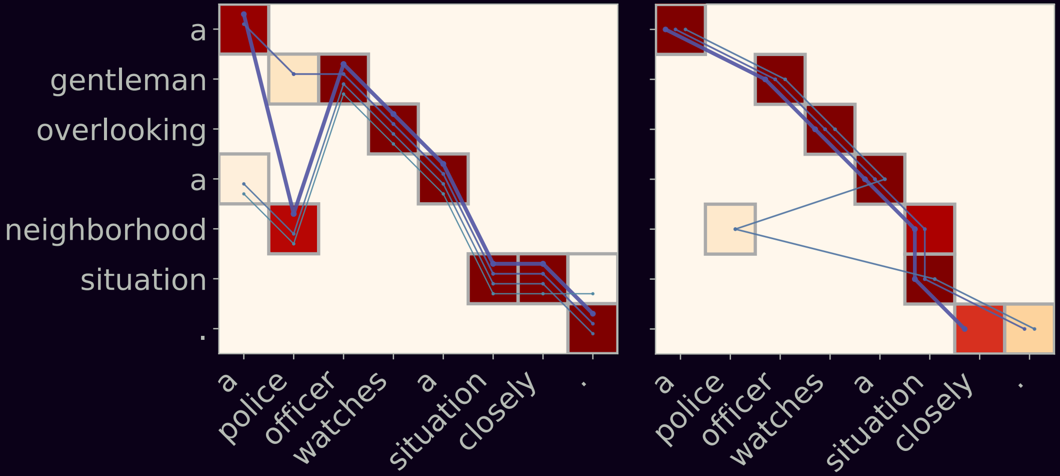






Extra slides

Some icons by Dave Gandy and Freepik via flaticon.com.


a
gentleman
overlooking
a
neighborhood
situation
.
a
police
officer
watches
a
situation
closely
.
a
police
officer
watches
a
situation
closely
.

Structured Output Prediction

SparseMAP

$$L_{\boldsymbol{A}}(\boldsymbol{\eta}, \bar{\boldsymbol{\mu}}) = \max_{\boldsymbol{\mu} \in \mathcal{M}} \left\{ \boldsymbol{\eta}^\top \boldsymbol{\mu} - 1/2 \|\boldsymbol{\mu}\|^2 \right\} - \boldsymbol{\eta}^\top \bar{\boldsymbol{\mu}} + 1/2 \|\bar{\boldsymbol{\mu}}\|^2$$

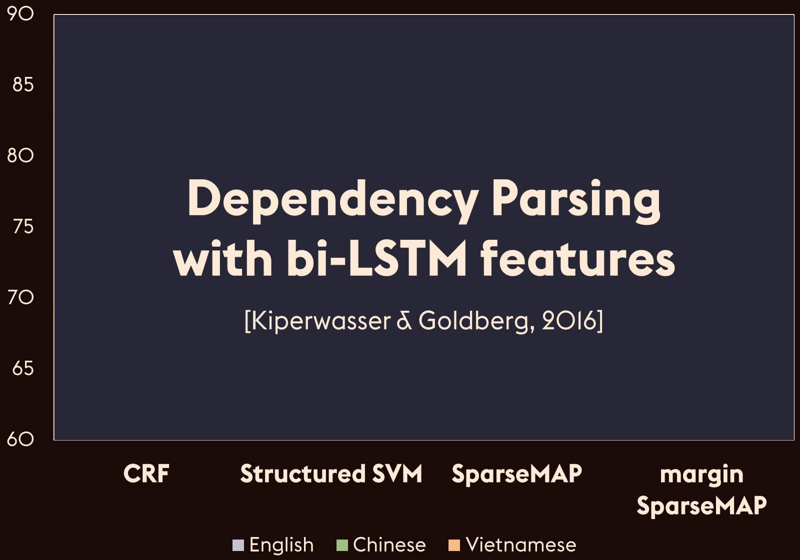
Instance of a structured Fenchel-Young loss, like CRF, SVM, etc. [Blondel, Martins, Niculae '18]

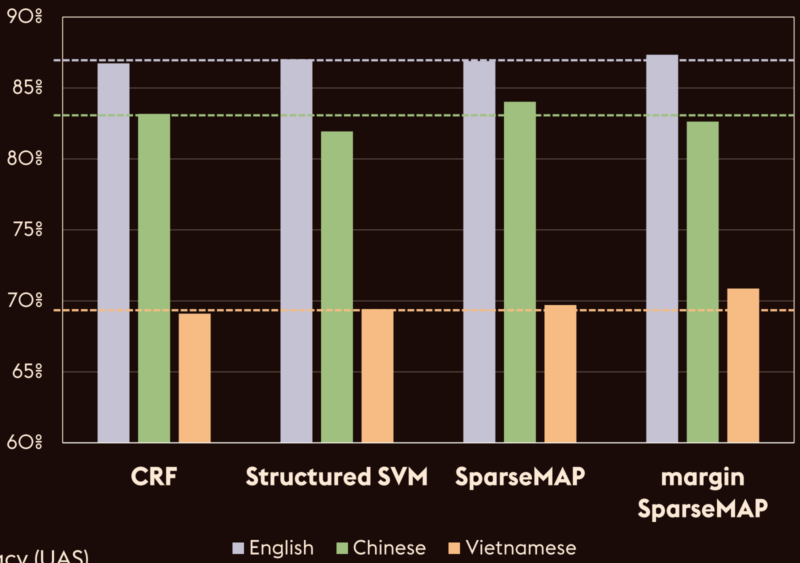
Structured Output Prediction

SparseMAP $$L_{\boldsymbol{A}}(\boldsymbol{\eta}, \bar{\boldsymbol{\mu}}) = \max_{\boldsymbol{\mu} \in \mathcal{M}} \left\{ \boldsymbol{\eta}^\top \boldsymbol{\mu} - 1/2 \|\boldsymbol{\mu}\|^2 \right\} - \boldsymbol{\eta}^\top \bar{\boldsymbol{\mu}} + 1/2 \|\bar{\boldsymbol{\mu}}\|^2$$

cost-SparseMAP $$L_{\boldsymbol{A}}^{\rho}(\boldsymbol{\eta}, \bar{\boldsymbol{\mu}}) = \max_{\boldsymbol{\mu} \in \mathcal{M}} \left\{ \boldsymbol{\eta}^\top \boldsymbol{\mu} - 1/2 \|\boldsymbol{\mu}\|^2 + \rho(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}) \right\} - \boldsymbol{\eta}^\top \bar{\boldsymbol{\mu}} + 1/2 \|\bar{\boldsymbol{\mu}}\|^2$$

Instance of a structured Fenchel-Young loss, like CRF, SVM, etc. [Blondel, Martins, Niculae ’18]

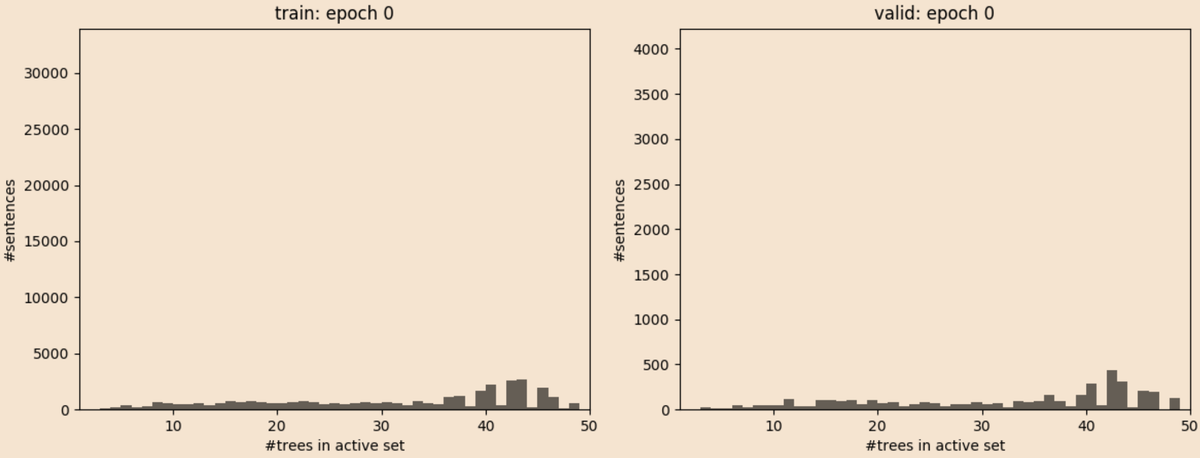

Dependency Parsing
with bi-LSTM features
[Kiperwasser & Goldberg, 2016]
90
85
80
75
70
65
60
CRF
Structured SVM
SparseMAP
margin SparseMAP
English
Chinese
Vietnamese



Unlabeled Accuracy (UAS)
Universal Dependencies dataset

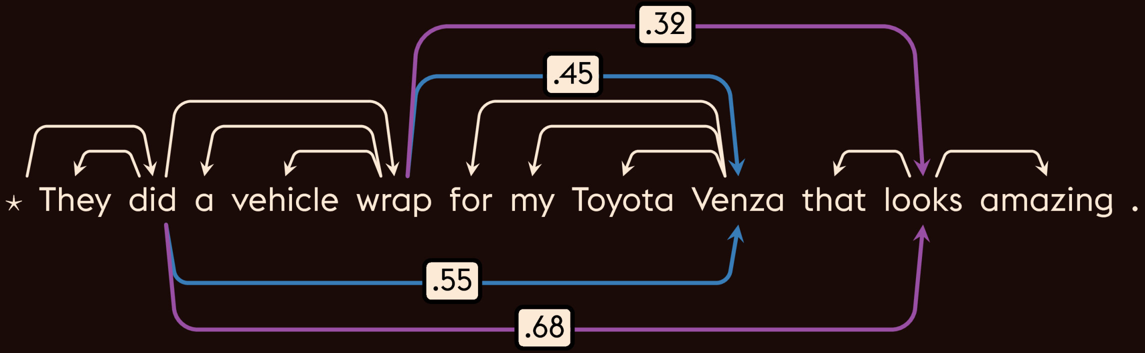
Sparse Structured Output Prediction

As models train, inference gets sparser!



Sparse Structured Output Prediction

Inference captures linguistic ambiguity!



Sparse Structured Output Prediction

Inference captures linguistic ambiguity!