Inference Networks (InfNets) and Structured Prediction Energy Networks (SPENs)

Richard Yuanzhe Pang yzpang.me

09/22/2020

Today's talk: mostly joint works with Lifu Tu, Kevin Gimpel (both currently at TTIC)

→ **Energy-based models**

Inference networks

Structured prediction energy networks

Feed-forward models vs. energy-based models

- Feed-forward model **at inference time**
  - $\boldsymbol{y} = f(\boldsymbol{x})$ where f could be any function (e.g., a complicated neural network)
- Energy-based model **at inference time**

$$\boldsymbol{y} = \operatorname{argmin}_{\boldsymbol{y}'} E(\boldsymbol{x}, \boldsymbol{y}')$$

- Feed-forward model **at training time**
  - One possibility: min $L(f(\boldsymbol{x}), \boldsymbol{y}_{\text{gold}})$ w.r.t. f 's params
- Energy-based model **at training time**
  - Goal: train E's params; more complicated

- Energy-based models
  - Cons: energy functions may be hard to formulate/train
  - Cons: inference may require dynamic programming or gradient descent
  - Pros: can better capture x-y dependencies; for example, multiple y's can be compatible with a single x
  - Pros: can inject expert knowledge to energy function; can produce parsimonious formulation => better generalization and better low-resource performance

Two classes of learning methods for energy-based models

- Contrastive: push down on $E(\boldsymbol{x}_i, \boldsymbol{y}_i)$; push up on other points $E(\boldsymbol{x}_i, \boldsymbol{y}')$
  - Examples: contrastive divergence, metric learning, noise contrastive estimation, generative adversarial networks, denoising auto-encoder, masked auto-encoder
- Architectural methods: build $E(\boldsymbol{x}, \boldsymbol{y})$ such that the volume of the low energy regions is limited or minimized through regularization
  - Examples: sparse coding, sparse auto-encoder, variational auto-encoders (VAEs), etc.

- Today: contrastive methods

based on Yann LeCun's slides

Energy-based models

→ **Inference networks**

Structured prediction energy networks

Inference networks (InfNets)

- Exact inference

$$\hat{\boldsymbol{y}} = \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y})$$

- Gradient descent for inference

$$GD(\boldsymbol{x}) = \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y})$$

- Inference network for inference
  - InfNet (can be a neural network):

$$A_{\Psi} : \mathcal{X} \to \mathcal{Y}_R$$

  - Inference time:

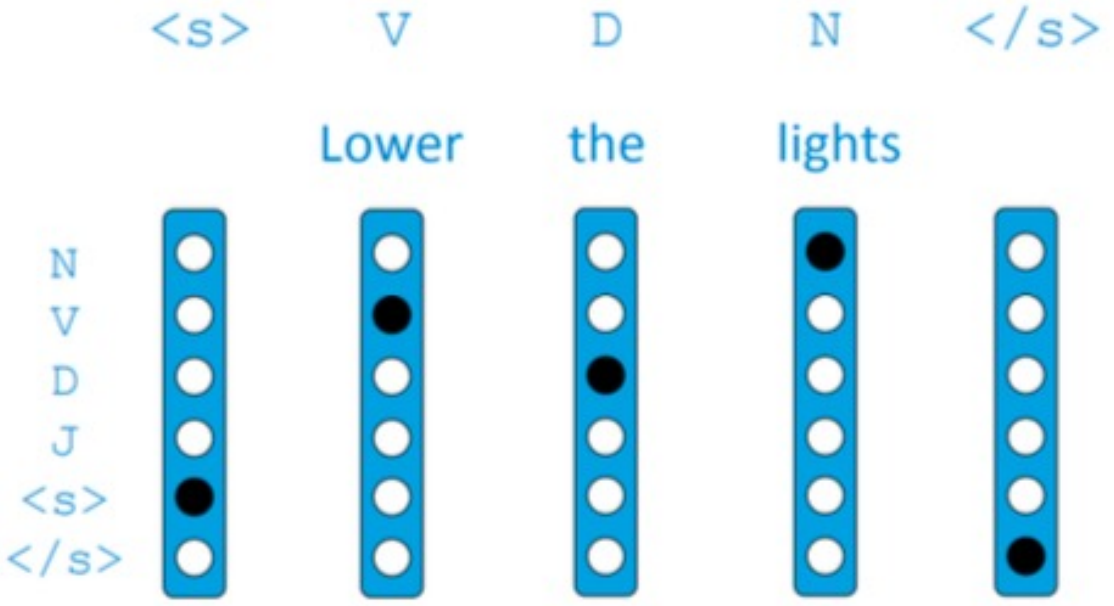
$$A_{\Psi}(\boldsymbol{x}) \approx \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y})$$

  - The question is, how do we train the InfNet?

Aside

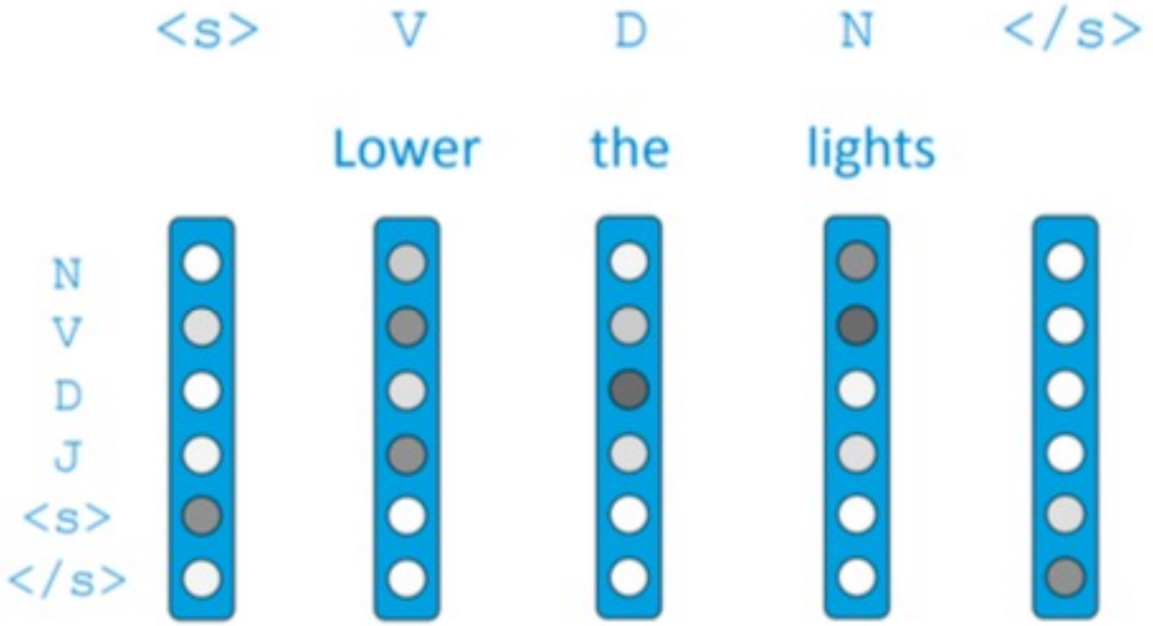
original "hard" space

- use 1 to represent V
- use 2 to represent D, etc.



"relaxed" space

- use a distribution to represent V
- e.g., (0, 1, 0, 0, 0, 0)
- or, (0.1, 0.8, 0.05, 0.05, 0, 0.1))



credit: Lifu Tu and Kevin Gimpel

Inference networks (InfNets)

- Exact inference

$$\hat{\boldsymbol{y}} = \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y})$$

- Gradient descent for inference

$$GD(\boldsymbol{x}) = \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y})$$

- Inference network for inference
  - InfNet (can be a neural network):

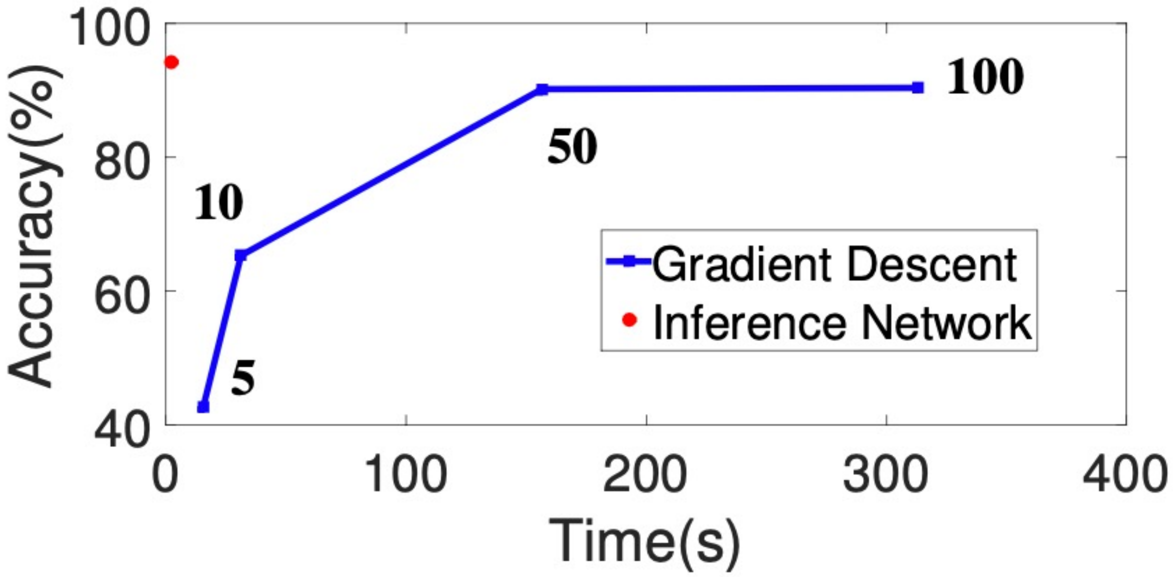
$$A_{\Psi} : \mathcal{X} \to \mathcal{Y}_R$$

  - Inference time:

$$A_{\Psi}(\boldsymbol{x}) \approx \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y})$$

  - The question is, how do we train the InfNet?

Inference network: fast and accurate



Experimental Details

CCG Supertagging with 400 labels
Energy function: BLSTM-CRF
Inference network architecture: BLSTM
Gradient descent run for $\{5, 10, 50, 100\}$ iterations

slide credit: Lifu Tu and Kevin Gimpel

Energy-based models
Inference networks
→ **Structured prediction energy networks**

$? = \text{argmin}_{\boldsymbol{y}'} E(\boldsymbol{x}, \boldsymbol{y}')$

Structured prediction energy networks (SPENs)

- Original SPEN (Belanger and McCallum, 2016)

  - Training:

$$\min_{\Theta} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[\max_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} (\Delta(\boldsymbol{y}, \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)) \right]_+$$

  - Inference:

$$GD(\boldsymbol{x}) = \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x}, \boldsymbol{y})$$

Structured prediction energy networks (SPENs)

- Original SPEN (Belanger and McCallum, 2016)
  - Training:

$$\min_{\Theta} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[\max_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} (\Delta(\boldsymbol{y}, \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)) \right]_+$$

- Approximate inference version
  - Training:

$$\min_{\Theta} \max_{\Phi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} [(\Delta(A_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, A_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i))]_+$$

Joint training of SPEN and InfNet

- Training:

$$\min_{\Theta} \max_{\Phi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} [(\Delta(A_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, A_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i))]_+$$

cost-augmented InfNet

test-time InfNet

- Inference:

$$A_\Psi(\boldsymbol{x})$$

Recall that InfNet is designed to approximate as follows

$$A_\Psi(\boldsymbol{x}) \approx \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x}, \boldsymbol{y})$$

Joint training of SPEN and InfNet

- Training:

$$\min_{\Theta} \max_{\Phi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[(\Delta(A_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, A_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)) \right]_+$$

- Fine-tuning InfNet parameters:

$$\hat{\Psi} \leftarrow \hat{\Phi}$$

$$\hat{\Psi} = \mathrm{argmin}_{\Psi} E_\Theta(\boldsymbol{x}, A_\Psi(\boldsymbol{x}))$$

- Inference:

$$A_\Psi(\boldsymbol{x})$$

- Cost-augmented inference

$$\mathrm{argmin}_{\boldsymbol{y}'} (E_\Theta(\boldsymbol{x}, \boldsymbol{y}') - \Delta(\boldsymbol{y}', \boldsymbol{y}))$$

- Test-time inference

$$\mathrm{argmin}_{\boldsymbol{y}'} E_\Theta(\boldsymbol{x}, \boldsymbol{y}')$$

Joint training of SPEN and (cost-augmented InfNet and test-time InfNet)

- Training:

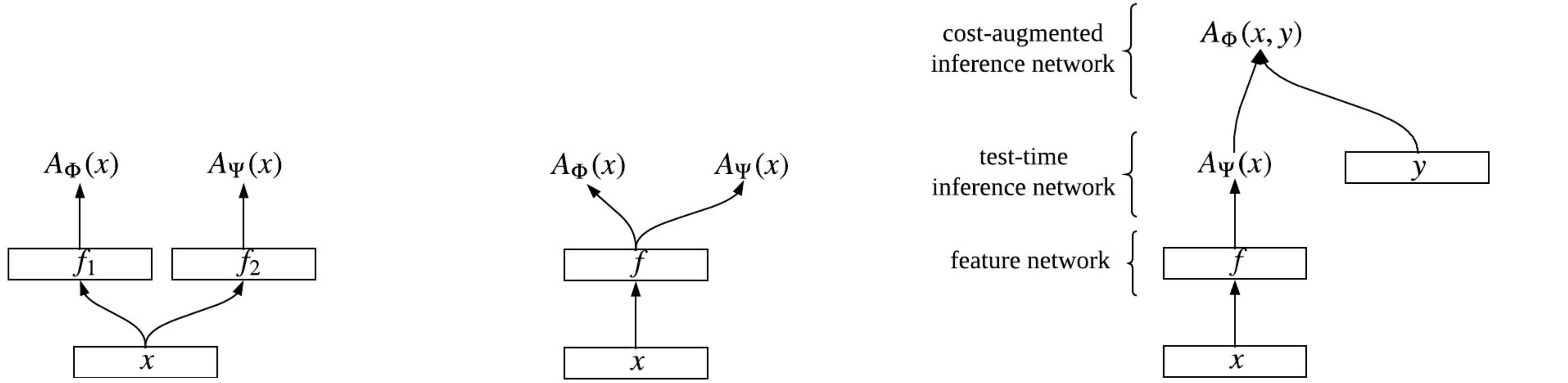
$$\min_{\Theta} \max_{\Phi,\Psi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[(\Delta(A_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, A_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)) \right]_+ + \lambda \left[-E_\Theta(\boldsymbol{x}_i, A_\Psi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i) \right]_+$$

- Inference:

$$A_\Psi(\boldsymbol{x})$$

- An example energy function for sequence labeling

$$E_\Theta(\boldsymbol{x}, \boldsymbol{y}) = -\left(\sum_{t=1}^{T} \sum_{j=1}^{L} y_{t,j} (U_j^\top b(\boldsymbol{x}, t)) + \sum_{t=1}^{T} \boldsymbol{y}_{t-1}^\top W \boldsymbol{y}_t \right)$$



(a) separated networks

(b) shared feature networks

(c) stacked networks with y as extra input to A_Φ

- Training:

$$\min_{\Theta} \max_{\Phi,\Psi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[(\Delta(A_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, A_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)) \right]_+ \\ + \lambda \left[-E_\Theta(\boldsymbol{x}_i, A_\Psi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i) \right]_+$$

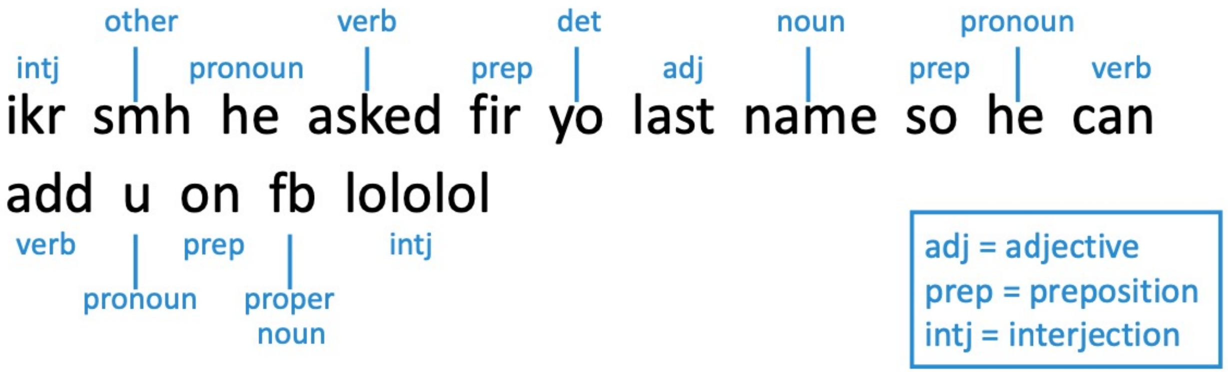
- Inference:

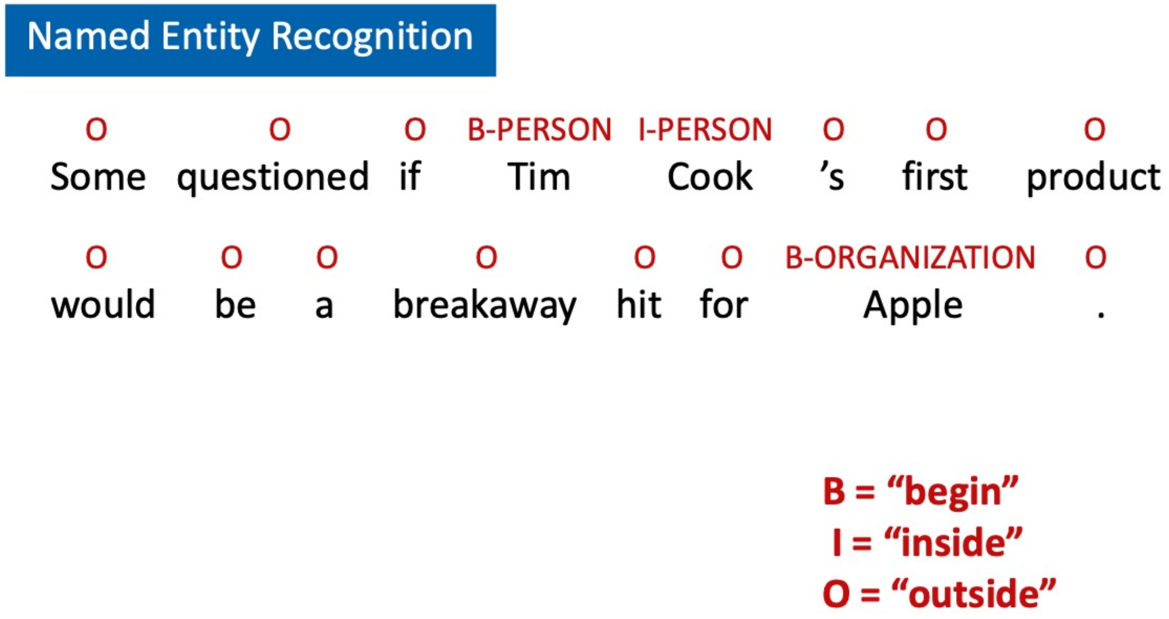
$$A_\Psi(\boldsymbol{x})$$

Experimental setup

1. Part-of-speech (POS) tagging
2. Named entity recognition (NER) labeling
3. Constituency parsing (skip)

- Only experimented on **sequence labeling** tasks. Tu and Gimpel (2018) has some multi-label classification results.
- Note that although these are NLP experiments, the approach can be applied to structured prediction tasks in general.





Credit: examples by Prof. Kevin Gimpel

Our hypotheses

1. **(InfNet + SPEN) > MLE**
2. (Cost-augmented InfNet + test-time InfNet + SPEN) > (InfNet + SPEN)
3. Tag language models help

	POS				NER				NER+
	acc (%)	$\|T\|$	$\|I\|$	speed	F1 (%)	$\|T\|$	$\|I\|$	speed	F1 (%)
BiLSTM	88.8	166K	166K	–	84.9	239K	239K	–	89.3

margin-rescaled	89.4	333K	166K	–	85.2	479K	239K	–	89.5
perceptron	88.6	333K	166K	–	84.4	479K	239K	–	89.0

Our hypotheses

1. (InfNet + SPEN) > MLE
2. **(Cost-augmented InfNet + test-time InfNet + SPEN) > (InfNet + SPEN)**
3. Tag language models help

	POS				NER				NER+
	acc (%)	$\|T\|$	$\|I\|$	speed	F1 (%)	$\|T\|$	$\|I\|$	speed	F1 (%)
BiLSTM	88.8	166K	166K	–	84.9	239K	239K	–	89.3
margin-rescaled	89.4	333K	166K	–	85.2	479K	239K	–	89.5
perceptron	88.6	333K	166K	–	84.4	479K	239K	–	89.0
SPENs with inference networks, compound objective, CE, no zero truncation (this paper):									
separated	89.7	500K	166K	66	85.0	719K	239K	32	89.8
shared	89.8	339K	166K	78	85.6	485K	239K	38	90.1
stacked	**89.8**	**335K**	**166K**	**92**	**85.6**	**481K**	**239K**	**46**	**90.1**

Our hypotheses

1. (InfNet + SPEN) > MLE
2. (Cost-augmented InfNet + test-time InfNet + SPEN) > (InfNet + SPEN)
3. **Tag language models help**

Joint training of SPEN and (cost-augmented InfNet and test-time InfNet)

- Training: $$\min_{\Theta} \max_{\Phi,\Psi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} [(\Delta(A_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, A_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i))]_+ + \lambda [-E_\Theta(\boldsymbol{x}_i, A_\Psi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)]_+$$
- Inference: $A_\Psi(\boldsymbol{x})$

- An example energy function for sequence labeling

$$E_\Theta(\boldsymbol{x}, \boldsymbol{y}) = -\left(\sum_{t=1}^{T} \sum_{j=1}^{L} y_{t,j} (U_j^\top b(\boldsymbol{x}, t)) + \sum_{t=1}^{T} \boldsymbol{y}_{t-1}^\top W \boldsymbol{y}_t \right)$$

$$\overline{\boldsymbol{y}_t} = h(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{t-1})$$

$$\overline{\boldsymbol{y}_t} = h(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{y}_0, \ldots, \boldsymbol{y}_{t-1})$$

$$E^{\mathrm{TLM}}(\boldsymbol{y}) = -\sum_{t=1}^{T+1} \log(\boldsymbol{y}_t^\top \overline{\boldsymbol{y}}_t)$$

Our hypotheses

1. (InfNet + SPEN) > MLE
2. (Cost-augmented InfNet + test-time InfNet + SPEN) > (InfNet + SPEN)
3. **Tag language models help**

	NER	NER+	NER++
margin-rescaled	85.2	89.5	90.2
compound, stacked, CE, no truncation	85.6	90.1	90.8
+ global energy GE(c)	**86.3**	**90.4**	**91.0**

Table 4: NER test F1 scores with global energy terms.

Conclusions and thoughts

1. SPENs are powerful but learning and inference are hard
2. Inference networks can make it easier and more efficient to use SPENs
3. Separating inference networks for the two inference problems (cost-augmented and test-time inference) improves accuracy and leads to complementary functionality
4. Adding global energy terms leads to further improvements
5. Next step: move to generation tasks and model other types of data (not only sequence labeling)