Benchmarking Approximate Inference Methods for Neural Structured Prediction

Lifu Tu **Kevin Gimpel**

Toyota Technological Institute at Chicago

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO

Overview

- Structured prediction is challenging due to exponentially-large output spaces.
- How to speed up the inference process? CRF layers are popular in sequence labeling tasks. However, it is slow when there is a large label space.
- Two approximate inference methods that we compare: **gradient descent and inference networks**[1]
- Inference networks achieve a better speed/accuracy/search error trade off than gradient descent.

Sequence Models

Conditional random fields(CRFs) define an energy function:

input sentence **input feature vector** **label transition matrix**

$$E_\Theta(\boldsymbol{x},\boldsymbol{y}) = -\left(\sum_t \sum_{i=1}^{L} y_{t,i}\left(U_i^\top f(\boldsymbol{x},t)\right) + \sum_t y_{t-1}^\top W y_t\right)$$

structure output sequence **L-dimensional one-hot vector**

using a pretrained BLSTM-CRF

Inference Methods

Exact(Viterbi) $$\operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} E_\Theta(\boldsymbol{x},\boldsymbol{y})$$

Gradient Descent $$\operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x},\boldsymbol{y})$$

relaxed continuous output space

Inference Network[1] $$\mathbf{A}_\Psi(\boldsymbol{x}) \approx \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x},\boldsymbol{y})$$

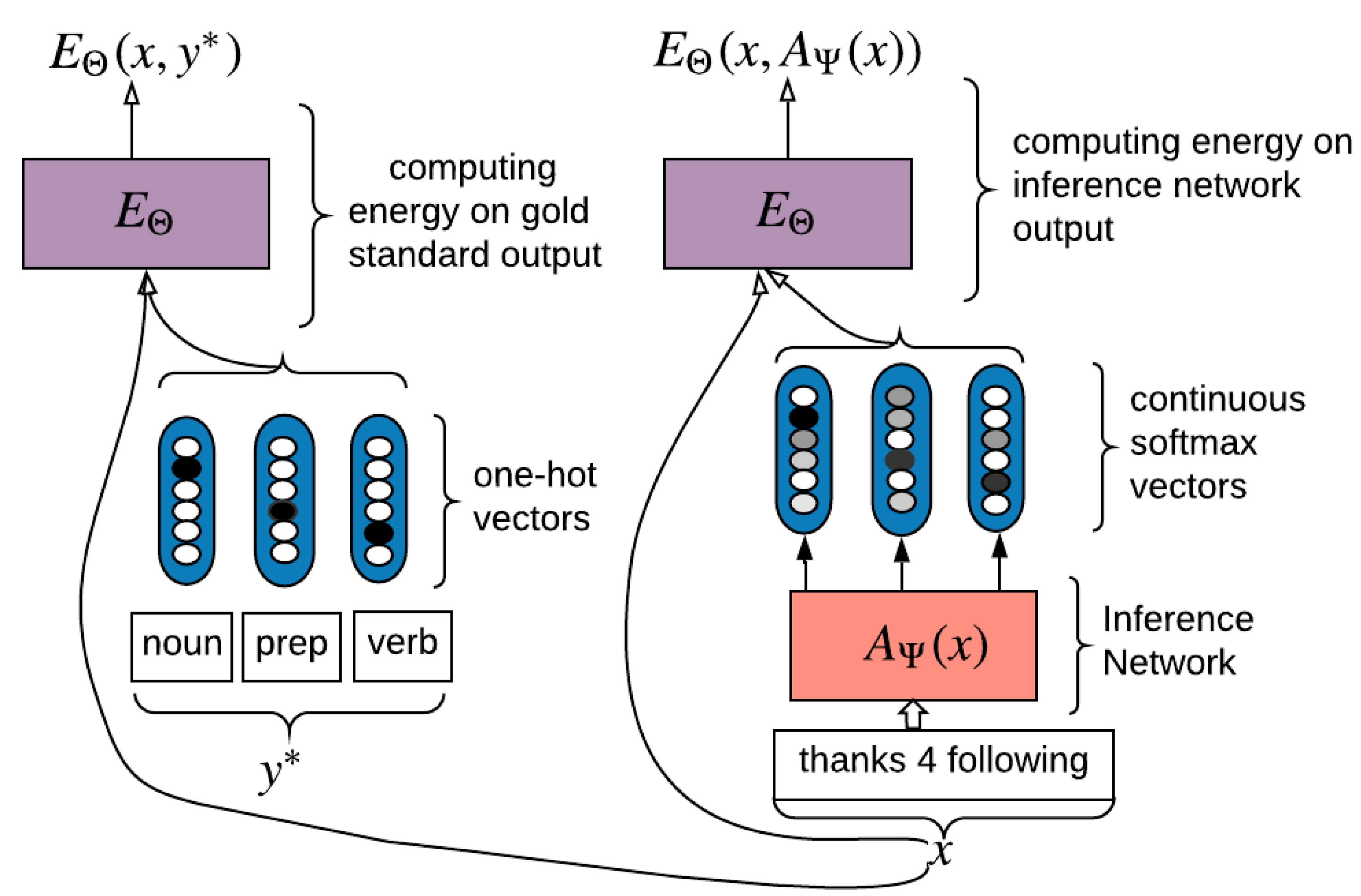
Inference Network Training

We use multi-task learning while training the **inference network**:

$$\operatorname*{argmin}_{\Psi} \sum_{\langle \boldsymbol{x},\boldsymbol{y}\rangle} E_\Theta(\boldsymbol{x},\mathbf{A}_\Psi(\boldsymbol{x})) + \lambda \ell_{\text{token}}(\boldsymbol{y},\mathbf{A}_\Psi(\boldsymbol{x}))$$

Sum of the Cross Entropy Loss at Each Position

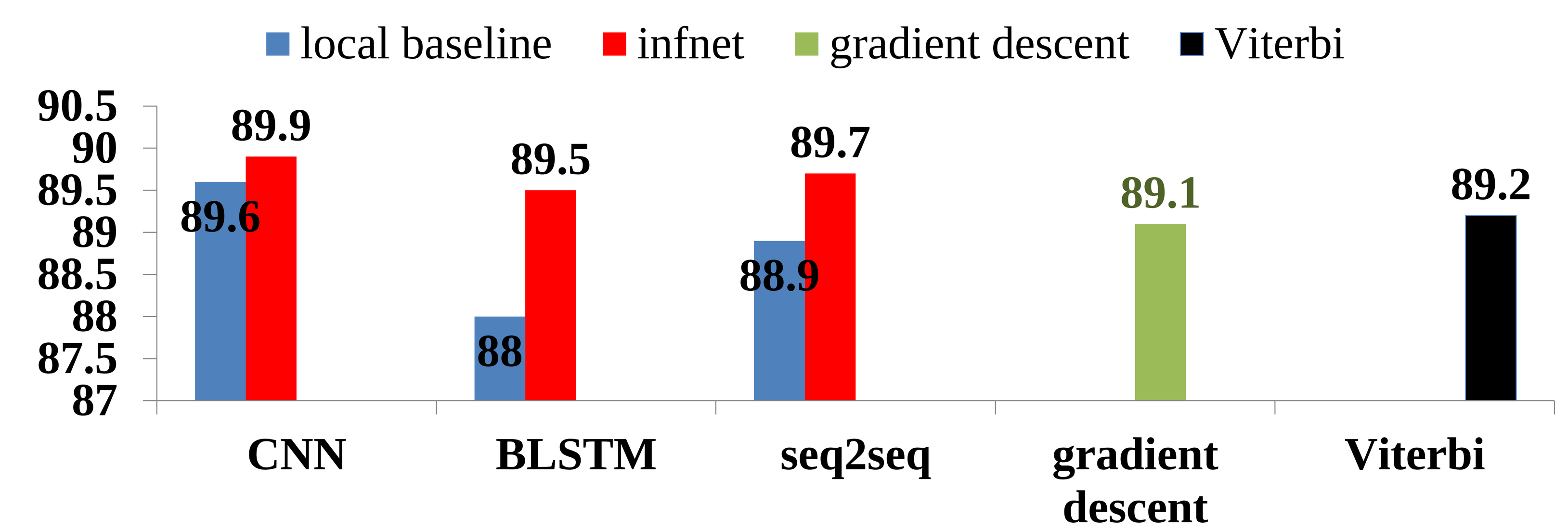
Framework



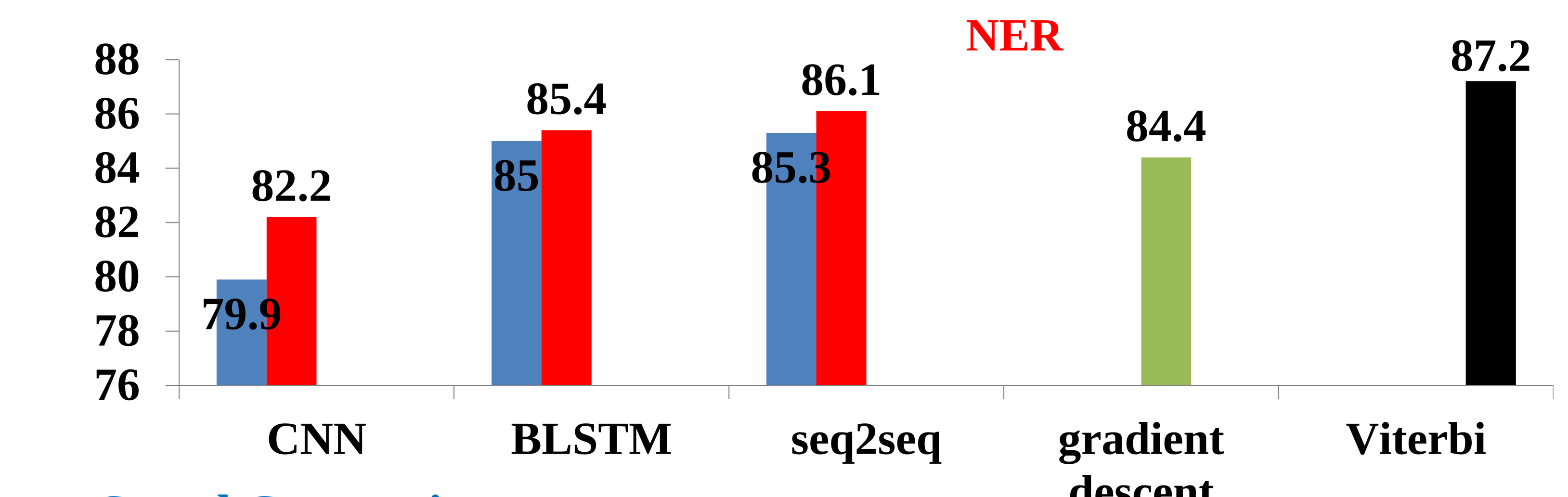
BLSTM-CRF Results For Different Inference Network Architectures, gradient descent and Viterbi

Performance Comparison

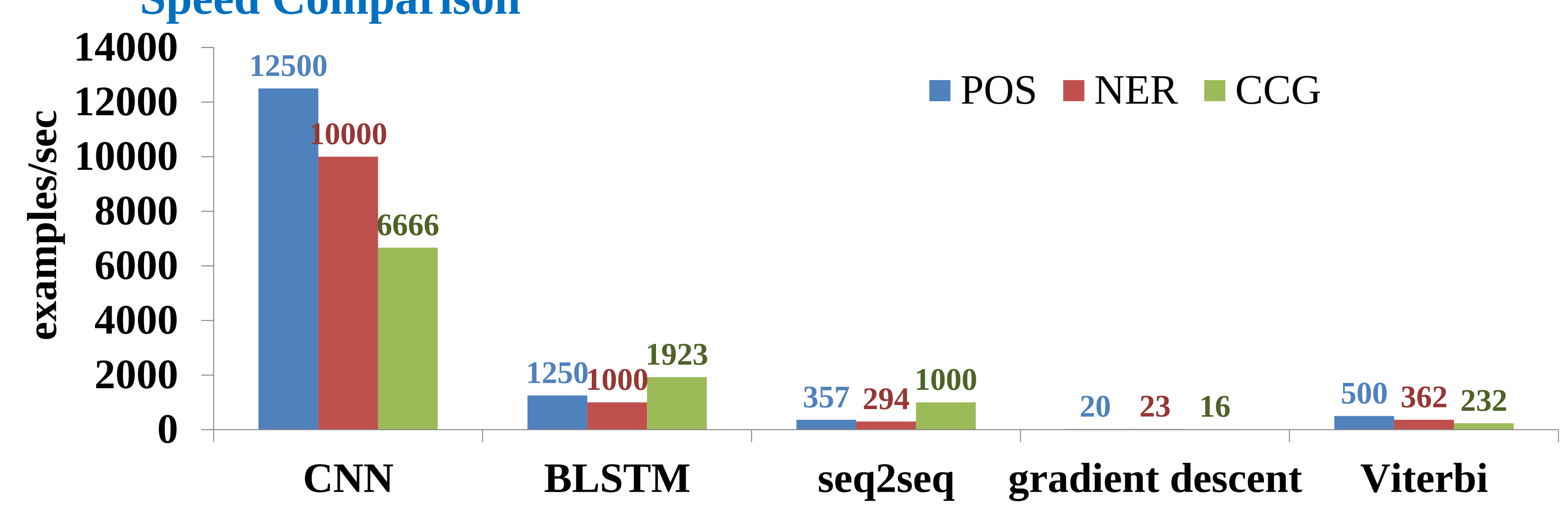
Twitter POS Tagging



NER



Speed Comparison



Three different inference network architectures

BLSTM-CRF+ Results

Additional techniques for improving the performance: Word Embedding Fine-Tuning, Character-Based Embedding, Dropout

BLSTM-CRF+: BLSTM-CRF with the above techniques
Infnet+: inference networks with the above techniques

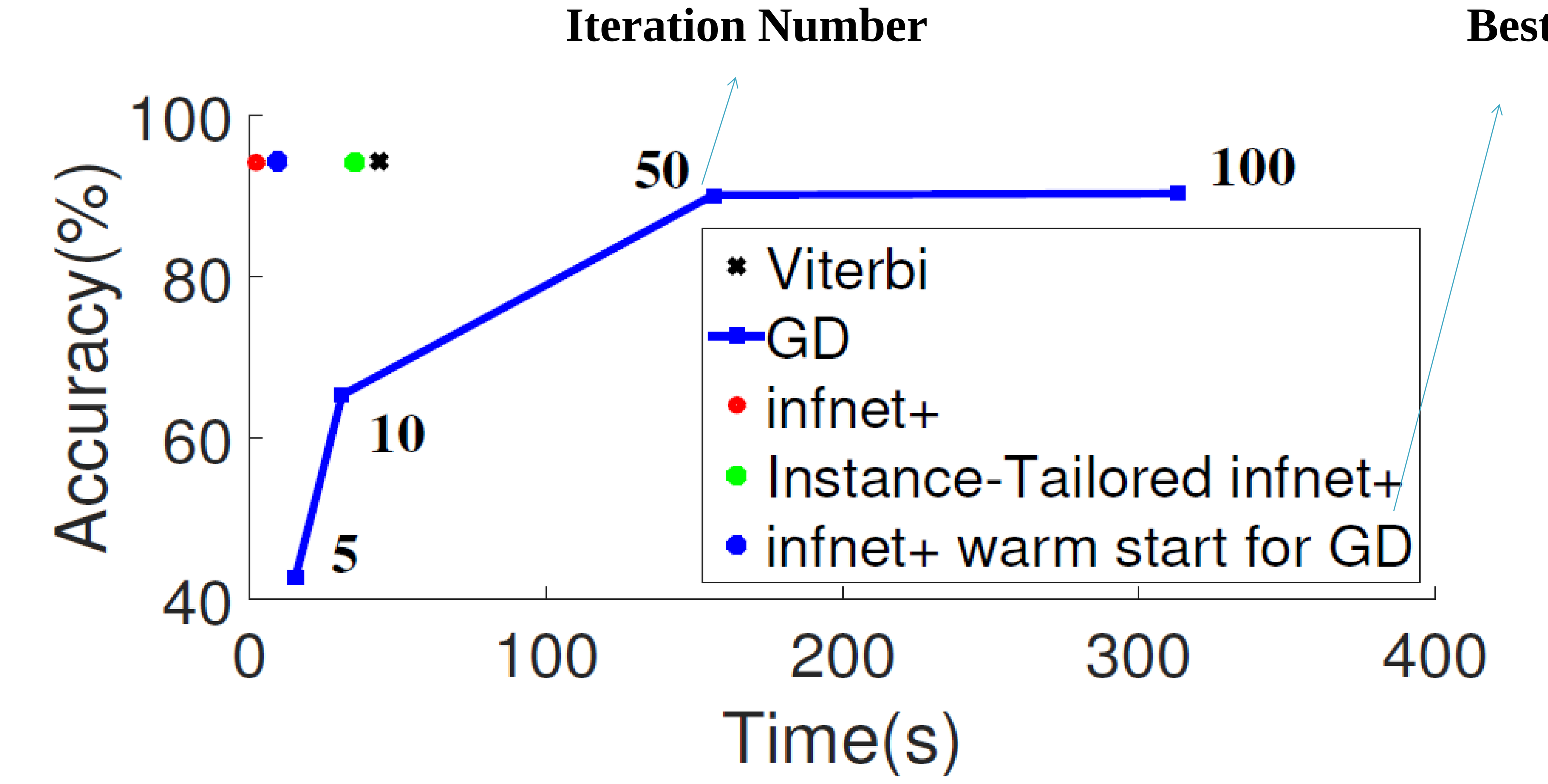
	POS	NER	CCG
local baseline	91.3	90.5	94.1
infnet+	91.3	90.8	94.2
gradient descent	90.8	89.8	90.4
Viterbi	90.9	91.6	94.3

	F1
local baseline(1-layer BLSTM)	90.3
Infnet+(1-layer BLSTM)	90.7
Infnet+(2-layer BLSTM)	91.1
Viterbi	91.6

Search Error Comparison

		Twitter POS Tagging		NER	
		Accuracy	Energy	F1	Energy
gold standard		100	-159.65	100	-230.63
Viterbi (BLSTM-CRF+)		90.9	-163.20	91.6	-231.53
gradient descent	10	89.2	-161.69	81.9	-227.92
	20	90.8	-163.06	89.6	-231.17
	30	90.7	-163.02	89.8	-231.30
infnet+		91.3	-162.59	90.8	-231.19
discretized output from infnet+		91.3	-160.87	90.8	-231.34
instance-tailored infnet+	10	91.3	-162.85	91.5	-231.39
Infnet+ as warm start for gradient descent	10	91.2	-163.15	91.5	-231.46

- **For POS, the inference network does not have lower energy but with higher performance due to the multi-task learning**
- **Instance tailoring and warm starting lead to lower energies and better performance than infnet+**



CCG Supertagging with 400 labels

- **Inference networks achieve a better speed/accuracy/search error trade off than gradient descent.**
- **Combining inference networks and gradient descent gets further benefit.**

References

1. Lifu Tu, Kevin Gimpel. Learning Approximate Inference Networks for Structured Prediction. ICLR 2018