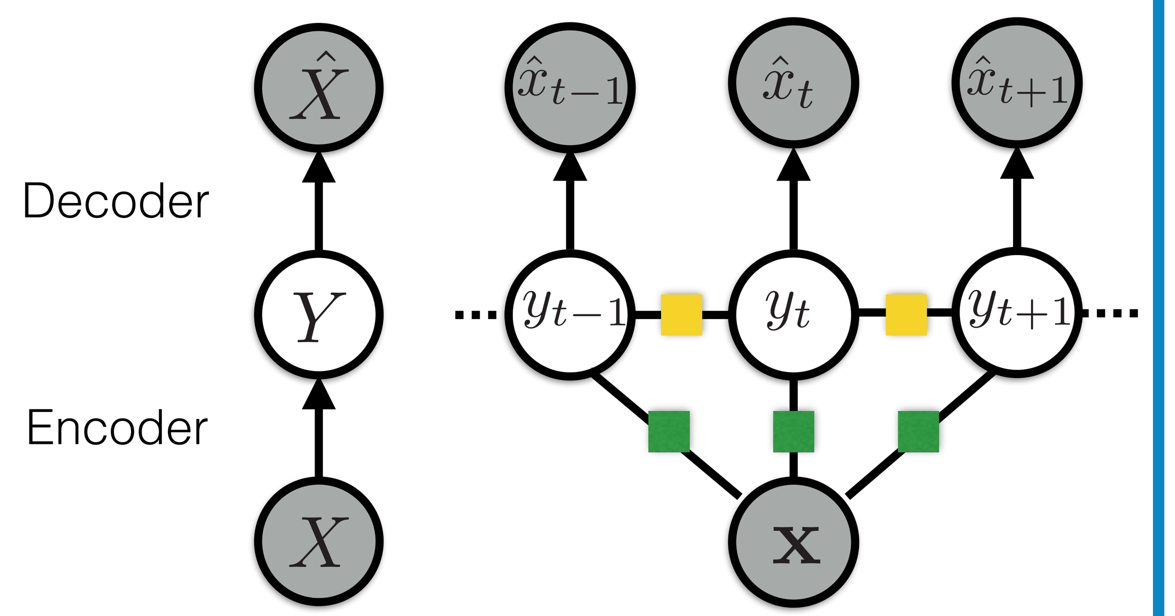
Semi-Supervised Structured Prediction with Neural CRF Autoencoder

Xiao Zhang[1], *Yong Jiang*[2], *Hao Peng*[1], *Kewei Tu*[2] and *Dan Goldwaswer*[1]
Purdue University[1], ShanghaiTech University[2]

emnlp2017

PURDUE UNIVERSITY

Model

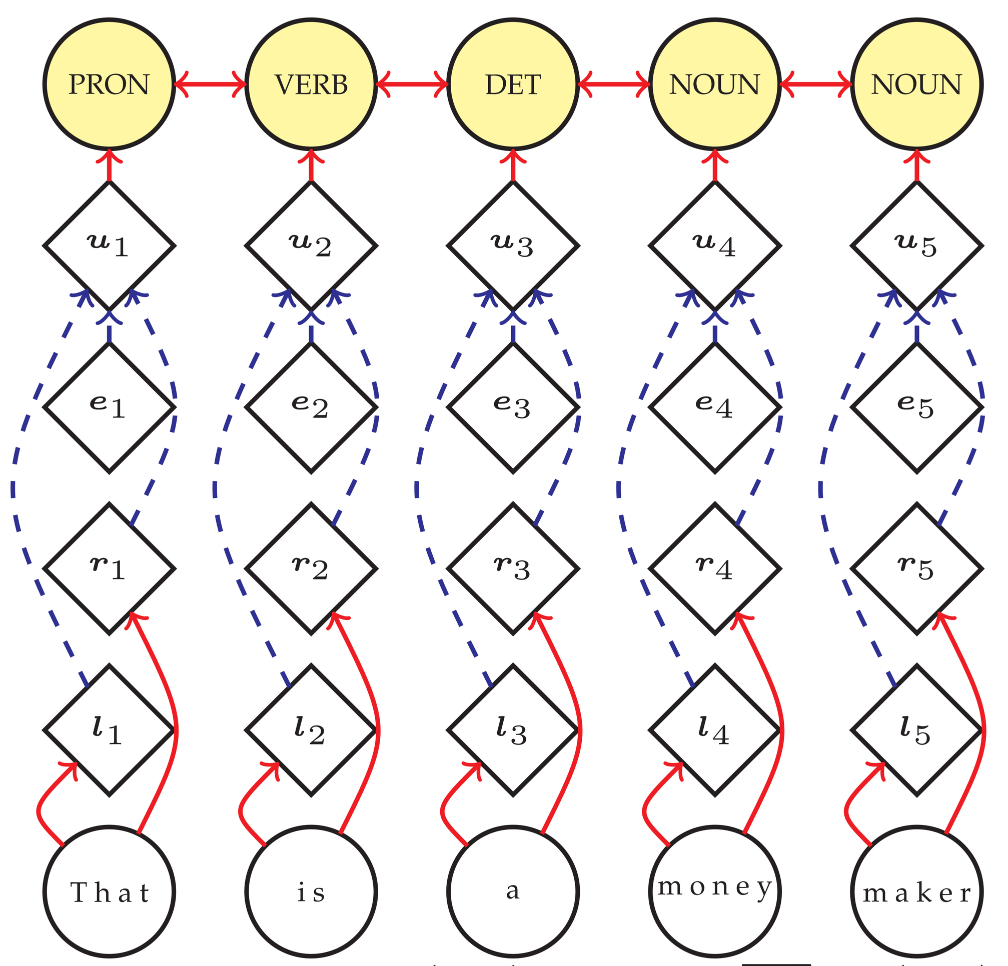


From autoencoder (left) to NCRF-AE (right).

- A generalized autoencoder (AE) models the conditional distribution $P(\hat{X}|X)$.
- A CRF autoencoder (CRF-AE) extends AE by modeling a sequence $P(\hat{\boldsymbol{x}}|\boldsymbol{x})$, using a CRF as an encoder and a generative model as a decoder.
- A neural CRF autoencoder (NCRF-AE) extends CRF-AE by using deep neural networks as feature extractors in the CRF encoder with the mixed EM algorithm.

$$P_{\Theta,\Lambda}(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \sum_{\boldsymbol{y}} P_{\Theta,\Lambda}(\hat{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{y}} \underbrace{P_{\Theta}(\hat{\boldsymbol{x}}|\boldsymbol{y})}_{\text{Decoder}} \underbrace{P_{\Lambda}(\boldsymbol{y}|\boldsymbol{x})}_{\text{Encoder}}.$$

Neural CRF encoder:



$$P_{\Lambda}(\boldsymbol{y}|\boldsymbol{x}) = e^{\Phi(\boldsymbol{x},\boldsymbol{y})}/Z, Z = \sum_{\tilde{\boldsymbol{y}}} e^{\Phi(\boldsymbol{x},\tilde{\boldsymbol{y}})};$$

$$\Phi(\boldsymbol{x},\boldsymbol{y}) = \sum_t \phi(\boldsymbol{x}, y_t) + \psi(y_{t-1}, y_t).$$

Abstract

We propose an end-to-end neural CRF autoencoder (NCRF-AE) model for semi-supervised learning of sequential structured prediction problems. Our NCRF-AE consists of two parts: an encoder which is a CRF model enhanced by deep neural networks, and a decoder which is a generative model trying to reconstruct the input. Our model has a unified structure with different loss functions for labeled and unlabeled data with shared parameters. We developed a variation of the EM algorithm for optimizing both the encoder and the decoder simultaneously by decoupling their parameters. Our experimental results over the Part-of-Speech (POS) tagging task on eight different languages show that the NCRF-AE model can outperform competitive systems in both supervised and semi-supervised scenarios.

Learning

Unified learning framework Loss functions for labeled and unlabeled data with *shared parameters*:
Labeled data: $loss_l = -\log P_{\Theta,\Lambda}(\hat{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{x})$; Unlabeled data: $loss_u = -\log P_{\Theta,\Lambda}(\hat{\boldsymbol{x}}|\boldsymbol{x})$.

Parameter learning using EM *Decoupled parameters* update in the E and M steps respectively:

$$\text{E:} \quad \sum_i \log P(\hat{\boldsymbol{x}}_i|\boldsymbol{x}_i) \geq \sum_i \sum_{\boldsymbol{y}_i} Q(\boldsymbol{y}_i) \log \frac{P(\hat{\boldsymbol{x}}_i, \boldsymbol{y}_i|\boldsymbol{x}_i)}{Q(\boldsymbol{y}_i)}; \quad \text{M:} \quad \arg\max_{\Theta^{(t)}} \sum_{y\to x} \log \theta^{(t)}_{y\to x} \mathrm{E}_{y\sim Q}[C(y,x)] \quad s.t. \sum_x \theta^{(t)}_{y\to x} = 1.$$

Decoding using Viterbi $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} P_{\Theta,\Lambda}(\hat{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{x})$.

Algorithm 1 Obtain Expected Count (T_e)

Require: the expected count table T_e
for an unlabeled data example $\boldsymbol{x}_i$ **do**
  Compute the forward messages: $\alpha(y,t) \quad \forall y,t$. ▷ t is the position in a sequence.
  Compute the backward messages: $\beta(y,t) \quad \forall y,t$.
  Calculate the expected count for each x in $\boldsymbol{x}_i$: $P(y_t|x_t) \propto \alpha(y,t) \times \beta(y,t)$.
  $T_e(x_t, y_t) \leftarrow T_e(x_t, y_t) + P(y_t|x_t)$ ▷ T_e is the expected count table.
end for

Algorithm 2 Mixed Expectation-Maximization

Initialize expected count table T_e using labeled data $\{\boldsymbol{x},\boldsymbol{y}\}^l_i$ and use it as $\Theta^{(0)}$ in the decoder.
Initialize $\Lambda^{(0)}$ in the encoder randomly.
for t in *epochs* **do**
  Train the encoder on labeled data $\{\boldsymbol{x},\boldsymbol{y}\}^l$ and unlabeled data $\{\boldsymbol{x}\}^u$ to update $\Lambda^{(t-1)}$ to $\Lambda^{(t)}$.
  Re-initialize expected count table T_e with 0s.
  Use labeled data $\{\boldsymbol{x},\boldsymbol{y}\}^l$ to calculate real counts and update T_e.
  Use unlabeled data $\{\boldsymbol{x}\}^u$ to compute the expected counts with parameters $\Lambda^{(t)}$ and $\Theta^{(t-1)}$ and update T_e.
  Obtain $\Theta^{(t)}$ globally and analytically based on T_e.
end for

Results

Models (Supervised)	English	French	German	Italian	Russian	Spanish	Indonesian	Croatian
HMM	86.28%	91.23%	85.59%	92.03%	79.82%	91.31%	89.40%	86.98%
CRF	89.96%	93.40%	86.83%	94.07%	83.38%	91.47%	88.63%	86.90%
LSTM	90.50%	94.16%	88.40%	94.96%	84.87%	93.17%	89.42%	88.95%
NCRF	91.52%	95.07%	90.27%	96.20%	93.37%	93.34%	92.32%	93.85%
NCRF-AE	**92.50%**	**95.28%**	**90.50%**	**96.64%**	**93.60%**	**93.86%**	**93.96%**	**94.32%**

Models (Semi-supervised)	English	French	German	Italian	Russian	Spanish	Indonesian	Croatian
NCRF-AE (Only Labeled)	88.41%	93.69%	90.75%	92.17%	87.82%	91.70%	89.06%	87.92%
HMM-EM	79.92%	88.15%	77.01%	84.57%	72.96%	86.77%	83.61%	77.20%
NCRF-AE (Hard EM)	86.79%	92.83%	89.78%	90.68%	86.39%	91.30%	88.86%	86.55%
NCRF-AE	**89.43%**	**93.89%**	**90.99%**	**92.85%**	**88.93%**	**92.17%**	**89.41%**	**89.14%**

Experiments

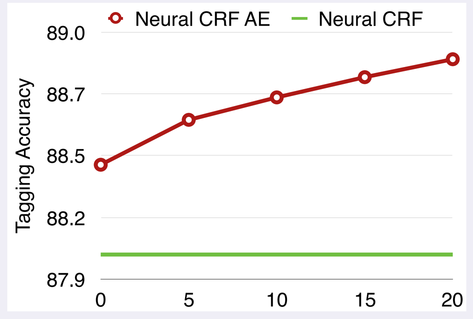
We evaluated our model on the POS tagging task, in both the supervised and semi-supervised learning settings, over 8 different languages from the UD (Universal Dependencies) 1.4 dataset.

Error analysis: an example form the test set

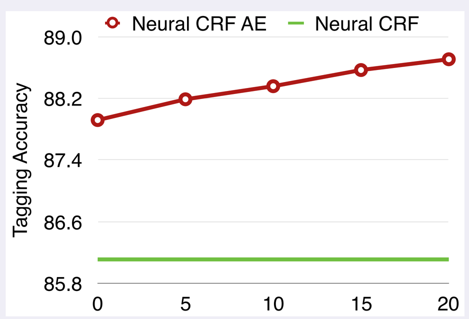
Text	Google	is	a	nice	search	engine	.
Gold	PROPN	VERB	DET	ADJ	NOUN	NOUN	PUNCT
NCRF-AE	PROPN	VERB	DET	ADJ	NOUN	NOUN	PUNCT
NCRF	NOUN	VERB	DET	ADJ	NOUN	NOUN	PUNCT
LSTM	PROPN	VERB	DET	ADJ	VERB	NOUN	PUNCT

Semi-supervised learning effect

UD POS tagging accuracy versus increasing proportion of unlabeled sequences using 20% labeled data. The green straight line is the performance of the neural CRF trained over the labeled data.



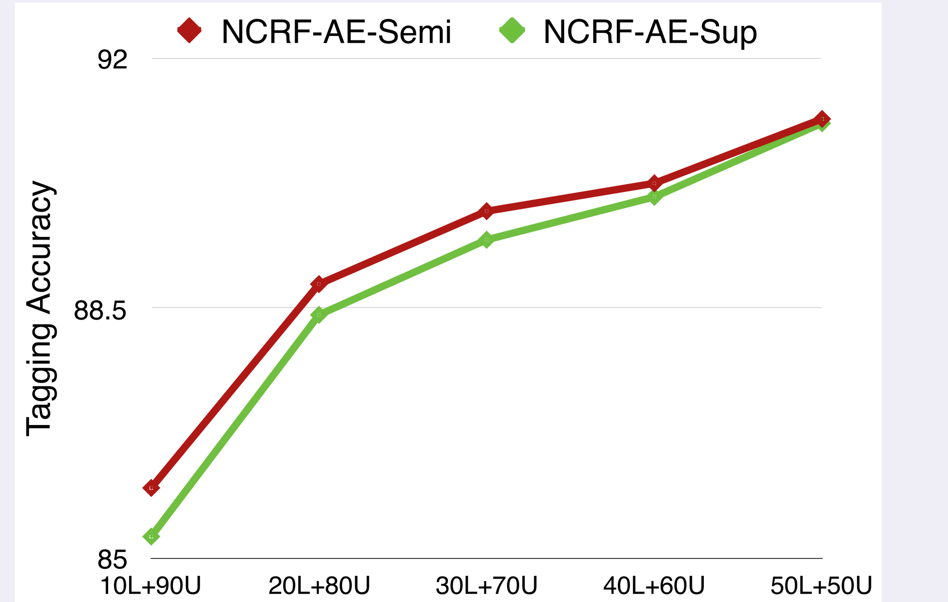
English



Croatian

Varying sizes of labeled data on English

We gradually increased the proportion of labeled data, and in accordance decreased the proportion of unlabeled.



Future Work

- Use embeddings for POS tags to compute both the transition score and the generative decoder score.
- Add a prior for predicted labels and cast it into the variational inference framework.