An Exploration of Arbitrary-Order Sequence Labeling via Energy-Based Inference Networks

Lifu Tu*[1] Tianyu Liu*[2] Kevin Gimpel[1]

[1]Toyota Technological Institute at Chicago

[2]Peking University

- Applying deep representation learning is popular to structured tasks.
  - DNN, LSTM, CNN, BERT, etc.
- Structured component is usually quite simplistic
  - Independent assumption
  - Linear chain CRF (first-order model) [Lafferty et al, 2001]
- Challenge with high-order model: time complexity of training and inference grow exponentially

Why global energies are still necessary?

[Finkel et al, 2005]

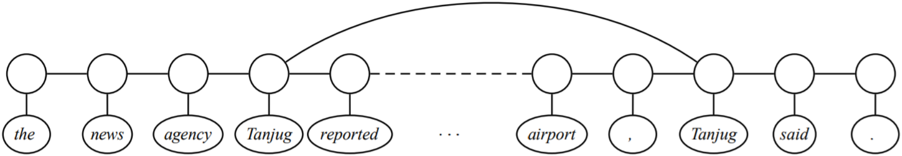


Figure: An example from CoNLL 2003 Named Entity Recognition

Enable label consistency

[Holtzman et al., 2020]

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Figure: Generated outputs from GPT-2 large language model.

Avoid repetition and incoherence

Training and Inference for Structured Models

Given an input sequence $\boldsymbol{x} = \langle x_1, x_2, ..., x_{|\boldsymbol{x}|} \rangle$, we wish to output a sequence $\boldsymbol{y} = \langle y_1, y_2, ..., y_{|\boldsymbol{y}|} \rangle \in \mathcal{Y}(\boldsymbol{x})$. $\mathcal{Y}(\boldsymbol{x})$ is the structured output space.

$$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}} E_\Theta(\boldsymbol{x}, \boldsymbol{y})$$

Where energy function $E_\Theta(\boldsymbol{x}, \boldsymbol{y})$ is a scalar that measures the compatibility of each configuration $\boldsymbol{x}$ and $\boldsymbol{y}$ [LeCun et al., 2006; Belanger and McCallum, 2016]

Inference for Structured Models

$$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}} E_\Theta(\boldsymbol{x}, \boldsymbol{y})$$

Gradient Descent for Inference

$$GD(\boldsymbol{x}) = \arg\min_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x}, \boldsymbol{y}).$$

Inference Networks [Tu et al., 2018]

A test-time inference network $\mathbf{A}_\Psi :\rightarrow \mathcal{Y}_R$ is parameterized by Ψ and trained with the goal that

$$\mathbf{A}_\Psi(\boldsymbol{x}) \approx \arg\min_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x}, \boldsymbol{y}).$$

- Achieving a better speed/accuracy/search error trade-off than gradient descent
- Faster than exact inference at similar accuracy levels

Training Objective

Learning the energy function and inference network jointly [Tu et al, 2018, 2020] :

$$\hat{\Theta}, \hat{\Phi}, \hat{\Psi} = \min_{\Theta} \max_{\Phi, \Psi} \sum_i \underbrace{[\triangle(\mathbf{F}_\Phi(\boldsymbol{x}), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \mathbf{F}_\Phi(\boldsymbol{x})) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)]_+}_{\text{margin-rescaled loss}}$$
$$+ \lambda \underbrace{[-E_\Theta(\boldsymbol{x}_i, \mathbf{A}_\Psi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)]_+}_{\text{perceptron loss}}$$

cost-augmented inference: $\mathbf{F}_\Phi \approx \arg\min_{\boldsymbol{y}'} (E_\Theta(\boldsymbol{x}, \boldsymbol{y}') - \triangle(\boldsymbol{y}', \boldsymbol{y}))$,

test-time inference: $\mathbf{A}_\Psi \approx \arg\min_{\boldsymbol{y}'} E_\Theta(\boldsymbol{x}, \boldsymbol{y}')$.

Setting

Θ: energy function

Φ: cost-augmented inference network

Ψ: test-time inference network

GAN Objective

Alternately optimize Θ, Φ and Φ (like adversarial training)

- Optimization is a min-max game.
- Inference network is analogous to the generator
- Energy function is analogous to the discriminator

We use the following energy:

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

L : label set size
$\boldsymbol{x}$: a length-T sequence
$y_{t,j}$: the jth entry of the output label $\boldsymbol{y}_t$ at position t
$f(\boldsymbol{x}, t)$: "input feature vector" at position t

We use the following energy:

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

L : label set size
$\boldsymbol{x}$: a length-T sequence
$y_{t,j}$: the jth entry of the output label $\boldsymbol{y}_t$ at position t
$f(\boldsymbol{x}, t)$: "input feature vector" at position t

Two special cases:

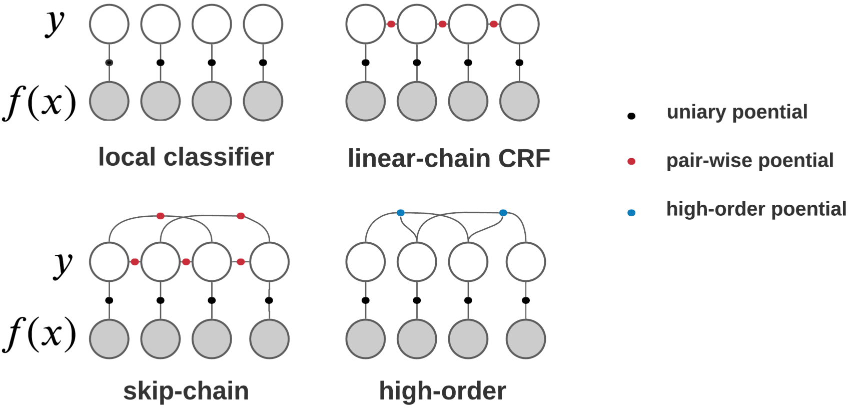
- Local Classifier :

$$E_W(\boldsymbol{y}) = 0$$

- Linear Chain Energies:

$$E_W(\boldsymbol{y}) = \sum_{t=1}^{T} \boldsymbol{y}_{t-1}^\top W \boldsymbol{y}_t$$

What are high-order energy terms?



$f(\boldsymbol{x})$: "input feature vector"
$\boldsymbol{y}$: output label sequence.

Skip-Chain Energies

$$E_W(\boldsymbol{y}) = \sum_{t=1}^{T} \sum_{i=1}^{M} \boldsymbol{y}_{t-i}^{\top} W_i \boldsymbol{y}_t$$

High-Order Energies

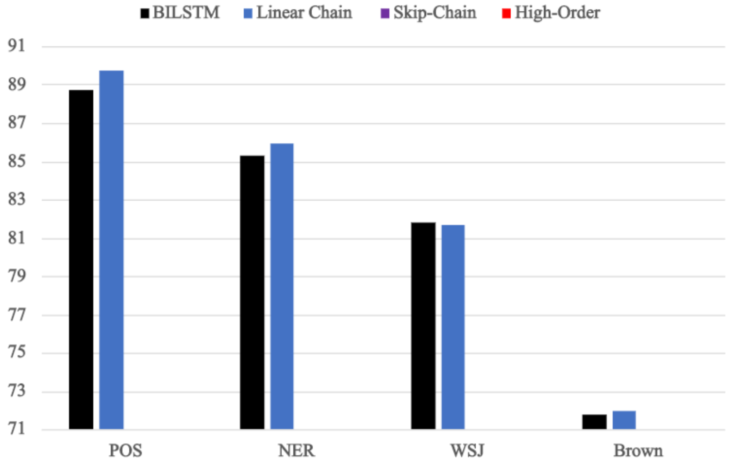
$$E_W(\boldsymbol{y}) = \sum_{t=M}^{T} F(\boldsymbol{y}_{t-M}, \ldots, \boldsymbol{y}_t)$$

We consider several different parameterizations of F:

- Vectorized Kronecker Product (VKP)
- CNN
- Tag Language model (TLM)
- Self-Attention (S-Att)

Fully-Connected Energies

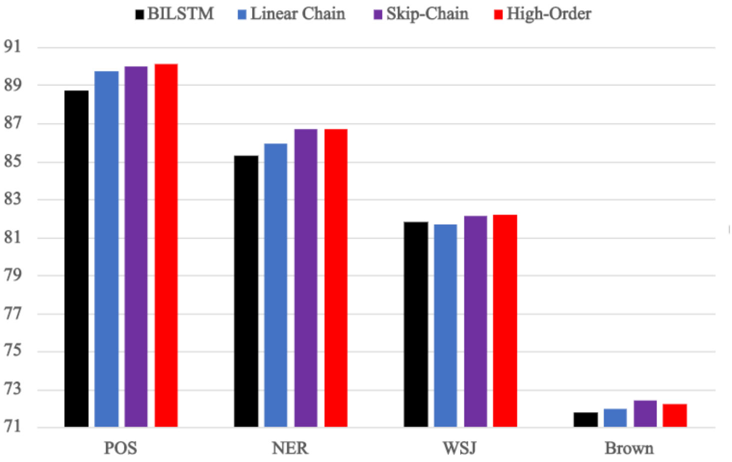
Results



Experimental Details

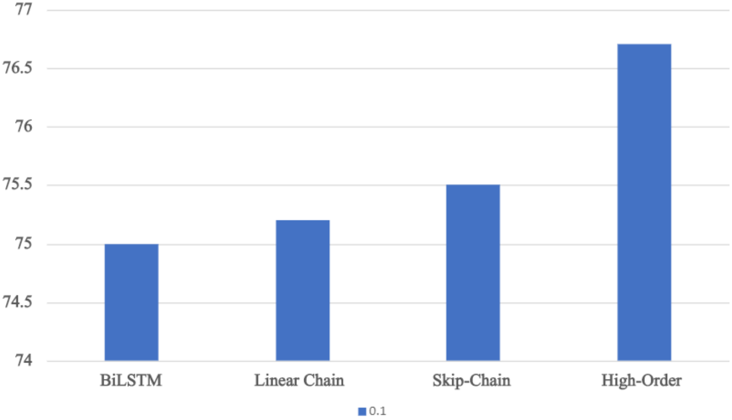
Baseline (1): local classifiers (BiLSTM)
Baseline (2): linear chain energy models.

Result



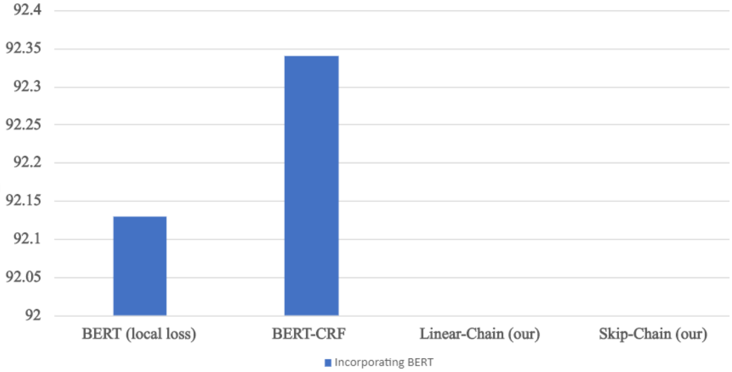
Skip-chain and high-order energy models yields higher performance.

Results on Noisy Datasets



High-order information helps the model recover from the noise.

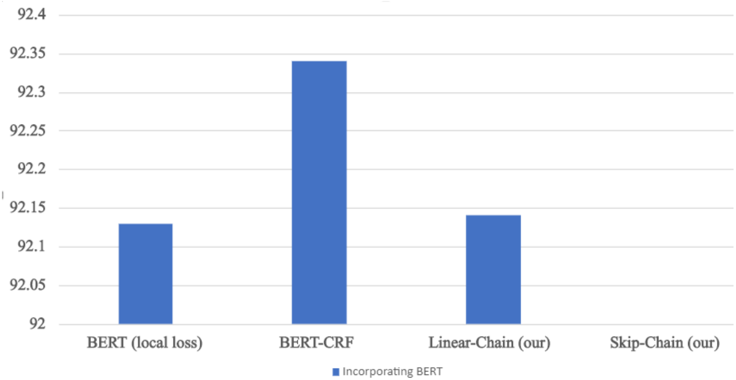
Incorporating BERT



Experimental Details

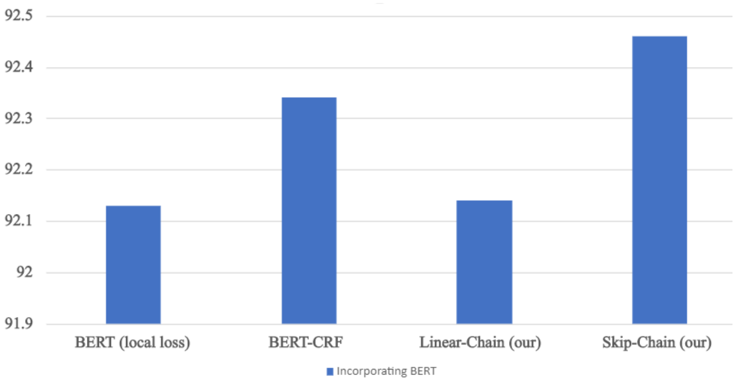
Tasks: NER
Baseline (1) BERT finetuned for NER using a local loss
Baseline (2) a CRF using BERT features ("BERT-CRF").

Incorporating BERT



Little difference between the locally-trained BERT and linear-chain energy function within our framework.

Incorporating BERT



Higher-order energy achieves much better than the locally-trained BERT model with framework.

Visualization of Learned Energies

The rows correspond to earlier labels,
the columns correspond to subsequent labels.

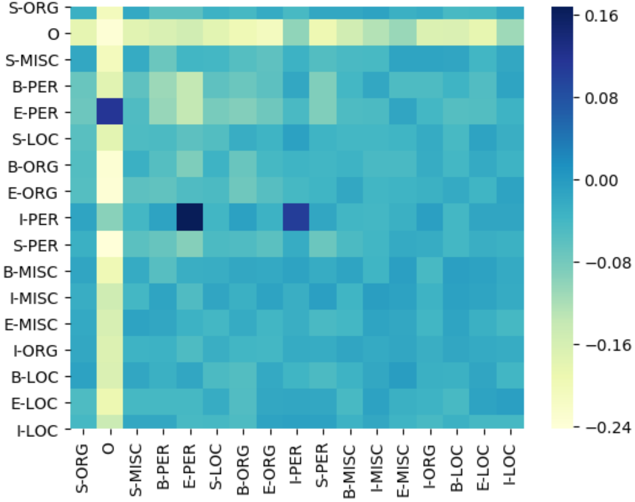


Figure: Learned 2nd-order VKP energy matrix beginning with B-PER

Conclusions

- Propose several high-order energy terms to capture complex dependencies
- Substantial improvements using high-order energy terms while keeping inference speed as the same as local classifiers
- Improvements even with BERT-like models
- High-order energy terms enrich the structured dependency on noisy settings.

Thanks!