Deep Active Learning for Named Entity Recognition

Yanyao Shen‡[a] , Hyokun Yun†, Zachary C. Lipton†¶, Yakov Kronrod†, Animashree Anandkumar†§

‡ University of Texas at Austin † Amazon AI ¶ Carnegie Mellon University § California Institute of Technology

shenyanyao@utexas.edu, yunhyoku@amazon.com, zlipton@cmu.edu, kronrod@amazon.com, anima@caltech.edu

[a]Work performed while interning at Amazon.

Overview

Deep Active Learning (DAL)

1. Goal: better-performing deep nets with fewer labels.
2. Methods: active learning techniques adapted to deep nets for NER.
3. Results: nearly match SOTA performance on NER with only 25% of the labels .
4. Impact: reduce costs on many data-thirsty applications.

Challenges

1. Training with sequential data is difficult.
2. Annotating sequential data is also difficult.

Named Entity Recognition

Our Focus

In this paper, we consider Named Entity Recognition (NER), which is a foundational technology that often underlies systems for:

1. Content classification
2. Content recommendation
3. Search

What is NER?

It's tough to imagine the Timberwolves being able to overcome their shortfalls on offense, which can't seem to get Karl-Anthony Towns going whatsoever, although a win Saturday can change things.

Figure 1: A labeled example from an NER dataset

Techniques and Results

Techniques

1. Design an efficient architecture (CNN-CNN-LSTM).
2. Incrementally train DNNs while actively selecting samples.
3. Use word-level budget in each round of annotation.
4. Adapt uncertainty-based active learning to sequential deep learning.

Results

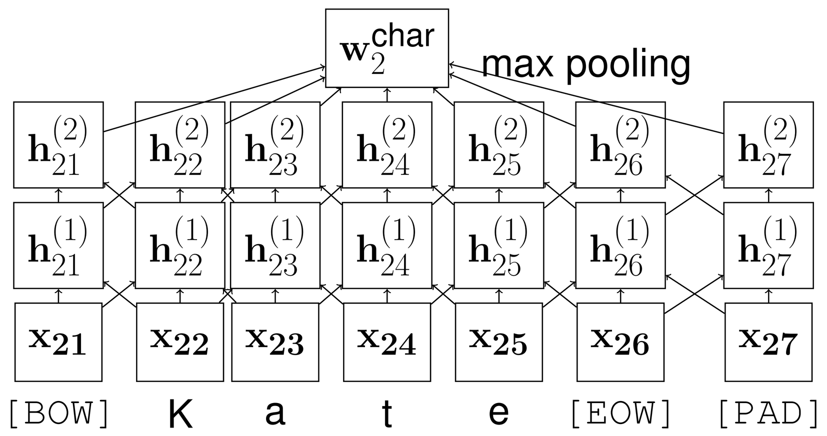
1. ***Nearly same accuracy with only*** $\sim 25\%$ ***training data.***
2. Simple active learning algorithms work well.

Lightweight Model Architecture

Design of the Architecture

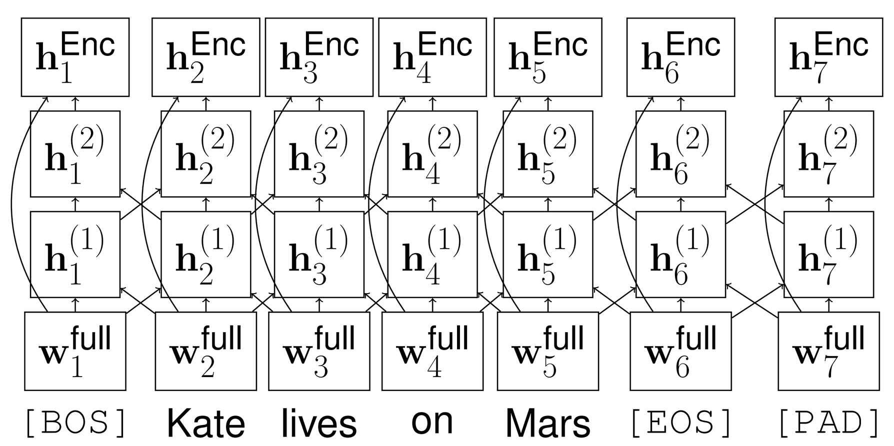
1. **Character-Level CNN Encoder**

- $\{x_{ij}\}$: formatted input sentence.
- x_{ij}: one-hot encoding of the j-th character in i-th word.
- $w_i^{\mathtt{char}}$: character-level feature.



2. **Word-Level CNN Encoder**

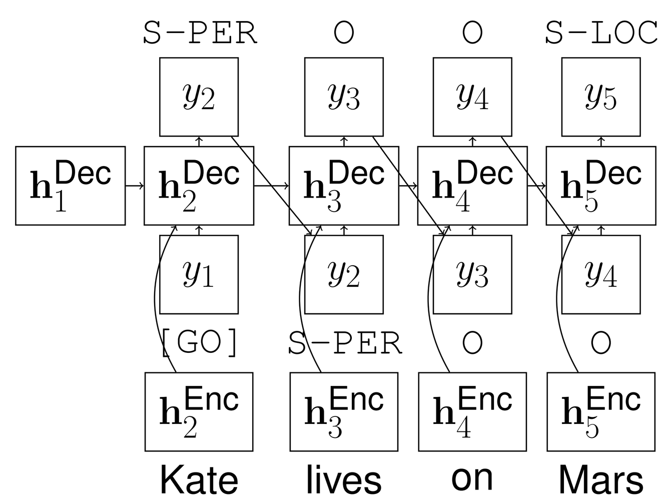
- $w_i^{\mathtt{emb}}$: word embedding vector. $\mathbf{w}_i^{\text{full}} := \left(\mathbf{w}_i^{\text{char}}, \mathbf{w}_i^{\text{emb}}\right)$.
- $h_i^{\mathtt{Enc}}$: word-level representation. $\mathbf{h}_i^{\text{Enc}} = \left(\mathbf{h}_i^{(l)}, \mathbf{w}_i^{\text{full}}\right)$



3. **Tag LSTM Decoder**

- Decoder induces a probability distribution over tags, conditioned on word-level encoder features.

$$\mathbb{P}\left[y_2, y_3, \ldots, y_{n-1} \mid \{\mathbf{h}_i^{\text{Enc}}\}\right]$$



Efficiency of the Architectures

OntoNotes-5.0 English dataset.

Char	Word	Tag	Reference	F1	Sec/Epoch
CNN	LSTM	CRF	[CN16]	86.28 ± 0.26	83*
None	Dilated CNN	CRF	[SVBM17]	86.84 ± 0.19	-
CNN	LSTM	LSTM		86.40 ± 0.48	76
CNN	CNN	LSTM		86.52 ± 0.25	22
CNN	CNN	CRF		86.15 ± 0.08	44
LSTM	LSTM	LSTM		86.63 ± 0.49	206

Observations

1. CNN is much more efficient than LSTM as an encoder.
2. LSTM is much more efficient than CRF as a decoder.

Deep Active Learning Choices

Under the uncertainty sampling framework, we explain four active learning strategies and how we use them in the sequential tagging task with NN-based models.

1. **Least Confidence (LC):**

$$1 - \max_{y_1,\ldots,y_n} \mathbb{P}\left[y_1, \ldots, y_n \mid \{\mathbf{x}_{ij}\}\right]. \tag{1}$$

- Intuition: sort examples in descending order by the probability of *not* predicting the most confident sequence from the current model.
- In practice: approximate (1) with the probability of a greedily decoded sequence.

2. **Maximum Normalized Log-Probability (MNLP):**

LC can be equivalently written as:

$$\max_{y_1,\ldots,y_n} \sum_{i=1}^{n} \log \mathbb{P}\left[y_i \mid y_1, \ldots, y_{n-1}, \{\mathbf{x}_{ij}\}\right]. \tag{2}$$

Normalize (2) as follows, and we get Maximum Normalized Log-Probability method:

$$\max_{y_1,\ldots,y_n} \frac{1}{n} \sum_{i=1}^{n} \log \mathbb{P}\left[y_i \mid y_1, \ldots, y_{n-1}, \{\mathbf{x}_{ij}\}\right].$$

- Intuition: (2) contains summation over words, LC naturally favors longer sentences.
- Our preliminary experiments verify that LC disproportionately selects longer sentences.

3. **Bayesian Active Learning by Disagreement (BALD):**

We sort the samples by $\frac{1}{n}\sum_{j=1}^{n} f_j$, where

$$f_i = 1 - \frac{\max_y \left|\{m : \mathrm{argmax}_{y'} \mathbb{P}^m\left[y_i = y'\right] = y\}\right|}{M}, \tag{3}$$

$\mathbb{P}^1, \mathbb{P}^2, \ldots \mathbb{P}^M$ are models sampled from the posterior. f_i is the measure of the ith word. $|\cdot|$ denotes cardinality of a set.

- Intuition: the fraction of models which disagreed with the most popular choice for each word.
- In practice: use Monte Carlo dropout to sample from model posterior with $M = 100$.

4. **Representative-based Uncertainty Sampling**:

$$f_w(\mathbb{S}) = \sum_{i \in \mathbb{X}^U} \mathtt{US}(i) \cdot \left[\max_{j \in \mathbb{S} \cup \mathbb{X}^L} w(i,j) - \max_{j \in \mathbb{X}^L} w(i,j)\right], \tag{4}$$

- Intuition: the uncertainty of a set is a reweighted sum based on representativeness of each sample.
- In practice: design streaming algorithm for submodular maximization under knapsack constraint to find a set of samples with high f_w score.

Results

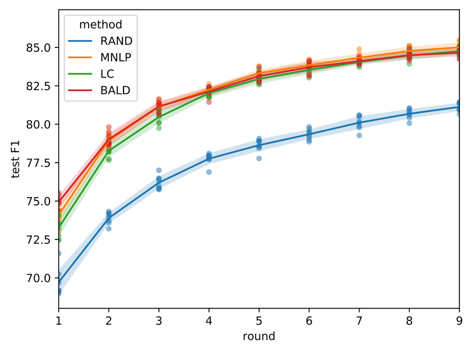


Figure 2: OntoNotes-5.0 English

Figure 3: Test of Uncertainty Measures

1. **Comparisons of selection algorithms:**
   - Among active learners, **MNLP, BALD** slightly outperformed others in early rounds. (MNLP is much cheaper.)
   - Impressively, active learning algorithms achieve **99%** performance of the best deep model trained on full data using only **24.9%** of the training data on the English dataset and **30.1%** on Chinese.
   - Also, **12.0%** and **16.9%** of training data were enough for deep active learning algorithms to surpass the performance of the shallow models trained on the full training data.
2. **Detection of under-explored genres:**
   - Experiment description: we design the experiment to better understand how DAL chooses informative examples.
     ✓ Select three datasets with same size but consist of different genres.
     ✓ Calculate the distribution of the top-1k samples for models trained with each dataset.
   - Impressively, although we did not provide the genre of sentences to the algorithm, it was able to automatically detect underexplored genres.
   - As is shown in Figure 3, A model trained using newswire (nw) data is more inclined to select uncertainty samples from broadcast conversation (bc) and telephone conversation (tc).

Conclusions

- We proposed deep active learning algorithms for NER, and empirically demonstrated that they achieve state-of-the-art performance with much less data than models trained in the standard supervised fashion.
- The proposed deep active learning algorithms are able to extend to other applications easily.

Future Work

- Explore more effective embeddings for sequential tasks.
- Combine with crowdsourcing and overcome label ambiguity.
- Extend to other applications.