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 Web Services

shenyanyao@utexas.edu, {yunhyoku,liptoz,kronrod,anima}@amazon.com

[a]the work was done while at Amazon

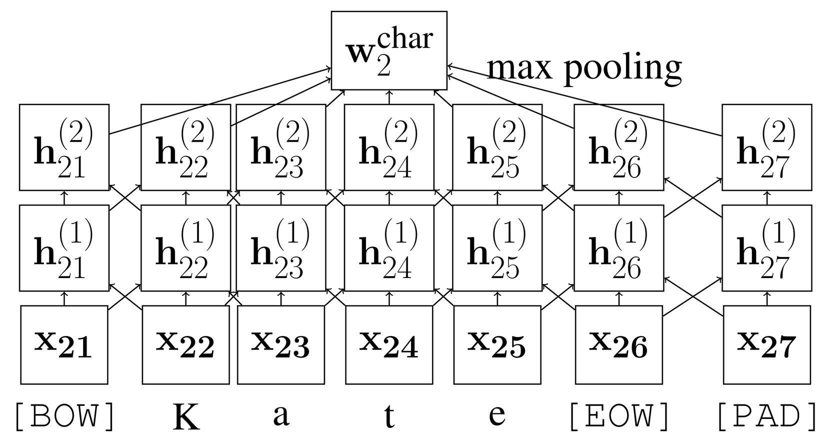
Introduction

1. Over the past several years, a series of papers have used deep neural networks (DNNs) to advance the state-of-the-art in named entity recognition (NER) over shallow models.
   - CoNLL-2003 English dataset: **0.4%** improvement (F1 score), small dataset, 0.2M words.
   - OntoNotes-5.0 English dataset: **2.24%** improvement (F1 score), large dataset, 1.09M words.
2. **Goal:** train DNNs using fewer samples.
   Approach: active learning.
   Practice: crowdsourcing platforms – Mechanical Turk.
   Impact: reduce sample requirements, lower the labeling costs.
3. Effectiveness of deep active learning:
   almost same accuracy with about 25-30% training data using active learning.

Model 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.



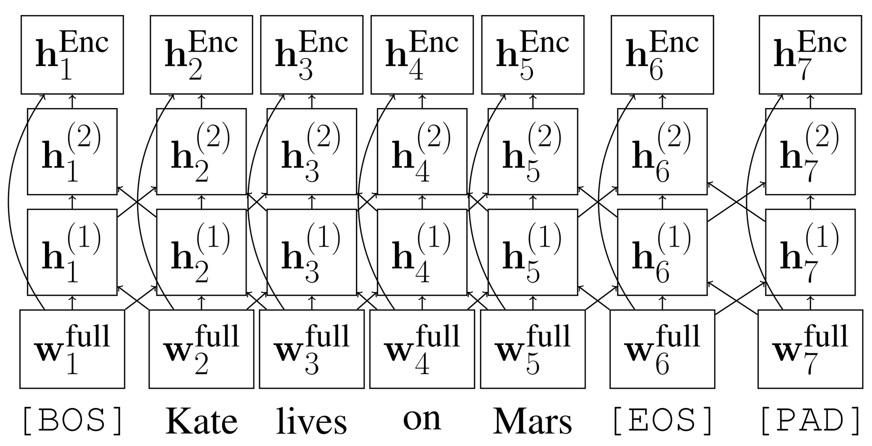
2. **Word-Level CNN Encoder**

- w_i^{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.

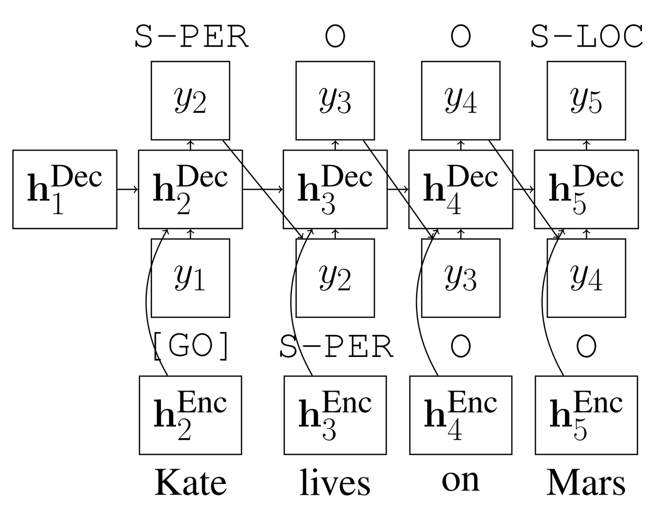
$$\mathbf{h}_i^{\text{Enc}} = \left(\mathbf{h}_i^{(l)}, \mathbf{w}_i^{\text{full}}\right)$$



3. **Tag LSTM Decoder**

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

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



Active Learning

Under the uncertainty sampling framework, we explain three 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:

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

   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

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

   $\mathbb{P}^1, \mathbb{P}^2, \ldots \mathbb{P}^M$ 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 $M = 100$.

Other techniques in deep active learning:

1. **Incremental training** of DNNs while actively selecting samples.
2. **Use word-level budget** in each round of selection.

Results

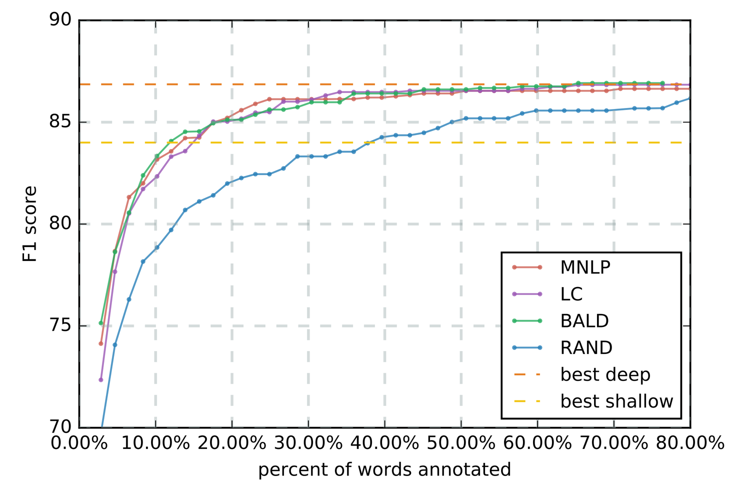


Figure 1: OntoNotes-5.0 English

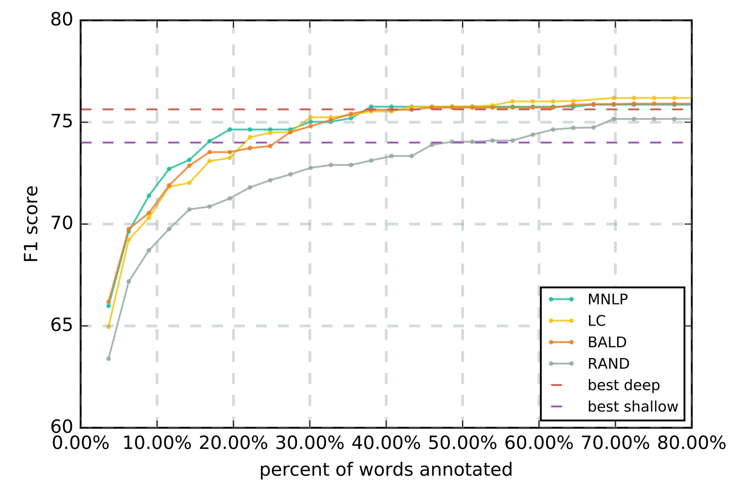


Figure 2: OntoNotes-5.0 Chinese

1. **Comparisons of selection algorithms:**
   - Among active learners, **MNLP** slightly outperformed others in early rounds.
   - 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.

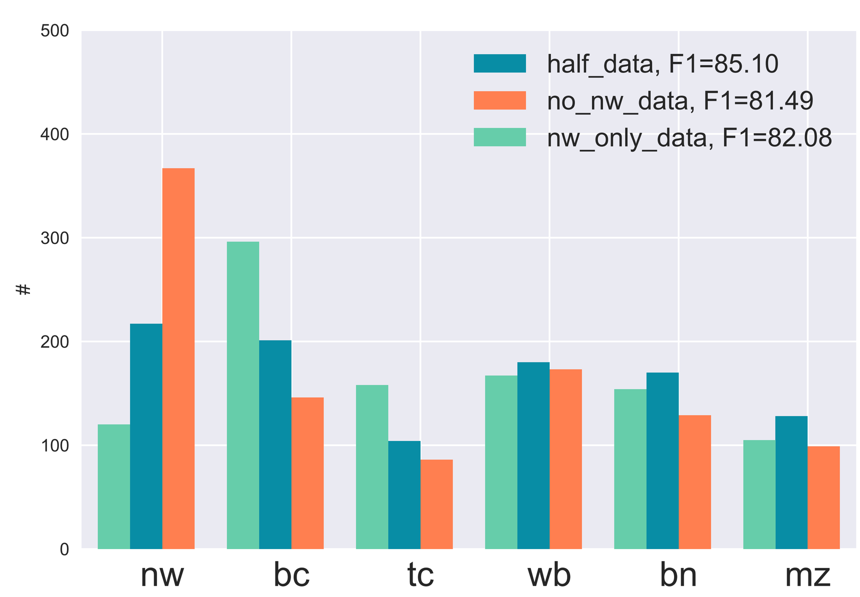


Figure 3: Genre distribution of top 1k sentences chosen by an active learning algorithm.

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 the effectiveness of subset selection in DAL setting.
- Combine with crowdsourcing and overcome label ambiguity.
- Extend to other applications.