Semi-Supervised Sequence Labeling with Self-Learned Features

Yanjun Qi¹, Pavel P. Kuksa², Ronan Collobert¹, Kunihiko Sadamasa, Koray Kavukcuoglu³, Jason Weston⁴

¹ Machine Learning Department, NEC Laboratories America, Inc.
² Computer Science Department, Rutgers University
³ Computer Science Department, New York University
⁴ Google Research New York
Roadmap

- Background
- Method (Self-Learned Features: SLF)
- Baseline Systems
- Experimental Results
Natural language processing (NLP) involves many machine learning tasks, especially sequential learning.

Learning: **Supervised** (classification, regression, etc.) vs. **Unsupervised** (clustering, etc.)

<table>
<thead>
<tr>
<th>Usage</th>
<th>Supervised learning</th>
<th>Unsupervised learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>{(x,y)} labeled data</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>{x*} unlabeled data</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
4 Background: Semi-Supervised Learning

- Labeled data are often **hard** to obtain
- Unlabeled data are often **easy** to obtain: **A Lot**

<table>
<thead>
<tr>
<th>Usage</th>
<th>Supervised learning</th>
<th>Semi-supervised learning</th>
<th>Unsupervised learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>{(x,y)} labeled data</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>{x*} unlabeled data</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- For instance, “Self-Training”
  - Popular semi-supervised method used in NLP
  - Induce self-labeled “pseudo” training “examples” from unlabeled set
5 Background: Semi-Supervised Learning (Cont’)

- Semi-supervised Learning (most not applicable for large scale NLP tasks)
  - Self-training or co-training
  - Transductive SVM
  - Graph-based regularization
  - Entropy regularization
  - EM with generative mixture models
  - Auxiliary task on unlabeled set through multi-task learning
  - Semi-supervised learning with “labeled features”
    - “Labeled features” \(\Rightarrow\) Prior class-bias of features from human annotation
    - Using “labeled features” to induce “pseudo” examples or enforce soft constraints on predictions of unlabeled examples
Individual **words** in NLP systems

- Carry **significant label** information
- Fundamental building blocks of NLP
- Many basic NLP tasks involve sequence modeling with word-level evaluation
  - For example, named entity recognition (NER), part-of-speech (POS) tagging

<table>
<thead>
<tr>
<th>Example</th>
<th>NLP Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>... former <em>captain</em> [Chris Lewis] ...</td>
<td>Name Entity [Person Name]</td>
</tr>
<tr>
<td>... the <em>state of</em> [washington] ...</td>
<td>Name Entity [Location Name]</td>
</tr>
</tbody>
</table>

Our target NLP problems: Information extraction

- **Assign labels to each word** in a sequence of text
- Essentially, classify **each word into multiple classes**
Provide “semi-supervision” at the level of features (e.g. words) related to target labels

- Through self-learned features (SLF) of words (basic case)

\[ \text{SLF}(w)_i = P(y = i | w, \text{ where } w \in x) \]

- SLF models the probability to each target class this word might be assigned with

- SLF is unknown (of course) \(\rightarrow\) re-estimate using unlabeled examples by applying a trained classifier

“semi-supervised” self-learned features (SLF)
Empirical SLF is estimated from unlabeled examples

- Each example is a sequence of words
- Thus, SLF of a word $w$, for class $i$ →
  - ($\#$ examples including word $w$ that are predicted as class $i$ / $\#$ examples including word $w$)

$$\overline{\text{SLF}}(w)_i = \frac{|\{j : f(x^*_j) = i \land w \in x^*_j\}|}{|\{k : w \in x^*_k\}|}$$

- Where $\{x^*\}$ represents unlabeled examples
- Where $f(-)$ represents a trained supervised sequence classifier
Method: Algorithm (Basic Case)

- **Pseudo-code**

1. Define the feature representation for a word as $\phi(\omega)$, and the representation for an example (a window of words) as $\Phi(x)$
2. Train a classifier $f(\cdot)$ on training examples $(x_i, y_i)$ using the feature representation $\Phi(\cdot)$
3. Use $f(\cdot)$ to estimate $\overline{SLF}(w)$ from unlabeled data $\{x^*\}$
4. Augment the representation of words to $\overline{\phi}(\omega)$ and refine $\Phi(x)$, where $\overline{\phi}(w) = (\phi(w), \overline{SLF}(w))$
5. Iterate steps 2 to 4 until stopping criterion is met.
Modified SLF: Word Sliding Window Case

Input Sentence (Text Window)

- **text**: the cat **sat** on the ...
- **words**: $x_1, x_2, x_3, x_4, x_5$
- **labels**: $y_1, y_2, y_3, y_4, y_5$

$x = (x_1, ..., x_5)$

each example $\Rightarrow$ a window of words

Self-Learned Features (SLF)

- $y_3 = i$
- Class-1, Class-2, Class-i, Class-K
Rare words are the hardest to label

Motivation: model those words happening frequently before or after a certain target class

Boundary SLF: extend basic SLF to incorporate the class boundary distribution

\[
\text{SLF}''(w)_{t,1} = P(y_i = t | w \in \{(x_i)_1, \ldots, (x_i)_{m-1}\})
\]
\[
\text{SLF}''(w)_{t,2} = P(y_i = t | w \in \{(x_i)_{m+1}, \ldots, (x_i)|x_i|\})
\]
Extension II: Clustered SLF

- Words exhibiting similar target class distribution have similar SLF features
- Group SLF features might give stronger indications of target class or class boundary
- $k$-means to cluster all words into $N$ clusters, and use cluster-ID as the new clustered-SLF features

Extension III: Attribute SLF

- Treat discrete attribute of words as the basic unit of sequence examples
- For instance, ‘stem-end’ for POS task
Why Useful?

- No incestuous bias since no examples are added
- No tricky parameters to tune (not like “self-training”)
- Supervised model learns SLF relevant or not
- Summarization over many potential labels, hence infrequent mistakes can be smoothed out
  - Potentially corrected on the next iteration
- Empirical SLF features for neighboring words are highly informative
- Highly scalable (adding a few features, not examples)
- A wrapper approach applicable on many other methods
A deep neural network (NN) based NLP system [Collobert 08]

Auxiliary task “LM” provides one type of semi-supervision

“Viterbi” training enforces local label dependencies among neighborhood

- SLF enforces local dependency as well
Conditional Random Field (CRF) [Lafferty 01]

- State-of-the-art performance on many sequence labeling tasks
- Discriminative probabilistic models over observation sequences and label sequences
- Apply SLF as a wrapper on CRF++ toolkit
Four Benchmark Data Sets

- CoNLL03 German Named Entity Recognition (NER)
- CoNLL03 English Name Entity Recognition (NER)
- English Part-of-Speech (POS) benchmark data [Toutanova 03]
- Gene Mention (GM) benchmark data [BioCreative II]

<table>
<thead>
<tr>
<th>Token Size</th>
<th>Training (Labeled)</th>
<th>Unlabeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>German NER</td>
<td>206,931</td>
<td>~60M</td>
</tr>
<tr>
<td>English NER</td>
<td>203,621</td>
<td>~200M</td>
</tr>
<tr>
<td>English POS</td>
<td>1,029,858</td>
<td>~300M</td>
</tr>
<tr>
<td>Bio GM</td>
<td>345,996</td>
<td>~900M</td>
</tr>
</tbody>
</table>

Evaluation Measurements

- Entity-level \( F1: \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \)
- Word-level error rate for POS task
Performance Comparison (German NER)

- IOBES style of class tag / 5 words sliding window
- All features case
  - (word, capitalization flag, prefix and suffix (length up to 4), part-of-speech tags, text chunk, string patterns)
- Best CoNLL03 team: test F1 - 74.17
- Baseline classifier: NN

<table>
<thead>
<tr>
<th>Setting</th>
<th>Test F1</th>
<th>+ Basic SLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>word only</td>
<td>45.89</td>
<td>51.10</td>
</tr>
<tr>
<td>word only + Viterbi</td>
<td>50.61</td>
<td>53.46</td>
</tr>
<tr>
<td>all features + LM</td>
<td>72.44</td>
<td>73.32</td>
</tr>
<tr>
<td>all features + LM + Viterbi</td>
<td>74.33</td>
<td>75.72</td>
</tr>
</tbody>
</table>
Performance Comparison (English NER)

- IOBES style of class tag / 7 words sliding window
- All features case
  - (word, cap, dictionary)
- Best CoNLL03 team: test F1 – 88.76
- Baseline classifier: NN

<table>
<thead>
<tr>
<th>Setting</th>
<th>Test F1</th>
<th>+ Basic SLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>word + cap</td>
<td>77.82</td>
<td>79.38</td>
</tr>
<tr>
<td>word + cap + Viterbi</td>
<td>80.53</td>
<td>81.51</td>
</tr>
<tr>
<td>word + cap + dict + LM</td>
<td>86.49</td>
<td>86.88</td>
</tr>
<tr>
<td>word + cap + dict + LM + Viterbi</td>
<td>88.40</td>
<td>88.69</td>
</tr>
</tbody>
</table>
Performance Comparison (English POS)

- IOBES style of class tag / 5 words sliding window
- All features case
  - (word, cap, stem-end)
- Best result (we know) : test error rate 2.76%
  - WER: token-level error rate
- Baseline classifier: NN

<table>
<thead>
<tr>
<th>Setting</th>
<th>WER</th>
<th>+ Basic SLF</th>
<th>+ Attribute SLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>4.99</td>
<td>4.06</td>
<td>-</td>
</tr>
<tr>
<td>word + LM</td>
<td>3.93</td>
<td>3.89</td>
<td>-</td>
</tr>
<tr>
<td>word + cap + stem</td>
<td>3.28</td>
<td>2.99</td>
<td>2.86</td>
</tr>
<tr>
<td>word + cap + stem + LM</td>
<td>2.79</td>
<td>2.75</td>
<td>2.73</td>
</tr>
</tbody>
</table>
Performance Comparison (Bio GM)

- Look for gene or protein name in bio-literature (two classes: gene or not)
- All features case
  - (word, cap, prefix and suffix (length up to 4). String pattern)
- Best BioCreativell team: test F1 – 87.21
  - Many other complex features + Bio-directional CRF training
- Baseline classifier: CRF++

<table>
<thead>
<tr>
<th>Setting</th>
<th>Test F1</th>
<th>+ Clustered SLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>word + cap</td>
<td>82.02</td>
<td>84.01 (on Basic SLF)</td>
</tr>
<tr>
<td>word + cap</td>
<td>82.02</td>
<td>85.24 (on Boundary SLF)</td>
</tr>
<tr>
<td>word + cap + pref + suf + str</td>
<td>86.34</td>
<td>87.16 (on Boundary SLF)</td>
</tr>
</tbody>
</table>
21 Performance Comparison to Self-Training

- Self training with random selection scheme:
  - Given $L$ training examples, choose $L/R$ ($R$ is a parameter to choose) unlabeled examples to add in next round’s training

- Self-Training on German NER

<table>
<thead>
<tr>
<th>Setting</th>
<th>Baseline</th>
<th>R=1</th>
<th>R=10</th>
<th>R=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words only + viterbi</td>
<td>50.61</td>
<td>47.07</td>
<td>47.92</td>
<td>47.9</td>
</tr>
<tr>
<td>All +LM+Viterbi</td>
<td>74.33</td>
<td>73.42</td>
<td>74.41</td>
<td>73.9</td>
</tr>
</tbody>
</table>

- Self-Training on English NER

<table>
<thead>
<tr>
<th>Setting</th>
<th>Baseline</th>
<th>R=1</th>
<th>R=20</th>
<th>R=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words only + Viterbi</td>
<td>80.53</td>
<td>79.51</td>
<td>81.01</td>
<td>80.85</td>
</tr>
<tr>
<td>Word +Cap+dict + LM+Viterbi</td>
<td>88.40</td>
<td>87.64</td>
<td>88.07</td>
<td>88.17</td>
</tr>
</tbody>
</table>

- SLF has better behavior than self-training (with a random selection strategy)
Conclusions

- Semi-supervised SLF is promising for sequence labeling tasks in NLP.

- Easily extendable for other cases, such as predicted class distributions (or related) for each n-gram.

- Easily extendable for other domains, such as sentimental analysis (word’s class distribution as the distribution of labels of documents containing this word):
  - “cash back” to class “shopping”