Semi-Supervised Sequence Labeling with Self-Learned Features

Yanjun Qi[1], Pavel P. Kuksa[2], Ronan Collobert[1], Kunihiko Sadamasa, Koray Kavukcuoglu[3], Jason Weston[4]

[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

Background: Learning

- Natural language processing (NLP) involves many machine learning tasks, especially sequential learning
- Learning: **Supervised** (classification, regression, etc.) vs. **Unsupervised** (clustering, etc)

Usage	Supervised learning	Unsupervised learning
{(x,y)} labeled data	Yes	No
{x*} unlabeled data	No	Yes

Background: Semi-Supervised Learning

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

Usage	Supervised learning	**Semi-supervised learning**	Unsupervised learning
{(x,y)} labeled data	Yes	**Yes**	No
{x*} unlabeled data	No	**Yes**	Yes

- 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" → Prior class-bias of features from human annotation
    - Using "labeled features" to induce "pseudo" examples or enforce soft constraints on predictions of unlabeled examples

Background: Word

- 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

Example	NLP Task
… former *captain* [Chris Lewis] …	Name Entity [Person Name]
… the *state of* [washington] …	Name Entity [Location Name]

- Our target NLP problems: Information extraction
  - Assign labels to each word in a sequence of text
  - Essentially, classify each word into multiple classes

Method: Semi-Supervised SLF

- Provide "semi-supervision" at the level of features (e.g. words) related to target labels
  - Through self-learned features (SLF) of words (basic case)

$$\mathrm{SLF}(w)_i = P(\mathbf{y} = i | w, \ \text{where } w \in \mathbf{x})$$

  - SLF models the probability to each target class this word might be assigned with
  - SLF is unknown (of course) → re-estimate using unlabeled examples by applying a trained classifier

"semi-supervised" self-learned features (SLF)

Method: Semi-Supervised SLF (Basic Case)

- 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)

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

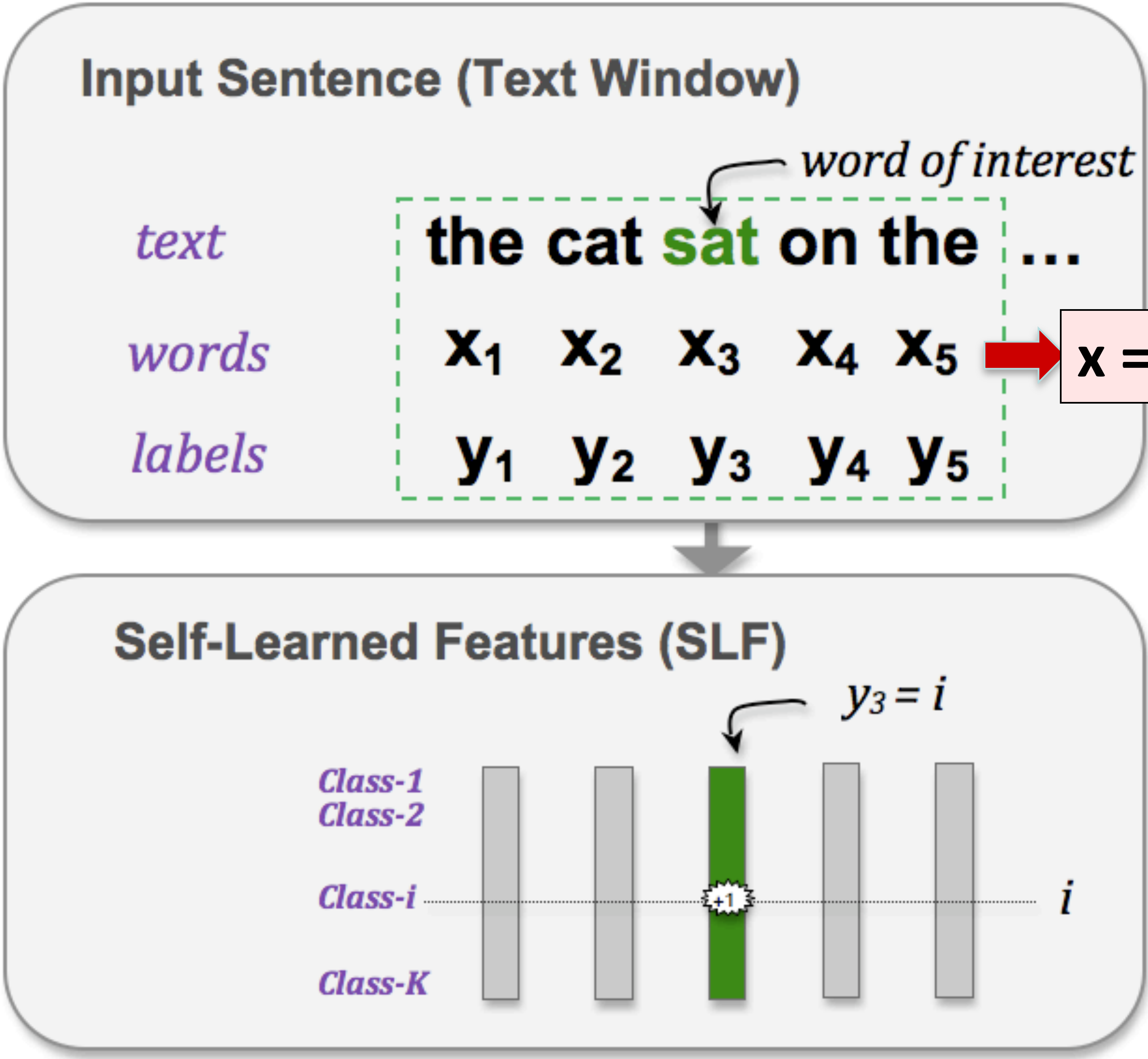
- Where {x*} represents unlabeled examples
- Where f(-) represents a trained supervised sequence classifier

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(·)* on training examples (x_i, y_i) using the feature representation $\Phi(\cdot)$
3. Use *f(·)* to estimate $\overline{\mathrm{SLF}}(w)$ from unlabeled data {x*}
4. Augment the representation of words to $\bar{\phi}(\omega)$ and refine $\Phi(x)$, where $\bar{\phi}(w) = (\phi(w), \overline{\mathrm{SLF}}(w))$
5. Iterate steps 2 to 4 until stopping criterion is met.

10 Modified SLF: Word Sliding Window Case



$\mathbf{x} = (x_1,\ldots,x_5)$

each example ➔ a window of words

Extension I: Boundary SLF

- Rare words are the hardest to label
- Motivation: model those words happening frequently before or after a certain target class

... former *captain* [Chris Lewis] ...
...[Hoddle] *said* ...
... [CRKL], an adapter protein ...
... [SH2-SH3-SH3] adapter protein ...

* Blue color words carry important label indications

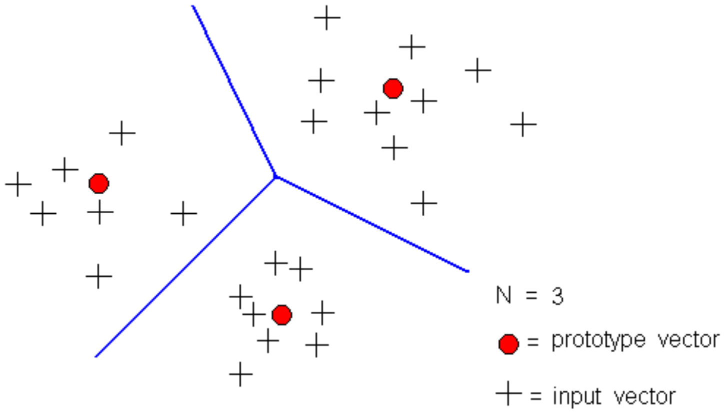
- Boundary SLF : extend basic SLF to incorporate the class boundary distribution

$$\begin{aligned}\mathrm{SLF}''(w)_{t,1} &= P(\mathbf{y}_i = t | w \in \{(\mathbf{x}_i)_1, \ldots, (\mathbf{x}_i)_{m-1}\}) \\ \mathrm{SLF}''(w)_{t,2} &= P(\mathbf{y}_i = t | w \in \{(\mathbf{x}_i)_{m+1}, \ldots, (\mathbf{x}_i)_{|\mathbf{x}_i|}\})\end{aligned}$$

Extension II & Extension III

❑ 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

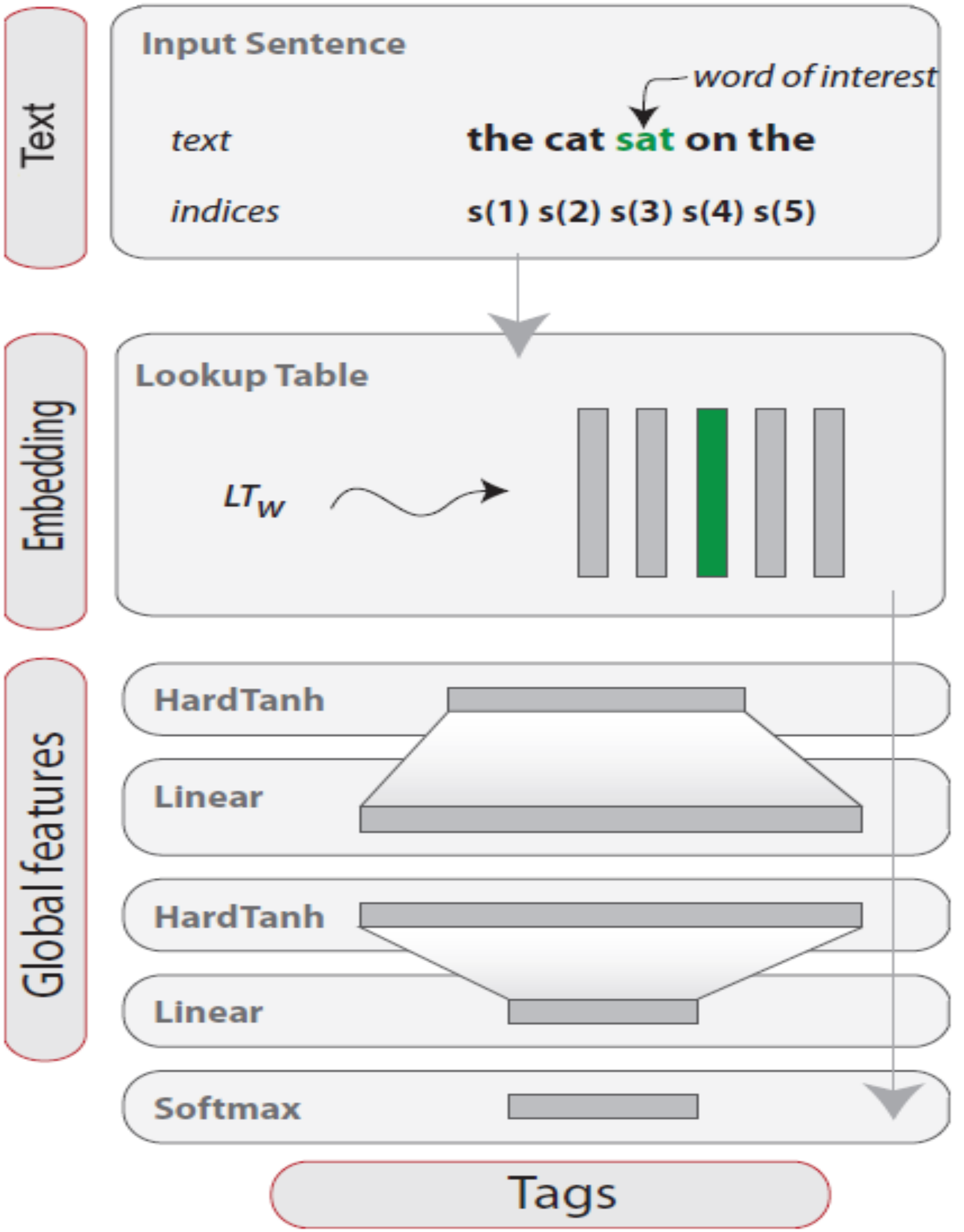
- Treat discrete attribute of words as the basic unit of sequence examples
- For instance, ‘stem-end’ for POS task

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

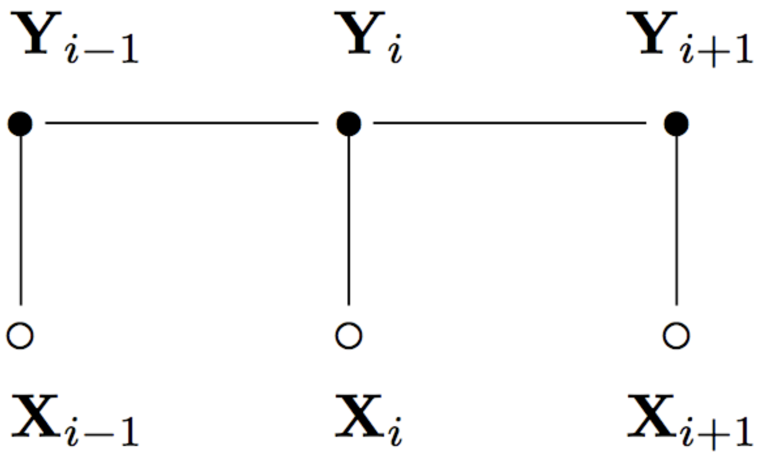
Baseline NLP System I - NN

- 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



Baseline NLP System II - CRF

- 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



Experimental Setting

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]

Token Size	Training (Labeled)	Unlabeled
German NER	206,931	~60M
English NER	203,621	~200M
English POS	1,029,858	~300M
Bio GM	345,996	~900M

Evaluation Measurements

- Entity-level F1: 2 (precision * recall) / (precision + recall)
- Word-level error rate for POS task

17 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

Setting	Test F1	+ Basic SLF
word only	45.89	**51.10**
word only + Viterbi	50.61	**53.46**
all features + LM	72.44	**73.32**
all features + LM + Viterbi	74.33	**75.72**

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

Setting	Test F1	+ Basic SLF
word + cap	77.82	**79.38**
word + cap + Viterbi	80.53	**81.51**
word + cap + dict + LM	86.49	**86.88**
word + cap + dict + LM + Viterbi	88.40	**88.69**

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

Setting	WER	+ Basic SLF	+ Attribute SLF
word	4.99	**4.06**	-
word + LM	3.93	**3.89**	-
word + cap + stem	3.28	**2.99**	**2.86**
word + cap + stem + LM	2.79	**2.75**	**2.73**

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 BioCreativeII team: test F1 – 87.21
  - Many other complex features + Bio-directional CRF training
- Baseline classifier: CRF++

Setting	Test F1	+ Clustered SLF
word + cap	82.02	**84.01 (on Basic SLF)**
word + cap	82.02	**85.24 (on Boundary SLF)**
word + cap + pref + suf + str	86.34	**87.16 (on Boundary SLF)**

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

Setting	Baseline	R=1	R=10	R=100
Words only + viterbi	**50.61**	47.07	47.92	47.9
All +LM+Viterbi	74.33	73.42	**74.41**	73.9

- Self-Training on English NER

Setting	Baseline	R=1	R=20	R=100
Words only + Viterbi	80.53	79.51	**81.01**	80.85
Word +Cap+dict + LM+Viterbi	**88.40**	87.64	88.07	88.17

- ➔ 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"