Self-training Models with Latent Variables for Natural Language Processing

Zhongqiang Huang

Department of Computer Science
University of Maryland
College Park, MD
Overview

- Self-training has helped for some NLP tasks
  - but seems to fail for tagging and parsing with moderate amounts of training data

- Models with latent variables
  - flexible at learning dependencies in different granularities
  - benefit much more from self-training
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w_1^n : \text{Does that flight serve dinner?} \]
\[ t_1^n : \text{VBZ DT NN VB NN .} \]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w^n_1: \text{Does that flight serve dinner?} \]

\[ t^n_1: \text{VBZ DT NN VB NN .} \]

Bigram HMM tagger

\[
P(w^n_1, t^n_1) \approx \prod_i P(t_i|t_{i-1})P(w_i|t_i)
\]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w^n_1 : \text{Does that flight serve dinner?} \]

\[ t^n_1 : \text{VBZ DT NN} \rightarrow \text{VB} \quad \text{NN} . \]

Bigram HMM tagger

\[ P(w^n_1, t^n_1) \approx \prod_i P(t_i | t_{i-1}) P(w_i | t_i) \]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w_1^n : \text{Does that flight serve dinner?} \]

\[ t_1^n : \text{VBZ DT NN } \text{[vb]} \text{ NN .} \]

Bigram HMM tagger

\[ P(w_1^n, t_1^n) \approx \prod_i P(t_i|t_{i-1})P(w_i|t_i) \]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w_1^n : \text{Does that flight serve dinner?} \]

\[ t_1^n : \text{VBZ DT NN VB NN}. \]

**Bigram HMM tagger**

\[ P(w_1^n, t_1^n) \approx \prod_i P(t_i|t_{i-1})P(w_i|t_i) \]

**Trigram HMM tagger**

\[ P(w_1^n, t_1^n) \approx \prod_i P(t_i|t_{i-1}, t_{i-2})P(w_i|t_i, t_{i-1}) \]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w^n_1 : \text{Does} \quad \text{that} \quad \text{flight} \quad \text{serve} \quad \text{dinner} \quad ? \]

\[ t^n_1 : \text{VBZ} \quad \text{DT} \quad \text{NN} \rightarrow \text{VB} \quad \text{NN} \quad . \]

Bigram HMM tagger

\[
P(w^n_1, t^n_1) \approx \prod_{i} P(t_i|t_{i-1})P(w_i|t_i)
\]

Trigram HMM tagger

\[
P(w^n_1, t^n_1) \approx \prod_{i} P(t_i|t_{i-1}, t_{i-2})P(w_i|t_i, t_{i-1})
\]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\[ w_1^n : \text{Does that flight serve dinner?} \]

\[ t_1^n : \text{VBZ DT NN VB NN .} \]

:\textbf{Bigram HMM tagger}

\[ P(w_1^n, t_1^n) \approx \prod_i P(t_i|t_{i-1})P(w_i|t_i) \]

:\textbf{Trigram HMM tagger}

\[ P(w_1^n, t_1^n) \approx \prod_i P(t_i|t_{i-1}, t_{i-2})P(w_i|t_i, t_{i-1}) \]
Part-of-Speech Tagging

Assigns Part-of-Speech tags to words in a sentence:

\( w^n_1: \) Does that flight serve dinner?

\[ t^n_1: \] VBZ DT NN VB NN

**Bigram HMM tagger**

\[
P(w^n_1, t^n_1) \approx \prod_i P(t_i|t_{i-1})P(w_i|t_i)
\]

**Trigram HMM tagger**

\[
P(w^n_1, t^n_1) \approx \prod_i P(t_i|t_{i-1}, t_{i-2})P(w_i|t_i, t_{i-1})
\]

**PROBLEMS**

- Bigram tagger:
  - Strong independence assumption
  - Poor performance
- Trigram tagger:
  - Some trigrams too sparse, need sophisticated smoothing
  - Some trigrams are abundant, could use higher n-grams
- Higher order taggers?
  - Data sparseness, harder to smooth
Tagging with Latent Variables

[ Huang et al. 2009 ]

\[ \text{NN} \rightarrow \begin{array}{c} \text{NN}_1 \\ \text{NN}_2 \\ \vdots \\ \text{NN}_m \end{array} \]

\[ \text{VB} \rightarrow \begin{array}{c} \text{VB}_1 \\ \text{VB}_2 \\ \vdots \\ \text{VB}_n \end{array} \]
Tagging with Latent Variables

[Huang et al. 2009]
Tagging with Latent Variables

[Huang et al. 2009]
Tagging with Latent Variables (cont.)

\[ \text{Does that flight serve dinner?} \]

\[ \begin{align*}
  t^n_1 & : \text{VBZ} \quad \text{DT} \quad \text{NN} \quad \text{VB} \quad \text{NN} \quad . \\
  w^n_1 & : \text{Does} \quad \text{that} \quad \text{flight} \quad \text{serve} \quad \text{dinner} \quad ?
\end{align*} \]
Tagging with Latent Variables (cont.)

\[ W^n_1: \]

Does that flight serve dinner?

\[ t^n_1: \]

VBZ DT NN VB NN .
Tagging with Latent Variables (cont.)

\[ W^n_1: \]

\[ t^n_1: \]

Trigram tagger

\[ X_1 \quad NN \quad \rightarrow \quad VB \]

\[ X_2 \quad NN \quad \rightarrow \quad VB \]

\[ \vdots \]

\[ X_n \quad NN \quad \rightarrow \quad VB \]
Tagging with Latent Variables (cont.)

$w^n_1$: Does that flight serve dinner?

$t^n_1$: V邹 DT NN VB NN.

Trigram tagger

$X_1$ NN $\Rightarrow$ VB

$X_2$ NN $\Rightarrow$ VB

$\vdots$

$X_n$ NN $\Rightarrow$ VB

for $1 \leq i \leq n$, $X_i$ NN $\Rightarrow$ NN

and

NN VB $\Rightarrow$ VB

5
Tagging with Latent Variables (cont.)

Does that flight serve dinner?

Trigram tagger

\[
X_1 \quad \text{NN} \quad \rightarrow \quad \text{VB}
\]
\[
X_2 \quad \text{NN} \quad \rightarrow \quad \text{VB}
\]
\[
\vdots
\]
\[
X_n \quad \text{NN} \quad \rightarrow \quad \text{VB}
\]

Bigram tagger with latent variables

\[
\text{NN}_1 \quad \rightarrow \quad \text{VB}_k
\]
\[
\text{NN}_2 \quad \rightarrow \quad \text{VB}_k
\]
\[
\vdots
\]
\[
\text{NN}_n \quad \rightarrow \quad \text{VB}_k
\]

for \(1 \leq i \leq n\),

\[
X_i \quad \text{NN} \Rightarrow \text{NN}
\]

and

\[
\text{NN} \quad \text{VB} \Rightarrow \text{VB}_k
\]
Performance of Chinese POS Taggers

EXPERIMENT SETUP

• The traditional bigram and trigram HMM taggers
• The bigram HMM tagger with latent variables: with increasing amounts of latent tags
• Penn Chinese Treebank 6.0 (CTB6)
• 28k labeled sentences in total
• 80% for training, 10% for dev, 10% for test
Performance of Chinese POS Taggers

EXPERIMENT SETUP

- The traditional bigram and trigram HMM taggers
- The bigram HMM tagger with latent variables: with increasing amounts of latent tags
- Penn Chinese Treebank 6.0 (CTB6)
- 28k labeled sentences in total
- 80% for training, 10% for dev, 10% for test

Token Accuracy

<table>
<thead>
<tr>
<th>94.5</th>
<th>94</th>
<th>93</th>
<th>92</th>
<th>91.5</th>
</tr>
</thead>
</table>

Bigram

Trigram
Performance of Chinese POS Taggers

EXPERIMENT SETUP
- The traditional bigram and trigram HMM taggers
- The bigram HMM tagger with latent variables: with increasing amounts of latent tags
- Penn Chinese Treebank 6.0 (CTB6)
- 28k labeled sentences in total
- 80% for training, 10% for dev, 10% for test
Performance of Chinese POS Taggers

EXPERIMENT SETUP

- The traditional bigram and trigram HMM taggers
- The bigram HMM tagger with latent variables: with increasing amounts of latent tags
- Penn Chinese Treebank 6.0 (CTB6)
- 28k labeled sentences in total
- 80% for training, 10% for dev, 10% for test
Performance of Chinese POS Taggers

EXPERIMENT SETUP
- The traditional bigram and trigram HMM taggers
- The bigram HMM tagger with latent variables: with increasing amounts of latent tags
- Penn Chinese Treebank 6.0 (CTB6)
- 28k labeled sentences in total
- 80% for training, 10% for dev, 10% for test
EXPERIMENT SETUP

- The traditional bigram and trigram HMM taggers
- The bigram HMM tagger with latent variables: with increasing amounts of latent tags
- Penn Chinese Treebank 6.0 (CTB6)
- 28k labeled sentences in total
- 80% for training, 10% for dev, 10% for test
Self-training (ST)

- A simple semi-supervised learning method

Hand Labeled Base Learner

Unlabeled
Self-training (ST)

- A simple semi-supervised learning method
Self-training (ST)

A simple semi-supervised learning method

- **Base Learner**
- **Hand Labeled**
- **Model**
- **Train**
- **Label**
- **Unlabeled**
- **Automatically Labeled**
Self-training (ST)

A simple semi-supervised learning method

Diagram:
- Unlabeled data
- Labeling
- Automatically labeled data
- Hand labeled data
- Model
- Base learner
- New model
Self-training
Chinese POS Taggers
[Huang et al. 2009]

EXPERIMENT SETUP

- Penn Chinese Treebank 6: 28k labeled sentences in total; 80% for training, 10% for dev, 10% for test
- 210k unlabeled Chinese newswire sentences for self-training
- Use 10%, 20%, 40%, 60%, and 80% of the hand labeled training data
- Always use the full unlabeled training set (210k sentences) for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
Self-training
Chinese POS Taggers
[Huang et al. 2009]

EXPERIMENT SETUP

- Penn Chinese Treebank 6: 28k labeled sentences in total; 80% for training, 10% for dev, 10% for test
- 210k unlabeled Chinese newswire sentences for self-training
- Use 10%, 20%, 40%, 60%, and 80% of the hand labeled training data
- Always use the full unlabeled training set (210k sentences) for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
**Self-training**

**Chinese POS Taggers**
[Huang et al. 2009]

**EXPERIMENT SETUP**

- Penn Chinese Treebank 6: 28k labeled sentences in total; 80% for training, 10% for dev, 10% for test
- 210k unlabeled Chinese newswire sentences for self-training
- Use 10%, 20%, 40%, 60%, and 80% of the hand labeled training data
- Always use the full unlabeled training set (210k sentences) for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
Self-training
Chinese POS Taggers
[Huang et al. 2009]

EXPERIMENT SETUP

• Penn Chinese Treebank 6: 28k labeled sentences in total; 80% for training, 10% for dev, 10% for test

• 210k unlabeled Chinese newswire sentences for self-training

• Use 10%, 20%, 40%, 60%, and 80% of the hand labeled training data

• Always use the full unlabeled training set (210k sentences) for self-training

• Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
**Self-training**

**Chinese POS Taggers**

[Huang et al. 2009]

---

**EXPERIMENT SETUP**

- Penn Chinese Treebank 6: 28k labeled sentences in total; 80% for training, 10% for dev, 10% for test
- 210k unlabeled Chinese newswire sentences for self-training
- Use 10%, 20%, 40%, 60%, and 80% of the hand labeled training data
- Always use the full unlabeled training set (210k sentences) for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
Self-training
Chinese POS Taggers
[Huang et al. 2009]

EXPERIMENT SETUP

- Penn Chinese Treebank 6: 28k labeled sentences in total; 80% for training, 10% for dev, 10% for test
- 210k unlabeled Chinese newswire sentences for self-training
- Use 10%, 20%, 40%, 60%, and 80% of the hand labeled training data
- Always use the full unlabeled training set (210k sentences) for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model

POS Taggers w/ and w/o Self-Training (ST)

Token Accuracy

Number of Hand Labeled Training Sentences

Latent Bigram
Latent Bigram+ST
**EXPERIMENT SETUP**

- PCFG parser without latent variables: Charniak’s lexicalized PCFG parser
- PCFG parser with latent variables: a modified Berkeley Parser
- Penn Treebank: sections 2-19 for training, 22 for dev, 23 for test.
- 210k unlabeled English newswire sentences for self-training
- Use 20%, 40%, 60%, and 80% of the labeled training data
- Always use the full unlabeled training set for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
Self-training
English Syntactic Parsers

[Huang and Harper 2009]

**EXPERIMENT SETUP**

- PCFG parser without latent variables: Charniak’s lexicalized PCFG parser
- PCFG parser with latent variables: a modified Berkeley Parser
- Penn Treebank: sections 2-19 for training, 22 for dev, 23 for test.
- 210k unlabeled English newswire sentences for self-training
- Use 20%, 40%, 60%, and 80% of the labeled training data
- Always use the full unlabeled training set for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
EXPERIMENT SETUP

- PCFG parser without latent variables: Charniak’s lexicalized PCFG parser
- PCFG parser with latent variables: a modified Berkeley Parser
- Penn Treebank: sections 2-19 for training, 22 for dev, 23 for test.
- 210k unlabeled English newswire sentences for self-training
- Use 20%, 40%, 60%, and 80% of the labeled training data
- Always use the full unlabeled training set for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model
EXPERIMENT SETUP

- PCFG parser without latent variables: Charniak’s lexicalized PCFG parser
- PCFG parser with latent variables: a modified Berkeley Parser
- Penn Treebank: sections 2-19 for training, 22 for dev, 23 for test.
- 210k unlabeled English newswire sentences for self-training
- Use 20%, 40%, 60%, and 80% of the labeled training data
- Always use the full unlabeled training set for self-training
- Combine the hand labeled and automatically unlabeled data in a weighted manner, so that each contributes 50% to the final model

### Syntactic Parsers w/ and w/o Self-Training (ST)

<table>
<thead>
<tr>
<th>Number of Hand Labeled Training Sentences</th>
<th>Parsing F Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>8k</td>
<td>87</td>
</tr>
<tr>
<td>16k</td>
<td>88</td>
</tr>
<tr>
<td>24k</td>
<td>89</td>
</tr>
<tr>
<td>32k</td>
<td>90</td>
</tr>
<tr>
<td>40k</td>
<td>91</td>
</tr>
<tr>
<td>48k</td>
<td>92</td>
</tr>
</tbody>
</table>

- Latent PCFG
- Latent PCFG+ST
Why?

- Conventional Models
  - Fixed parameterization
  - Limited benefit from more labeled training data, self-labeled data
- Models with latent variables
  - Flexible parameterization with varying granularities
  - Less data → fewer latent annotations
  - More data → more latent annotations
  - Benefit more from more labeled training data, even imperfect self-labeled data.
Why?

- Conventional Models
  - Fixed parameterization
  - Limited benefit from more labeled training data, self-labeled data

- Models with latent variables
  - Flexible parameterization with varying granularities
  - Less data → fewer latent annotations
  - More data → more latent annotations
  - Benefit more from more labeled training data, even imperfect self-labeled data.
Why?

- Conventional Models
  - Fixed parameterization
  - Limited benefit from more labeled training data, self-labeled data
- Models with latent variables
  - Flexible parameterization with varying granularities
  - Less data $\rightarrow$ fewer latent annotations
  - More data $\rightarrow$ more latent annotations
  - Benefit more from more labeled training data, even imperfect self-labeled data.

![PCFG Parser with Latent Variables](chart.png)

**Parsing F Score**

- **Test**
- **Train**

**Number of Latent Tags**

- Test (ST)
- Train (ST)
Future Work

- Theoretical understanding of self-training
- Introducing latent variables for other NLP tasks
- Scaling up models to use more unlabeled data
- Automatic data selection
- Domain adaptation