Learning by Bootstrapping

Anoop Sarkar
School of Computing Science
Simon Fraser University
http://natlang.cs.sfu.ca

Acknowledgements

• This is joint work with my students Gholamreza Haffari (Ph.D.) and Max Whitney (B.Sc.) at SFU.
• Thanks to Michael Collins for providing the named-entity dataset and answering our questions.
• Thanks to Damianos Karakos and Jason Eisner for providing the word sense dataset and answering our questions.
Supervised &
Unsupervised
Machine Learning

Supervised Learning

Word sense disambiguation:

- ... company said the plant is still operating.

  factory sense +

- ...and divide life into plant and animal kingdom.

  living organism sense -
Supervised Learning

Word sense disambiguation:
- ... company said the plant is still operating.
- ... and divide life into plant and animal kingdom.

Supervised Word Sense Disambiguation:
1. Label a large number of sentences with the correct sense y
2. Each sentence x is mapped to predictive features $f_k(x,y)$
3. Using labeled data, learn a weight $w_k$ for each $f_k$
4. Weighted features active in x provide $score(x,y)$ for any given x
5. Label any new input x using $score(x,y)$: output y with best score
Unsupervised Learning

• Can we learn the labels without any supervision?

• Assumptions about unsupervised learning
  – Clustering (group by similarity)
  – Maximum Likelihood (generative models)
  – Co-training (learn from agreement with others)
  – Self-training (learn from agreement between features)
Problem with Clustering

- Having similar elements grouped is not enough
- Which class corresponds to which cluster?
- **Identifiable problem** = a problem in which a small number of labeled examples help identify the class of a cluster
- Problem: natural language learning tasks are not easily identifiable

**Maximum Likelihood (EM)**

Word sense disambiguation:

\[
\begin{align*}
\text{(sense +/−)} & \ \text{y} \\
\text{(company, operating)} & \ f_{20} f_{12} \\
\text{(life, animal, kingdom)} & \ f_{64} f_{2} f_{35}
\end{align*}
\]

Construct a probabilistic model \( P(y, x) \):

1. \( P(y, x) = P(y, f_1(x), ..., f_m(x)) = P(y) \times P(f_1 | y) \times ... \times P(f_m | y) \)
2. Some examples are labeled (\( y \) is known) others are not
3. Likelihood of the data \( L = P(y_1, x_1) \times ... \times P(y_m, x_m) \times [ P(+, x_{m+1}) + P(-, x_{m+1}) ] \times ... \times [ P(+, x_n) + P(-, x_n) ] \) labeled & unlabeled
4. The EM algorithm: searches for values of \( P(y) \) and \( P(f_k | y) \) to give the maximum value for \( L \)
Co-training

... says \texttt{[NE Maury Cooper]}, a vice \texttt{[CONTEXT president]} at S. & P.

- Is \texttt{Maury Cooper} a PERSON name?
- Assume a feature in the context (\texttt{president}) predicts that \texttt{Maury Cooper} is a PERSON name
- This creates a newly labeled item, the feature \texttt{Cooper} can now be associated with PERSON
- In another example, the feature \texttt{Cooper} can now be sufficient to label \texttt{Mr. Cooper} as a PERSON
- More importantly, this new example indicates that the feature \texttt{old} is now likely to modify a PERSON
- The feature \texttt{old} modifying other noun phrases can then be used to label them as PERSON, and so on ...

... hired \texttt{[NE Mr. Cooper]} , 61 years \texttt{[CONTEXT old]}, as director .

Self-Training / Yarowsky Algorithm

- **Example:** disambiguate 2 senses of \texttt{sentence}
- **Seed rules:**
  - If \texttt{context contains served}, label +1, conf = 1.0
  - If \texttt{context contains reads}, label -1, conf = 1.0
- **Seed rules label 8 out of 303 unlabeled examples**
- **Create new rules from these 8 pseudo-labeled examples**
  - If feature \texttt{f} co-occurs with served, label +1, conf = \Pr(+1|f)
  - If feature \texttt{f} co-occurs with reads, label -1, conf = \Pr(-1|f)
  - Feature \texttt{f} could co-occur with both served & reads
- **These 8 pseudo-labeled examples provide 6 rules above 0.95 conf threshold** (including the original seed rules) e.g.
  - If \texttt{context contains read}, label -1, conf = 0.953
- **These 6 rules label 151 out of 303 unlabeled examples**
**Example:** disambiguate 2 senses of *sentence*

- These 151 pseudo-labeled examples provide 60 rules above the threshold, e.g.
  - If *context contains prison*, label +1, conf = 0.989
  - If *prev word is life*, label +1, conf = 0.986
  - If *prev word is his*, label +1, conf = 0.983
  - If *next word is from*, label -1, conf = 0.982
  - If *context contains relevant*, label -1, conf = 0.953
  - If *context contains page*, label -1, conf = 0.953

- After 5 iterations, 297/303 unlabeled examples are permanently labeled (no changes possible)
- Building final classifier gives 67% accuracy on test set of 515 sentences. With some “tricks” we can get 76% accuracy.

**Semi-supervised Learning**

- Use few supervised examples to start the learning process
- These labeled examples provide the desired class labels for the categories we will discover
- Four methods to compare:
  - Baseline (knowledge-free)
  - Maximum Likelihood using EM
  - Co-training (requires two views to bootstrap)
  - Self-training (Yarowsky algorithm)
Experiments

Named Entity Classification
(Collins and Singer, 1999)

- 971,476 sentences from the NYT were provided a full syntactic parse
  - Using a statistical parser (Collins parser)
- The task is to identify three types of named entities:
  1. Location (LOC)
  2. Person (PER)
  3. Organization (ORG)
  - 1. not a NE or “don’t know”
Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Appositive Prepositional Phrase attachment

### Syntax -> Lexical Semantics

#### Syntax Tree

```
TOP
  \( \rightarrow \)
  s
  \( \rightarrow \)
  NP-DEC
    \( \rightarrow \)
    NNP, NNP
    \( \rightarrow \)
    NP
      \( \rightarrow \)
      CD, NNS
      \( \rightarrow \)
      JJ
        \( \rightarrow \)
        MD
          \( \rightarrow \)
          VB
            \( \rightarrow \)
            NP
              \( \rightarrow \)
              DT, NN, IN
                \( \rightarrow \)
                DT
                  \( \rightarrow \)
                  JJ
                    \( \rightarrow \)
                    NN
                      \( \rightarrow \)
                      PP-CLA
```

### Named Entity Classification

- **Noun phrases** were extracted that met the following conditions:
  1. The NP contained only words tagged as proper nouns
  2. The NP appeared in the following two syntactic contexts:
     - Modified by an appositive whose head is a singular noun
     - In a prepositional phrase modifying an NP whose head is a singular noun
Named Entity Classification

(Collins and Singer, 1999)

- The task: classify NPs into LOC, PER, ORG
- 89,305 training examples with 68,475 distinct feature types
  - 88,962 was used in CS99 experiments
- 1000 test data examples (includes NPs that are not LOC, PER or ORG)
  - Month names are easily identifiable as not named entities: leaves 962 examples
  - Still 85 NPs that are not LOC, PER, ORG.
  - Clean accuracy over 877; Noisy over 962
Yarowsky Variants

(ABney 2004, Collins and Singer, 1999)

• A trick from Co-training (Blum and Mitchell 1998) is to be cautious. Don’t add all rules above the 0.95 threshold
• Add only \( n \) rules per label (say 5) and increase this amount by \( n \) in each iteration
• Changes the dynamics of learning in the algorithm but not the objective fn
• Two variants: Yarowsky (basic), Yarowsky (cautious)
• Without a threshold: Yarowsky (no threshold)

Results

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Accuracy (Clean)</th>
<th>Accuracy (Noisy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (all organization)</td>
<td>45.8</td>
<td>41.8</td>
</tr>
<tr>
<td>EM</td>
<td>83.1</td>
<td>75.8</td>
</tr>
<tr>
<td>Yarowsky (basic)</td>
<td>80.7</td>
<td>73.5</td>
</tr>
<tr>
<td>Yarowsky (no threshold)</td>
<td>80.3</td>
<td>73.2</td>
</tr>
<tr>
<td>Yarowsky (cautious)</td>
<td>91</td>
<td>83</td>
</tr>
<tr>
<td>Co-Training</td>
<td>91</td>
<td>83</td>
</tr>
</tbody>
</table>
Word Sense Disambiguation

• Data from (Eisner and Karakos 2005)
• Disambiguate two senses each for drug, duty, land, language, position, sentence (Gale et. al 1992)
• Source of unlabeled data: 14M word Canadian Hansards (English only)
• Two seed rules for each disambiguation task from (Eisner and Karakos 2005)

Word Sense Disambiguation

• Just as people become addicted to drugs and alcohol, they become addicted to gambling.
• Why are the socialists and their spouse, the Liberals, acting like intoxicated drug addicts?
• The NDP is the only group in this House which does not need drugs to suffer from fantasies.
• Our young Canadians are not all a bunch of drug addicts, alcoholics and suicidal people.

• Drug information to physicians is being distributed exclusively by the drug companies themselves
• Does the Minister think that the people of Canada are being hosed by these drug companies?
Word Sense Disambiguation

• Just as people become addicted to drugs and alcohol, they become addicted to gambling.
• Why are the socialists and their spouse, the Liberals, acting like intoxicated drug addicts?
• The NDP is the only group in this House which does not need drugs to suffer from fantasies.
• Our young Canadians are not all a bunch of drug addicts, alcoholics and suicidal people.

• Drug information to physicians is being distributed exclusively by the drug companies themselves
• Does the Minister think that the people of Canada are being hosed by these drug companies?

Results

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>drug</th>
<th>land</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeds</td>
<td>alcohol</td>
<td>medical</td>
<td>acres</td>
</tr>
<tr>
<td>Train / Test size</td>
<td>134 / 386</td>
<td>1604 / 1488</td>
<td>303 / 515</td>
</tr>
<tr>
<td>Yarowsky (basic)</td>
<td>53.3</td>
<td>79.3</td>
<td>67.7</td>
</tr>
<tr>
<td>Yarowsky (no threshold)</td>
<td>52</td>
<td>79</td>
<td>64.8</td>
</tr>
<tr>
<td>Yarowsky (cautious)</td>
<td>55.9</td>
<td>79</td>
<td>76.1</td>
</tr>
<tr>
<td>DL-CoTrain (2 views = long distance v.s. immediate context)</td>
<td>53.1</td>
<td>77.7</td>
<td>75.9</td>
</tr>
</tbody>
</table>
Summary

• Start from a small set of seed rules.
• Bootstrapping works by trading precision for recall – very cautiously.
  – Precision: number of correct predictions (be conservative = make fewer predictions)
  – Recall: how many correct examples were recovered (be rash = make lots of predictions)
• Effective in learning diverse natural language tasks (finding names, identifying word senses, etc.)
• Questions that I did not address (yet):
  – Does the choice of seed rules matter in bootstrapping?
  – Can bootstrapping be used for complex tasks like translation?
  – Is there a theoretical analysis of bootstrapping?

Seed Rules
Seeds


• Selecting seed rules: what is a good strategy?
  - **Frequency**: sort by frequency of feature occurrence
  - **Contexts**: sort by number of other features a feature was observed with
  - **Weighted**: sort by a weighted count of other features observed with feature.
    
    $\text{Weight}(f) = \frac{\text{count}(f)}{\Sigma_{f'} \text{count}(f')}$

• In each case the frequencies were taken from the unlabeled training data
• Seeds were extracted from the sorted list of features by manual inspection and assigned a label (the entire example was used)
• Location (LOC) features appear infrequently in all three orderings
• It is possible that some good LOC seeds were missed
## Seeds

<table>
<thead>
<tr>
<th>Number of Rules</th>
<th>Frequency</th>
<th>Contexts</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n/3) rules/label</td>
<td>Clean</td>
<td>Noisy</td>
<td>Clean</td>
</tr>
<tr>
<td>3</td>
<td>84</td>
<td>77</td>
<td>84</td>
</tr>
<tr>
<td>9</td>
<td><strong>91</strong></td>
<td><strong>83</strong></td>
<td>90</td>
</tr>
<tr>
<td>15</td>
<td><strong>91</strong></td>
<td><strong>83</strong></td>
<td><strong>91</strong></td>
</tr>
<tr>
<td>7 (CS99)</td>
<td>Clean: <strong>91</strong></td>
<td>Noisy: <strong>83</strong></td>
<td></td>
</tr>
</tbody>
</table>
Statistical Machine Translation

Learn to translate from previously translated text.
Align words in a parallel text.
Extract phrases based on the word alignment.
Translate by using a probabilistic model to combine and then reorder phrases.

Self-training for MT

• Can a machine translation system learn by translating twice?
• Translate a second time by observing its own output translation.
• Why does it work? Reinforces parts of the phrase translation model which are relevant for test corpus
• Glue phrases from test data used to compose new phrases (most phrases still from original phrase table)

<table>
<thead>
<tr>
<th>eval-04</th>
<th>editorials</th>
<th>newswire</th>
<th>speeches</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentences</td>
<td>449</td>
<td>901</td>
<td>438</td>
</tr>
<tr>
<td>selected translations</td>
<td>101</td>
<td>187</td>
<td>113</td>
</tr>
<tr>
<td>size of adapted phrase table</td>
<td>1,981</td>
<td>3,591</td>
<td>2,321</td>
</tr>
<tr>
<td>adapted phrases used</td>
<td>707</td>
<td>1,314</td>
<td>815</td>
</tr>
<tr>
<td>new phrases</td>
<td>679</td>
<td>1,359</td>
<td>657</td>
</tr>
<tr>
<td>new phrases used</td>
<td>23</td>
<td>47</td>
<td>25</td>
</tr>
</tbody>
</table>
Self-training for MT

Table X. Translation examples* from the 2006 GALE corpus.

| baseline    | [the report said] [that the] [united states] [is] [a potential] [problem] [the] [practice of] [china 's] [foreign policy] [is] [likely to] [weaken] [us] [influence] [.] |
| adapted     | [the report] [said that] [this is] [a potential] [problem] [in] [the] [united states] [,] [china] [is] [likely to] [weaken] [the impact of] [american foreign policy] [.] |
| reference   | the report said that this is a potential problem for america. china 's course of action could possibly weaken the influence of american foreign policy. |

| baseline    | [what we advocate] [his] [name] |
| adapted     | [we] [advocate] [him] [.] |
| reference   | we advocate him. |
Analysis of Self-Training

(Features) $F$  

(Instances) $X$

We propose to view bootstrapping as propagating the labels of initially labeled nodes to the rest of the graph nodes.
Converges in Poly time \( O(|F|^2 |X|^3) \)