Statistical parsing

Fei Xia
Feb 27, 2009
CSE 590A
Statistical parsing

• History-based models (1995-2000)

• Recent development (2000-present):
  – Supervised learning: reranking and label splitting
  – Semi-supervised learning: self-training, adaptation
  – Unsupervised learning: prototype learning, etc.

  – Dependency parsing
History-based models
Major studies

English Penn Treebank was released in 1992 (v0.5) and 1995 (v2).

• (Magerman, 1995)
• (Collins 1996, 1997, 1999)
• (Ratnaparkhi 1997)
• (Charniak 1997, 2000)
• (Henderson 2003)
• ...
History-based model

\[ y^* = \text{argmax}_y P(x, y) \quad y^* = \text{argmax}_y P(y|x) \]

A 1-to-1 mapping is defined between \((x, y)\) and a decision sequence \(<d_1, d_2, \ldots, d_n>\)

\[ P(x, y) = \prod_{i=1}^{n} P(d_i|\phi(d_1, d_2, \ldots, d_{i-1})) \]

The models vary w.r.t. learner, \(d_i\) and \(\phi\).

Ex: PCFG: \( P(x, y) = \prod_{i}^{n} P(A \Rightarrow \beta|A) \)
Base model in (Collins, 1999)

\[ VP(saw) \rightarrow VBD\ (saw)\ NP-C\ (her)\ NP\ (today) \]
The first two passes in (Ratnaparkhi, 1997)

1\textsuperscript{st} pass: POS tagging: \texttt{tagset} = \{NN, VBD, DT, ...\}

\begin{verbatim}
PRP  VBD  DT  NN  IN  DT  NN
  |    |    |    |    |    |    |
  I   saw  the  man  with  the  telescope
\end{verbatim}

2\textsuperscript{nd} pass: chunking: \texttt{tagset} = \{start-X, join-X, Other\}

\begin{verbatim}
Start NP  Other  Start NP  Join NP  Other  Start NP  Join NP
  |        |        |        |        |        |        |
  PRP     VBD     DT     NN     IN     DT     NN
  |        |        |        |        |        |        |
  I       saw     the     man     with     the     telescope
\end{verbatim}

\begin{verbatim}
NP  VBD
  |    |
  PRP saw
  |    |
  I

NP
  |   |
  DT  NN
  |   |
  the  man

IN
  |   |
  with

NP
  |   |
  DT  NN
  |   |
  the  telescope
\end{verbatim}
The third pass

BUILD: Start X, Join X
CHECK: Yes, No
The third pass
The third pass

Start S
NP
PRP
I

VP
VBD
saw

NP
DT
NN
the
man

IN
with

PP
IN
NP

with
DT
NN
telescope

NP

PRP
I

VP
VBD
saw

NP
DT
NN
the
man
The third pass

Start S
  NP
    PRP
      I

VBD
  saw

Join S
  VP
    VBD
      saw
    NP
      DT
        the
      NN
        man
    IN
      with
    NP
      DT
        the
      NN
        telescope
The third pass

I saw the man with the telescope.
• Three passes:
  – POS tagging: tagset = \{NN, VB, DT, \ldots\}
  – Chunking: tagset = \{Start-X, Join-X, other\}
  – Build a parse tree: shift-reduce parser
    • BUILD: Start-X, Join-X
    • CHECK: Yes, No

• Learner: four MaxEnt classifiers
  – POS tagger
  – chunking
  – “BUILD”
  – “CHECK”

• Use breadth-first search to find a good decision sequence.
Summary of history-based model

• There is a 1-to-1 mapping between \((x,y)\) and a decision sequence.

\[
P(x, y) = \prod_{i=1}^{n} P(d_i | \phi(d_1, d_2, \ldots, d_{i-1}))
\]

• The probability of each decision is estimated from the training data (e.g., MaxEnt, MLE).

• A good decision sequence is found by breath-first search etc.
Comparing different models

• Dataset: English Penn Treebank, WSJ data
  – Training data: Section 02-21
  – Dev data: Section 01, 22, 24
  – Final test data: Section 23

• Evaluation measures: precision, recall

\[ f\text{-score} = \frac{2PR}{P+R}, \text{ P is precision, R is recall} \]
# Results

<table>
<thead>
<tr>
<th>Parser</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCFG</td>
<td>74.8%</td>
<td>70.6%</td>
</tr>
<tr>
<td>(Magerman, 1995)</td>
<td>84.3%</td>
<td>84.0%</td>
</tr>
<tr>
<td>(Collins, 1996)</td>
<td>85.7%</td>
<td>85.3%</td>
</tr>
<tr>
<td>(Charniak, 1997)</td>
<td>86.6%</td>
<td>86.7%</td>
</tr>
<tr>
<td>(Ratnaparkhi, 1997)</td>
<td>87.5%</td>
<td>86.3%</td>
</tr>
<tr>
<td>(Collins, 1997)</td>
<td>88.1%</td>
<td>87.5%</td>
</tr>
<tr>
<td>(Collins, 1999)</td>
<td>88.3%</td>
<td>88.1%</td>
</tr>
<tr>
<td>(Charniak, 2000)</td>
<td>89.5%</td>
<td>89.6%</td>
</tr>
</tbody>
</table>

Test data: Section 23, Sentences with <= 100 words, labeled precision/recall
Remaining issues

• Adding new features (esp. global features) to the model could be difficult: e.g., each sentence should have at least one VP.
  ➔ Reranking

• The models can be very complicated.
  ➔ PCFG with split labels

• Parsers trained on one domain does not work well in another domain.
  ➔ Parsing adaptation

• Creating a treebank is very expensive.
  ➔ semi-supervised and/or unsupervised parsing.
Statistical parsing

• History-based models (1995-2000)

• Recent development (2000-present):
  – Supervised learning: reranking and label splitting
  – Semi-supervised learning: self-training, adaptation
  – Unsupervised learning: prototype learning, etc.
  – Dependency parsing
Reranking

a sentence -> Parser -> Reranker 

topN parse trees -> Reranker 

reranked list
Main ideas

• Training data:
  \( \{(s_i, t_i)\} \) and for each \( s_i \), a set of candidates \( C(s_i) = \{x_{ij}\} \).

• Test data:
  \( \{s_i\} \) and for \( C(s_i) \) for each \( s_i \)

• Represent each parse \( x \) as a feature vector:
  \[
  h(x) = (h_1(x), h_2(x), \ldots, h_n(x))
  \]
  Training: calculate \( \mathbf{\tilde{w}} \)
  Decoding: \( x^* = \arg\max_{x \in C(s)} \mathbf{\tilde{w}} \cdot h(x) \)

Choice of features/kernels, learner, modeling (e.g., objective functions)
Major studies

• (Collins, 2000):
• (Collins and Duffy, 2001)
• (Shen et al., 2003)
• (Charniak and Johnson, 2005)
• ...
(Collins and Duffy, 2001)

- Features: subtrees in the parse tree

- Reranker learner: SVM and voted perceptron
  - Define a tree kernel

- Results: 74% (unlexicalized PCFG) => 80% (f-score)
(Charniak and Johnson, 2005)

• Features:
  – Probability of the parse tree
  – Subtrees
  – The conjunct parallelism at various levels: “XP1 and XP2”
  – ...

• Reranker learner: MaxEnt

\[
P(x_{i,j} | C(s_i)) = \frac{e^{\sum_k \lambda_k f_k(x_{i,j})}}{Z}
\]

\[
L(\lambda^n_1) = - \sum_i \log P(x_{i1} | C(s_i)) \quad \text{where } x_{i1} = t_i
\]

\[
\lambda^* = \text{argmin}_\lambda L(\lambda)
\]

• Results: 0.897 \(\Rightarrow\) 0.91 (oracle result for top50 is 0.968)
Label splitting
The Game of Designing a Grammar

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Parent annotation [Johnson ’98]

```
S
  /\   \\
NP^S     VP
   /\   /\   \\
PRP VBD NP^VP
   /   /   \\
She heard the noise
```
The Game of Designing a Grammar

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Parent annotation [Johnson ’98]
  - Head lexicalization [Collins ’99, Charniak ’00]
The Game of Designing a Grammar

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Parent annotation [Johnson ’98]
  - Head lexicalization [Collins ’99, Charniak ’00]
  - Automatic clustering?
Major studies

• (Klein and Manning, 2003)

• (Matsuzaki et al., 2005)

• (Petrov et al., 2006)
• Manually split categories
  – NP: subject vs object
  – DT: determiners vs demonstratives
  – IN: sentential vs prepositional

• Advantages:
  – Fairly compact grammar
  – Linguistic motivations

• Disadvantages:
  – Performance leveled out
  – Manually annotated

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Treebank grammar</td>
<td>72.6</td>
</tr>
<tr>
<td>Klein &amp; Manning ’03</td>
<td>86.3</td>
</tr>
</tbody>
</table>
(Petrov et al., 2006)

• Automatic splitting:
  – $S \rightarrow NP\ VP$     vs.     $S_{25} \rightarrow NP_{12}\ VP_{5}$
  – Learn PCFG with modified EM algorithm: brackets are known, base categories are known. We only need to induce subcategories.

• Automatic splitting could result in large grammar and run into sparse data problem
  – Merging back symbols when the gain is small
  – Smoothing
Split-and-merge algorithm

• Extracting the baseline PCFG from the treebank

• Repeat
  – Split each category (e.g., NP, N) into two
  – Initialize EM with the result of the previous grammar, and re-train the new grammar.
  – Merge some subcategories back
Parsing results

- Naïve treebank grammar: 72.6
- Splitting: 88.4
- Merging: 89.5
- Smoothing: 90.2
- (Klein and Manning, 2003): 86.3
- (Matsuzaki et al. 2005): 86.7
- (Collins, 1999): 88.6
- (Charniak and Johnson, 2005): 90.1
Semi-supervised learning
Self-training
(McClosky et al., NAACL 2006)

• Setting:
  – Training data:
    • Labeled data: WSJ
    • Unlabeled data: NANC
  – Test data: WSJ

• Self-training procedure:
  – Train a stage-1 parser and a reranker with WSJ data
  – Parse NANC data and add the best parse to re-train stage-1 parser

• Best parses for NANC sentences come from
  – the stage-1 parser ("Parser-best")
  – the reranker ("Reranker-best")
Conclusion:
• Self-training alone does not help
• Self-training with reranking provides a modest gain
### Self-training for parsing adaptation

<table>
<thead>
<tr>
<th>Sentences added</th>
<th>Parser</th>
<th>Reranking Parser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline BROWN</td>
<td>86.4</td>
<td>87.4</td>
</tr>
<tr>
<td>Baseline WSJ</td>
<td>83.9</td>
<td>85.8</td>
</tr>
<tr>
<td>wsJ+50k</td>
<td>84.8</td>
<td>86.6</td>
</tr>
<tr>
<td>wsJ+250k</td>
<td>85.7</td>
<td>87.2</td>
</tr>
<tr>
<td>wsJ+500k</td>
<td>86.0</td>
<td>87.3</td>
</tr>
<tr>
<td>wsJ+750k</td>
<td>86.1</td>
<td>87.5</td>
</tr>
<tr>
<td>wsJ+1,000k</td>
<td>86.2</td>
<td>87.3</td>
</tr>
<tr>
<td>wsJ+1,500k</td>
<td>86.2</td>
<td>87.6</td>
</tr>
<tr>
<td>wsJ+2,000k</td>
<td>86.1</td>
<td>87.7</td>
</tr>
<tr>
<td>wsJ+2,500k</td>
<td>86.4</td>
<td>87.7</td>
</tr>
</tbody>
</table>

→ Adding NANC data helps: 83.9% => 86.4%
Summary

• History-based models: 0.73 => 0.897
  – It is hard to add more features
  – It does not work well for another domain
  – It requires labeled training data

• Reranking:
  – (Collins and Duffy, 2001): 0.73 => 0.80 (PCFG, subtree features)
  – (Charniak and Johnson, 2005): 0.897 => 0.91

• Label splitting: 0.73 => 0.90

• Self training (reranker-best):
  – In-domain (WSJ): 0.903 => 0.91
  – Out-of-domain (WSJ+NANC, Brown): 83.9 => 86.4

• Unsupervised learning:
  – Prototype learning: 0.263 => 0.651
References
History-based models


Reranking


Label splitting


• Petrov et al., 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In ACL 2006.
Semi-supervised learning


