Latent TAG derivations for Semantic Role Labeling

Anoop Sarkar
(Joint work with Yudong Liu & Gholamreza Haffari)

School of Computing Science
Simon Fraser University
British Columbia, Canada
http://natlang.cs.sfu.ca
Semantic Role Labeling
Semantic Role Labeling (SRL)

• For a given verb (predicate), SRL aims to identify and label all its arguments with semantic roles, such as Agent, Patient, and Theme

[Ports of Call Inc.] reached agreements to sell [its remaining seven aircraft] [to buyers that weren't disclosed].

A0: seller

A1: goods

A2: buyer
SRL in NLP applications

Document summarization
(SFU team: SQUASH: Melli et al. DUC-2005)
- sentence selection
- sentence compression

Semantic entailment
(Braz et al., 2005)

Machine translation
(Wu and Fung, 2009)

Co-reference resolution
(Ponzetto and Strube, 2006)

Question answering
(Shen and Lapata, 2007, Stenchikova et al. 2005)

Information retrieval
(Surdeanu et al., 2003)

Verb sense disambiguation
(Dang and Palmer, 2005)

Automatic case marker prediction in Japanese
(Suzuki and Touranova, 2006)

...
High accuracy is achieved by

Proposing **new types of features** from **different syntactic views**
- chunks (Hacioglu et al., 2004)
- parses (Gildea and Jurafsky, 2002, Gildea and Palmer, 2002; Punyakanok et al., 2005)
- CCG derivations (Gildea and Hockenmaier, 2003)
- dependency trees (Hacioglu et al., 2004)

Modeling the predicate **frameset** between arguments: A0 A0 V A2 A1
(Gildea and Jurafsky, 2002; Pradhan et al., 2004; Toutanova et al., 2008; Punyakanok et al., 2008)

Dealing with incorrect parser output by **using more than one parse**
(Punyakanok et al., 2005; Toutanova et al., 2008; Pradhan et al., 2005)
Our work

Proposing new types of features from different syntactic views

- chunks (Hacioglu et al., 2004)
- parses (Gildea and Jurafsky, 2002; Gildea and Palmer, 2002; Punyakanok et al., 2005)
- CCG derivations (Gildea and Hockenmaier, 2003)
- dependency trees (Hacioglu et al., 2004)
- Lexicalized Tree Adjoining Grammars (TAG) derivations (Liu and Sarkar EMNLP 2007)
Architecture of our SRL system
Tree adjoining Grammars (TAG)
TAG derivations from Treebanks or phrase-structure parses

Magerman-Collins head percolation heuristic rules

China’s 14 open border cities marked economic achievements
TAG derivations from Treebanks or phrase-structure parses

S

NP

NP

NNP

POS

CD

JJ

NN

NNS

VBD

JJ

NNS

Magerman-Collins head percolation heuristic rules

China 's 14 open border cities marked economic achievements
TAG derivations from Treebanks or phrase-structure parses

China ‘s 14 open border cities marked economic achievements
China’s 14 open border cities marked economic achievements
SRL and TAG

• TAG is closely related to SRL due to its *extended domain of locality*

• TAG provides an alternative syntactic view for SRL feature selection
TAG derivations for SRL

sentence → TAG parser → TAG derivations → SRL

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>phrase-structure</td>
<td>85.8</td>
<td>87.7</td>
</tr>
<tr>
<td>TAG</td>
<td>85.8</td>
<td>85.6</td>
</tr>
</tbody>
</table>
TAG derivations for SRL

1. Sentence
2. TAG parser
3. TAG derivations
4. State of the art phrase structure parser
5. Converter
6. SRL

Phrase structure parses
TAG derivations for SRL

Sentence → TAG parser → TAG derivations

TAG parser → state of the art phrase structure parser

state of the art phrase structure parser → phrase structure parses → converter

Converter → std

TAG derivations → SRL

SRL → tag
TAG for SRL

Extended domain of locality (EDL)

(Chen and Rambow, 2003)

only $\sim 87\%$ of dependencies between predicate and (core) argument are captured in gold trees.
TAG for SRL

(Liu and Sarkar, EMNLP 2007)

- Magerman-Collins head percolation rules (Chiang, 2000)
- Sister-adjunction operation (Schabes and Shieber, 1994)
- path feature less sparse
the example revisited: [seller *Ports of Call Inc.*] reached agreements to sell [goods *its remaining seven aircraft*] [buyer to buyers that weren't disclosed].
Ports of Call Inc. reached agreements to sell its remaining seven aircraft to buyers that weren't disclosed.

Argument-adjunct distinction:
All elementary trees are in spinal form
Sister-adjunction
TAG features

Example \(<\text{sell, NP(agreement)}>\>

- **P**redicate elementary tree features
- **A**rgument elementary tree features
- **I**ntermediate elementary tree features

- **T**opological relations in TAG derivations:
  - distance between e-trees
  - relative position
  - modifying relations.

- Feature analysis shows that adding all feature types improves accuracy.
Motivation for latent derivations

(Chen and Rambow, 2003)

(Liu and Sarkar, EMNLP 2007)
Motivation for latent derivations

• Observations:
  – For a single derived tree, **multiple** TAG derivations exist and can be treated as latent structures
  – TAG derivations can **localize** long distance dependencies and provide useful features for SRL

• Hypothesis:
  – For different SRL instances, possibly **different** latent TAG derivations can provide discriminative features
  – Use TAG features to search for **more accurate SRL** classifiers. Do **not** search for “good” TAG derivations.

• Head choice in head-percolation and Lexical choice:
  – Extend head-percolation heuristics to generate multiple **predicate** e-trees and associated argument e-trees
  – Enumerate **all possible lexical heads** for argument constituents
Generating latent TAG derivations <x, NP>

- p1: Magerman-Collins head percolation rule
- p2: consecutive VPs from predicate x to the 1st non-VP node
- p3: from predicate to the root node

Wi:

\[ \vdots \]

Wi+1:

\[ \vdots \]

Wi+2:

\[ \vdots \]

Wi+3:

\[ \vdots \]

\[ p_1 \]

\[ p_1 \]

\[ p_2 \]

\[ p_3 \]

\[ a_1 \]

\[ a_2 \]

\[ a_6 \]

\[ a_7 \]

\[ a_{11} \]

\[ a_{16} \]

\[ a_{16} \]

\[ a_{16} \]

\[ p_i: \text{variant } i \text{ of predicate etree} \]

\[ a_i: \text{variant } i \text{ of argument etree} \]
Derivation using p1

NP(\text{support})

SBAR(\text{was})

VP(\text{struggling})

S(\text{to})

VP(\text{rebuild})

Derivation using p2

NP(\text{support})

SBAR(\text{struggling})

VP(\text{rebuild})

S

\text{pred}
Generating latent TAG derivations

• The set of features includes the three intermediate elementary trees closest to the predicate (if they exist)
• The average number of TAG derivations per SRL decision is ~130
• Problems with using latent features:
  – Scaling to millions of features and unlimited input length
  – Effectively use such a large number of latent features
  – Focus on discriminative features for each SRL instance
• Solution:
  – Latent support vector machines (LSVM)
  – Train several binary classifiers using LSVM and combine them using one vs. all for the full SRL task
Latent Support Vector Machines
SRL as binary classification

- <pred, arg candidate> pair
- arg candidates taken from the original derived tree
- all depth-1 node in the pruned tree
SRL as binary classification

\[ L_D(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i)) \]

Accurate SRL involves finding the right feature vector with a large classification margin.

\[ f_w(x) = w \cdot \Phi(x) \]
Latent SVM

\[ L_D(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i)) \]

\[ f_w(x) = \max_{h \in H(x)} w \cdot \Phi(x, h) \]
Previous work

Object detection in images
[P. Felzenszwalb et al. 2008]

Sentence classification for language modeling (in MT)
[Cherry & Quirk, 2008]
Semi-convexity (Felzenszwalb et al. 2008)

- \( f_w(x) = \max_{h \in H(x)} w \cdot \Phi(x, h) \)
- Maximum of convex function is convex, thus
  \( f_w(x) = \max_{h \in H(x)} w \cdot \Phi(x, h) \) is convex in \( w \), thus
  \[ \max(0, 1 - y_i f_w(x_i)) \] is convex for negative examples

- \( L_D(w) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i)) \)

Objective \( L_D(w) \) becomes convex if we fix the latent structure \( h \) for positive examples.
Latent SVM training

- Two-step optimization algorithm (Felzenszwalb et al. 2008):
  - Initialize $w$ and iterate:
    1. Pick best $h$ for each positive example. For each training example $x$ pick $h = \text{argmax}_h w \cdot \Phi(x,h)$
    2. Find $w$ for objective function with fixed $h$ optimized using online learning (stochastic gradient descent)
      - $\text{svmsgd}$ (Léon Bottou)
  - In our implementation:
    - examples: $<$predicate, argument candidate$>$ pair
    - $h$: best latent TAG derivation, picked for each positive and negative SRL instance
Latent SVM training

- For each training example, the phrase-structure tree remains fixed
  - Gold Treebank phrase-structure tree is used for training
  - Charniak parser output (from CoNLL 2005 shared task) is used for test data
- All the latent TAG derivations for a given sentence produce the same phrase-structure tree
- Each word lexicalizes one tree each and so all derivations have same number of steps
Experimental Results
Experimental Setup

• Data:
  – CoNLL-2005 shared task released data
  – PropBank Section 02-21 for training, 23 for testing

• Argument Set Under Consideration:
  – \{A0, A1, A2, A3, A4, A5, AM-* \& R-A*\}

• Model: one-vs-all binary classifiers
  – Svmmsgd (linear kernel)

• Evaluation metrics: Precision/Recall/F-score

• Baseline1: std

• Baseline2: std + tag (Liu and Sarkar 2007)

• Initial weights for LSVM iterations are from Baseline2
Architecture of our SRL system

• On a given parse tree, run the pruning component: some candidate spans are potential arguments, the others are labeled NONE
• Run a binary classifier for identification and have some spans labeled ARG and the rest NONE
• Run binary classifiers for classification: A0 vs not-A0, A1 vs not-A1, etc. on the nodes labeled ARG
• Combine output of binary classifiers using one vs all
  – for each ARG node pick binary classifier with highest confidence and decide the label of each node: A0, A1, A2, ...
• Convert output to CoNLL 2005 shared task format and run CoNLL05 evaluation script.
### CoNLL 2005 Shared Task / Charniak parser

<table>
<thead>
<tr>
<th></th>
<th>Toutanova et al. (2008)</th>
<th>LSVM-SRL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>Overall</td>
<td>81.90</td>
<td>78.81</td>
</tr>
<tr>
<td>A0</td>
<td>88.37</td>
<td>88.91</td>
</tr>
<tr>
<td>A1</td>
<td>81.50</td>
<td>81.27</td>
</tr>
<tr>
<td>A2</td>
<td>73.44</td>
<td>68.74</td>
</tr>
<tr>
<td>A3</td>
<td>75.00</td>
<td>55.49</td>
</tr>
<tr>
<td>A4</td>
<td>74.74</td>
<td>69.61</td>
</tr>
<tr>
<td>A5</td>
<td>100.00</td>
<td>80.00</td>
</tr>
<tr>
<td>AM-*</td>
<td>78.19</td>
<td>69.98</td>
</tr>
<tr>
<td>R-AM-*</td>
<td>73.91</td>
<td>61.44</td>
</tr>
</tbody>
</table>

4/23/10
Experimental results: F-score

<table>
<thead>
<tr>
<th>A2</th>
<th>Multiple TAG dv (LSVM)</th>
<th>1 TAG dv</th>
<th>0 TAG dv</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F-score values:

- A2: 78.72, 79.09, 84.36
- Id: 81.13, 83.34, 89.99
- A1: 87.01, 86.26, 92.74
- A0: 87.68, 85.87, 94.57
- A4: 79.01, 78.26, 87.91
- AM: 81.09, 81.11, 91.10
- A3: 71.00, 74.39, 86.09

4/23/10
## Analysis: Individual binary classifiers, id, A0 vs. not-A0, etc.

<table>
<thead>
<tr>
<th>class</th>
<th>No TAG (p/r%)</th>
<th>1 TAG deriv</th>
<th>Latent TAG derivs</th>
<th>stop</th>
<th>Recall bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>87.71</td>
<td>84.86</td>
<td>89.00</td>
<td>85.21</td>
<td>98.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>86.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>86.90</td>
</tr>
<tr>
<td>A0</td>
<td>86.46</td>
<td>85.30</td>
<td>87.87</td>
<td>87.50</td>
<td>99.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>94.24</td>
</tr>
<tr>
<td>A1</td>
<td>78.70</td>
<td>83.72</td>
<td>84.56</td>
<td>82.16</td>
<td>99.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>82.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.37</td>
</tr>
<tr>
<td>A2</td>
<td>85.04</td>
<td>73.91</td>
<td>83.00</td>
<td>74.86</td>
<td>99.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76.24</td>
</tr>
<tr>
<td>A3</td>
<td>77.04</td>
<td>65.82</td>
<td>83.46</td>
<td>67.09</td>
<td>98.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78.24</td>
</tr>
<tr>
<td>A4</td>
<td>77.42</td>
<td>79.12</td>
<td>90.14</td>
<td>70.33</td>
<td>98.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80.20</td>
</tr>
<tr>
<td>AM</td>
<td>80.85</td>
<td>81.39</td>
<td>82.10</td>
<td>81.87</td>
<td>97.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.75</td>
</tr>
</tbody>
</table>

4/23/10
Baseline TAG derivation

NP (support)
  | SBAR (was)
  | VP (struggling)
  | S (to)
  | VP (rebuild)

Derivation picked by LSVM

NP (support)
  | SBAR (struggling)
  | VP (rebuild)

arg = A1
Baseline TAG derivation

S(caught)

NP(viewer) S(to)

VP(be)

VP(urge)

S(to)

VP(stay)

Derivation picked by LSVM

PP(In)

NP(viewer) S(stay)
Analysis: F-scores across LSVM iterations
Analysis: change of derivation trees over LSVM iterations

Derivation Trees (%)

- Different from ALL previous
- Different from just previous

Iteration number of LSVM
Analysis: distribution of active features over LSVM iterations
Summary

• Latent TAG derivations and LSVM provide predictive features and very high precision and similar recall.
• LSVM boosts SRL F-score from 80% to 89%
• LSVM picked 8K features from the pool of 1,242,869 all possible.
• Further analysis of LSVM derivation trees is in our NAACL 2010 paper
  – Careful v.s. random initialization in LSVM training
  – How is LSVM taking advantage of the latent derivations?
Current Work

• Log linear models: sum over all latent TAG derivations per SRL decision
• Release of code and output of our system on CoNLL dev and test data
• Learn something about the SRL task from the derivations selected by LSVM
• LSVM is a general learning framework that can be potentially applied to other NLP tasks
• LSVM for (TAG) parsing
Thank you!
Analysis: Does initialization matter?

- Does picking the initial derivation tree carefully matter? Or can we simply select one at random.
- We compared A0-vs-not-A0 classifier with and without random choice of initial TAG derivation with identification classifier remaining the same.

<table>
<thead>
<tr>
<th>1 TAG dv, Magerman-Collins (A0: Baseline2)</th>
<th>1 TAG dv, Random (A0: avg over 5 runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>87.87</td>
<td>87.50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Magerman-Collins init+LSVM (A0: Baseline2)</th>
<th>Random init+LSVM (A0: avg over 5 runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>99.26</td>
<td>90.31</td>
</tr>
</tbody>
</table>
Analysis: Why LSVM does better?

- Compare the LSVM argmax derivation tree with the Baseline2 Magerman-Collins derivation tree.
- Track changes when LSVM was correct and Baseline2 was incorrect.
- Five major categories of changes in derivation trees.

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>A0</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical choice</td>
<td>26.1</td>
<td>44.5</td>
<td>47.7</td>
<td>74.4</td>
<td>28.6</td>
<td>18.8</td>
</tr>
<tr>
<td>Distance</td>
<td>79.9</td>
<td>18.4</td>
<td>15.9</td>
<td>5.3</td>
<td>3.9</td>
<td>6.3</td>
</tr>
<tr>
<td>Predicate etree</td>
<td>90.7</td>
<td>43.4</td>
<td>58.1</td>
<td>74.6</td>
<td>80.5</td>
<td>81.3</td>
</tr>
<tr>
<td>Argument etree</td>
<td>47.1</td>
<td>56.6</td>
<td>65.8</td>
<td>80.6</td>
<td>28.6</td>
<td>25.0</td>
</tr>
<tr>
<td>Intermediate etree</td>
<td>5.7</td>
<td>17.1</td>
<td>14.7</td>
<td>2.3</td>
<td>14.3</td>
<td>6.2</td>
</tr>
</tbody>
</table>
CoNLL 2005 Shared Task / Charniak parser

<table>
<thead>
<tr>
<th></th>
<th>Corr.</th>
<th>Excess</th>
<th>Missed</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>11360</td>
<td>506</td>
<td>2245</td>
<td>95.90</td>
<td>84.05</td>
<td>89.59</td>
</tr>
<tr>
<td>A0</td>
<td>3322</td>
<td>41</td>
<td>241</td>
<td>98.78</td>
<td>93.24</td>
<td>95.93</td>
</tr>
<tr>
<td>A1</td>
<td>3907</td>
<td>342</td>
<td>1020</td>
<td>91.95</td>
<td>79.30</td>
<td>85.16</td>
</tr>
<tr>
<td>A2</td>
<td>821</td>
<td>6</td>
<td>289</td>
<td>99.27</td>
<td>73.96</td>
<td>84.77</td>
</tr>
<tr>
<td>A3</td>
<td>129</td>
<td>13</td>
<td>44</td>
<td>90.85</td>
<td>74.57</td>
<td>81.90</td>
</tr>
<tr>
<td>A4</td>
<td>80</td>
<td>4</td>
<td>22</td>
<td>95.24</td>
<td>78.43</td>
<td>86.02</td>
</tr>
<tr>
<td>A5</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>40.00</td>
<td>80.00</td>
<td>53.33</td>
</tr>
<tr>
<td>AM-*</td>
<td>3101</td>
<td>76</td>
<td>612</td>
<td>97.60</td>
<td>83.51</td>
<td>90.01</td>
</tr>
<tr>
<td>R-AM-*</td>
<td>305</td>
<td>126</td>
<td>3</td>
<td>70.76</td>
<td>99.02</td>
<td>82.54</td>
</tr>
<tr>
<td>V</td>
<td>5126</td>
<td>141</td>
<td>141</td>
<td>97.32</td>
<td>97.32</td>
<td>97.32</td>
</tr>
</tbody>
</table>

4/23/10