Non-local Dependencies and Semantic Role Labeling

Scott Wen-tau Yih
Kristina Toutanova
Microsoft Research (formerly UIUC, Stanford)
Roger Levy
UCSD (formerly Stanford)

Syntactic Variations versus Semantic Roles

Yesterday, Kristina hit Scott with a baseball
Scott was hit by Kristina yesterday with a baseball
Yesterday, Scott was hit with a baseball by Kristina
With a baseball, Kristina hit Scott yesterday
Yesterday Scott was hit by Kristina with a baseball
The baseball with which Kristina hit Scott yesterday was hard
Kristina hit Scott with a baseball yesterday

Syntactic Variations (as trees)

Semantic Role Labeling – Giving Semantic Labels to Phrases

AGENT John broke THEME the window
AGENT Sotheby’s offered THEME a money-back guarantee to THEME a money-back guarantee offered by AGENT Sotheby’s
AGENT Sotheby’s offered THEME a money-back guarantee to THEME a money-back guarantee
AGENT Sotheby’s offered THEME a money-back guarantee to THEME a money-back guarantee

Recovering non-local dependencies

We want to interpret non-local dependencies:

Who did Mary want to see? (WH question)
What did Paul expect to arrive?

We know how to disambiguate (in parsing) with probabilistic context-free grammars
But non-local dependencies aren’t transparently represented by context-free grammars

What is the state of the art in robust computation of linguistic meaning?

Probabilistic context-free grammars trained on syntactically annotated corpora (Treebanks) yield robust, high-quality syntactic parse trees
Nodes of these parse trees are often reliable indicators of phrases corresponding to semantic units (Gildea & Palmer 2002)
Dependency trees from CF trees

- Alternatively, syntactic parse trees can directly induce dependency trees
- Can be interpreted pseudo-propositionally; high utility for Question Answering (Pasca and Harabagiu 2001)

Parses to dependencies: limits

Nonlocal annotation in Penn Treebank

- Intuitively, much more nonlocal dependency in German
- NEGRA directly annotates crossing dependencies, algorithmically maps to a context-free representation.

Nonlocal annotation in NEGRA (German)

- Null complementizers (mediate relativization)
  A. Identify sites for null node insertion
  B. Find best daughter position and insert.
- Dislocated dependencies
  A. Identify dislocated nodes
  B. Relocate to original/deep mother node
  C. Find best daughter position and insert
- Shared dependencies
  A. Identify sites of nonlocal shared dependency
  B. Identify best daughter position and insert
  C. Find controller for each control locus

Three methods of non-local dependency recovery

- Approximate dependency recovery with a context-free parser; correct the output post-hoc (Johnson 2002; present work; also akin to traditional LFG parsing)
- Incorporate non-local dependency information into the category structure of chart parser entries (Collins 1999; Dienes 2003; also akin to traditional G/HPSG, CCG parsing)
- Incorporate non-local dependency information into the edge structure of chart parser entries (Plaehn 2000; TAG)

Tree reshaping via cascaded classification

1. Null complementizers (mediate relativization)
   A. Identify sites for null node insertion
   B. Find best daughter position and insert.
2. Dislocated dependencies
   A. Identify dislocated nodes
   B. Relocate to original/deep mother node
   C. Find best daughter position and insert
3. Shared dependencies
   A. Identify sites of nonlocal shared dependency
   B. Identify best daughter position and insert
   C. Find controller for each control locus.
Use sequence of maxent classifiers. Feature types:
- Syntactic category, parent, grandparent (subj vs obj extraction; VP finiteness)
- Head words (wanted vs to vs. eat)
- Presence of daughters (NP under S)
- Syntactic path (Gildea & Jurafsky 2002): <SBAR,JS,V,P,JS,VP>

Plus: feature conjunctions, specialized features for expletive subject dislocations, passivizations, passing feature information properly through coordinations, etc., etc.


Why is SRL Important?
Applications as a simple meaning rep’n
- Question Answering
  - Q: When was Napoleon defeated?
    Look for: [PATIENT Napoleon] \{defeat-synset\} \{ARGM-TMP *ANS*\}
- Machine Translation
  English (SVO)
  [AGENT The little boy] \{cut\} [THEME the red ball] \{ARGM-MNR hard\} \{PRED zaad-e\}
  Farsi (SOV)
- Document Summarization
  - Predicates and Heads of Roles summarize content
- Information Extraction
  - SRL can be used to construct useful rules for IE

Evaluation on dependency metric: gold-standard input trees

Why is SRL Important?
Applications as a simple meaning rep’n
- Question Answering
  - Q: When was Napoleon defeated?
    Look for: [PATIENT Napoleon] \{defeat-synset\} \{ARGM-TMP *ANS*\}
- Machine Translation
  English (SVO)
  [AGENT The little boy] \{cut\} [THEME the red ball] \{ARGM-MNR hard\} \{PRED zaad-e\}
  Farsi (SOV)
- Document Summarization
  - Predicates and Heads of Roles summarize content
- Information Extraction
  - SRL can be used to construct useful rules for IE

Application: Semantically precise search

Query: afghans destroying opium poppies

Underlying hypothesis: verbal meaning determines syntactic realizations

Beth Levin analyzed thousands of verbs and defined hundreds of classes.

Examples of verbs for each class:
- Break: bork, bang, crack
- Cut: chop, hack, saw
- Eat: bite, nibble, gag
Frames in FrameNet

FrameNet [Fillmore et al. 01]

Methodology for FrameNet

While (remaining funding > 0) do
1. Define a frame (eg DRIVING)
2. Find some sentences for that frame
3. Annotate them

- Corpora
  - FrameNet I – British National Corpus only
  - FrameNet II – LDC North American Newswire corpora

- Size
  - >8,900 lexical units, >125 frames, >135,000 sentences

Annotations in PropBank

- Based on Penn TreeBank
- Goal is to annotate every tree systematically
- so statistics in the corpus are meaningful
- Based on Levin’s verb classes (via VerbNet)
- But annotated with lowest common denominator ARG0, ARG1 roles.
- Generally more data-driven & bottom up
- No level of abstraction beyond verb senses
- Annotate every verb you see, whether or not it seems to be part of a frame

Some verb senses and “framesets” for propbank

FrameNet vs PropBank -1

FRAMENET ANNOTATION:
[Agent Chuck] bought [Agent a car] [Goal from Jerry] [Preposition for $1000].
[Agent Jerry] sold [Agent a car] [Agent to Chuck] [Preposition for $1000].

PROPBANK ANNOTATION:
[Agent Chuck] bought [Agent a car] [Agent from Jerry] [Preposition for $1000].
[Agent Jerry] sold [Agent a car] [Agent to Chuck] [Preposition for $1000].
FrameNet vs PropBank

**FrameNet**
- FRAMENET ANNOTATION:
  - [Good] A car was bought [Buyer by Chuck].
  - [Good] A car was sold [Seller to Chuck] [Sells by Jerry].

**PropBank**
- PROPANK ANNOTATION:
  - [Arg1] A car was bought [Large by Chuck].
  - [Arg2] A car was sold [Large to Chuck] [Large by Jerry].

Proposition Bank (PropBank) [Palmer et al. 05]

- Transfer sentences to propositions
  - Kristina hit Scott → hit(Kristina, Scott)
- Penn TreeBank → PropBank
  - Add a semantic layer on Penn TreeBank
  - Define a set of semantic roles for each verb
  - Each verb’s roles are numbered
    - ... [A0 the company] to ... offer [A1 a 15% to 20% stake] [A2 to the public]
    - ... [A0 Sotheby’s] ... offered [A2 the Dorrance heirs] [A1 a money-back guarantee]
    - ... [A1 an amendment offered] [A0 by Rep. Peter DeFazio] ...
    - ... [A2 Subcontractors] will be offered [A1 a settlement] ...

Proposition Bank (PropBank)

**Define the Set of Semantic Roles**

- It’s difficult to define a general set of semantic roles for all types of predicates (verbs).
- PropBank defines semantic roles for each verb and sense in the frame files.
- The (core) arguments are labeled by numbers.
  - A0 – Agent; A1 – Patient or Theme
  - Other arguments – no consistent generalizations
- Adjunct-like arguments — universal to all verbs
  - AM-LOC, TMP, EXT, CAU, DIR, PNC, ADV, MNR, NEG, MOD, DIS

**Frame Files**

- hit.01 “strike”
  - A0: agent, hitter; A1: thing hit; A2: instrument, thing hit by or with
- look.02 “seeming”
  - A0: seemer; A1: seemed like; A2: seemed to
    - looked [to her] like [he deserved this].
- deserve.01 “deserve”
  - A0: deserving entity; A1: thing deserved;
    - It looked to her like [he deserved this].

Proposition Bank (PropBank)

**Add a Semantic Layer**

- Proposition:
  - A sentence and a target verb

**Add a Semantic Layer – Continued**

- Proposition:
  - A sentence and a target verb

- Kristina hit Scott with a baseball yesterday
- The worst thing about him, "said Kristina, "is his laziness"
Proposition Bank (PropBank)

Final Notes

- Current release (Mar 4, 2005): Proposition Bank I
  - Verb Lexicon: 3,324 frame files
  - Annotation: ~113,000 propositions
    http://www.cis.upenn.edu/~mptalmer/project_pages/ACE.htm
- Alternative format: CoNLL-04,05 shared task
  - Represented in table format
  - Has been used as standard data set for the shared tasks on semantic role labeling
    http://www.lsi.upc.es/~srlconll/soft.html

Information Extraction versus Semantic Role Labeling

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>IE</th>
<th>SRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>narrow</td>
<td>broad</td>
</tr>
<tr>
<td>Depth of semantics</td>
<td>shallow</td>
<td>shallow</td>
</tr>
<tr>
<td>Directly connected to</td>
<td>sometimes</td>
<td>no</td>
</tr>
<tr>
<td>application</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Overview of SRL Systems

- Definition of the SRL task
- Evaluation measures
- General system architectures
- Machine learning models
  - Features & models
  - Performance gains from different techniques

Subtasks

- Identification: \(2^{[1,2,\ldots,n]} \setminus \{\text{NONE, ARG}\}\)
  - Very hard task: to separate the argument substrings from the rest in this exponentially sized set
  - Usually only 1 to 9 (avg. 2.7) substrings have labels ARG and the rest have NONE for a predicate
- Classification: \(2^{[1,2,\ldots,n]} \setminus \{\text{NONE}\} \rightarrow L\)
  - Given the set of substrings that have an ARG label, decide the exact semantic label
- Core argument semantic role labeling: (easier)
  - Label phrases with core argument labels only. The modifier arguments are assumed to have label NONE.

Evaluation Measures

Correct: [The queen] broke [the window] [yesterday]
Guess: [The queen] broke [the window] [yesterday]

<table>
<thead>
<tr>
<th>Correct</th>
<th>Guess</th>
</tr>
</thead>
<tbody>
<tr>
<td>[The queen]</td>
<td>[The queen]</td>
</tr>
<tr>
<td>[the window]</td>
<td>[window]</td>
</tr>
<tr>
<td>[yesterday]</td>
<td>[yesterday]</td>
</tr>
</tbody>
</table>

- Precision, Recall, F-Measure (tp=1,fp=2,fn=2) prc/f1=1/3
- Measures for subtasks
  - Identification (Precision, Recall, F-measure) (tp=2,fp=1,fn=1) prc/f1=2/3
  - Classification (Accuracy) acc = .5 (labeling of correctly identified phrases)
  - Core arguments (Precision, Recall, F-measure) (tp=1,fp=1,fn=1) prc/f1=1/2
Basic Architecture of a Generic SRL System

Annotations Used

- Syntactic Parsers
  - Collins’, Chamiak’s (most systems)
  - CCG parses
    - (Ghassan & Hockenmaier 03) [Pradhan et al. 03]
  - TAG parses (Collier & Rambow 03)

- Shallow parsers

- Semantic ontologies (WordNet, automatically derived), and named entity classes
  - (vi) hit (cause to move by striking)
  - WordNet hypernym
  - propel, impel (cause to move forward with force)

Annotations Used - Continued

Most commonly, substrings that have argument labels correspond to syntactic constituents

- In Propbank, an argument phrase corresponds to exactly one parse tree constituent in the correct parse tree for 95.7% of the arguments:
  - when more than one constituent correspond to a single argument (4.3%), simple rules can join constituents together (in 80% of these cases. (Tsarfaty 00))
  - In Propbank, an argument phrase corresponds to exactly one parse tree constituent in Chamiak’s automatic parse tree for approx 90.0% of the arguments.
  - Some cases (about 30% of the mismatches) are easily recoverable with simple rules that join constituents (Tsarfaty 00)
  - In FrameNet, an argument phrase corresponds to exactly one parse tree constituent in Collins’ automatic parse tree for 87% of the arguments.

Labeling Parse Tree Nodes

- Given a parse tree t, label the nodes (phrases) in the tree with semantic labels
- To deal with discontinuous arguments:
  - In a post-processing step, join some phrases using simple rules
  - Use a more powerful labeling scheme, i.e., C-A0 for continuation of A0

Another approach: labeling chunked sentences. Will not describe in this section.

Combining Identification and Classification Models

- Step 2. Identification. Identification model:
  - filters out candidates with high probability of NONE.
- Step 3. Classification. Classification model assigns one of the argument labels to selected nodes (or sometimes possibly NONE).

Combining Identification and Classification Models – Continued

\[
P(l|c, t, p) = P(l|Id(l), \Phi(c, t, p)) + P_{Id}(l|Id(l), \Phi(c, t, p)) \sim P(l|c, t, p) = P(l|\Phi(c, t, p))
\]
Joint Scoring Models

- These models have scores for the whole labeling of a tree (not just individual labels)
- Encode some dependencies among the labels of different nodes

$$P_{\text{JOINT}}(l_1, \ldots, l_n|n, t, p) = \prod P(l_i|n, t, p)$$

Combining Local and Joint Scoring Models

- Tight integration of local and joint scoring in a single probabilistic model and exact search [Cohn&Blunsom 05] [Martínez et al. 09][Thompson et al. 03]
- When the joint model makes strong independence assumptions

- Re-ranking or approximate search to find the labeling which maximizes a combination of local and a joint score [Gildea&Jurafsky 02] [Pradhan et al. 04][Toutanova et al. 05]
- Usually exponential search required to find the exact maximizer

- Exact search for best assignment by local model satisfying hard joint constraints
  - Using Integer Linear Programming [Punyakanok et al 04,05] (worst case NP-hard)
  - More details later

Gildea & Jurafsky (2002) Features

- Key early work
- Future systems use these features as a baseline

- Constituent Independent
  - Target predicate (lemma)
  - Voice
  - Subcategorization
- Constituent Specific
  - Path
  - Position (left, right)
  - Phrase Type
  - Governing Category (S or VP)
  - Head Word

Performance with Baseline Features using the G&J Model

- Better ML: 67.6 → 80.8 using SVMs [Pradhan et al. 04].
- Content Word (different from head word)
- Head Word and Content Word POS tags
- NE labels (Organization, Location, etc.)
- Structural/lexical context (phrase/words around parse tree)
- Head of PP Parent

  - If the parent of a constituent is a PP, the identity of the preposition

Performance with Baseline Features using the G&J Model

- Machine learning algorithm: interpolation of relative frequency estimates based on subsets of the 7 features introduced earlier

Performance with Baseline Features using the G&J Model

- 31% error reduction from baseline due to these + Surdeanu et al. features
Joint Scoring: Enforcing Hard Constraints

- Constraint 1: Argument phrases do not overlap.
  By this working [he, hard], he said, you can achieve a lot.

- Pradhan et al. (04) – greedy search for a best set of non-overlapping arguments.

- Toutanova et al. (05) – exact search for the best set of non-overlapping arguments (dynamic programming, linear in the size of the tree).

- Punyakanon et al. (05) – exact search for the best non-overlapping arguments using integer linear programming.

- Other constraints (Punyakanon et al. 04, 05):
  - no repeated core arguments (good heuristic)
  - phrases do not overlap the predicate.

Joint Scoring: Integrating Soft Preferences

- Gilead and Jurafsky (02) – a smoothed relative frequency estimate of the probability of frame element multi-sets:
  \[ P(A0, AM\_TMP, A1, AM\_TMP | \text{hit}) \]
  - Gains relative to local model 62.9 → 62.9 FrameNet automatic parses.

- Pradhan et al. (04) – a language model on argument label sequences (with the predicate included):
  \[ P(A0, AM\_TMP, \text{hit}, A1, AM\_TMP) \]
  - Small gains relative to local model for a baseline system 88.0 → 88.9 on core arguments (PlopropBank correct parses).

- Toutanova et al. (05) – a joint model based on CRFs with a rich set of joint features of the sequences of labeled arguments:
  - Gains relative to local model on PropBank correct parses 88.4 → 91.2 (24% error reduction), gains on automatic parses 79.2 → 80.0.

- Also tree CRFs (Cohn & Brunson) have been used.

Semantic roles: joint models boost results [Toutanova et al. 2005]

Accuracies of local and joint models on core arguments

<table>
<thead>
<tr>
<th></th>
<th>XAP</th>
<th>Local</th>
<th>Integrated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>90.1</td>
<td>56.3</td>
<td>90.5</td>
</tr>
<tr>
<td>Class</td>
<td>98.6</td>
<td>55.7</td>
<td>98.8</td>
</tr>
<tr>
<td>Integrated</td>
<td>94.6</td>
<td>51.0</td>
<td>95.0</td>
</tr>
</tbody>
</table>

Error reduction from best published result: 44.6% on Integrated 52% on Classification

System Properties

- Learning Methods:
  - SNoW, MaxEnt, AdaBoost, SVM, CRFs, etc.
  - The choice of learning algorithms is less important.

- Features:
  - All teams implement more or less the standard features with some variations.
  - A must-do for building a good system!
  - A clear feature study and more feature engineering will be helpful.
System Properties – Continued

- Syntactic Information
  - Charniak’s parser, Collins’ parser, clauser, chunker, etc.
  - Top systems use Charniak’s parser or some mixture

- Quality of syntactic information is very important!

- System/Information Combination
  - 8 teams implement some level of combination
  - Greedy, Re-ranking, Stacking, ILP inference

- Combination of systems or syntactic information is a good strategy to reduce the influence of incorrect syntactic information!

Per Argument Performance
CoNLL-05 Results on WSJ-Test

- Core Arguments (Freq. ~70%)

<table>
<thead>
<tr>
<th></th>
<th>Best F</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG</td>
<td>88.31</td>
<td>25.56%</td>
</tr>
<tr>
<td>A1</td>
<td>79.91</td>
<td>35.36%</td>
</tr>
<tr>
<td>A2</td>
<td>70.28</td>
<td>8.26%</td>
</tr>
<tr>
<td>A3</td>
<td>65.28</td>
<td>1.36%</td>
</tr>
<tr>
<td>A4</td>
<td>77.25</td>
<td>1.09%</td>
</tr>
</tbody>
</table>

- Adjuncts (Freq. ~30%)

<table>
<thead>
<tr>
<th></th>
<th>Best F</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP</td>
<td>78.21</td>
<td>8.66%</td>
</tr>
<tr>
<td>AOV</td>
<td>59.73</td>
<td>3.46%</td>
</tr>
<tr>
<td>DIS</td>
<td>80.45</td>
<td>2.05%</td>
</tr>
<tr>
<td>MNR</td>
<td>59.22</td>
<td>2.67%</td>
</tr>
<tr>
<td>LOC</td>
<td>60.99</td>
<td>2.48%</td>
</tr>
<tr>
<td>MOD</td>
<td>98.47</td>
<td>3.83%</td>
</tr>
<tr>
<td>CAU</td>
<td>64.62</td>
<td>0.52%</td>
</tr>
<tr>
<td>NEG</td>
<td>98.91</td>
<td>1.36%</td>
</tr>
</tbody>
</table>

Arguments that need to be improved

Data from Carreras&Màrquez’s slides (CoNLL 2005)