Introduction

So far we have introduced (discriminative, incremental) parsing as a fundamental task in NLP. But what is parsing actually good for?

Parsing is used to break up sentences into meaningful parts, which can then feed into downstream semantic tasks:

- semantic role labeling (figure out who did what to whom);
- semantic parsing (turn a sentence into a logical form);
- word sense disambiguation (figure out what the words in a sentence mean);
- compositional semantics (compute the meaning of a sentence based on the meaning of its parts).

Reading: Gildea and Jurafsky (2002).
So far we have introduced (discriminative, incremental) parsing as a fundamental task in NLP. But what is parsing actually good for?

Parsing is used to break up sentences into meaningful parts, which can then feed into downstream semantic tasks:

- semantic role labeling (figure out who did what do whom);
- semantic parsing (turn a sentence into a logical form);
- word sense disambiguation (figure out what the words in a sentence mean);
- compositional semantics (compute the meaning of a sentence based on the meaning of its parts).

Let’s start with semantic role labeling.

Frame Semantics

Due to Fillmore (1976):

- a frame describes a prototypical situation;
- it is evoked by a frame evoking element (FEE);
- it can have several frame elements (semantic roles).

Matilde fried the catfish in a heavy iron skillet.

- Apply_heat

FEE

Matilde fried the catfish in a heavy iron skillet.
Frame Semantics

- Due to Fillmore (1976);
- a frame describes a prototypical situation;
- it is evoked by a frame evoking element (FEE);
- it can have several frame elements (semantic roles).

### Apply_heat

Roles

Matilde fried the catfish in a heavy iron skillet.

FrameNet Corpus

FrameNet is a corpus with frame semantics markup:
- uses a tagset of 76 semantic roles (frame elements) from 12 general semantic domains (body, cognition, communication);
- consists of a sample of sentences from the BNC annotated with frame elements;
- 49,013 sentences and 99,232 frame elements in total;
- this includes 927 verbs, 339 nouns, 175 adjectives.

The sentences in the corpus were not chosen from the BNC at random; rather representative usages were selected.

Properties of Frame Semantics

- Provides a shallow semantic analysis (no modality, scope);
- granularity in between “universal” and “verb specific” roles;
- generalizes well across languages;
- can benefit various NLP applications (IR, QA).

### How much did Google pay for YouTube?

Google snapped up YouTube for $1.65 billion.

Matching

Gildea and Jurafsky (2002) break down the role labeling task:
- parse the training corpus using the Collins parser;
- match frame elements to constituents;
- extract features from the parse tree;
- train probabilistic model on the features.

The start and end word of each parsed constituent is found and matched against a frame element with the same start and end. No match is possible in 13% of the cases (parsing errors).
Features

Assume the sentences are parsed, then the following features can be extracted for role labeling:

- **Phrase Type**: syntactic type of the phrase expressing the semantic role (e.g., NP, VP, S);
- **Governing Category**: syntactic type of the phrase governing the semantic role (NP, VP), only used for NPs;
- **Parse Tree Path**: path through the parse tree from the target word to the phrase expressing the role;
- **Position**: whether the constituent occurs before or after the predicate; useful for incorrect parses;
- **Voice**: active or passive; use heuristics to identify passives;
- **Head Word**: the lexical head of the constituent.

Probabilistic Model

Divide the FrameNet corpus into:

- 10% test set;
- 10% development set;
- 80% training set;

Relative small training set: average number of sentences per target word is 34, number of sentences per frame is 732.
Probabilistic Model

Build a classifier by combining conditional distributions of the features. Compute the distribution from the training data, e.g.:

\[ P(r|pt, t) = \frac{\#(r, pt, t)}{\#(pt, t)} \]  

\[ r \text{ semantic role} \]
\[ pt \text{ phrase type} \]
\[ gov \text{ governing category} \]
\[ pos \text{ position} \]
\[ voice \text{ voice} \]
\[ h \text{ head word} \]
\[ t \text{ target word (predicate)} \]

Evaluation

Measure the performance of a distribution using the following metrics:

- **Coverage**: percentage of the test data for which the conditioning event has been seen in the training data.
- **Accuracy**: percentage of covered test data for which the correct role is predicted.
- **Performance**: product of coverage and accuracy.

These metrics are similar to familiar precision and recall. But: multi-way classification, not binary decision is being made.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Coverage</th>
<th>Accuracy</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P(r</td>
<td>t))</td>
<td>100</td>
<td>40.9</td>
</tr>
<tr>
<td>(P(r</td>
<td>pt, t))</td>
<td>92.5</td>
<td>60.1</td>
</tr>
<tr>
<td>(P(r</td>
<td>pt, gov, t))</td>
<td>92.0</td>
<td>66.6</td>
</tr>
<tr>
<td>(P(r</td>
<td>pt, pos, voice))</td>
<td>98.8</td>
<td>57.1</td>
</tr>
<tr>
<td>(P(r</td>
<td>pt, pos, voice, t))</td>
<td>90.8</td>
<td>70.1</td>
</tr>
<tr>
<td>(P(r</td>
<td>h))</td>
<td>80.3</td>
<td>73.6</td>
</tr>
<tr>
<td>(P(r</td>
<td>h, t))</td>
<td>56.0</td>
<td>86.6</td>
</tr>
<tr>
<td>(P(r</td>
<td>h, pt, t))</td>
<td>50.1</td>
<td>87.4</td>
</tr>
</tbody>
</table>
**Interpolation and Backoff**

**Backoff:** if there is no data for a given distribution, back off to a more general one.

\[
P(r \mid h, pt, t) = \lambda_1 P(r \mid h) + \lambda_2 P(r \mid pt, t) + \lambda_3 P(r \mid h, t)
\]

The features used are **Parse Tree Path**, **Head Word**, and **Target Word**, as previously introduced.

**Evaluation**

<table>
<thead>
<tr>
<th>Combining Method</th>
<th>% Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal linear interpolation</td>
<td>79.5</td>
</tr>
<tr>
<td>EM linear interpolation</td>
<td>79.3</td>
</tr>
<tr>
<td>Dev set: Geometric mean</td>
<td>79.6</td>
</tr>
<tr>
<td>Backoff, linear interpolation</td>
<td>80.4</td>
</tr>
<tr>
<td>Backoff, geometric mean</td>
<td>79.6</td>
</tr>
<tr>
<td>Baseline: most common role</td>
<td>40.9</td>
</tr>
</tbody>
</table>

Note: varying the weights \(\lambda_i\) changed performance only marginally.

<table>
<thead>
<tr>
<th>Combining Method</th>
<th>% Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM linear interpolation</td>
<td>78.5</td>
</tr>
<tr>
<td>Backoff, linear interpolation</td>
<td>76.9</td>
</tr>
<tr>
<td>Baseline: most common role</td>
<td>40.6</td>
</tr>
</tbody>
</table>

Excludes the 13% FEs that don’t match constituents in the parse.
Generalizing Lexical Statistics

Head words are good predictors of semantic role, but data is sparse. This can be overcome using:

- **clustering**: find words that are similar to head words that fail to occur in the training data; increases performance to 85%;
- **WordNet**: if a word is not in the training data, use its hypernym in WordNet; percolate co-occurrence counts up the WordNet hierarchy (problem: multiple hierarchies and multiple word senses); increases accuracy to 84.3%;
- **bootstrapping**: label unannotated data with the automatic system, use the resulting data as training data; increases accuracy to 83.2%.

Coverage is always 100%, which explains boost in performance.

Summary

- Semantic role labeling means identifying the constituents (frame elements) that participate in a prototypical situation (frame) and labeling them with their roles;
- this provides a shallow semantic analysis that can benefit various NLP applications;
- FrameNet is a corpus marked up with semantic roles;
- a simple probabilistic model combining lexical and syntactic features (from parse tree) performs well on the task;
- the model interpolates distributions or performs backoff;
- similar features can be used for identifying frame elements;
- in both models, lexical statistics are sparse, this can be addressed with clustering, WordNet, or bootstrapping.

References
