Parsing with Lexical-Functional Grammars

Nikos Engonopoulos

Universität des Saarlandes

January 19, 2012
Outline

1. Introduction

2. LFG parsing using Discriminative Estimation
   - Robust LFG parsing
   - Discriminative Estimation

3. PCFG-based LFG approximations
   - Automatic f-structure annotation
   - Inducing LFG approximations
   - Long-distance dependencies

4. Experimental Evaluation

5. Discussion
Outline

1 Introduction

2 LFG parsing using Discriminative Estimation
   - Robust LFG parsing
   - Discriminative Estimation

3 PCFG-based LFG approximations
   - Automatic f-structure annotation
   - Inducing LFG approximations
   - Long-distance dependencies

4 Experimental Evaluation

5 Discussion
LFG basics

Lexical Functional Grammar:

- **lexical** (not transformational): richly structured lexicon which encodes grammatical structure (e.g. verbal frames)
- **functional** (not configurational): grammatical functions (e.g. subject) object are explicit primitives

LFG distinguishes (minimally) two representations for a sentence:

- **c(onstituent)-structure** Phonological realization
  - Represented as a tree structure
  - Similar to CFG structures

- **f(unctional)-structure** Abstract functional organization:
  - predicate-argument structures, functional relations
  - Represented as an attribute-value matrix (AVM)
  - Similar to dependency structures
LFG structure example

LFG structure example

```
S  →  NP  VP
    ↑SUBJ=↓  ↑=↓

VP  →  V  NP
    ↑=↓  ↑OBJ=↓

NP  →  U.N  V  →  signs
    ↑PRED=U.N.  ↑PRED=sign
```

S

<table>
<thead>
<tr>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.N.</td>
</tr>
<tr>
<td>V</td>
</tr>
<tr>
<td>NP</td>
</tr>
</tbody>
</table>

| VP |
|    |
| signs |
| treaty |

| SUBJ |
| PRED |
| U.N. |

| OBJ |
| PRED |
| treaty |
Deep parsing with LFG (or HPSG) presents a number of challenges:

- Annotation with LFG structures is very difficult and time-consuming
- Very small LFG-annotated training corpora
- Difficult to automatically induce stochastic LFG grammars directly using the same data-driven approaches as for PCFGs
- Hand-crafted grammars have usually low coverage, and are extremely time-consuming and expensive

Two different approaches will be presented:

**The PARC approach**  Carefully building a hand-crafted LFG, then using stochastic disambiguation guided by treebank PSG annotations (*linguistically motivated*)

**The DCU approach**  Automatically extracting an approximation of an LFG using treebank PCFG and functional annotations, plus advanced heuristics for building missing information (*data-driven*)
1 Introduction

2 LFG parsing using Discriminative Estimation
   - Robust LFG parsing
   - Discriminative Estimation

3 PCFG-based LFG approximations
   - Automatic f-structure annotation
   - Inducing LFG approximations
   - Long-distance dependencies

4 Experimental Evaluation

5 Discussion
Riezler et al. (2002) address the LFG parsing problem by using:

- A hand-crafted LFG grammar (ParGram, 1999)
- A widely used PSG-annotated corpus (WSJ), also containing some partial functional annotations (e.g. NP-SBJ, ADJP-PRD)
- A deterministic LFG parser which produces packed representations of all possible parses for each input
- A discriminative statistical estimation method to stochastically disambiguate between the possible parses
An LFG grammar with 314 rules is used, compiled into a collection of FSMs with \( \sim 8,700 \) states and \( \sim 20,000 \) transitions.

Grammar modified to parse POS tags and labeled bracketings, instead of raw text.

The XLE parser (Maxwell and Kaplan, 1993) is used to tag partially labeled data from WSJ and creating packed representations of all possible parses for each input.

- **Robustness:** When no parse is found, a partial one is assigned (FRAGMENT parse).
- **Efficiency:** If a parsing time threshold is exceeded, a SKIMMED parse is produced (full or partial).
The conditional probability of a parse $x$ given a sentence $y$ is $p_\lambda(x|y) = Z_\lambda(y)^{-1} e^{\lambda \cdot f(x)}$, where:

- $f = (f_1, ..., f_n)$ is a vector of feature functions which map parse to numeric values
- $\lambda = (\lambda_1, ..., \lambda_n)$ is a vector of log-parameters for these property functions
- $Z_\lambda(y) = \sum_{x \in X(y)} e^{\lambda \cdot f(x)}$ is a normalization constant

Feature functions can represent various complex properties of the parses.
Around 1000 feature functions are used, for example:

- c-structure nodes and subtrees
- number of recursively embedded phrases (to account for high-vs-low attachment)
- values of specific f-structure attributes
- presence of particular attribute-value pairs
- branching behaviour of c-structures
- properties based on soft clustering of grammatical relations encoded as lexicalized head-argument-relation tuples
- ...
Discriminative Estimation method

Goal: maximize the conditional likelihood of the correct parse given the training example

\[ P(\lambda) = -L(\lambda) - G(\lambda) = -\log \prod_{j=1}^{m} p_\lambda(z_j|y_j) + \sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma_i^2} \quad (1) \]

- In this case: no gold-standard LFG parse
- Instead, consider for each training sentence \( y \) the set of parses \( X(y,z) \) consistent with the partial treebank annotation \( z \):

\[ P(\lambda) = -\sum_{j=1}^{m} \log \frac{\sum_{X(y_j,z_j)} e^{\lambda \cdot f(x)}}{\sum_{X(y_j)} e^{\lambda \cdot f(x)}} + \sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma_i^2} \quad (2) \]

\[ = -\sum_{j=1}^{m} \log \sum_{X(y_j,z_j)} e^{\lambda \cdot f(x)} + \sum_{j=1}^{m} \log \sum_{X(y_j)} e^{\lambda \cdot f(x)} + \sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma_i^2} \quad (3) \]
Outline

1. Introduction

2. LFG parsing using Discriminative Estimation
   - Robust LFG parsing
   - Discriminative Estimation

3. PCFG-based LFG approximations
   - Automatic f-structure annotation
   - Inducing LFG approximations
   - Long-distance dependencies

4. Experimental Evaluation

5. Discussion
Data-driven, PCFG-based LFG approximations

- **Problem:** Building a broad-coverage, large-scale LFG (or HPSG) grammar is time-consuming and expensive.
- However, large PCFG annotated corpora exist (e.g. Penn-II treebank), with some functional, trace, head annotations.

**The DCU approach:**
- Automatically generate PCFG-based approximations of LFG grammars, using an shallow f-structure annotation algorithm.
- Automatically extract semantic subcategorization frames and finite approximations of Functional Uncertainty equations from the training data.
- Resolve long-distance dependencies using the induced f-structures, frames and FU equations.
Automatic f-structure annotation algorithm

An algorithm exploiting Penn-II annotations to produce f-structures.

1. The Penn-II trees are automatically head-lexicalized
2. All local subtrees are partitioned into left-right context (relative to head)
3. A pipeline of 4 modules is applied for annotating with f-equations:
   - L/R Context $\Rightarrow$ Coordination $\Rightarrow$ Traces $\Rightarrow$ Catch-All
4. A constraint solver produces full f-structures

The algorithm has 99.82% coverage and 96.57% F-1 on Penn-II treebank data.
Automatic extraction of LFG approximations using PCFG annotations

Two alternative architectures used:

1. The **pipeline** model
   - extract a PCFG from a treebank
   - parse unseen data
   - generate f-structures using the f-structure annotation algorithm

2. The **integrated** model
   - annotate treebank with f-structure equations
   - extract a PCFG from the augmented treebank annotations
   - parse unseen data using the extended PCFG
   - resolve f-structure equations into f-structures using a constraint solver
Long-distance dependencies

**Problem:** Resulting f-structures are still “shallow”: they don’t account for long-distance dependencies

![Diagram of a parse tree]

**Solution:** Resolve LDDs by using *Functional uncertainty (FU)* equations and subcategorization frames
LDD resolution with extracted frames and FU equations

1. Subcategorization frames:

\[
V \rightarrow \text{said} \quad \uparrow\text{PRED}=\text{say}\left(\uparrow\text{SUBJ}, \uparrow\text{COMP}\right)
\]

\[
V \rightarrow \text{signs} \quad \uparrow\text{PRED}=\text{sign}\left(\uparrow\text{SUBJ}, \uparrow\text{OBJ}\right)
\]

2. Functional uncertainty (FU) equations:

\[
S \rightarrow S \quad \uparrow\text{TOPIC}=\downarrow \quad \text{NP} \quad \text{VP} \\
\quad \uparrow\text{TOPIC}=\downarrow\text{COMP}\star\text{COMP}
\]

3. Instead of being hand-coded, they are learned from the training data and assigned probabilities.

4. The LDD resolution algorithm then recursively traverses the f-structure looking for paths that match FU equations, while satisfying subcategorization constraints.
Outline

1. Introduction

2. LFG parsing using Discriminative Estimation
   - Robust LFG parsing
   - Discriminative Estimation

3. PCFG-based LFG approximations
   - Automatic f-structure annotation
   - Inducing LFG approximations
   - Long-distance dependencies

4. Experimental Evaluation

5. Discussion
Experimental Evaluation: PARC approach

Two evaluation methods for produced f-structures:

1. LFG f-relations
2. Dependency relations (to enable comparison with other formalisms)

Results on the PARC 700 dataset (700 randomly selected sentences from section 23 of WSJ):

<table>
<thead>
<tr>
<th></th>
<th>LFG</th>
<th>DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>upper bound</td>
<td>84.1</td>
<td>80.7</td>
</tr>
<tr>
<td>stochastic</td>
<td>78.6</td>
<td>73.0</td>
</tr>
<tr>
<td>lower bound</td>
<td>75.5</td>
<td>68.8</td>
</tr>
</tbody>
</table>

Later results with a better disambiguation technique (Kaplan et al., 2004):

<table>
<thead>
<tr>
<th></th>
<th>time</th>
<th>prec.</th>
<th>rec.</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFG core</td>
<td>298.88</td>
<td>79.1</td>
<td>76.2</td>
<td>77.6</td>
</tr>
<tr>
<td>LFG complete</td>
<td>985.3</td>
<td>79.4</td>
<td>79.8</td>
<td>79.6</td>
</tr>
<tr>
<td>Collins 1000</td>
<td>199.6</td>
<td>78.3</td>
<td>71.2</td>
<td>74.6</td>
</tr>
</tbody>
</table>
Experimental Evaluation: DCU approach

On the same dataset (PARC 700), the DCU approach achieves better F-1 than the PARC approach.

<table>
<thead>
<tr>
<th></th>
<th>Pipeline PCFG</th>
<th>P-PCFG</th>
<th>Integrated A-PCFG</th>
<th>PA-PCFG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># Parses</td>
<td>2416</td>
<td>2416</td>
<td>2416</td>
<td>2414</td>
</tr>
<tr>
<td>Lab. F-Score</td>
<td>75.83</td>
<td>80.80</td>
<td>79.17</td>
<td>81.32</td>
</tr>
<tr>
<td>Unlab. F-Score</td>
<td>78.28</td>
<td>82.70</td>
<td>81.49</td>
<td>83.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCU 105 F-Strs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All GFs F-Score (before LDD resolution)</td>
<td>79.82</td>
<td>79.24</td>
<td>81.12</td>
<td>81.20</td>
</tr>
<tr>
<td>All GFs F-Score (after LDD resolution)</td>
<td>83.79</td>
<td>84.59</td>
<td>86.30</td>
<td>87.04</td>
</tr>
<tr>
<td>Preds only F-Score (before LDD resolution)</td>
<td>70.00</td>
<td>71.57</td>
<td>73.45</td>
<td>74.61</td>
</tr>
<tr>
<td>Preds only F-Score (after LDD resolution)</td>
<td>73.78</td>
<td>77.43</td>
<td>78.76</td>
<td><strong>80.97</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARC 700 Dependency Bank</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All GFs F-Score (before LDD resolution)</td>
<td>81.98</td>
<td>81.49</td>
<td>83.32</td>
<td>82.78</td>
</tr>
<tr>
<td>All GFs F-Score (after LDD resolution)</td>
<td>84.16</td>
<td>84.37</td>
<td>86.45</td>
<td>86.00</td>
</tr>
<tr>
<td>Preds only F-Score (before LDD resolution)</td>
<td>72.00</td>
<td>73.23</td>
<td>75.22</td>
<td>75.10</td>
</tr>
<tr>
<td>Preds only F-Score (after LDD resolution)</td>
<td>74.07</td>
<td>76.12</td>
<td>78.36</td>
<td><strong>78.40</strong></td>
</tr>
<tr>
<td>Subset of GFs following (Kaplan et al., 2004)</td>
<td>77.86</td>
<td><strong>80.24</strong></td>
<td>77.68</td>
<td>78.60</td>
</tr>
</tbody>
</table>
Outline

1. Introduction

2. LFG parsing using Discriminative Estimation
   - Robust LFG parsing
   - Discriminative Estimation

3. PCFG-based LFG approximations
   - Automatic f-structure annotation
   - Inducing LFG approximations
   - Long-distance dependencies

4. Experimental Evaluation

5. Discussion
Discussion

- LFG parsing is challenging due to costly grammar engineering and annotating

- Two approaches were presented:
  1. Linguistically motivated (PARC): hand-crafted LFG using stochastic disambiguation guided by existing treebank annotations
     - more robust and consistent
     - no annotation effort
     - huge engineering effort, over-reliance on a single person’s expertise
  2. Data-driven (DCU): automatically induced LFG approximation from existing treebank annotations
     - no grammar engineering or annotation effort
     - more fragmented grammar, no linguistic motivation

- Both approaches achieve similar evaluation results
References

