A Generative Model for Parsing Natural Language to Meaning Representations

Jake Vasilakes

March 9, 2015
# Outline

**Background**
- Key Concepts
- Purpose and Structure

**Generative Model**
- Process
- Tree probability
- Parameters
- Decoding

**Discriminative reranking**
- Averaged Perceptron

**Evaluation**
- Methodology
- Results
A Generative Model for Parsing Natural Language to Meaning Representations

Outline

Background

Key Concepts

Purpose and Structure

Generative Model

Process

Tree probability

Parameters

Decoding

Discriminative reranking

Averaged Perceptron

Evaluation

Methodology

Results
Key Concepts

Key Concepts

  - Semantic Category
  - Function Symbol
  - Arguments
Key Concepts

  - Semantic Category
  - Function Symbol
  - Arguments

NUM : count(STATE)

- Semantic Category
- Function Symbol
- Arguments

\[
\text{NUM} : \text{count(STATE)}
\]

Semantic Parsing: Mapping of natural language (NL) sentences to meaning representations.
Outline

Background
- Key Concepts

Purpose and Structure

Generative Model
- Process
- Tree probability
- Parameters
- Decoding

Discriminative reranking
- Averaged Perceptron

Evaluation
- Methodology
- Results

A Generative Model for Parsing Natural Language to Meaning Representations
Purpose

- Learn a generative model to map NL sentences to MR trees.
- Learn an implicit grammar.
Purpose

- Learn a generative model to map NL sentences to MR trees.
- Learn an implicit grammar.

System Structure
Outline

Background
   Key Concepts
   Purpose and Structure

Generative Model
   Process
      Tree probability
      Parameters
      Decoding

Discriminative reranking
   Averaged Perceptron

Evaluation
   Methodology
   Results
Goal

Simultaneous generation of NL sentence and MR structure.
Goal

Simultaneous generation of NL sentence and MR structure.

How many states do not have rivers?
A Generative Model for Parsing Natural Language to Meaning Representations

Outline

Background
- Key Concepts
- Purpose and Structure

Generative Model
- Process
- Tree probability
- Parameters
- Decoding

Discriminative reranking
- Averaged Perceptron

Evaluation
- Methodology
- Results
Tree probability

\[
P(\hat{w}, \hat{m}, T) = P(M_a) \times P(m_a|M_a) \times P(w_1M_bw_2M_c|m_a) \\
\times P(m_b|m_a, \text{arg} = 1) \times P(\ldots|m_b) \\
\times P(m_c|m_a, \text{arg} = 2) \times P(\ldots|m_c)
\]

\(\hat{w}\): words \quad \hat{m}\): MR structures \quad T: hybrid tree
A Generative Model for Parsing Natural Language to Meaning Representations
Outline

Background
  Key Concepts
  Purpose and Structure

Generative Model
  Process
  Tree probability

Parameters
  Decoding

Discriminative reranking
  Averaged Perceptron

Evaluation
  Methodology
  Results
- **MR model parameters:** \( \sum_{m'} \rho(m'|m_j, \text{arg} = k) = 1 \) for all \( j \) and \( k = 1,2 \)

- **Pattern parameters:** \( \sum_r \phi(r|m_j) = 1 \) for all \( j \)
  
  \( r \): hybrid pattern, e.g. \( wYwZ \)

- **Emission parameters:** \( \sum_t \theta(t|m_j, \Lambda) = 1 \) for all \( j \)
  
  \( t \): any node in \( \mathcal{T} \)
  
  \( \Lambda \): preceding context
Different contexts (Λ) result in different models.

- **Model I**: $\theta(t_k | m_j, \Lambda) = P(t_k | m_j)$ (Unigram)

- **Model II**: $\theta(t_k | m_j, \Lambda) = P(t_k | m_j, t_{k-1})$ (Bigram)

- **Model III**: $\theta(t_k | m_j, \Lambda) = \frac{1}{2} \times (\text{Model I} + \text{Model II})$ (Interpolation)
Estimation

- **MR model parameters**: count and normalize.

- **Pattern and Emission parameters**: EM algorithm
  Unknown alignment between NL words and MR structures in training data.
EM: inside and outside probabilities

- Inside and outside probabilities used to calculate estimated counts.

- $O(n^6m)$ time for 1 EM iteration, where $n$ is length of NL sentence and $m$ the size of the MR structure.

- Modification implemented to bring complexity down to $O(n^3m)$. 
Modification

- **Idea**: aggregate probabilities of NL-MR subsequences to use in subsequent computations.

- Aggregate probabilities for a given NL-MR subsequence $\langle m_v, w_v \rangle$ and a given pattern $r$, e.g. $w \gamma w \zeta$.

- This aggregate probability can be used to calculate the partial inside or outside probability for a given $\langle m_v, w_v \rangle$.

- By summing over all $r$, we get the total inside or outside probability.
Outline

Background
  Key Concepts
  Purpose and Structure

Generative Model
  Process
  Tree probability
  Parameters

Decoding

Discriminative reranking
  Averaged Perceptron

Evaluation
  Methodology
  Results
Goal: Most probable MR structure $\hat{m}^*$ given NL sentence $\hat{w}$.

$$\hat{m}^* = \arg\max_{\hat{m}} \sum_T P(\hat{m}, T|\hat{w})$$

But summing over all possible trees $T$ is expensive. Approximate with the most likely tree (Viterbi approximation).

$$\hat{m}^* = \arg\max_{\hat{m}} \max_T P(\hat{m}, T|\hat{w}) = \arg\max_{\hat{m}} \max_T P(\hat{w}, \hat{m}, T)$$

In practice, ranked list of $k$ best trees is output.
Outline

Background
- Key Concepts
- Purpose and Structure

Generative Model
- Process
- Tree probability
- Parameters
- Decoding

Discriminative reranking
- Averaged Perceptron

Evaluation
- Methodology
- Results
Generative model cannot model long range dependencies within trees.

Use discriminative classifier to rerank the list of $k$ best trees generated by the generative model ($k = 50$).

Averaged Perceptron
Generative model cannot model long range dependencies within trees.

Use discriminative classifier to rerank the list of $k$ best trees generated by the generative model ($k = 50$).

Averaged perceptron with separating plane.
A Generative Model for Parsing Natural Language to Meaning Representations

Averaged Perceptron

- Feature function maps a given tree $\mathcal{T}$ to a feature vector $\Phi(\mathcal{T})$.
- Weight vector $w$ associated with $\Phi(\mathcal{T})$.
- $\mathcal{T}$ with highest score based on weights is picked as output.
A Generative Model for Parsing Natural Language to Meaning Representations

Feature function maps a given tree $\mathcal{T}$ to a feature vector $\Phi(\mathcal{T})$.

Weight vector $\mathbf{w}$ associated with $\Phi(\mathcal{T})$.

$\mathcal{T}$ with highest score based on weights is picked as output.

Separating Plane

After $\mathbf{w}$ is learned, set a threshold score value $b$.

Reject a given $\mathcal{T}$ if it’s score is less than $b$.

Choose $b$ that results in maximum F-score.
### Features

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Hybrid Rule</td>
<td>A MR production and its child hybrid form</td>
</tr>
<tr>
<td>2. Expanded Hybrid Rule</td>
<td>A MR production and its child hybrid form expanded</td>
</tr>
<tr>
<td>3. Long-range Unigram</td>
<td>A MR production and a NL word appearing below in tree</td>
</tr>
<tr>
<td>4. Grandchild Unigram</td>
<td>A MR production and its grandchild NL word</td>
</tr>
<tr>
<td>5. Two Level Unigram</td>
<td>A MR production, its parent production, and its child NL word</td>
</tr>
<tr>
<td>6. Model Log-Probability</td>
<td>Logarithm of base model’s joint probability</td>
</tr>
</tbody>
</table>

Features 1-5 are binary \( \{0,1\} \). Feature 6 is real valued.
Outline

Background
  Key Concepts
  Purpose and Structure

Generative Model
  Process
  Tree probability
  Parameters
  Decoding

Discriminative reranking
  Averaged Perceptron

Evaluation
  Methodology
  Results

A Generative Model for Parsing Natural Language to Meaning Representations
Evaluated on two corpora: GEOQUERY and ROBOCUP.

- Precision, recall, and F-score reported.
- GEOQUERY: MR structure considered correct if it retrieves the same answer as the reference MR structure when used as a query to the database, regardless of differences in the string representation.
- ROBOCUP: MR structure considered correct if it has the same string representation as the reference MR structure.
Outline

Background
  Key Concepts
  Purpose and Structure

Generative Model
  Process
  Tree probability
  Parameters
  Decoding

Discriminative reranking
  Averaged Perceptron

Evaluation
  Methodology

Results
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>GEOQUERY (880)</th>
<th>ROBOCUP (300)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>I</td>
<td>81.3</td>
<td>77.1</td>
</tr>
<tr>
<td>II</td>
<td><strong>89.0</strong></td>
<td>76.0</td>
</tr>
<tr>
<td>III</td>
<td>86.2</td>
<td><strong>81.8</strong></td>
</tr>
<tr>
<td>I+R</td>
<td>87.5</td>
<td>80.5</td>
</tr>
<tr>
<td>II+R</td>
<td><strong>93.2</strong></td>
<td>73.6</td>
</tr>
<tr>
<td>III+R</td>
<td>89.3</td>
<td><strong>81.5</strong></td>
</tr>
</tbody>
</table>
Comparison to previous work

<table>
<thead>
<tr>
<th>System</th>
<th>Geoquery (880)</th>
<th></th>
<th>Robocup (300)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
<td>F</td>
<td>Prec.</td>
</tr>
<tr>
<td>SILT</td>
<td>89.0</td>
<td>54.1</td>
<td>67.3</td>
<td>83.9</td>
</tr>
<tr>
<td>WASP</td>
<td>87.2</td>
<td>74.8</td>
<td>80.5</td>
<td>88.9</td>
</tr>
<tr>
<td>KRISP</td>
<td>93.3</td>
<td>71.7</td>
<td>81.1</td>
<td>85.2</td>
</tr>
<tr>
<td>Model III+R</td>
<td>89.3</td>
<td>81.5</td>
<td>85.2</td>
<td>82.5</td>
</tr>
</tbody>
</table>
Comparison to previous work

<table>
<thead>
<tr>
<th>System</th>
<th>GEOQUERY (880)</th>
<th>ROBOCUP (300)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>SILT</td>
<td>89.0</td>
<td>54.1</td>
</tr>
<tr>
<td>WASP</td>
<td>87.2</td>
<td>74.8</td>
</tr>
<tr>
<td>KRISP</td>
<td>93.3</td>
<td>71.7</td>
</tr>
<tr>
<td>Model III+R</td>
<td>89.3</td>
<td><strong>81.5</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>WASP</td>
<td>95.42</td>
<td>70.00</td>
</tr>
<tr>
<td>Model III+R</td>
<td>91.46</td>
<td>72.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Japanese</th>
<th>Turkish</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>WASP</td>
<td>91.98</td>
<td>74.40</td>
</tr>
<tr>
<td>Model III+R</td>
<td>87.56</td>
<td>76.00</td>
</tr>
</tbody>
</table>

(Evaluated on a subset of GEOQUERY.)
Summary

- Learn a generative model which outputs a list of $k$ best NL-MR hybrid trees from a given NL sentence.

- Rerank the $k$ best list according to score assigned by the averaged perceptron with separating plane.

- Choose tree with highest score as output.
W. Lu, H. T. Ng, W. S. Lee, L. S. Zettlemoyer.
“A Generative Model for Parsing Natural Language to Meaning Representations”.