A Generative Model for Parsing Natural Language to Meaning Representations

Jake Vasilakes

March 9, 2015

Outline

Outline

- Meaning Representation (MR): Formal representation of meaning. Written using a meaning representation language (MRL).

- Meaning Representation (MR): Formal representation of meaning. Written using a meaning representation language (MRL).
  - Semantic Category
  - Function Symbol
  - Arguments

- Meaning Representation (MR): Formal representation of meaning. Written using a meaning representation language (MRL).
  - Semantic Category
  - Function Symbol
  - Arguments

NUM : count(STATE)

- Meaning Representation (MR): Formal representation of meaning. Written using a meaning representation language (MRL).
  - Semantic Category
  - Function Symbol
  - Arguments

NUM : count(STATE)

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

Outline

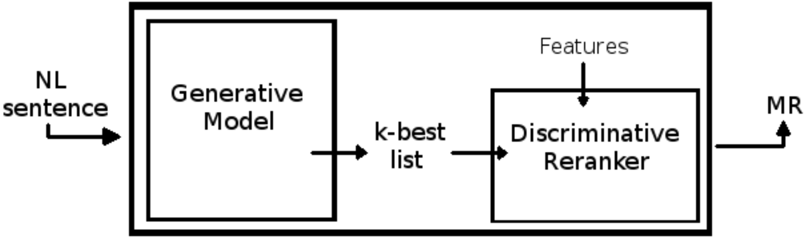
Purpose

- 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



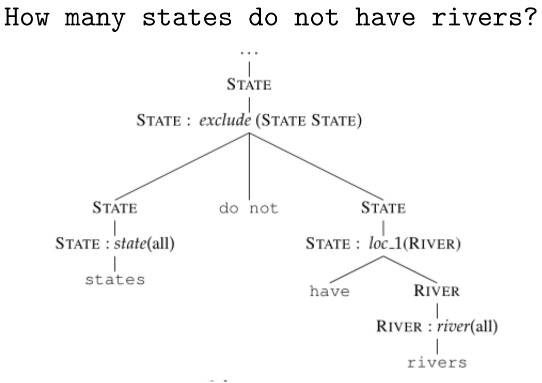
Outline

Goal

Simultaneous generation of NL sentence and MR structure.

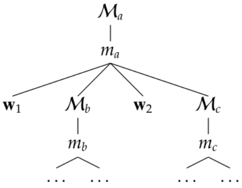
Goal

Simultaneous generation of NL sentence and MR structure.



Outline

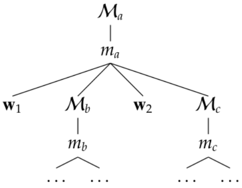
Tree probability



$$P(\widehat{\mathbf{w}}, \widehat{\mathbf{m}}, \mathcal{T}) = P(\mathcal{M}_a) \times P(m_a|\mathcal{M}_a) \times P(\overline{w_1 \mathcal{M}_b w_2 \mathcal{M}_c}|m_a)$$
$$\times P(m_b|m_a, \texttt{arg} = 1) \times P(\ldots|m_b)$$
$$\times P(m_c|m_a, \texttt{arg} = 2) \times P(\ldots|m_c)$$

$\widehat{\mathbf{w}}$: words $\quad$ $\widehat{\mathbf{m}}$: MR structures $\quad$ $\mathcal{T}$: hybrid tree

Tree probability



$$
\begin{aligned}
P(\widehat{\mathbf{w}}, \widehat{\mathbf{m}}, \mathcal{T}) = P(\mathcal{M}_a) \times P(m_a|\mathcal{M}_a) \times P(\overline{w_1 \mathcal{M}_b w_2 \mathcal{M}_c}|m_a) \\
\times P(m_b|m_a, \texttt{arg} = 1) \times P(\ldots|m_b) \\
\times P(m_c|m_a, \texttt{arg} = 2) \times P(\ldots|m_c)
\end{aligned}
$$

$\widehat{\mathbf{w}}$: words $\widehat{\mathbf{m}}$: MR structures $\mathcal{T}$: hybrid tree

$$
\begin{aligned}
P(\overline{w_1 \mathcal{M}_b w_2 \mathcal{M}_c}|m_a) = & P(m \rightarrow w \mathcal{Y} w \mathcal{Z}|m_a) \times P(w_1|m_a) \\
& \times P(\mathcal{M}_b|m_a, w_1) \times P(w_2|m_a, w_1, \mathcal{M}_b) \\
& \times P(\mathcal{M}_c|m_a, w_1, \mathcal{M}_b, w_2) \\
& \times P(\texttt{END}|m_a, w_1, \mathcal{M}_b, w_2, \mathcal{M}_c)
\end{aligned}
$$

Outline

- **MR model parameters**: $\sum_{m'} \rho(m'|m_j, arg = k) = 1$ for all j and k = 1,2
- **Pattern parameters**: $\sum_r \phi(r|m_j) = 1$ for all j
  r: hybrid pattern, e.g. $w\mathcal{Y}w\mathcal{Z}$
- **Emission parameters**: $\sum_t \theta(t|m_j, \Lambda) = 1$ for all j
  t: any node in $\mathcal{T}$ Λ: preceding context

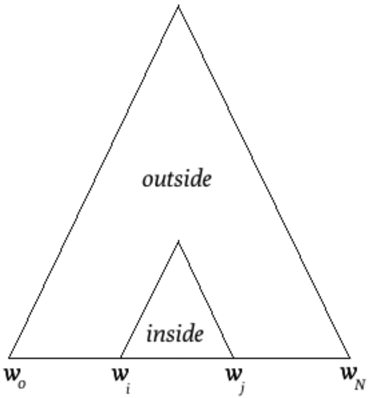
Different contexts (Λ) result in different models.

- **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**: $\theta(t_k|m_j, \Lambda) = \frac{1}{2} \times$ (Model I + 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.
- $O(n^6 m)$ 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 $O(n^3 m)$.

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. $w\mathcal{Y}w\mathcal{Z}$.
- 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

Goal: Most probable MR structure $\widehat{\mathbf{m}}^*$ given NL sentence $\widehat{\mathbf{w}}$.

$$\widehat{\mathbf{m}}^* = \underset{\widehat{\mathbf{m}}}{\operatorname{argmax}} \sum_{\mathcal{T}} P(\widehat{\mathbf{m}}, \mathcal{T} | \widehat{\mathbf{w}})$$

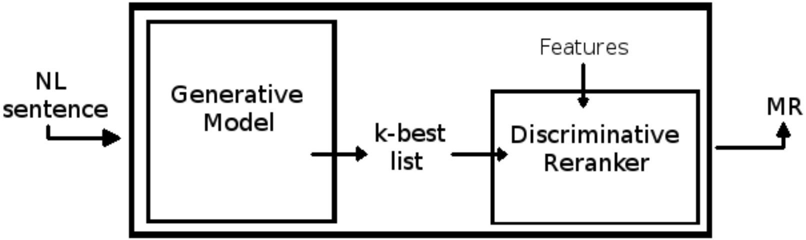
But summing over all possible trees $\mathcal{T}$ is expensive. Approximate with the most likely tree (Viterbi approximation).

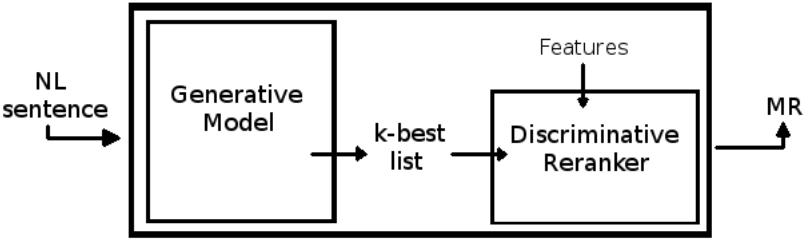
$$\widehat{\mathbf{m}}^* = \underset{\widehat{\mathbf{m}}}{\operatorname{argmax}} \max_{\mathcal{T}} P(\widehat{\mathbf{m}}, \mathcal{T} | \widehat{\mathbf{w}}) = \underset{\widehat{\mathbf{m}}}{\operatorname{argmax}} \max_{\mathcal{T}} P(\widehat{\mathbf{w}}, \widehat{\mathbf{m}}, \mathcal{T})$$

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

Outline

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.

- 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.

- 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

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

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

Outline

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

Results

Model	GEOQUERY (880)			ROBOCUP (300)		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
I	81.3	77.1	79.1	71.1	**64.0**	67.4
II	**89.0**	76.0	82.0	**82.4**	57.7	**67.8**
III	86.2	**81.8**	**84.0**	70.4	63.3	66.7
I+R	87.5	80.5	83.8	79.1	67.0	72.6
II+R	**93.2**	73.6	82.3	**88.4**	56.0	68.6
III+R	89.3	**81.5**	**85.2**	82.5	**67.7**	**74.4**

Comparison to previous work

System	GEOQUERY (880)			ROBOCUP (300)		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
SILT	89.0	54.1	67.3	83.9	50.7	63.2
WASP	87.2	74.8	80.5	**88.9**	61.9	73.0
KRISP	**93.3**	71.7	81.1	85.2	61.9	71.7
Model III+R	89.3	**81.5**	**85.2**	82.5	**67.7**	**74.4**

Comparison to previous work

System	GEOQUERY (880)			ROBOCUP (300)		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
SILT	89.0	54.1	67.3	83.9	50.7	63.2
WASP	87.2	74.8	80.5	**88.9**	61.9	73.0
KRISP	**93.3**	71.7	81.1	85.2	61.9	71.7
Model III+R	89.3	**81.5**	**85.2**	82.5	**67.7**	**74.4**

System	English			Spanish		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
WASP	95.42	70.00	80.76	91.99	72.40	81.03
Model III+R	91.46	72.80	**81.07**	95.19	79.20	**86.46**

System	Japanese			Turkish		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
WASP	91.98	74.40	**82.86**	96.96	62.40	75.93
Model III+R	87.56	76.00	81.37	93.82	66.80	**78.04**

(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.

References I

W. Lu, H. T. Ng, W. S. Lee, L. S. Zettlemoyer.
"A Generative Model for Parsing Natural Language to Meaning Representations".
Conference on Empirical Methods on Natural Language Processing, 2008.