Methods in Unsupervised Dependency Parsing

Mohammad Sadegh Rasooli

Candidacy exam
Department of Computer Science
Columbia University

April 1st, 2016
Overview

1 Introduction
   Dependency Grammar
   Dependency Parsing

2 Fully Unsupervised Parsing Models
   Unsupervised Parsing
   Dependency Model with Valence (DMV)
   Common Learning Algorithms for DMV
   Discussion

3 Syntactic Transfer Models
   Approaches in Syntactic Transfer
   Direct Syntactic Transfer
   Annotation Projection
   Discussion

4 Conclusion
A formal grammar introduced by [Tesnière, 1959] inspired from the valency theory in Chemistry.

In a dependency tree, each word has exactly one parent and can have as many dependents.

Benefit: explicit representation of syntactic roles.

Economic news had little effect on financial markets.
A formal grammar introduced by [Tesnière, 1959] inspired from the valency theory in Chemistry

In a dependency tree, each word has exactly one parent and can have as many dependents

Benefit: explicit representation of syntactic roles

Economic news had little effect on financial markets.
State-of-the-art parsing models are very accurate

Requirement: large amounts of annotated trees

- ≤50 treebanks available, ~7000 languages without any treebank
- Treebank development: an expensive and time-consuming task
  - Five years of work for the Penn Chinese Treebank [Hwa et al., 2005]

Unsupervised dependency parsing is an alternative approach when no treebank is available
State-of-the-art parsing models are very accurate

**Requirement:** large amounts of annotated trees

- \( \leq 50 \) treebanks available, \( \approx 7000 \) languages without any treebank
- Treebank development: an expensive and time-consuming task
  - *Five years* of work for the Penn Chinese Treebank [Hwa et al., 2005]

Unsupervised dependency parsing is an alternative approach when no treebank is available
Overview

1 Introduction
   Dependency Grammar
   Dependency Parsing

2 Fully Unsupervised Parsing Models
   Unsupervised Parsing
   Dependency Model with Valence (DMV)
   Common Learning Algorithms for DMV
   Discussion

3 Syntactic Transfer Models
   Approaches in Syntactic Transfer
   Direct Syntactic Transfer
   Annotation Projection
   Discussion

4 Conclusion
Unsupervised Parsing

- **Goal:** Develop an accurate parser *without* annotated data
- **Common assumptions**
  - Part-of-speech (POS) information is available
  - Raw data is available
Initial Attempts

- The seminal work of [Carroll and Charniak, 1992] and [Paskin, 2002] tried different techniques and achieved interesting results.
- Their models could not beat the baseline of attaching every word to the next word.
DMV: the First Breakthrough

- **Dependency model with valence (DMV)**
  
  [Klein and Manning, 2004] is the first model that could beat the baseline

- Most papers extended the DMV either in the inference method or parameter definition
The Dependency Model with Valence

- Input \( x \), output \( y \), \( p(x, y|\theta) = p(y^{(0)}|$, \( \theta $)
- \( \theta_c \) for dependency attachment
- \( \theta_s \) for stopping getting dependents
- \( \text{adj}(j) \): true iff \( x_j \) is adjacent to its parent
- \( \text{dep}_{\text{dir}}(j) \) set of dependents for \( x_j \) in direction \( \text{dir} \)

Recursive calculation

\[
P(y^{(i)}|x_i, \theta) = \prod_{\text{dir} \in \{\leftarrow, \rightarrow\}} \theta_s(\text{stop}|x_i, \text{dir}, [\text{dep}_{\text{dir}}(i) \neq \emptyset]) \\
\times \prod_{j \in \text{y}_{\text{dir}}(i)} (1 - \theta_s(\text{stop}|x_i, \text{dir}, \text{adj}(j))) \\
\times \theta_c(x_j|x_i, \text{dir}) \times P(y^{(j)}, \theta)
\]
DMV: A Running Example

\[ P(y^{(0)}) = \theta_c(VB|\text{ROOT}, \rightarrow) \times P(y^{(2)}|VB, \theta) \]
DMV: A Running Example

\[
P(y^{(0)}) = \theta_c(VB|\text{ROOT}, \rightarrow) \times P(y^{(2)}|VB, \theta)
\]

\[
P(y^{(2)}|VB, \theta) = \theta_s(\text{stop}|VB, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \leftarrow, \text{false}))
\]

\[
\times \theta_c(\text{PRN}|VB, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta)
\]
DMV: A Running Example

\[ P(y^{(0)}) = \theta_c(VB|\text{ROOT}, \rightarrow) \times P(y^{(2)}|VB, \theta) \]

\[ P(y^{(2)}|VB, \theta) = \theta_s(\text{stop}|VB, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \leftarrow, \text{false})) \]
\[ \times \theta_c(\text{PRN}|VB, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta) \]

\[ P(y^{(1)}|\text{PRN}, \theta) = \theta_s(\text{stop}|\text{PRN}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{PRN}, \rightarrow, \text{false}) \]
DMV: A Running Example

\[ P(y^{(0)}) = \theta_c(VB|ROOT, \rightarrow) \times P(y^{(2)}|VB, \theta) \]

\[ P(y^{(2)}|VB, \theta) = \theta_s(stop|VB, \leftarrow, true) \times (1 - \theta_s(stop|VB, \leftarrow, false)) \times \theta_c(PRN|VB, \leftarrow) \times P(y^{(1)}|PRN, \theta) \]

\[ \times \theta_s(stop|VB, \rightarrow, true) \times (1 - \theta_s(stop|VB, \rightarrow, false)) \times \theta_c(NN|VB, \rightarrow) \times P(y^{(4)}|NN, \theta) \]

\[ P(y^{(1)}|PRN, \theta) = \theta_s(stop|PRN, \leftarrow, false) \times \theta_s(stop|PRN, \rightarrow, false) \]
DMV: A Running Example

\[
P(y^{(0)}) = \theta_c(VB|\text{ROOT}, \rightarrow) \times P(y^{(2)}|VB, \theta)
\]

\[
P(y^{(2)}|VB, \theta) = \theta_s(\text{stop}|VB, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \leftarrow, \text{false})) \times \theta_c(\text{PRN}|VB, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta)
\]

\[
\times \theta_s(\text{stop}|VB, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \rightarrow, \text{false})) \times \theta_c(\text{NN}|VB, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta)
\]

\[
P(y^{(1)}|\text{PRN}, \theta) = \theta_s(\text{stop}|\text{PRN}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{PRN}, \rightarrow, \text{false})
\]

\[
P(y^{(4)}|\text{NN}, \theta) = \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{false})) \times \theta_c(\text{DT}|\text{NN}, \leftarrow) \times P(y^{(3)}|\text{DT}, \theta)
\]
DMV: A Running Example

\[
P(y^{(0)}) = \theta_c(VB|\text{ROOT}, \rightarrow) \times P(y^{(2)}|VB, \theta)
\]

\[
P(y^{(2)}|VB, \theta) = \theta_s(\text{stop}|VB, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \leftarrow, \text{false}))
\]

\[
\times \theta_c(\text{PRN}|VB, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta)
\]

\[
\times \theta_s(\text{stop}|VB, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \rightarrow, \text{false}))
\]

\[
\times \theta_c(\text{NN}|VB, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta)
\]

\[
P(y^{(1)}|\text{PRN}, \theta) = \theta_s(\text{stop}|\text{PRN}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{PRN}, \rightarrow, \text{false})
\]

\[
P(y^{(4)}|\text{NN}, \theta) = \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{false}))
\]

\[
\times \theta_c(\text{DT}|\text{NN}, \leftarrow) \times P(y^{(3)}|\text{DT}, \theta)
\]

\[
P(y^{(3)}|\text{DT}, \theta) = \theta_s(\text{stop}|\text{DT}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{DT}, \rightarrow, \text{false})
\]
DMV: A Running Example

\[
P(y^{(0)}) = \theta_c(VB|\text{ROOT}, \rightarrow) \times P(y^{(2)}|VB, \theta)
\]

\[
P(y^{(2)}|VB, \theta) = \theta_s(\text{stop}|VB, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \leftarrow, \text{false})) \times \theta_c(\text{PRN}|VB, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta)
\]

\[
\times \theta_s(\text{stop}|VB, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|VB, \rightarrow, \text{false})) \times \theta_c(\text{NN}|VB, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta)
\]

\[
P(y^{(1)}|\text{PRN}, \theta) = \theta_s(\text{stop}|\text{PRN}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{PRN}, \rightarrow, \text{false})
\]

\[
P(y^{(4)}|\text{NN}, \theta) = \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{false})) \times \theta_c(\text{DT}|\text{NN}, \leftarrow) \times P(y^{(3)}|\text{DT}, \theta)
\]

\[
\times \theta_s(\text{stop}|\text{NN}, \rightarrow, \text{false})
\]

\[
P(y^{(3)}|\text{DT}, \theta) = \theta_s(\text{stop}|\text{DT}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{DT}, \rightarrow, \text{false})
\]
DMV: Parameter Estimation

- Parameter estimation based on occurrence counts; e.g.

\[
\theta_c(w_j|w_i, \rightarrow) = \frac{\text{count}(w_i \rightarrow w_j)}{\sum_{w' \in \mathcal{V}} \text{count}(w_i \rightarrow w')} 
\]

- In an unsupervised setting, we can use dynamic programming (the Inside-Outside algorithm [Lari and Young, 1990]) to estimate model parameters \( \theta \)
Problems with DMV

- A non-convex optimization problem for DMV
  - Local optima is not necessarily a global optima
  - Very sensitive to the initialization

- Encoding constraints is not embedded in the original model
- Lack of expressiveness
- Low supervised accuracy (upperbound)
- Needs inductive bias
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement [Klein and Manning, 2004]
Problems with DMV

- A non-convex optimization problem for DMV
  - Local optima is not necessarily a global optima
  - Very sensitive to the initialization
- Encoding constraints is not embedded in the original model
  - Lack of expressiveness
  - Low supervised accuracy (upperbound)
- Needs inductive bias
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement [Klein and Manning, 2004]
Problems with DMV

- A **non-convex** optimization problem for DMV
  - Local optima is not necessarily a global optima
  - Very sensitive to the *initialization*
- Encoding constraints is not embedded in the original model
- Lack of **expressiveness**
  - Low supervised accuracy (upperbound)
- Needs **inductive bias**
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement
  - [Klein and Manning, 2004]
Problems with DMV

- A non-convex optimization problem for DMV
  - Local optima is not necessarily a global optima
  - Very sensitive to the initialization
- Encoding constraints is not embedded in the original model
- Lack of expressiveness
- Low supervised accuracy (upperbound)
- Needs inductive bias
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement
  
  [Klein and Manning, 2004]
Problems with DMV

- A **non-convex** optimization problem for DMV
  - Local optima is not necessarily a global optima
  - Very sensitive to the **initialization**
- Encoding constraints is not embedded in the original model
- Lack of **expressiveness**
- Low supervised accuracy (upperbound)
- Needs **inductive bias**
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement [Klein and Manning, 2004]
Extensions to DMV

- Changing the learning algorithm from EM
  - Contrastive estimation [Smith and Eisner, 2005]
  - Bayesian models [Headden III et al., 2009, Cohen and Smith, 2009a, Blunsom and Cohn, 2010, Naseem et al., 2010, Mareček and Straka, 2013]

- Local optima problem
  - Switching between different objectives [Spitkovsky et al., 2013]

- Lack of expressiveness
  - Lexicalization [Headden III et al., 2009]
  - Parameter tying [Cohen and Smith, 2009b, Headden III et al., 2009]
  - Tree substitution grammars [Blunsom and Cohn, 2010]
  - Rereanking with a richer model [Le and Zuidema, 2015]
Extensions to DMV

- Changing the learning algorithm from EM
  - Contrastive estimation [Smith and Eisner, 2005]
  - Bayesian models [Headden III et al., 2009, Cohen and Smith, 2009a, Blunsom and Cohn, 2010, Naseem et al., 2010, Mareček and Straka, 2013]
- Local optima problem
  - Switching between different objectives [Spitkovsky et al., 2013]
- Lack of expressiveness
  - Lexicalization [Headden III et al., 2009]
  - Parameter tying [Cohen and Smith, 2009b, Headden III et al., 2009]
  - Tree substitution grammars [Blunsom and Cohn, 2010]
  - Rereanking with a richer model [Le and Zuidema, 2015]
Extensions to DMV

- Changing the learning algorithm from EM
  - Contrastive estimation [Smith and Eisner, 2005]
  - Bayesian models [Headden III et al., 2009, Cohen and Smith, 2009a, Blunsom and Cohn, 2010, Naseem et al., 2010, Mareček and Straka, 2013]
- Local optima problem
  - Switching between different objectives [Spitkovsky et al., 2013]
- Lack of expressiveness
  - Lexicalization [Headden III et al., 2009]
  - Parameter tying [Cohen and Smith, 2009b, Headden III et al., 2009]
  - Tree substitution grammars [Blunsom and Cohn, 2010]
  - Rereanking with a richer model [Le and Zuidema, 2015]
Extensions to DMV

- Inductive bias
  - Adding constraints
    - Posterior regularization [Gillenwater et al., 2010]
    - Forcing unambiguity [Tu and Honavar, 2012]
    - Universal knowledge [Naseem et al., 2010]
  - Stop probability estimation from raw text
    [Mareček and Straka, 2013]

- Alternatives to DMV
  - Convex objective based on convex hull of plausible trees
    [Grave and Elhadad, 2015]
Extensions to DMV

- Inductive bias
  - Adding constraints
    - Posterior regularization [Gillenwater et al., 2010]
    - Forcing unambiguity [Tu and Honavar, 2012]
    - Universal knowledge [Naseem et al., 2010]
  - Stop probability estimation from raw text
    [Mareček and Straka, 2013]

- Alternatives to DMV
  - Convex objective based on convex hull of plausible trees
    [Grave and Elhadad, 2015]
Common Learning Algorithms for DMV

- Expectation maximization (EM) [Dempster et al., 1977]
- Posterior regularization (PR) [Ganchev et al., 2010]
- Variational Bayes (VB) [Beal, 2003]
- PR + VB [Naseem et al., 2010]
Common Learning Algorithms for DMV

▶ Expectation maximization (EM) [Dempster et al., 1977]
▶ Posterior regularization (PR) [Ganchev et al., 2010]
▶ Variational Bayes (VB) [Beal, 2003]
▶ PR + VB [Naseem et al., 2010]
Expectation Maximization (EM) Algorithm

- Start with initial parameters $\theta^{(t)}$ in iteration $t = 1$
- Repeat until $\theta^{(t)} \cong \theta^{(t+1)}$
  - **E step**: Maximize the posterior probability
    \[
    \forall i = 1 \ldots N; \forall y \in \mathcal{Y}_{x_i}
    q^{(t)}_i \leftarrow p_{\theta^{(t)}}(y|x) = \frac{p_{\theta^{(t)}}(x_i, y)}{\sum_{y' \in \mathcal{Y}_{x_i}} p_{\theta^{(t)}}(x_i, y')}
    \]
  - **M step**: Maximize the parameter values $\theta$
    \[
    \theta^{(t+1)} \leftarrow \arg \max_{\theta} \sum_{i=1}^{N} \sum_{y \in \mathcal{Y}_{x_i}} q^{(t)}_i(y) \log p_{\theta}(x_i, y)
    \]
- $t \leftarrow t + 1$
Expectation Maximization (EM) Algorithm

- Start with initial parameters $\theta^{(t)}$ in iteration $t = 1$
- Repeat until $\theta^{(t)} \simeq \theta^{(t+1)}$
  - **E step:** Maximize the posterior probability

\[
\forall i = 1 \ldots N; \forall y \in \mathcal{Y}_{x_i} \\
q_i^{(t)} \leftarrow p_{\theta^{(t)}}(y|x) = \frac{p_{\theta^{(t)}}(x_i, y)}{\sum_{y' \in \mathcal{Y}_{x_i}} p_{\theta^{(t)}}(x_i, y')}
\]

Another interpretation of the E step [Neal and Hinton, 1998]

\[
q^{(t)} \leftarrow \arg \min_q KL(q(Y) \mid \mid p_{\theta^{(t)}}(Y|X))
\]

- $t \leftarrow t + 1$
Expectation Maximization (EM) Algorithm

- Start with initial parameters $\theta^{(t)}$ in iteration $t = 1$
- Repeat until $\theta^{(t)} \simeq \theta^{(t+1)}$

**M step**

Optimal parameters for a categorical distribution is achieved by normalization:

$$
\theta^{(t+1)}(y|x) = \frac{\sum_{i=1}^{N} q_i^{(t)}(y|x)}{\sum_{y'} \sum_{i=1}^{N} q_i^{(t)}(y'|x)}
$$

- **M step**: Maximize the parameter values $\theta$

$$
\theta^{(t+1)} \leftarrow \arg \max_{\theta} \sum_{i=1}^{N} \sum_{y \in Y_{x_i}} q_i^{(t)}(y) \log p_{\theta}(x_i, y)
$$

- $t \leftarrow t + 1$
Common Learning Algorithms for DMV

- Expectation maximization (EM) [Dempster et al., 1977]
- Posterior regularization (PR) [Ganchev et al., 2010]
- Variational Bayes (VB) [Beal, 2003]
- PR + VB [Naseem et al., 2010]
Posterior Regularization

- Prior knowledge as constraint
- Just affects the \textbf{E step} and the \textbf{M step} remains unchanged
Posterior Regularization

Original objective

\[ q^{(t)} \leftarrow \arg \min_q \text{KL}(q(Y) \mid\mid p_{\theta^{(t)}}(Y|X)) \]

Modified objective

\[ q^{(t)} \leftarrow \arg \min_q \text{KL}(q(Y) \mid\mid p_{\theta^{(t)}}(Y|X)) + \sigma \sum_i b_i \]

\[ s.t. \quad \|E_q[\phi_i(X, Y)]\|_\beta \leq b_i \]

\( \sigma \) is the regularization coefficient and \( b_i \) is the proposed numerical constraint for sentence \( i \).
Modified objective

\[ q^{(t)} \leftarrow \arg \min_q KL(q(Y) \mid\mid p_{\theta(t)}(Y \mid X)) + \sigma \sum_i b_i \]

Types of constraints:

- Number of unique child-head tag pairs in a sentence (less is better) [Gillenwater et al., 2010]
- Number of preserved pre-defined linguistic rules in a tree (more is better) [Naseem et al., 2010]
- Information entropy of the sentence (less is better) [Tu and Honavar, 2012]
Common Learning Algorithms for DMV

- Expectation maximization (EM) [Dempster et al., 1977]
- Posterior regularization (PR) [Ganchev et al., 2010]
- **Variational Bayes (VB)** [Beal, 2003]
- PR + VB [Naseem et al., 2010]
Variational Bayes

- A Bayesian model that encodes prior information
- Just affects the M step and the E step remains unchanged
### Variational Bayes

**M step**

\[
\theta^{(t+1)}(y|x) = \frac{\sum_{i=1}^{N} q_i^{(t)}(y|x)}{\sum_{y'} \sum_{i=1}^{N} q_i^{(t)}(y'|x)}
\]

**Modified M step in VB**

\[
\theta^{(t+1)}(y|x) = \frac{F(\alpha_y + \sum_{i=1}^{N} q_i^{(t)}(y|x))}{F(\sum_{y'} \alpha_{y'} + \sum_{i=1}^{N} q_i^{(t)}(y'|x))}
\]

\(\alpha\) is the prior

\[F(v) = e^{\Psi(v)}\]

\(\Psi\) is the digamma function
Common Learning Algorithms for DMV

- Expectation maximization (EM) [Dempster et al., 1977]
- Posterior regularization (PR) [Ganchev et al., 2010]
- Variational Bayes (VB) [Beal, 2003]
- PR + VB [Naseem et al., 2010]
VB + PR

- Makes use of both methods [Naseem et al., 2010]:
  - E step as in PR
  - M step as in VB
Discussion

- Significant improvements?
  - Yes!
- Satisfying performance?
  - No!
  - Mostly optimized for English
  - Far less than a supervised model
Discussion

- Significant improvements?
  - Yes!
- Satisfying performance?
  - No!
    - Mostly optimized for English
    - Far less than a supervised model
Discussion

- Significant improvements?
  - Yes!

- Satisfying performance?
  - No!
    - Mostly optimized for English
    - Far less than a supervised model
Discussion

- Significant improvements?
  - Yes!

- Satisfying performance?
  - No!
    - Mostly optimized for English
    - Far less than a supervised model
Unsupervised Parsing Improvement Over Time

[Klein and Manning, 2004]
Unsupervised Parsing Improvement Over Time

Unlabeled dependency accuracy on WSJ testing data

- Random: 30.1
- Adjacent: 33.6
- DMV: 35.9
- 2008: 40.5

[Cohen et al., 2008]
Unsupervised Parsing Improvement Over Time

- Random: 30.1
- Adjacent: 33.6
- DMV: 35.9
- 2008: 40.5
- 2009: 41.4

[Cohen and Smith, 2009a]
Unsupervised Parsing Improvement Over Time

Unlabeled dependency accuracy on WSJ testing data

[Blunsom and Cohn, 2010]

Methods in Unsupervised Dependency Parsing

Mohammad Sadegh Rasooli
Unsupervised Parsing Improvement Over Time

[Spitkovsky et al., 2011]

unlabeled dependency accuracy on WSJ testing data

- Random: 30.1
- Adjacent: 33.6
- DMV: 35.9
- 2008: 40.5
- 2009: 41.4
- 2010: 55.7
- 2011: 59.1

Unsupervised Parsing Improvement Over Time

Mohammad Sadegh Rasooli  Methods in Unsupervised Dependency Parsing
Unsupervised Parsing Improvement Over Time

[Spitkovsky et al., 2013]
Unsupervised Parsing Improvement Over Time

[Le and Zuidema, 2015]
Unsupervised Parsing Improvement Over Time

15 minutes of programming to write down rules gives ≈ 60% accuracy!
Introduction

Fully Unsupervised Parsing Models
Syntactic Transfer Models
Conclusion

Unsupervised Parsing Improvement Over Time

Unlabeled dependency accuracy on WSJ testing data

<table>
<thead>
<tr>
<th>Year</th>
<th>Random</th>
<th>Adjacent</th>
<th>DMV</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2015</th>
<th>DMV-supervised</th>
<th>Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>30.1</td>
<td>33.6</td>
<td>35.9</td>
<td>40.5</td>
<td>41.4</td>
<td>55.7</td>
<td>59.1</td>
<td>61.2</td>
<td>64.4</td>
<td>66.2</td>
<td>76.3</td>
<td>94.4</td>
</tr>
</tbody>
</table>

Mohammad Sadegh Rasooli  
Methods in Unsupervised Dependency Parsing
Overview

1 Introduction
   Dependency Grammar
   Dependency Parsing

2 Fully Unsupervised Parsing Models
   Unsupervised Parsing
   Depndency Model with Valence (DMV)
   Common Learning Algorithms for DMV
   Discussion

3 Syntactic Transfer Models
   Approaches in Syntactic Transfer
   Direct Syntactic Transfer
   Annotation Projection
   Discussion

4 Conclusion
Syntactic Transfer Models

- **Transfer Learning**: learn a problem $X$ and apply to a similar (but not the same) problem $Y$.

- **Challenges**: feature mismatch, domain mismatch, and lack of sufficient similarity between the two problems.

- **Syntactic transfer**: Learn a parser for languages $\mathcal{L}_1 \ldots \mathcal{L}_m$ and use them for parsing language $\mathcal{L}_{m+1}$.

- **Challenges**: mismatch in lexical features, difference in word order.
Syntactic Transfer Models

- **Transfer Learning:** learn a problem $X$ and apply to a similar (but not the same) problem $Y$

- **Challenges:** feature mismatch, domain mismatch, and lack of sufficient similarity between the two problems

- **Syntactic transfer:** Learn a parser for languages $L_1 \ldots L_m$ and use them for parsing language $L_{m+1}$

- **Challenges:** mismatch in **lexical features**, difference in **word order**
Approaches in Syntactic Transfer

- **Direct transfer**: train directly on treebanks for languages $\mathcal{L}_1 \ldots \mathcal{L}_m$ and apply it to language $\mathcal{L}_{m+1}$
- **Annotation projection**: use parallel data and project supervised parse trees in language $\mathcal{L}_s$ to target language through word alignment
- **Treebank translation**: develop an SMT system, translate source treebanks to the target language, and train on the translated treebank [Tiedemann et al., 2014]
Approaches in Syntactic Transfer

- **Direct transfer:** train directly on treebanks for languages $\mathcal{L}_1 \ldots \mathcal{L}_m$ and apply it to language $\mathcal{L}_{m+1}$

- **Annotation projection:** use parallel data and project supervised parse trees in language $\mathcal{L}_s$ to target language through word alignment

- **Treebank translation:** develop an SMT system, translate source treebanks to the target language, and train on the translated treebank [Tiedemann et al., 2014]
Approaches in Syntactic Transfer

- **Direct transfer:** train directly on treebanks for languages $\mathcal{L}_1 \ldots \mathcal{L}_m$ and apply it to language $\mathcal{L}_{m+1}$

- **Annotation projection:** use parallel data and project supervised parse trees in language $\mathcal{L}_s$ to target language through **word alignment**

- **Treebank translation:** develop an **SMT** system, translate source treebanks to the target language, and train on the **translated treebank** [Tiedemann et al., 2014]
A supervised parser gets input $x$ and outputs the best tree $y^*$, using lexical features $\phi^{(l)}(x, y)$ and unlexicalized features $\phi^{(p)}(x, y)$:

$$y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} \theta_l \cdot \phi^{(l)}(x, y) + \theta_p \cdot \phi^{(p)}(x, y)$$

A direct transfer model cannot make use of lexical features. Direct delexicalized transfer only uses unlexicalized features [Cohen et al., 2011, McDonald et al., 2011]
A **supervised parser** gets input $x$ and outputs the best tree $y^*$, using *lexical features* $\phi^{(l)}(x, y)$ and *unlexicalized features* $\phi^{(p)}(x, y)$:

$$y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} \theta_l \cdot \phi^{(l)}(x, y) + \theta_p \cdot \phi^{(p)}(x, y)$$

A **direct transfer** model cannot make use of lexical features.

**Direct delexicalized transfer** only uses unlexicalized features.

[Cohen et al., 2011, McDonald et al., 2011]
A supervised parser gets input $x$ and outputs the best tree $y^*$, using lexical features $\phi^{(l)}(x, y)$ and unlexicalized features $\phi^{(p)}(x, y)$:

$$y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} \theta_l \cdot \phi^{(l)}(x, y) + \theta_p \cdot \phi^{(p)}(x, y)$$

A direct transfer model cannot make use of lexical features. Direct delexicalized transfer only uses unlexicalized features [Cohen et al., 2011, McDonald et al., 2011]
Direct Delexicalized Transfer: Pros and Cons

**Pros**

- **Simplicity**: can employ any supervised parser
- More accurate than **fully unsupervised** models

**Cons**

- No treatment for word order difference
- Lack of lexical features
Direct Delexicalized Transfer: Pros and Cons

**Pros**

- **Simplicity**: can employ any supervised parser
- More accurate than *fully unsupervised* models

**Cons**

- No treatment for *word order* difference
- Lack of *lexical features*
Addressing problems in direct delexicalized transfer

- Word order difference
- Lack of lexical features
Addressing problems in direct delexicalized transfer

- **Word order difference**
- **Lack of lexical features**
The World Atlas of Language Structures (WALS)

[Dryer and Haspelmath, 2013] is a large database of structural (phonological, grammatical, lexical) properties for near 3000 languages.
Selective Sharing: Addressing Words Order Problem

- Use **typological features** such as the subject-verb order for each source and target language.
- In addition to the **original parameters**, share **typological features** for languages that have specific orderings in common
  - Added features: original features conjoined with each typological feature
  - **Discriminative models with selective sharing** gain very high accuracies [Täckström et al., 2013, Zhang and Barzilay, 2015]
Selective Sharing: Addressing Words Order Problem

- Use **typological features** such as the subject-verb order for each source and target language.
- In addition to the **original parameters**, share **typological features** for languages that have specific orderings in common.
  - Added features: original features conjoined with each typological feature
- **Discriminative** models with **selective sharing** gain very high accuracies [Täckström et al., 2013, Zhang and Barzilay, 2015]
Addressing problems in direct delexicalized transfer

- Word order difference
- Lack of lexical features
Addressing the Lack of Lexical Features

- Using **bilingual dictionaries** to transfer lexical features
  [Durrett et al., 2012, Xiao and Guo, 2015]

- Creating **cross-lingual word representations**
  - **without** parallel text [Duong et al., 2015]
  - **using** parallel text [Zhang and Barzilay, 2015, Guo et al., 2016]

- Successful models use **cross-lingual word representations using parallel text**
  - Could we leverage more if we have parallel text?
    - Yes!
Addressing the Lack of Lexical Features

- Using **bilingual dictionaries** to transfer lexical features
  
  [Durrett et al., 2012, Xiao and Guo, 2015]

- Creating **cross-lingual word representations**
  
  - **without** parallel text [Duong et al., 2015]
  
  - **using** parallel text [Zhang and Barzilay, 2015, Guo et al., 2016]

- Successful models use **cross-lingual word representations using parallel text**
  
  - Could we leverage more if we have parallel text?
    
    - Yes!
Addressing the Lack of Lexical Features

- Using *bilingual dictionaries* to transfer lexical features
  [Durrett et al., 2012, Xiao and Guo, 2015]

- Creating *cross-lingual word representations*
  - *without* parallel text [Duong et al., 2015]
  - *using* parallel text [Zhang and Barzilay, 2015, Guo et al., 2016]

- Successful models use *cross-lingual word representations* using *parallel text*
  - Could we leverage more if we have parallel text?
    - Yes!
Annotation Projection

Steps in annotation projection

1. Prepare bitext
2. Align bitext
3. Parse source sentence with a supervised parser
4. Project dependencies
5. Train on the projected dependencies
Annotation Projection

Steps in annotation projection

1. Prepare bitext
2. Align bitext
3. Parse source sentence with a supervised parser
4. Project dependencies
5. Train on the projected dependencies
Annotation Projection

Steps in annotation projection

1. Prepare bitext
2. Align bitext
3. Parse source sentence with a supervised parser
4. Project dependencies
5. Train on the projected dependencies
Annotation Projection

Steps in annotation projection

1. Prepare bitext
2. Align bitext
3. Parse source sentence with a supervised parser
4. Project dependencies
5. Train on the projected dependencies
Annotation Projection

Steps in annotation projection

1. Prepare bitext
2. Align bitext
3. Parse source sentence with a supervised parser
4. Project dependencies
5. Train on the projected dependencies
Annotation Projection

- Steps in annotation projection
  1. Prepare *bitext*
  2. Align *bitext*
  3. Parse source sentence with a supervised parser
  4. Project dependencies
  5. Train on the *projected* dependencies
Projecting Dependencies from Parallel Data

Bitext

Prepare bitext

The political priorities must be set by this House and the MEPs.  
Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Projecting Dependencies from Parallel Data

Align

Align bitext (e.g. via Giza++)

The political priorities must be set by this House and the MEPs.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Projecting Dependencies from Parallel Data

Parse

Parse source sentence with a supervised parser

The political priorities must be set by this House and the MEPs.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Projecting Dependencies from Parallel Data

Project dependencies

The political priorities must be set by this House and the MEPs.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Train

Train on the projected dependencies

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.  

ROOT
Practical Problems

- Most translations are **not word-to-word**
  - Partial alignments
- Alignment errors
- Supervised parsers are not perfect
- Difference in **syntactic behavior** across languages
Approaches in Annotation Projection

- Post-processing alignments with *rules* and *filtering* sparse trees [Hwa et al., 2005]
- Use projected dependencies as *constraints in posterior regularization* [Ganchev et al., 2009]
- Use projected dependencies to *lexicalize a direct model* [McDonald et al., 2011]
- *Entropy regularization* on projected trees [Ma and Xia, 2014]
- Start with *fully projected trees and self-train on partial trees* [Rasooli and Collins, 2015]
Approaches in Annotation Projection

- Post-processing alignments with **rules** and **filtering** sparse trees [Hwa et al., 2005]
- Use projected dependencies as **constraints** in **posterior regularization** [Ganchev et al., 2009]
- Use projected dependencies to **lexicalize a direct model** [McDonald et al., 2011]
- **Entropy regularization** on projected trees [Ma and Xia, 2014]
- Start with **fully projected trees and self-train on partial trees** [Rasooli and Collins, 2015]
Approaches in Annotation Projection

- Post-processing alignments with **rules** and **filtering** sparse trees [Hwa et al., 2005]
- Use projected dependencies as **constraints** in **posterior regularization** [Ganchev et al., 2009]
- Use projected dependencies to **lexicalize** a **direct** model [McDonald et al., 2011]
- **Entropy regularization** on projected trees [Ma and Xia, 2014]
- Start with **fully projected trees** and self-train on **partial trees** [Rasooli and Collins, 2015]
Approaches in Annotation Projection

- Post-processing alignments with **rules** and **filtering** sparse trees [Hwa et al., 2005]
- Use projected dependencies as **constraints** in **posterior regularization** [Ganchev et al., 2009]
- Use projected dependencies to **lexicalize a direct model** [McDonald et al., 2011]
- **Entropy regularization** on projected trees [Ma and Xia, 2014]
- Start with **fully projected trees and self-train on partial trees** [Rasooli and Collins, 2015]
Approaches in Annotation Projection

- Post-processing alignments with **rules** and **filtering** sparse trees [Hwa et al., 2005]
- Use projected dependencies as **constraints** in **posterior regularization** [Ganchev et al., 2009]
- Use projected dependencies to **lexicalize a direct model** [McDonald et al., 2011]
- **Entropy regularization** on projected trees [Ma and Xia, 2014]
- Start with **fully projected trees** and **self-train on partial trees** [Rasooli and Collins, 2015]
Discussion

- Significant improvements?
  - Yes!
- Satisfying performance?
  - Yes!
  - Mostly optimized for rich-resource languages
Discussion

- Significant improvements?
  - Yes!

- Satisfying performance?
  - Yes!
  - Mostly optimized for rich-resource languages
Discussion

- Significant improvements?
  - Yes!

- Satisfying performance?
  - Yes!
  - Mostly optimized for rich-resource languages
Discussion

- Significant improvements?
  - Yes!
- Satisfying performance?
  - Yes!
  - Mostly optimized for rich-resource languages
Unsupervised Parsing Best Models Comparison

[Grave and Elhadad, 2015]

average unlabeled dependency accuracy on 6 EU languages

Unsupervised: 56.1
Unsupervised Parsing Best Models Comparison

![Graph showing average unlabeled dependency accuracy on 6 EU languages.]

- **Unsupervised:** 56.1
- **Direct:** 77.8
- **Annotation Projection:** [Ammar et al., 2016]
- **Supervised**

Mohammad Sadegh Rasooli

Methods in Unsupervised Dependency Parsing
Unsupervised Parsing Best Models Comparison

- **Unsupervised**: 56.1
- **Direct**: 77.8
- **Ann. Proj.**: 82.2
- **Supervised**: [Rasooli and Collins, 2015]

Average unlabeled dependency accuracy on 6 EU languages.
Unsupervised Parsing Best Models Comparison

- Unsupervised: 56.1
- Direct: 77.8
- Annotation Projection: 82.2
- Supervised: 87.5

Average unlabeled dependency accuracy on 6 EU languages.
Overview

1. Introduction
   - Dependency Grammar
   - Dependency Parsing

2. Fully Unsupervised Parsing Models
   - Unsupervised Parsing
   - Dependency Model with Valence (DMV)
   - Common Learning Algorithms for DMV
   - Discussion

3. Syntactic Transfer Models
   - Approaches in Syntactic Transfer
   - Direct Syntactic Transfer
   - Annotation Projection
   - Discussion

4. Conclusion
Conclusion

- Read 28+ papers about
  - Unsupervised dependency parsing
  - Direct cross-lingual transfer of dependency parsers
  - Annotation projection for cross-lingual transfer

- Seems that more effort may decrease the need for new treebanks!
Conclusion

- Read 28+ papers about
  - Unsupervised dependency parsing
  - Direct cross-lingual transfer of dependency parsers
  - Annotation projection for cross-lingual transfer
- Seems that more effort may decrease the need for new treebanks!
Thanks

Thanks a lot

Danke sehr


References VI


References VII


References IX


