Methods in Unsupervised Dependency Parsing

Mohammad Sadegh Rasooli

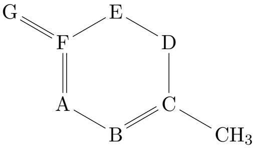
Candidacy exam
Department of Computer Science
Columbia University

April 1st, 2016

Overview

Dependency Grammar

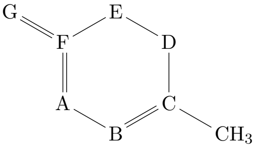
- A formal grammar introduced by [Tesnière, 1959] inspired from the **valency theory** in **Chemistry**



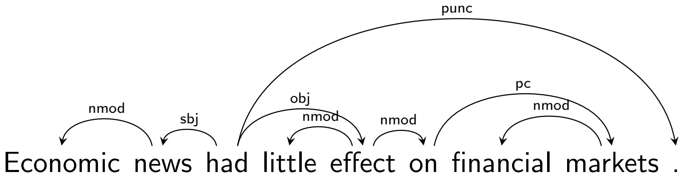
- In a **dependency tree**, each word has exactly one parent and can have as many dependents
- **Benefit**: explicit representation of syntactic roles

Dependency Grammar

- 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



Dependency Parsing

- State-of-the-art parsing models are very accurate
- **Requirement:** large amounts of **annotated trees**
  - $\leq$50 treebanks available, $\simeq$**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

Dependency Parsing

- State-of-the-art parsing models are very accurate
- **Requirement:** large amounts of **annotated trees**
  - $\leq$50 treebanks available, $\simeq$**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

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

Recursive calculation

$$P(\mathbf{y}^{(i)}|x_i, \theta) = \prod_{\mathsf{dir}\in\{\leftarrow,\rightarrow\}} \theta_s(\mathsf{stop}|x_i, \mathsf{dir}, [\mathsf{dep}_{\mathsf{dir}}(i) \stackrel{?}{=} \emptyset]) \times \prod_{j\in\mathbf{y}_{\mathsf{dir}}(i)} (1 - \theta_s(\mathsf{stop}|x_i, \mathsf{dir}, \mathsf{adj}(j))) \times \theta_c(x_j|x_i, \mathsf{dir}) \times P(\mathbf{y}^{(j)}, \theta)$$

DMV: A Running Example

ROOT VB

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

DMV: A Running Example

ROOT PRN VB

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

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

DMV: A Running Example

ROOT PRN VB

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

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

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

DMV: A Running Example

ROOT PRN VB NN

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

$$\begin{aligned} P(y^{(2)}|VB, \theta) = & \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{false})) \\ & \times \theta_c(\text{PRN}|\text{VB}, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta) \\ & \times \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{false})) \\ & \times \theta_c(\text{NN}|\text{VB}, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta) \end{aligned}$$

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

DMV: A Running Example

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

$$\begin{aligned}P(y^{(2)}|VB, \theta) = & \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{false})) \\ & \times \theta_c(\text{PRN}|\text{VB}, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta) \\ & \times \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{false})) \\ & \times \theta_c(\text{NN}|\text{VB}, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta)\end{aligned}$$

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

$$\begin{aligned}P(y^{(4)}|NN, \theta) = & \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{false})) \\ & \times \theta_c(DT|NN, \leftarrow) \times P(y^{(3)}|DT, \theta)\end{aligned}$$

DMV: A Running Example

ROOT PRN VB DT NN

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

$$\begin{aligned} P(y^{(2)}|VB, \theta) =& \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{false})) \\ & \times \theta_c(\text{PRN}|\text{VB}, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta) \\ & \times \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{false})) \\ & \times \theta_c(\text{NN}|\text{VB}, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta) \end{aligned}$$

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

$$\begin{aligned} P(y^{(4)}|NN, \theta) =& \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{false})) \\ & \times \theta_c(DT|NN, \leftarrow) \times P(y^{(3)}|DT, \theta) \end{aligned}$$

$$P(y^{(3)}|DT, \theta) = \theta_s(\text{stop}|\text{DT}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{DT}, \rightarrow, \text{false})$$

DMV: A Running Example

ROOT PRN VB DT NN

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

$$\begin{aligned} P(y^{(2)}|VB, \theta) =& \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \leftarrow, \text{false})) \\ &\times \theta_c(\text{PRN}|\text{VB}, \leftarrow) \times P(y^{(1)}|\text{PRN}, \theta) \\ &\times \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{VB}, \rightarrow, \text{false})) \\ &\times \theta_c(\text{NN}|\text{VB}, \rightarrow) \times P(y^{(4)}|\text{NN}, \theta) \end{aligned}$$

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

$$\begin{aligned} P(y^{(4)}|NN, \theta) =& \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{true}) \times (1 - \theta_s(\text{stop}|\text{NN}, \leftarrow, \text{false})) \\ &\times \theta_c(DT|NN, \leftarrow) \times P(y^{(3)}|DT, \theta) \\ &\times \theta_s(\text{stop}|\text{NN}, \rightarrow, \text{false}) \end{aligned}$$

$$P(y^{(3)}|DT, \theta) = \theta_s(\text{stop}|\text{DT}, \leftarrow, \text{false}) \times \theta_s(\text{stop}|\text{DT}, \rightarrow, \text{false})$$

DMV: Parameter Estimation

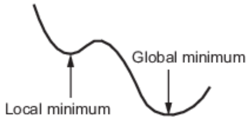
- Parameter estimation based on occurrence counts; e.g.

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

  - In an unsupervised setting, we can use **dynamic programming** (the **Inside-Outside algorithm** [Lari and Young, 1990]) to estimate model parameters θ

Problems with DMV

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

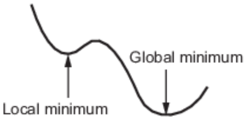


- Encoding constraints is not embedded in the original model
- Lack of **expressiveness**
- Low supervised accuracy (upperbound)
- Needs **inductive bias**
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement [Klein and Manning, 2004]

DET NOUN

Problems with DMV

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

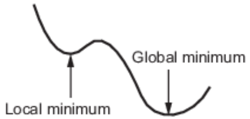


- Encoding constraints is not embedded in the original model
- Lack of **expressiveness**
- Low supervised accuracy (upperbound)
- Needs **inductive bias**
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement [Klein and Manning, 2004]

DET NOUN

Problems with DMV

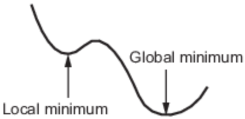
- A **non-convex** optimization problem for DMV
  - Local optima is not necessarily a global optima
  - Very sensitive to the **initialization**



- 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**

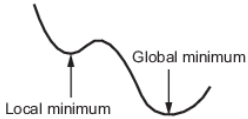


- Encoding constraints is not embedded in the original model
- Lack of **expressiveness**
- Low supervised accuracy (upperbound)
- Needs **inductive bias**
  - Post-processing the DMV output by fixing the determiner-noun direction gave a huge improvement [Klein and Manning, 2004]

DET NOUN

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]

DET NOUN

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 $\theta^{(t)} \simeq \theta^{(t+1)}$
  - **E step:** Maximize the posterior probability

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

  - **M step:** Maximize the parameter values θ

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

  - $t \leftarrow t + 1$

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]

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

  - $t \leftarrow t + 1$

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 θ

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

- $t \leftarrow t + 1$

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 **E step** and the **M step** remains unchanged

Posterior Regularization

Original objective

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

Modified objective

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

$$s.t. \quad ||\mathbb{E}_q[\phi_i(X,Y)]||_\beta \leq b_i$$

σ is the regularization coefficient and b_i is the proposed numerical constraint for sentence i.

Posterior Regularization Constraints

Modified objective

$$q^{(t)} \leftarrow \arg\min_{q} \mathbf{KL}(q(Y) \mid\mid p_{\theta^{(t)}}(Y|X)) + \sigma \sum_{i} b_i$$

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

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

α is the prior

$$\mathbb{F}(v) = e^{\Psi(v)}$$

Ψ 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

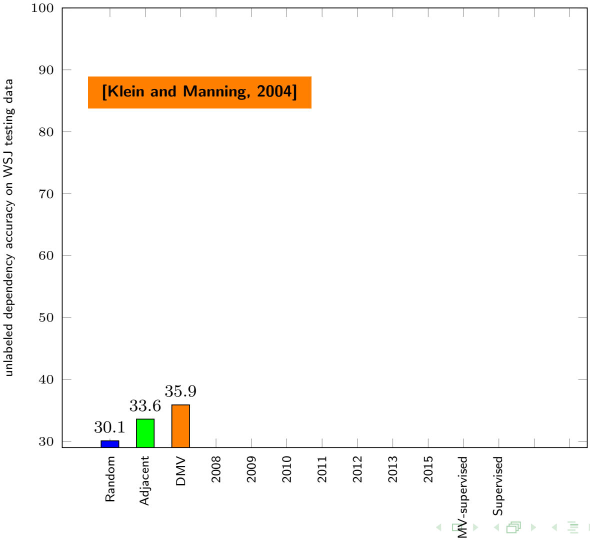
Discussion

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

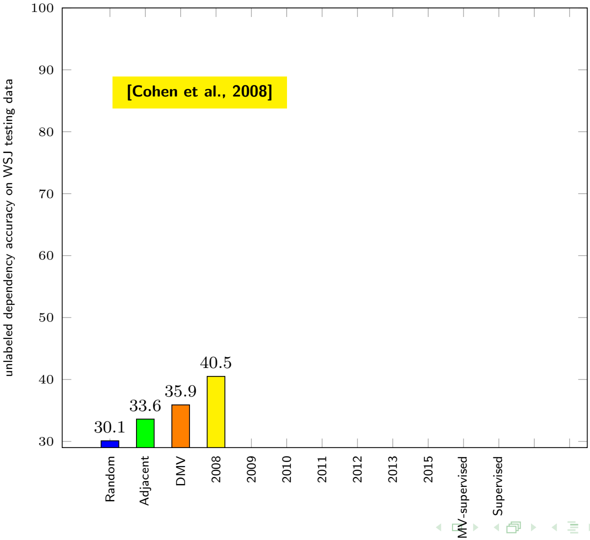
Discussion

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

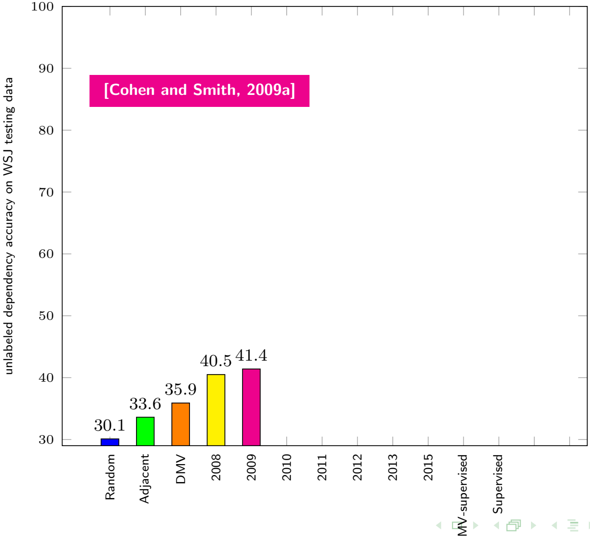
Unsupervised Parsing Improvement Over Time



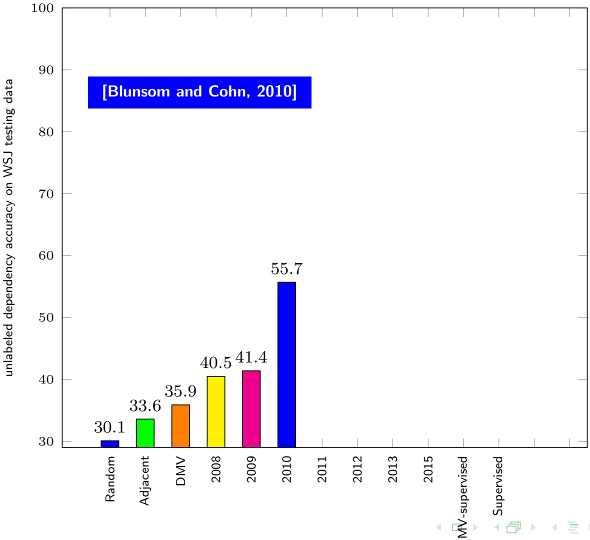
Unsupervised Parsing Improvement Over Time



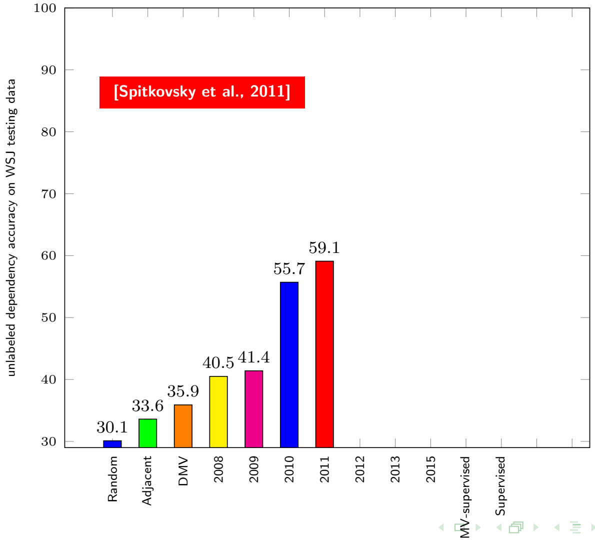
Unsupervised Parsing Improvement Over Time



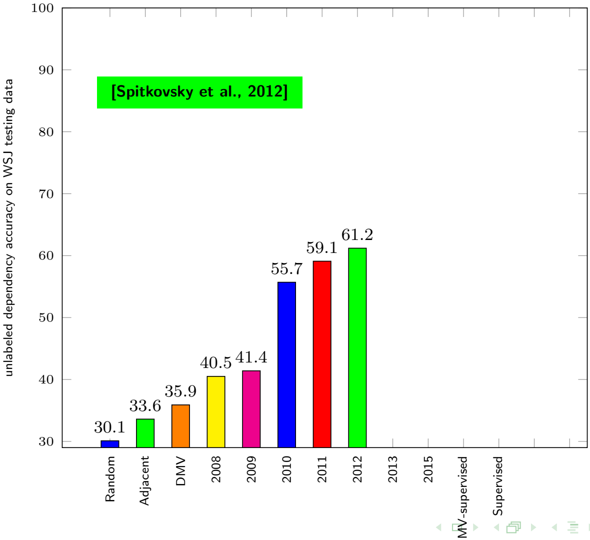
Unsupervised Parsing Improvement Over Time



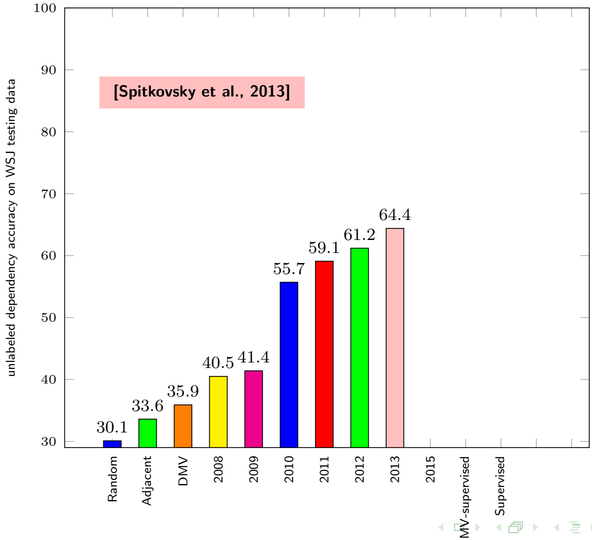
Unsupervised Parsing Improvement Over Time



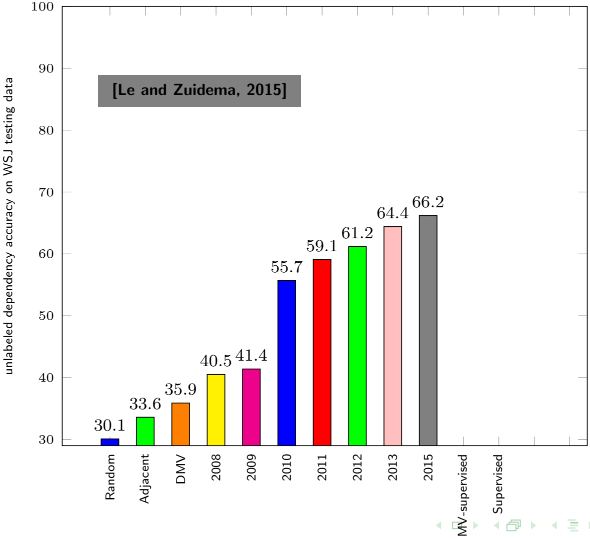
Unsupervised Parsing Improvement Over Time



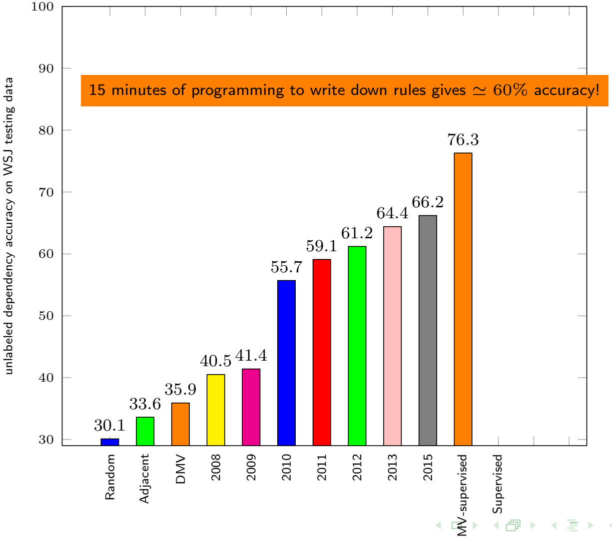
Unsupervised Parsing Improvement Over Time



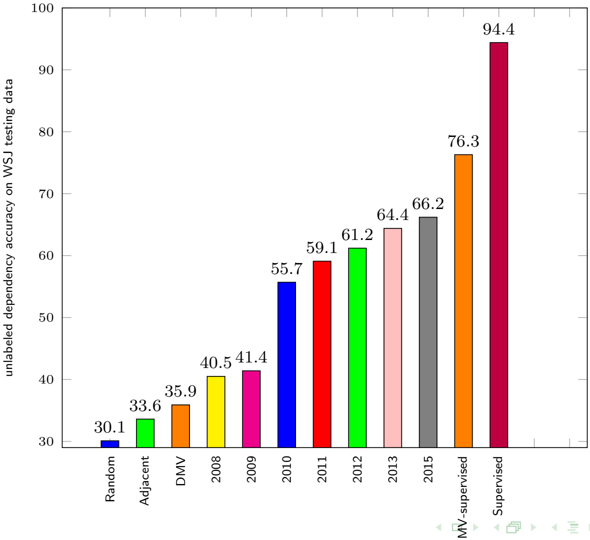
Unsupervised Parsing Improvement Over Time



Unsupervised Parsing Improvement Over Time



Unsupervised Parsing Improvement Over Time



Overview

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 $\mathcal{L}_1 \dots \mathcal{L}_m$ and use them for parsing language $\mathcal{L}_{m+1}$
- **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 $\mathcal{L}_1 \dots \mathcal{L}_m$ 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]

Direct Syntactic Transfer

- A **supervised parser** gets input x and outputs the best tree y^*, using **lexical features** $\phi^{(l)}(x, y)$ and **unlexicalized features** $\phi^{(p)}(x, y)$:

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

- 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 Syntactic Transfer

- A **supervised parser** gets input x and outputs the best tree y^*, using **lexical features** $\phi^{(l)}(x,y)$ and **unlexicalized features** $\phi^{(p)}(x,y)$:

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

- 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 Syntactic Transfer

- A **supervised parser** gets input x and outputs the best tree y^*, using **lexical features** $\phi^{(l)}(x, y)$ and **unlexicalized features** $\phi^{(p)}(x, y)$:

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

- 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

Addressing problems in direct delexicalized transfer

- Word order difference
- Lack of lexical features

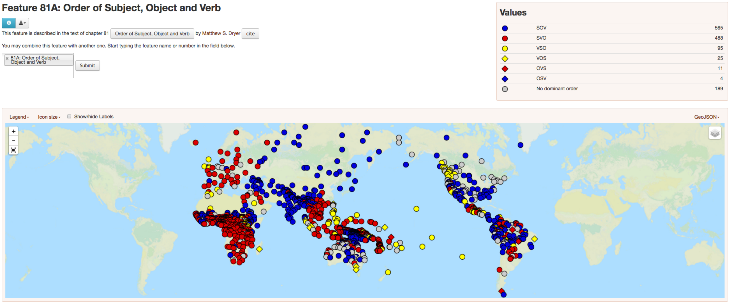
Addressing Problems in Direct Delexicalized Transfer

Addressing problems in direct delexicalized transfer

- **Word order difference**
- Lack of lexical features

The World Atlas of Language Structures (WALS)

- **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

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

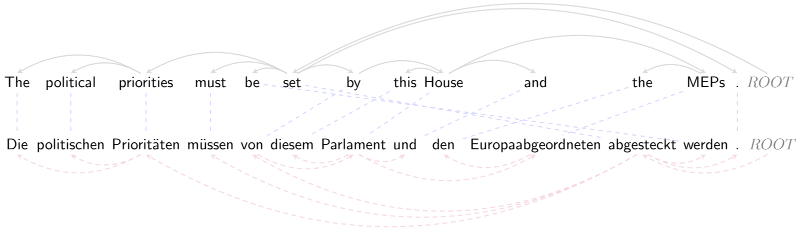
Annotation Projection

- 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

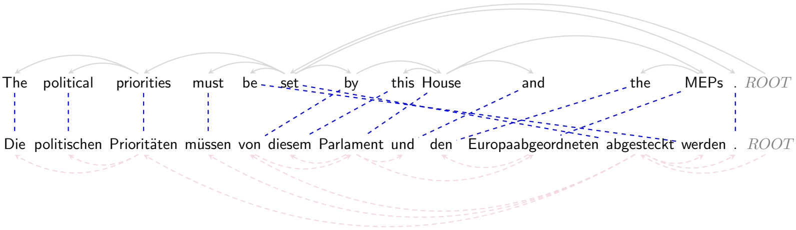
Prepare bitext



Projecting Dependencies from Parallel Data

Align

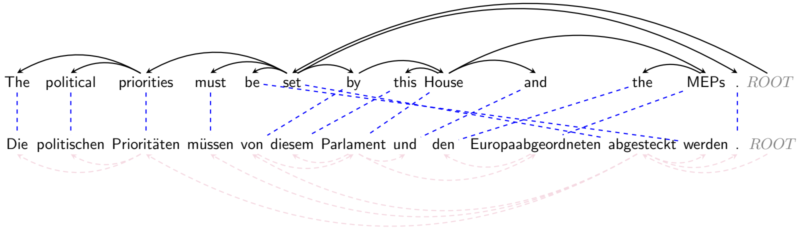
Align bitext (e.g. via Giza++)



Projecting Dependencies from Parallel Data

Parse

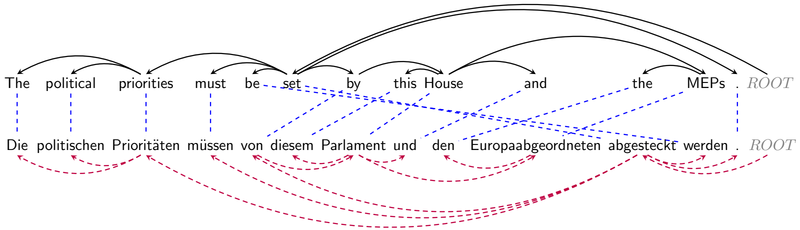
Parse source sentence with a supervised parser



Projecting Dependencies from Parallel Data

Project

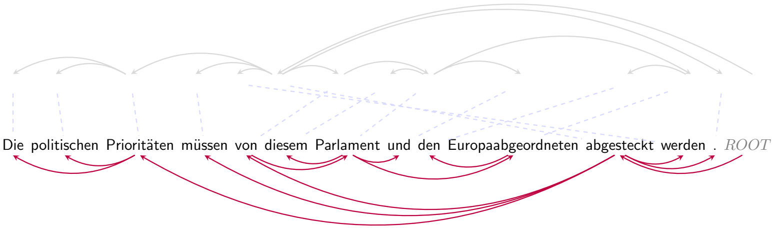
Project dependencies



Projecting Dependencies from Parallel Data

Train

Train on the projected dependencies



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

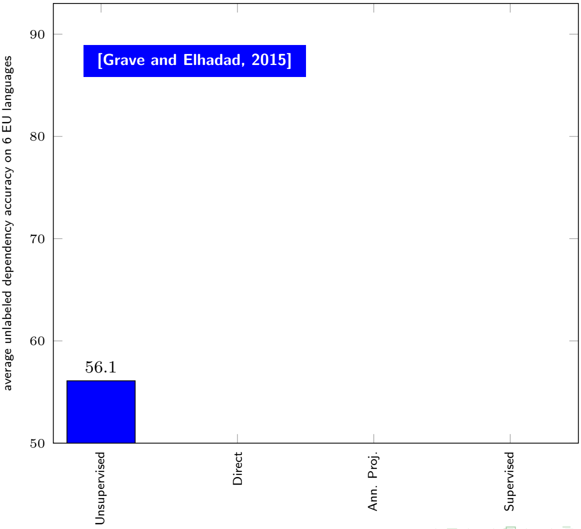
Discussion

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

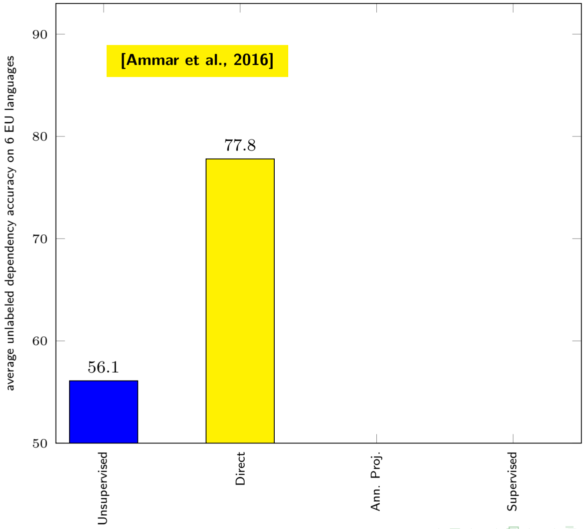
Discussion

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

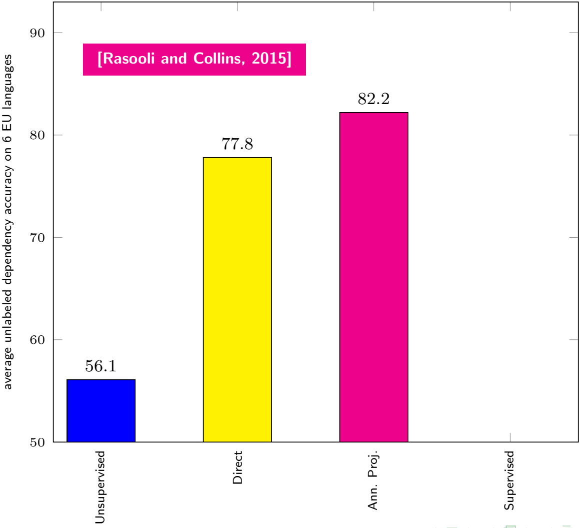
Unsupervised Parsing Best Models Comparison



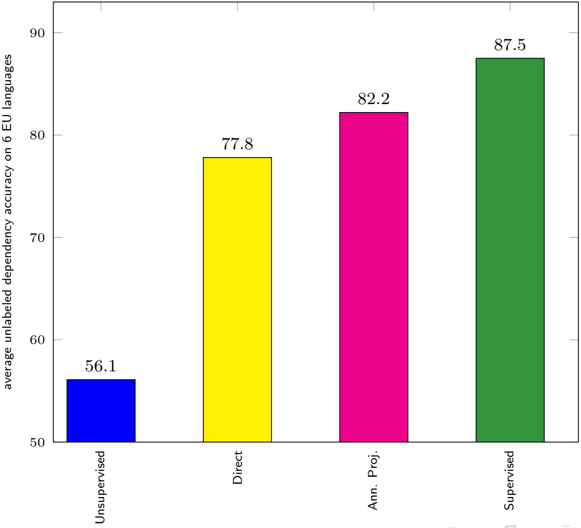
Unsupervised Parsing Best Models Comparison



Unsupervised Parsing Best Models Comparison



Unsupervised Parsing Best Models Comparison



Overview

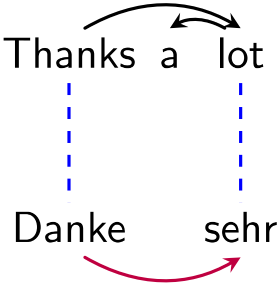
Conclusion

- 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



References I

Ammar, W., Mulcaire, G., Ballesteros, M., Dyer, C., and Smith, N. A. (2016).
One parser, many languages.
arXiv preprint arXiv:1602.01595.

Beal, M. J. (2003).
Variational algorithms for approximate Bayesian inference.
University of London London.

Blunsom, P. and Cohn, T. (2010).
Unsupervised induction of tree substitution grammars for dependency parsing.
In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1213, Cambridge, MA. Association for Computational Linguistics.

Carroll, G. and Charniak, E. (1992).
Two experiments on learning probabilistic dependency grammars from corpora.
Department of Computer Science, Univ.

References II

Cohen, S. B., Das, D., and Smith, N. A. (2011).
Unsupervised structure prediction with non-parallel multilingual guidance.
In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 50–61, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Cohen, S. B., Gimpel, K., and Smith, N. A. (2008).
Logistic normal priors for unsupervised probabilistic grammar induction.
In *Advances in Neural Information Processing Systems*, pages 321–328.

Cohen, S. B. and Smith, N. A. (2009a).
Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction.
In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 74–82, Stroudsburg, PA, USA. Association for Computational Linguistics.

References III

Cohen, S. B. and Smith, N. A. (2009b).
Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction.
In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82. Association for Computational Linguistics.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977).
Maximum likelihood from incomplete data via the em algorithm.
Journal of the royal statistical society. Series B (methodological), pages 1–38.

Dryer, M. S. and Haspelmath, M., editors (2013).
WALS Online.
Max Planck Institute for Evolutionary Anthropology, Leipzig.

Duong, L., Cohn, T., Bird, S., and Cook, P. (2015).
Cross-lingual transfer for unsupervised dependency parsing without parallel data.
In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 113–122, Beijing, China. Association for Computational Linguistics.

References IV

Durrett, G., Pauls, A., and Klein, D. (2012).
Syntactic transfer using a bilingual lexicon.
In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11, Jeju Island, Korea. Association for Computational Linguistics.

Ganchev, K., Gillenwater, J., and Taskar, B. (2009).
Dependency grammar induction via bitext projection constraints.
In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 369–377, Suntec, Singapore. Association for Computational Linguistics.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010).
Posterior regularization for structured latent variable models.
The Journal of Machine Learning Research, 11:2001–2049.

References V

Gillenwater, J., Ganchev, K., Graça, J., Pereira, F., and Taskar, B. (2010).
Sparsity in dependency grammar induction.
In *Proceedings of the ACL 2010 Conference Short Papers*, pages 194–199.
Association for Computational Linguistics.

Grave, E. and Elhadad, N. (2015).
A convex and feature-rich discriminative approach to dependency grammar induction.
In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1375–1384, Beijing, China.
Association for Computational Linguistics.

Guo, J., Che, W., Yarowsky, D., Wang, H., and Liu, T. (2016).
A representation learning framework for multi-source transfer parsing.
In *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix, Arizona, USA.

References VI

Headden III, W. P., Johnson, M., and McClosky, D. (2009).
Improving unsupervised dependency parsing with richer contexts and smoothing.
In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109, Boulder, Colorado. Association for Computational Linguistics.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005).
Bootstrapping parsers via syntactic projection across parallel texts.
Natural language engineering, 11(03):311–325.

Klein, D. and Manning, C. D. (2004).
Corpus-based induction of syntactic structure: Models of dependency and constituency.
In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

References VII

Lari, K. and Young, S. J. (1990).
The estimation of stochastic context-free grammars using the inside-outside algorithm.
Computer speech & language, 4(1):35–56.

Le, P. and Zuidema, W. (2015).
Unsupervised dependency parsing: Let's use supervised parsers.
In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 651–661, Denver, Colorado. Association for Computational Linguistics.

Ma, X. and Xia, F. (2014).
Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization.
In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348, Baltimore, Maryland. Association for Computational Linguistics.

References VIII

Mareček, D. and Straka, M. (2013).
Stop-probability estimates computed on a large corpus improve unsupervised dependency parsing.
In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 281–290, Sofia, Bulgaria. Association for Computational Linguistics.

McDonald, R., Petrov, S., and Hall, K. (2011).
Multi-source transfer of delexicalized dependency parsers.
In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Naseem, T., Chen, H., Barzilay, R., and Johnson, M. (2010).
Using universal linguistic knowledge to guide grammar induction.
In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244, Cambridge, MA. Association for Computational Linguistics.

References IX

Neal, R. M. and Hinton, G. E. (1998).
A view of the em algorithm that justifies incremental, sparse, and other variants.
In *Learning in graphical models*, pages 355–368. Springer.

Paskin, M. A. (2002).
Grammatical digrams.
Advances in Neural Information Processing Systems, 14(1):91–97.

Rasooli, M. S. and Collins, M. (2015).
Density-driven cross-lingual transfer of dependency parsers.
In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 328–338, Lisbon, Portugal. Association for Computational Linguistics.

Smith, N. A. and Eisner, J. (2005).
Guiding unsupervised grammar induction using contrastive estimation.
In *Proceedings of IJCAI Workshop on Grammatical Inference Applications*, pages 73–82.

References X

Spitkovsky, V. I., Alshawi, H., Chang, A. X., and Jurafsky, D. (2011).
Unsupervised dependency parsing without gold part-of-speech tags.
In *Proceedings of the conference on empirical methods in natural language processing*, pages 1281–1290. Association for Computational Linguistics.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. (2012).
Bootstrapping dependency grammar inducers from incomplete sentence fragments via austere models.
In *ICGI*, pages 189–194.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. (2013).
Breaking out of local optima with count transforms and model recombination: A study in grammar induction.
In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1983–1995, Seattle, Washington, USA. Association for Computational Linguistics.

Täckström, O., McDonald, R., and Nivre, J. (2013).
Target language adaptation of discriminative transfer parsers.
Transactions for ACL.

References XI

Tesnière, L. (1959).
Eléments de syntaxe structurale.
Librairie C. Klincksieck.

Tiedemann, J., Agić, v., and Nivre, J. (2014).
Treebank translation for cross-lingual parser induction.
In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140, Ann Arbor, Michigan. Association for Computational Linguistics.

Tu, K. and Honavar, V. (2012).
Unambiguity regularization for unsupervised learning of probabilistic grammars.
In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1324–1334. Association for Computational Linguistics.

References XII

Xiao, M. and Guo, Y. (2015).
Annotation projection-based representation learning for cross-lingual dependency parsing.
In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 73–82, Beijing, China. Association for Computational Linguistics.

Zhang, Y. and Barzilay, R. (2015).
Hierarchical low-rank tensors for multilingual transfer parsing.
In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal.