Motivation
Almost all previous work on unsupervised dependency parsing focuses on learning generative models. We propose an discriminative and tractable model for this task.

Proposed Solution
We develop an unsupervised dependency parsing model based on the CRF autoencoder. We propose an exact algorithm for parsing as well as a tractable learning algorithm.

Model
Our model based on CRF autoencoder consists of two part:
- Encoder: a log-linear model represented by a first-order dependency parser. The score of a dependency tree can be factorized as the sum of scores of its dependencies. The encoder models the distribution \( P(y|x) \).
- Decoder: a set of categorial conditional distributions \( \theta_{yt} \), which represents the probability of generating token \( t \) conditioned on the token of its parent \( y \) given on its parse tree \( x \). The decoder models the distribution \( P(\hat{x}|y) \).

The conditional distribution of \( \hat{x}, y \) given \( x \) is modeled by the whole model.

\[
P(y, \hat{x}|x) = P(y|x)P(\hat{x}|y)
\]

Training
The goal of the training procedure is to maximize the Vertibi-like conditional reconstruction probability.

Our objective function is

\[
J(w, \theta) = - \sum_{i=1}^{N} \log \left( \max_{y \in \mathcal{Y}(x)} P(\hat{x},y|x)Q^{\alpha}(x, y) \right) + \lambda \Omega(w)
\]

- \( \lambda \): a hyperparameter controlling the strength of regularization
- \( \Omega(w) \): the regularization term of the encoder parameter \( w \)
- \( \alpha \): a hyperparameter controlling the strength of the constraint factor
- \( Q(x, y) \): a soft constraint factor over the parse tree

The purpose of adding the soft constraint factor \( Q(x, y) \) to the objective function is to encourage learning of dependency relations that satisfy universal linguistic knowledge. The constraint factor \( Q(x, y) \) is calculated based on the universal syntactic rules following Naseem et al. (2010) and Grave et al. (2015).

We apply coordinate descent to minimize the objective function, which alternately updates parameters of encoder \( w \) and parameters of decoder \( \theta \).

- In each optimization step of \( w \), we run several epochs of stochastic gradient descent,
- In each optimization step of \( \theta \), we run several iterations of the Viterbi EM algorithm.

Results on Seven Other Languages
We evaluated our model on seven languages from the PASCAL Challenge on Grammar Induction.

<table>
<thead>
<tr>
<th>Language</th>
<th>Basque</th>
<th>Catalan</th>
<th>Danish</th>
<th>Dutch</th>
<th>English</th>
<th>Portuguese</th>
<th>Spanish</th>
<th>Turkish</th>
<th>Urdu</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMV(EVM)</td>
<td>41.1</td>
<td>31.3</td>
<td>30.8</td>
<td>37.1</td>
<td>36.7</td>
<td>36.7</td>
<td>43.5</td>
<td>43.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMV(Viterbi)</td>
<td>46.5</td>
<td>33.1</td>
<td>35.6</td>
<td>45.0</td>
<td>30.4</td>
<td>42.2</td>
<td>44.3</td>
<td>43.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural DMV (EM)</td>
<td>48.1</td>
<td>32.6</td>
<td>39.8</td>
<td>37.2</td>
<td>36.5</td>
<td>39.9</td>
<td>47.9</td>
<td>39.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural DMV (Viterbi)</td>
<td>49.4</td>
<td>36.3</td>
<td>49.7</td>
<td>41.3</td>
<td>46.4</td>
<td>43.7</td>
<td>38.5</td>
<td>39.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convex-MST (No Prior)</td>
<td>50.1</td>
<td>41.1</td>
<td>51.6</td>
<td>55.5</td>
<td>55.4</td>
<td>63.7</td>
<td>50.9</td>
<td>47.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convex-MST (With Prior)</td>
<td>52.0</td>
<td>43.9</td>
<td>52.8</td>
<td>59.3</td>
<td>47.6</td>
<td>54.7</td>
<td>51.3</td>
<td>40.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRFAE (No Prior)</td>
<td>49.0</td>
<td>39.1</td>
<td>35.0</td>
<td>37.9</td>
<td>28.8</td>
<td>43.5</td>
<td>32.5</td>
<td>34.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRFAE (With Prior)</td>
<td>50.9</td>
<td>42.1</td>
<td>36.2</td>
<td>47.9</td>
<td>68.8</td>
<td>62.5</td>
<td>64.7</td>
<td>54.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Parsing accuracy on seven languages. Our model is compared with DMV, Neural DMV, and Convex-MST.

Future work
- Using higher order dependency models as the encoder.
- Introducing neural networks into the encoder.