Making Latent SVM\textsuperscript{struct} Practical for Coreference Resolution
Iryna Haponchyk\textsuperscript{1}  Alessandro Moschitti\textsuperscript{2,1}
\textsuperscript{1}Department of Information Engineering and Computer Science, University of Trento,
\textsuperscript{2}Qatar Computing Research Institute
iryna.haponchyk@unitn.it amoschitti@gmail.com

Motivation
The complex task of Coreference Resolution (CR) has recently been approached by learning algorithms with structured output.
Finding optimal mention clusters is NP-hard. \( \rightarrow \) CR has been reformulated as a spanning graph problem.
1. The system of Fernandes et al. (2012) (latent trees + structured perceptron) delivered the state-of-the-art results in the CoNLL-2012 Shared Task.
2. Earlier, in 2009, Yu and Joachims proposed their Latent SVM\textsuperscript{struct} (LSVM) approach which learns to predict graph structures in a similar manner.
• SVMs are known to generally provide higher accuracy than a perceptron!
• LSVM has not been applied to the CoNLL data.
• There is no previous work on a comparison of the two methods.

Structured Perceptron vs. SVM\textsuperscript{struct}
Target: \( f : X \times Y \rightarrow \mathbb{R}, f(x, y) = \langle w, \Phi(x, y) \rangle \) \( \text{argmax}_{y \in Y} f(x, y) \)
• Structured Perceptron (Collins (2002))
\( \{ (x_i, y_i) \}_{i=1,...,J} : \)
\[ \hat{y} = \text{argmax}_{y \in Y} f(x_i, y) \]
\[ w \leftarrow w + \phi(x_i, y_i) - \phi(x_i, \hat{y}) \]
• Structural SVMs (Tschantaridis et al. (2004))
\[ \min_{w} \frac{1}{2} \|w\|^2 \]
\[ s.t. \forall i, \forall y \in Y \setminus \hat{y}, (w, \Phi(x_i, y) - \Phi(x_i, \hat{y})) \geq 1 \]

Latent SVM\textsuperscript{struct} of Yu and Joachims (2009)
• For each document, the authors construct an undirected graph of mention pair relations.
• The learning involves running Kruskal's spanning algorithm for finding a maximum spanning forest. Each connected component corresponds to a cluster.
• There is no one-to-one correspondence between clusterings and spanning forests.
\( (x, y, h) \) – input/output example
• latent variable \( h \) – spanning forest
\[ \Phi(x, y, h) = \sum_{e|h} \phi(e) \]
• In the LSVM implementation\textsuperscript{a}, the inference is done on a fully-connected graph.
• However, a large portion of mention pair links do not convey significant information, e.g., very distant mentions are very improbable to corefer.
• Preliminary filtering has been a common practice in coreference research, e.g., Fernandes et al. (2012).

Latent Perceptron of Fernandes et al. (2012)
• CR is formulated as a spanning tree problem on a directed candidate graph with an additional root node.
• Edmonds' algorithm is used for finding a maximum spanning tree. The subtrees directly connected to the root node of that tree form clusters.
• The trees are implicit (latent) in data, this modelling is incorporated into a latent perceptron framework.
• For each node, Edmonds' algorithm chooses the best incoming edge, i.e. the best antecedent for each mention. This strategy fits the nature of the CR task very well.

Experiments
• Dataset: corpus from the CoNLL 2012–Shared Task – English part
• Evaluation: MELA score – versions 4 and 7 of the official CoNLL scorer
• Features: BART\textsuperscript{\textsuperscript{b}} and some Fernandes et al. features

<table>
<thead>
<tr>
<th>Scorer Version</th>
<th>All edges</th>
<th>Filtered edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original LSVM</td>
<td>59.61</td>
<td>55.19</td>
</tr>
<tr>
<td>Modified LSVM</td>
<td>59.61</td>
<td>55.19</td>
</tr>
<tr>
<td>Latent Perceptron</td>
<td>61.21</td>
<td>57.64</td>
</tr>
<tr>
<td>Latent Perceptron+Kruskal\textsuperscript{c}</td>
<td>58.79</td>
<td>56.69</td>
</tr>
</tbody>
</table>

Table 1: Performance on the test set.

<table>
<thead>
<tr>
<th>Scorer Version</th>
<th>v4</th>
<th>v7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latent Perceptron</td>
<td>61.37</td>
<td>58.33</td>
</tr>
<tr>
<td>Latent Perceptron+Kruskal\textsuperscript{c}</td>
<td>58.79</td>
<td>55.43</td>
</tr>
<tr>
<td>Modified LSVM</td>
<td>59.51</td>
<td>56.22</td>
</tr>
</tbody>
</table>

Table 2: Performance on the development set.

http://ikernels-portal.disi.unitn.it

Summary
• We have performed a comparative analysis of the structured prediction frameworks for CR.
  ✓ The graph modelling of Fernandes et al. and Edmonds' spanning algorithm seem to tackle the task more specifically.
  ✓ Future work: we intend to verify if LSVM benefits from using Edmonds' algorithm.
• We have extended the LSVM implementation to partial graphs which facilitates
  combining the framework with different filtering strategies,
  comparison with other systems.

http://ikernels-portal.disi.unitn.it/