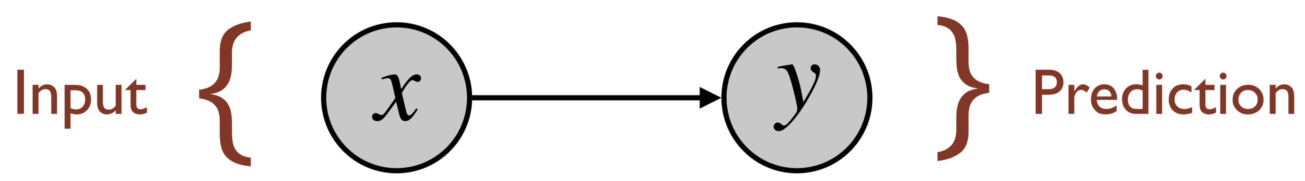
Learning Latent Trees with Stochastic Perturbations and Differentiable Dynamic Programming

Caio Corro, Ivan Titov

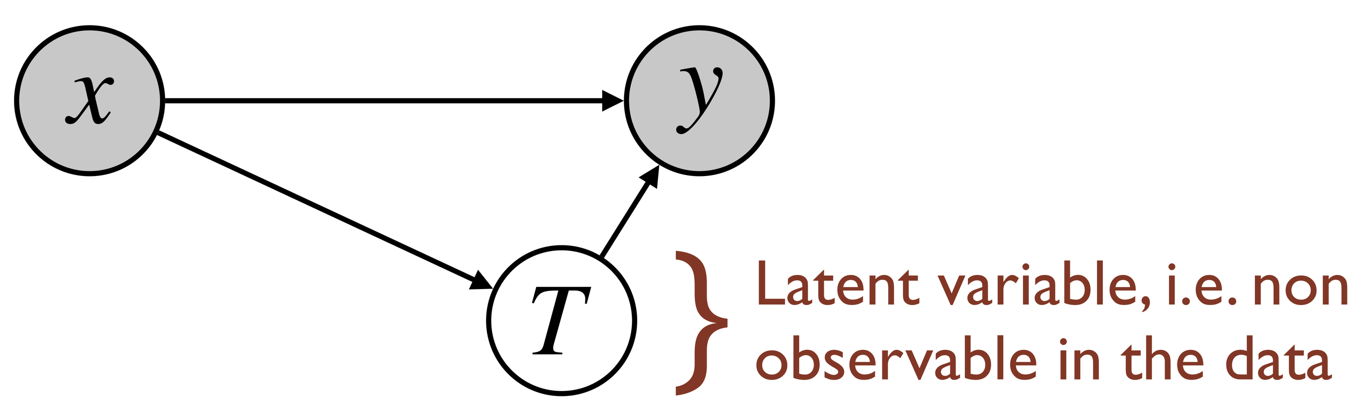
ILCC, School of Informatics, University of Edinburgh   ILLC, University of Amsterdam

Latent Variable Models

Supervised learning can be understood as inferring the probability distribution corresponding to a directed graphical model.

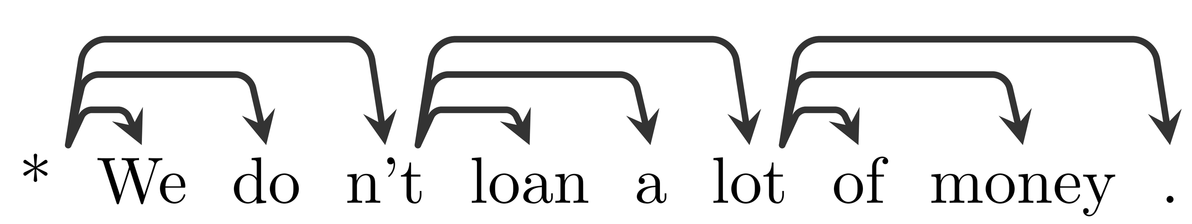


Latent variables can model unobserved inter-dependencies or introduce knowledge about the structure of a given problem.



Projective Dependency Tree

We are interested in **latent projective dependency trees** that implicitly encode hierarchical decomposition of a sentence into spans.



Distribution over Trees

The probability distribution over dependency trees is a log-linear model factored over arc weights.

W : matrix of arc weights computed with a NN

T : boolean adjacency matrix,
i.e $T_{h,m} = 1$ iff arc $x_h \rightarrow x_m$ is in the tree

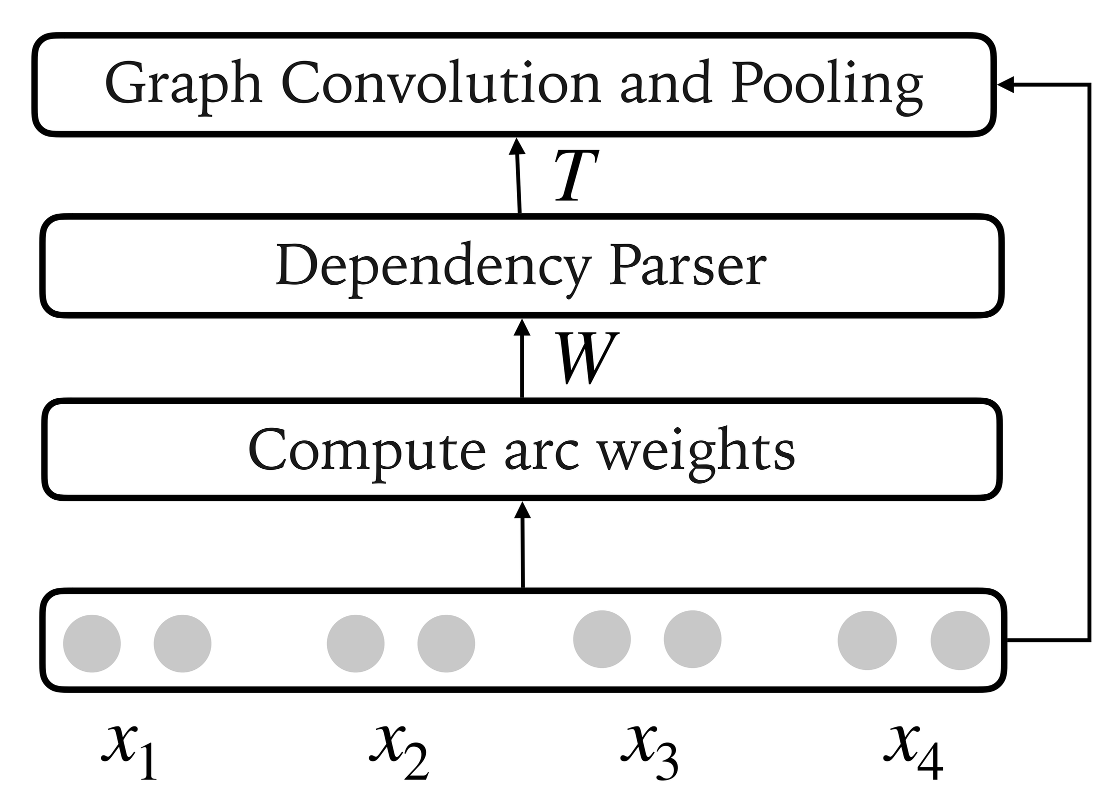
$$p(T|x) = \frac{\sum_{h,m} T_{h,m} \times W_{h,m}}{Z(T,x)}$$

Contributions

1. We show that a **latent tree model** can be estimated by drawing global approximate samples via **Gumbel perturbation and differentiable dynamic programming**
2. We demonstrate that constraining the structures to be projective dependency trees is beneficial
3. We show the effectiveness of our approach on two standard tasks and on a synthetic dataset

Neural Architecture

A **Graph Convolutional Network** [Kipf and Welling, 2017] is used to compute the sentence representation w.r.t. the dependency tree.



Training Loss

We maximise the likelihood of training data via SGD:

$$\max \sum_{x,y} p(y|x) \quad \Big\} \text{ T does not appear here}$$

where:

$$\log\ p(y|x) = \log\ \mathbb{E}_{T\sim p(T|x)}[\ p(y|T,x)\ ]$$

Unfortunately, exact marginalisation is intractable:

$$= \log \sum_{T} p(T|x) \times p(y|T,x)$$

Therefore, we derive a bound using Jensen's inequality:

$$\geq \mathbb{E}_{T\sim p(T|x)}[\ \log\ p(y|T,x)\ ]$$

Which can be approximated via Monte-Carlo method.

Perturb-and-MAP

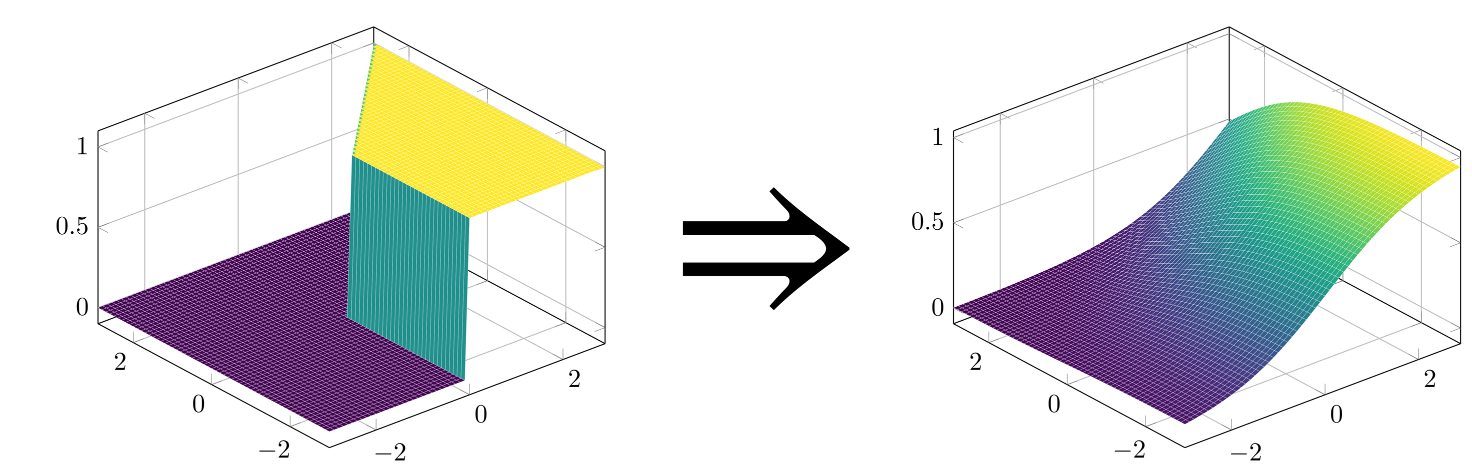
Approximate sampling method for log-linear models:

$$\left.\begin{array}{l} G \sim \mathcal{G}(0,1) \\ \widetilde{W} = W + G \end{array}\right\} \text{Arc weight perturbation with Gumbel noise [Papandreou \& Yuille, 2011]}$$

$$\left.\underset{T\in\mathcal{T}(s)}{argmax} \sum_{h,m} T_{h,m} \times \widetilde{W}_{h,m}\right\} \text{Solved with dynamic programming [Eisner, 1996]}$$

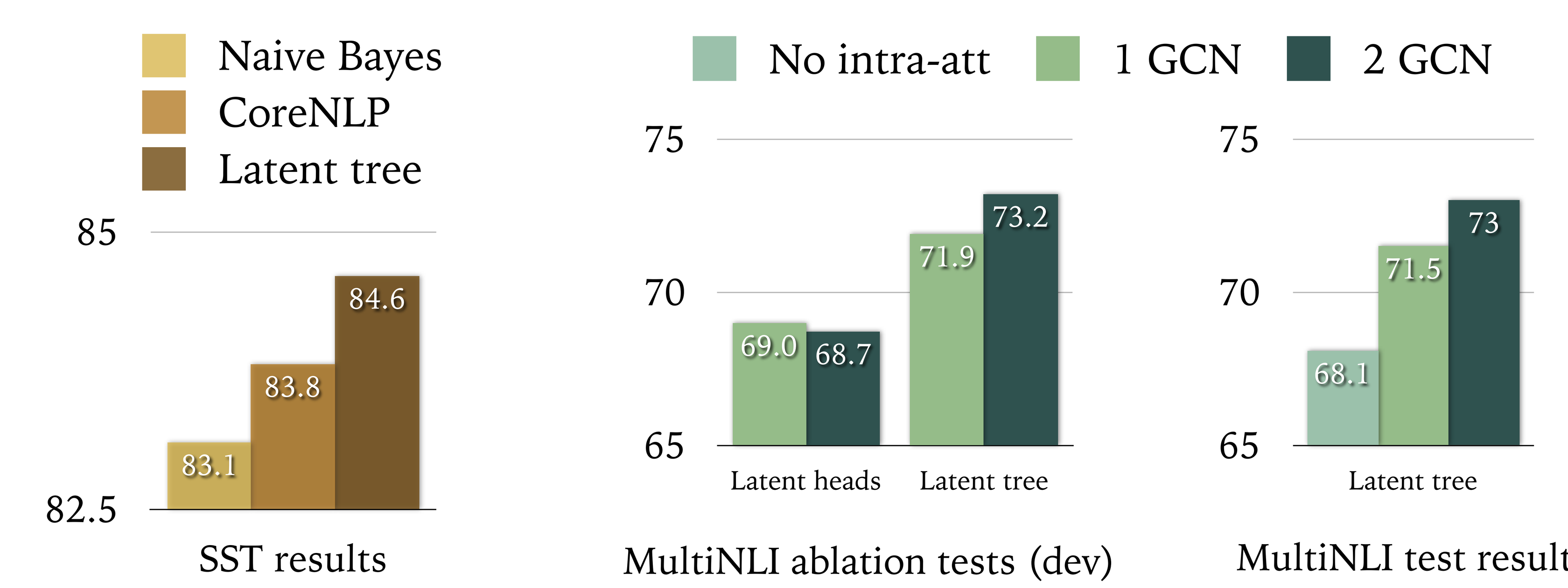
Differentiable Dynamic Programming

The dynamic programming approach for parsing relies on recursive calls to the *one-hot-argmax* op, which introduces ill-defined derivatives during the backward pass. We replace *one-hot-argmax* ops with *softmax* ops to smooth the optimization landscape.



Experimental Results

Experimentally, we observe that our **Latent Tree (LT)** model improves comparable baselines on **sentiment analysis** with syntactic trees predicted by CoreNLP and on **Natural Language Inference** datasets.



Acknowledgments

NWO VIDI 639.022.518
ERC BroadSem 678254