Tree-structured decoding with doubly-recurrent neural networks

David Alvarez-Melis & Tommi S. Jaakkola
MIT CSAIL
(davidam,tommi)@csail.mit.edu

Summary

We propose a novel neural network architecture specifically tailored to tree-structured decoding, which:
- maintains separate depth and width recurrent states and combines them to obtain hidden states for every node in the tree.
- has a mechanism to predict tree topology explicitly (as opposed to implicitly by adding nodes with special tokens).

Our experiments show that this architecture:
- is capable of recovering trees from encoded representations
- achieves state-of-the-art performance in a task consisting of mapping sentences to simple functional programs
- exhibits desirable invariance properties over sequential architectures

Why tree-structured?

- RNNs are a natural model for sequential data
- But many types of data are non-sequential, e.g.
  - natural language sentences or associated parse trees
  - programs, executable queries, etc.
- Even sentences, which can be modeled as if they were linear sequences, have an underlying compositional process.

Previous work

Current neural architectures for non-sequential data usually assume:
- a) the full tree structure is given (e.g. [5, 6]), or
- b) at least the nodes are known (e.g. [1, 3])

In case (a), the network aggregates the node information in a manner that is coherent with a given tree structure. In case (b), generation is reduced to an attachment problem, i.e., sequentially deciding which pairs of nodes to join with an edge until a tree is formed.

Full decoding with structure is much less explored. Models so far remained relatively close to their sequential counterparts, e.g. using alternating RNNs coupled with external classifiers to predict branching [7] and introducing special tokens [2] to signal stopping.

Two downsides to using special tokens to control topology are:
- (i) tree growth (up to O(n^2) padding nodes in an n-node tree)
- (ii) single stopping token selected competitively with other tokens

Challenges of tree-structured decoding

As opposed to seq-to-seq, encoding and decoding are intrinsically asymmetrical. Decoding requires multiple design choices:
- In which order should the tree be generated?
- What information should each node receive? Parent, sibling(s), etc.
- How to terminate generation?

Our approach

Grow trees root-to-Leaves, encode parent-to-child and sibling-to-sibling information in separate recurrent states and model topological (stopping) decisions explicitly with a dedicated module

Tree 1: Left: A cell in the DRNN corresponding to node i with parent p and sibling s. Right: Structure-unrolled DRNN network in an encoder-decoder setting. Solid (dashed) lines indicate ancestral (fraternal) connections. Crossed arrows indicate production halted by the topology module.

Doubly-Recurrent Neural Networks

Cell recurrent states

\[ h_{i}^{\text{pred}} = \sigma( h_{i}^{\text{pred}} + \sigma( h_{p}^{\text{pred}} + \alpha_{i} v_{p}^{\text{pred}} + \alpha_{i} v_{i}^{\text{pred}})) \]

These are combined to obtain a predictive hidden state:

\[ h_{i}^{\text{pred}} = \text{tanh}(W_{h} h_{i}^{\text{pred}} + b_{h}^{\text{pred}}) \]

Training DRNNs

- With (reverse) back-propagation through structure (BPTS)
- Forward pass: top-down, on the structure-unrolled network
- Backward pass: bottom-up, feeding into every node gradients from children and sibling, computing internally gradients with respect to both topology and label prediction.
- Two loss terms: label and topology prediction

Topological Prediction

Instead of using stopping tokens, our model makes topological decisions explicitly, by computing:

\[ p_{i}^{\text{pred}} = \sigma( u_{i}^{\text{pred}}) \]

where \( p_{i}^{\text{pred}} \in [0,1] \) is interpreted as the probability that node i has children. Analogously, the probability of stopping at level \( j \) is:

\[ p_{j}^{\text{pred}} = \sigma( u_{j}^{\text{pred}}) \]

Topological decisions \( \alpha_{i}, v_{i} \in [0,1] \) are included for label prediction:

\[ \alpha_{i} \in \text{softmax}(W_{h} h_{i}^{\text{pred}} + \alpha_{i} v_{p}^{\text{pred}} + \alpha_{i} v_{i}^{\text{pred}}) \]

In practice, during training, we perform teacher forcing, replacing topological predictions \( p_{i}^{\text{pred}}, p_{j}^{\text{pred}} \) for true values \( (\alpha_{i}, v_{i}) \) after computing loss and before computing \( \alpha_{i} \).

Synthetic tree recovery

Task: Recovering tree structure from flattened (encoding) representations

Dataset: 5000 trees labeled with letters A-Z. We generate trees in a top-down fashion, conditioning every node’s label and topology on the state of its ancestors and siblings.

Model: A DRNN as decoder, paired with a (sequential) RNN as encoder.

Evaluation: To give partial credit to correct substructures, we use an IR approach to evaluation, measuring F1-score of node and edge recovery.

Experiments

Synthetic tree recovery

Train

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Tree Width (# nodes)</th>
<th>Log-Likelihood relative change (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRNN</td>
<td>Small</td>
<td>12</td>
<td>70.1 51.2 62.7</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>12</td>
<td>78.8 50.5 60.4</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>Small</td>
<td>12</td>
<td>90.1 70.5 70.5</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>12</td>
<td>89.7 70.5 70.5</td>
</tr>
<tr>
<td>DRNN</td>
<td>Small</td>
<td>12</td>
<td>90.1 70.5 70.5</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>12</td>
<td>89.7 70.5 70.5</td>
</tr>
</tbody>
</table>

Table 1: Results on the IFTTT task. Left: Non-English/untagged removed, Right: at least 3 human-agreed gold recipes. (758 recipes)

Machine Translation

Can decoding with structure bring benefits to a task traditionally approached as a sequence-to-sequence problem, such as MT?

Training data: 50K En->Fr sentences from the WMT14 dataset.

Models:
- DRNN: L/R children distinction, paired w/ LSTM encoder
- Seq2Seq: LSTM units, roughly same # of params as DRNN

Evaluation:
- i) Insensitivity to structural perturbations in output, measuring A in LL
- ii) Quality of translations at different resolutions (max target “size”)

Selected References