SD214 : Reading Report
Do latent tree learning models identify meaningful structure in sentences?[1]

by Anna van Elst

I. Highlight and explanation of the main contributions of the paper

Latent tree learning allows neural networks to be trained to parse a sentence, that is, to break it down into its constituent components, and then use the resultant parse to understand the sentence for tasks such as sentiment analysis, textual entailment, and translation. This type of model is trained without the use of existing parse trees, which are rooted trees that reflect the syntactic structure of a string using a context-free grammar.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman[1] discovered interesting findings on the performance, consistency, and grammar of latent tree learning models. They trained two latent tree learning models, RL-SPINN and ST-Gumbel. These models have been proven to outperform similar models based on supervised parsing, that is, models using already existing parse trees. ST-Gumbel is an abbreviation for the Straight-Through Gumbel-Softmax estimator, whereas RL-SPINN is an abbreviation for Shift-reduce ParserInterpreter Neural Network with Reinforcement Learning.

Both of these models were trained and tested for textual entailment using the SNLI and MultiNLI corpus. Textual entailment is the task of determining the relationship between two text fragments called “text” and “hypothesis”: text can entail hypothesis, contradict hypothesis or be neutral. The SNLI and MultiNLI corpus contain thousands of sentence pairs annotated with textual entailment information (labels can be entailment, contradiction and neutral).

Here are the key findings. First of all, both models produce viable sentence representations, but only ST-Gumbel beats standard tree-structured models on sentence classification, such as a similar TreeRNN baseline and basic LSTM RNN. Moreover, ST-Gumbel grammar is inconsistent across random restarts. However, there are certain regularities in the resultant grammar, such as a predilection for shallow trees (compared to PTB parses), a fairly systematic handling of negation, and a predisposition to regard pairs of neighboring words at the borders of a sentence as constituents.

Let’s take a closer look at the outcomes for each model. While RL-SPINN performs comparably to the TreeLSTM baseline but falls short of the LSTM RNN baseline, ST-Gumbel outperforms both baselines.

TABLE I
Performance of RL-SPINN and ST-Gumbel compared to baseline models

Model	SNLI	MultiNLI
RL-SPINN 300D	82.3	67.4
ST-Gumbel 300D	83.3	69.5
TreeLSTM 300D	82.2	67.5
LSTM 300D	82.6	69.1

Let us now examine the consistency and structure of the grammar in both models. RL-SPINN generates trees that are remarkably consistent across random restarts. The created trees are quite consistent with left-branching trees (a score of almost 100%!) but not with right-branching trees (less than random and baseline trees) and not with PTB-style trees, also known as Penn TreeBank trees. ST-Gumbel, on the other hand, develops trees that are more likely to branch to the right than baseline. They are also incompatible with PTB-style trees while being consistent with left-branching trees. The baseline models is very consistent with PTB-style trees, as it reaches a score above 70%.

TABLE II
F1 scores of RL-SPINN and ST-Gumbel compared to baseline models

Model	L-Branch	R-Branch	Stanford PTB Parser
RL-SPINN 300D	99.8	11.1	18.2
ST-Gumbel 300D	35.6	40.3	25.2
TreeLSTM 300D	19.2	38.2	73.1
Random Trees	27.9	28.1	27.1

In terms of qualitative findings, the top RL-SPINN trials use only left-branching parses, which explains why the resulting trees are inconsistent with PTB — because English favors right-branching trees. The model is analogous to a sequential RNN.

ST-Gumbel trees, on the contrary, are balanced and shallow — that is, of a small depth. Moreover, the first and last words are almost always components. Another interesting observation is that ST-Gumbel parses are inconsistent when it comes to prepositions and determiner attachments. For example, "The students reacted with horror." would be divided into "The students", "reacted", "with", and "horror." although it is divided into "The students", "reacted", "with horror," and "." in PTB. Additionally, negation is consistently associated with the right-neighbor, as in "it", "’", "not predictable," ". ", while PTB gives "It’, "’s not," "predictable," ". " Although ST-Gumbel trees are less informative on sentence structure, their shallow depth reduces the number of layers a word must pass through to reach the classifier, ensuring that word information is not dissolved by multiple compositions and may make learning a suitable composition function easier.

The conclusion of this research is that latent tree learning models that enhance downstream tasks like textual entailment do not always create the most grammatical syntactic trees. So, why do they continue to enhance performance? This article provides no actual explanation, and further research will be required to build new coding structures that take advantage of the syntactic tree characteristics provided in this research.

There are two alternatives. If PTB grammar rules aren’t always the best for natural language inference (NLI) tasks, the question becomes, what kind of grammar rules are needed for NLI tasks? If the PTB grammar rules are important for improving the performance of downstream tasks, how may this structure be extracted?

II. An analysis of the advantages and issues of the proposed method

The article’s most interesting contribution is that it evaluated the quality of the trees created by the latent tree learning models both quantitatively and qualitatively, which has reportedly never been done previously. The objective is to find out if the models learn consistent and principled latent grammars. To answer this point, they examined the f1 scores in relation to the Stanford Parser trees to see if they were compatible with the PTB grammar.

Their method for analyzing the induced trees was indeed valuable for other researchers, such as Jean Maillard and Stephen Clark[2], who explored the tree structures produced by their novel model in the same way. These researchers obtained the same f1 scores for the identical task and data, confirming the study's conclusion: the trained trees do not reflect PTB grammar.

The metrics used, namely the f1 scores, are appropriate and provided meaningful insights into the structure of the trained trees. However, I noticed that they only used five runs of each model to evaluate some metrics like mean and standard deviation of f1 and tree depth. This surprised me because I believe 5 runs are insufficient and perhaps more runs should have been used to compute these metrics.

Furthermore, by displaying the results for both SNLI and MultiNLI corpus, they demonstrated the relevance of the choice of training data used for latent tree learning models, even though the downstream task is the same. This was crucial since the accuracy measures revealed a considerable variation in performance scores. In the example of 300D ST-Gumbel, we find 83.3% accuracy for the SNLI corpus and 69.5% accuracy for the MultiNLI, a difference of more than 10%. This can be explained by the fact that MultiNLI, which comprises longer sentences from multiple genres on average, may be impeding the capacity of existing models to learn consistent grammars.

Finally, the qualitative analysis of the learned trees for the trained models was relevant, specifically the negation, the Initial and Final Two-Word Constituents, Function Words and Modifiers, and the tree types (left and right-branching) because it illustrates that, even if the induced grammar does not match PTB grammar, the models still have some form of strategy for generating trees.

III. Your personal take, the contribution, the relevance of the paper and how it could be improved on

This paper was interesting to me because it presented a basic analysis of the grammar induced by latent tree learning models as well as directions for future research. Their problem is well-motivated and has the potential to produce meaningful results. While I thought it was interesting that the authors focused on textual entailment, I think it's unfortunate that they didn't focus on other tasks like sentiment analysis and translation. It would have been interesting to compare the performance of both models on various tasks. The classifier, according to the authors, performs better on balanced and shallow trees (ST-Gumbel), but this may not be true for other tasks. It might be effective for textual entailment because this particular task necessitates less semantic and syntactic understanding.

IV. Your understanding of the related works and impact of the paper

As I aforementioned, this research paper contributed to the improvement of the quantitative and qualitative analysis of induced trees as some researchers[2] reused their approach for assessing induced trees because it was relatively new in the domain. Some researchers also implemented the latent tree models proposed in this paper as a baseline to assess the performance of their new model.

Another study[3], also co-authored by Samuel R.

Bowman, looked into the parsing of latent tree learning models. Bowman and another researcher, Nikita Nangia, demonstrated that current latent tree learning models perform worse in sentence understanding than purely sequential RNNs. They made a new dataset, ListOps, which is composed of sequences in the style of prefix arithmetic. The toy dataset is intended to have a single correct parsing strategy that a system must learn in order to complete the task. They proved that the latent tree learning models performed poorly on this new dataset because they were unable to parse the sequences and use that parse to build the sentence representation.

Overall, I believe that this research paper had little influence in the area of textual entailment and latent tree learning because it did not propose a novel model that significantly improves performance scores and it did not provide a simple explanation of induced tree structure.

V. References

[1] **Adina Williams, Andrew Drozdov, and Samuel R. Bowman**, 2017, Do latent tree learning models identify meaningful structure in sentences?

[2] **Jean Maillard and Stephen Clark**, 2018, Latent Tree Learning with Differentiable Parsers: Shift-Reduce Parsing and Chart Parsing

[3] **Nikita Nangia and Samuel R. Bowman**, 2018, ListOps: A Diagnostic Dataset for Latent Tree Learning