Probing syntactic hierarchies in internal representations of neural language models.

Tim Ottens
University of Amsterdam
t_im1996@live.nl

Thomas van Osch
University of Amsterdam
t.vanosch@hotmail.com

Abstract

Prior studies showed neural language models to be capable of incorporating representations of linguistic phenomena in their internal states. However, a direct comparison between state-of-the-art language models is lacking. We propose probing two linguistic tasks on a Transformer and a recurrent neural network (RNN) to evaluate their ability to capture syntactic hierarchies in the hidden representations. With the inclusion of control tasks based on selectivity (Hewitt and Liang, 2019), the Transformer architectures are notably favored over the RNN, suggesting improved structural hierarchies embedded in the hidden representation of the Transformer.

1 Introduction

In natural language processing, deep learning has influenced not only the usage of deep models but also the tools for NLP, such as deep contextual embeddings. To attain more interpretability into the learning process and the internal representations of these increasingly complex models, an external supervised model is built upon the original model to interpret these hidden representations. This is also known as a probing task (Conneau et al., 2018) or diagnostic classification (Hupkes et al., 2018). Although Hewitt and Manning (2019) found convincing evidence of parse trees being encoded in internal representations of Transformer models, no investigation was performed on recurrent neural networks (RNN). Comparisons between these types of architectures have been studied on syntactic phenomena (Peters et al., 2018b), but their work did not consider complete structural parse trees encodings or governing the role of the probing model by employing control tasks. Moreover, since Peters et al. (2018b), additional novel Transformers have been proposed. Our work aims to investigate to what extent parse trees and part-of-speech information are structurally encoded in the internal representations of autoregressive Transformers and RNN models and subsequently evaluating these experiments by employing control tasks (Hewitt and Liang, 2019).

2 Related Work

Prior studies emphasized the applicability of probing models. Linzen et al. (2016) studied the existence of subject-verb incongruity agreement encoded in the internal representation of an LSTM. Moreover, an LSTM has also shown its ability to capture morphological and part-of-speech tags (Belinkov et al., 2017) and sentence-level linguistic information (Conneau et al., 2018) in its encodings. Søgaard and Goldberg (2016) modeled the importance of linguistic hierarchies captured by LSTMs in automatic POS-tagging, chunking, dependency parsing and CCG super tagging.

Recently, attention-based language models - Transformers (Vaswani et al., 2017) - have achieved state-of-the-art performance on NLP downstream tasks, specifically ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2018). Variants of these improved parallelizable models have shown their power in various domains of NLP, such as machine translation and language modeling, achieving competitive results to the LSTM. Consequently, probing tasks have been applied to Transformer architectures to reveal their incorporation of linguistic phenomena in the encoded representation, such as parse trees (Hewitt and Manning, 2019), part-of-speech tagging, coreference and other syntactic relationships (Peters et al., 2018b).

Our work extends prior developments in terms of comparing the syntactic representation hidden in the internal encodings of both an LSTM and a Transformer model. Unlike Peters et al. (2018b), parse trees are also considered and more recent Transformer models are evaluated, also by control tasks.

3 Methods

In this section, three sub-tasks are formulated that contribute to answering our main research question, that is, to what extent a Transformer architecture and an LSTM model are able in developing hierarchical representations based on part-of-speech (POS) tagging and dependency parsing.

3.1 Part-of-speech tagging

One syntactic linguistic phenomenon, which depends on the hierarchy, is the task of assigning the correct POS-tag to a given word or word group. By employing a diagnostic linear classifier as the probing model, the encodings of the word are mapped to a specific part-of-speech tag. This simple probing model should be sufficient in finding a linear mapping based on the internal hierarchical representations of the models. To govern the probing model of learning the task itself, control tasks are introduced to prevent this (Hewitt and Liang, 2019).

3.2 Structural dependency

Another task that tests this hierarchical representation in the neural language models is the structural dependency tree task. To test the presence of parse trees, a projection matrix B is learned to map the squared distances in the hidden representations $\mathbf{h}_\mathrm{i}^\ell$ to a tree distance structure in the following way:

$$d_B\left(\mathbf{h}_i^\ell, \mathbf{h}_j^\ell\right)^2 = \left(B\left(\mathbf{h}_i^\ell - \mathbf{h}_j^\ell\right)\right)^T \left(B\left(\mathbf{h}_i^\ell - \mathbf{h}_j^\ell\right)\right) \tag{1}$$

So the predicted distances, as a result of projection matrix B, are d_B. To define a loss function on which the projection B is trained, the normalized squared distance between the gold tree distances d_T and predicted distances d_B is taken. So the function to minimize will become:

$$\min_B \sum_\ell \frac{1}{|s^\ell|^2} \sum_{i,j} \left| d_{T^\ell}\left(w_i^\ell, w_j^\ell\right) - d_B\left(\mathbf{h}_i^\ell, \mathbf{h}_j^\ell\right)^2 \right| \tag{2}$$

where w_i^ℓ are the words and s^ℓ is the sentence length.

The performance of the architectures on the structural task is evaluated on the undirected unlabeled attachment score (UUAS) which measures a percentage of edges labeled correctly by the model divided by the edges of the gold parse tree.

3.3 Control tasks

This task aims to analyze the role of a probing model on the original linguistics task of the language model by introducing randomness in the training process. For the control tasks, we employ the definitions of Hewitt and Liang (2019).

In the **POS-tagging control task**, there will be a function that maps every word to a randomly sampled POS-tag, defined as:

$$f_{control}(x_{1:T}) = f(C(x_1), C(x_2), ...C(x_T)) \tag{3}$$

where $C(x_i)$ is the control behavior that maps the word token x_i to an output y_i of POS-tags. This random mapping is also structural because it maps every word token in the whole vocabulary V to one output such that the same words also have the same POS-tag. The selectivity of a network is now defined as the score that is obtained with the normal linguistic task minus the score that is obtained with the control task (which is the accuracy that can be achieved with only random input). Thus, a language model is selective if it has a low control task score and a high linguistic task score.

For the **structural dependency control task**, the task is shifted to an edge prediction task in which the model is trained on predicting the parent node. The predictions are performed similarly to the POS-tagging control task but now we have the following mapping:

$$f_{\mathrm{DEP}}\left(\mathbf{x}_{1:T}\right) = \mathbf{y}_{1:T} \tag{4}$$

where the input $\mathbf{x}_{1:T}$ is mapped to a predicted edge with parent $\mathbf{y}_{1:T}$, for which we define a control behavior $C(v)$ such that it maps to fake gold labels as follows:

attach to self: token is mapped to its own token, so $\mathbf{y}_i = i$

attach to first: token is mapped to the first token, so $\mathbf{y}_i = 0$

attach to last: token is mapped to the last token, so $\mathbf{y}_i = T$

These definitions of the control task can again be applied to the POS-tagging control function, which is defined in Eqn. 3

4 Experiments and Results

Treebank The Universal Dependencies (UD, v2.0) (Nivre et al., 2016) serves as the treebank corpora and is post-processed identically as described by Gulordava et al. (2018).

Data The English training data for the language models is obtained from Gulordava et al. (2018) which employed TreeTagger for tokenization and considered only the 50k most common words, replacing less frequent words with the <UNK> token.

4.1 Models

The two different neural language architectures are described below.

RNN A long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) represents the recurrent model in the comparison. The network consists of two layers with 650 hidden units each and a dropout of 0.2. The pre-trained architecture is derived from Gulordava et al. (2018).

Transformers The base transformer is a pre-trained GPT-2 network (Radford et al., 2019), consisting of 12 hidden layers with a corresponding pre-trained Byte-level Byte-Pair-Encoding tokenizer. Additionally, a distilled version (6-layer) and a medium (24-layer with higher embedding dimensions) of the GPT-2 are considered. Furthermore, the 12-layer XLNet (Yang et al., 2019) is evaluated which has been pre-trained on bidirectional contexts in a sentence and uses a tokenizer based on SentencePiece (Kudo and Richardson, 2018).

Both autoregressive architectures employ default parameters based on the `transformer` library of Huggingface (Wolf et al., 2019) and are selected on their different pre-training objectives.

4.2 Results

First, the results of the linguistic probing tasks are presented. In Table 1 the POS-tagging task is depicted for the different architectures. The GPT-2 scores the highest accuracy of 90.1%, for the base and medium variant, and the distilled version 89.0%, outperforming both the LSTM and the XLNet. However, for the control task, the XLNet is significantly better with 46.1% as compared to the recurrent model (74.9%) and all GPT-2 models (62.2%, 59.0% and 59.6%). Achieving a low score for the control task accuracy is ideal since it suggests that by introducing randomness to the probing model, performance predictably decreases. Therefore, the desired selectivity should be high. Based on selectivity, the Transformer models, achieving between 26.8 and 39.3, perform notably better than the LSTM with a poor selectivity of 10.5.

Network	Accuracy	Control	Selectivity
GPT-2-distil (TF)	89.0%	62.2%	26.8
GPT-2-base (TF)	90.1%	59.0%	31.1
GPT-2-medium (TF)	90.1%	59.6%	30.5
LSTM (RNN)	85.4%	74.9%	10.5
XLNet-base (TF)	83.5%	46.1%	39.3

Table 1: Part-of-speech for the three architectures for the original task accuracy, the control task accuracy and selectivity.

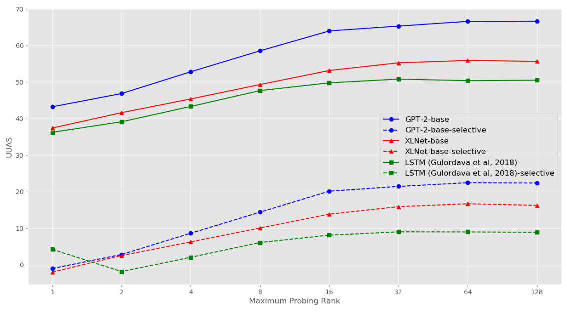
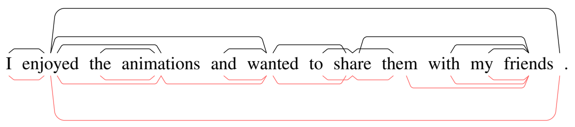
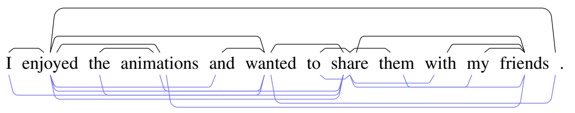


Figure 1: UUAS performance on the different architectures and the impact of the constraint of the probing rank.



I enjoyed the animations and wanted to share them with my friends .

Figure 2: Minimum spanning tree of GPT-2 with an UUAS of 83.3 and sentence length of 13



I enjoyed the animations and wanted to share them with my friends .

Figure 3: Minimum spanning tree of LSTM with an UUAS of 16.7 and sentence length of 13

The linear transformation for the structural task is defined by the dimension of $B \in \mathbb{R}^{k \times m}$ where k is the rank for the projection of B onto the hidden representation $\mathbf{h}$. The higher k is, the greater the expressive power of the probe. Figure 1 shows the effect of increasing rank k on the overall performance in terms of UUAS. Generally, after a rank of 64, the performance does not increase notably, therefore, all experiments are based on a probing rank of 64. Furthermore, at rank 64, the difference in UUAS is significant between the GPT-2 (66.6%) and the XLNet (55.9%) and LSTM (50.4%). This trend applies also for the visualized selectivity.

In Figures 2 and 3 a minimum spanning tree is constructed by both the GPT-2 and the LSTM,

	UUAS scores for sentence lengths			
Network	**≤5**	**6-10**	**11-15**	**≥16**
GPT-2-base (TF)	87.7%	63.6%	59.6%	49.8%
LSTM (RNN)	81.8%	47.8%	38.5%	24.8%
XLNet-base (TF)	81.1%	47.5%	46.3%	39.1%

Table 2: UUAS scores measured on different sentence lengths, that are evenly distributed, for the three architectures (with rank 64).

respectively. Both models create 12 edges which is also the total of the golden tree edges, nevertheless, GPT-2 predicts 10 correctly and the LSTM only 2. What we observe is that the LSTM has difficulties in connecting far ends with each other but also prefers long distances where small edges would be correct. Again, the Transformer outperforms the recurrent model on the linguistic structural task.

As a result to the constructed dependency trees, we further examined the UUAS scores on different sentence lengths, which are displayed in Table 2. For all models, the ability to correctly construct the dependency tree decreases as the sentence length increases, however, the amount of decrease from length 5-10 to both length 11-15 and ≥16 is significantly larger for the LSTM as for the Transformer networks. This indicates that the LSTM has difficulties with representing dependencies over longer sentences.

5 Discussion

In this section, the results are discussed in more detail and limitations that the proposed methods bring about will be amplified. Finally, possible avenues for future work will be proposed that would complement and extend our research.

5.1 Evaluating the results

The overall conclusion on the results of all subtasks is that the Transformer models outperform the LSTM model by a significant margin. Which transformer model performed best on these sub-tasks and possible explanations for these performances will be explained here in detail.

The **POS-tagging task** results show that the Transformer models outperform the LSTM on this sub-task in Table 1. The GPT-2 performs better in terms of accuracy on the POS-tagging task but the XLNet obtained a higher selectivity indicating that its embeddings are better in capturing the representation of POS-tags. However, this can be argued for since a lower percentage on the control task accuracy could also be the result of more fixed hidden representations towards this sub-task but not necessarily representations that contain more information about the POS-tags.

The **structural dependency task** results show that GPT-2 outperforms all other models by a significant margin even if we subtract the control task scores (Figure 1). When comparing the selectivity, we observe that the XLNet approaches the GPT-2 UUAS score slightly but still performs substantially worse on this sub-task. What we can conclude from Table 2 is that the LSTM can not handle long distances as well as the Transformer networks. This could be because of the model differences but also the pre-trained objectives of these networks. In Figure 3 the LSTM does not span an edge from the word 'enjoyed' to '.', a possible explanation for this could be that this information can not be propagated far enough in the network to correctly predict that these are connected syntactically.

5.2 Limitations and future work

Our work solely analyzed the ultimate hidden representation and not intermediate hidden layers. Peters et al. (2018b) has shown different linguistic phenomena to be encoded in either low-level or high-level layers, implying that perhaps one linguistic task may be encoded earlier and not being passed through the model sufficiently for the probing task to model properly.

Although performing probing tasks with linear classifiers seemingly uncover an existing hierarchical representation in Transformers and LSTMs, one could imagine such a simple diagnostic classifier to not always be sufficient due to the complexity and non-linearity of the inherent structural dependency of a model. A more intricate non-linear probing model may be capable of successfully probing more complex structural representations but comes at the risk of the probing model assisting the language model in its training objective, thus losing the grasp of the transparency of the language model. This trade-off is studied by (Hewitt and Liang, 2019) on the notion of selectivity, but requires other measurements and probing models to assess the balance.

Lastly, by standardizing probing models for deep neural language models these architectures are able to grow more complex while maintaining proper assessment in terms of transparency.

References

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. *CoRR*, abs/1808.08949.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.