Can auto-regressive language models understand language?

Anonymous ACL submission

Abstract

Large language models made a major breakthrough in the NLP field, demonstrating excellent performance on many linguistic tasks. However, the neural networks on which these models are based largely form a black box. The extent to which LMs actually have a grasp on the underlying linguistics of natural language remains unknown. This paper uses existing probing techniques to show that LSTM and GPT2 models are capable of detecting POS-tags and dependency graphs features. More research is needed to fortify these claims.

1 Introduction

Linguistics and deep learning nowadays are tightly linked in NLP and are mutually beneficial (Linzen, 2018). Especially the recent rise of large language models has the potential to achieve a new understanding of language. These language models construct sentences based on probabilities assigned to tokens which are determined by prior probabilites. These probabilities are learned by the use of neural models. Large neural models are capable of performing excellently on many NLP tasks. Multiple researchers have investigated the behaviour of such language models (e.g. Marvin and Linzen, 2018). However, the inner workings of language models largely remain a black box. This lack of transparancy reduces the interpretability of the model output. In this paper we attempt to shed more light on the exact functioning of these models to increase transparancy and interpretability.

Two types of neural models are discussed and compared in this paper: recurrent models and transformer models. Recurrent models pass through a sentence word for word whilst keeping track of a hidden state. The most well known recurrent model is the Long-Short-Term-Memory (LSTM) model (Hochreiter and Schmidhuber, 1997). Transformer models are attention based models, using self-attention to represent sentences (Vaswani et al., 2017). Both types of models are inherently unstructured, while language is of nature a hierarchically structured phenomenon. In this paper we will investigate whether transformer models have a stronger notion of syntactic structure than recurrent models. Considering their susceptibility to long distance depencies, transformer models are expected to better capture syntactic structure than recurrent models, especially for longer, more complex sentences. In order to investigate the models capability of recognising structure, a technique is used called probing (Conneau et al., 2018), structural probing in particular (Hewitt and Manning, 2019).

In addition to syntactic structure, the models are tested on their ability to recognise Part-of-Speech (POS) tags (Martinez, 2012). Then these two interpretability results are assessed on their mutual correlation. This provides new insights into how these models represent language to come to their outputs.

As a final test the models are assessed on their performance on a control task (Hewitt and Liang, 2019). In doing this the model performance is analysed on it's grasp of the underlying linguistic structure, as opposed to it having learned a random pattern that produces good results.

Both GPT2 and LSTM models were found to perform better on POS-tagging than the control task, indicating an understanding of the linguistic structure of language. According to our results, LSTM models are better at retrieving POS-tags and transformer models are better at capturing the hierarchical structure of natural language.

2 Related Work

RNNs are ideal for sequential analysis as they incrementally learn from input to produce their output, taking previous entries into account when producing a new output (Lipton et al., 2015). However, as the gap between interdependent entries increases, regular RNNs struggle to effectively learn these

dependencies (Bengio et al., 1994). Currently, the most used neural networks in NLP are LSTMs. These RNN models were introduced by Hochreiter and Schmidhuber (1997) and are better able to capture long-distance dependencies than regular RNNs because they keep track of a cell state. This state is the long-term memory of the algorithm that runs through the entries largely unchanged to ensure long-term consistency.

A more recently introduced neural network is the Transformer model (Vaswani et al., 2017). This model also works on sequences, but instead of working through a sequence from start to finish it takes a more global approach using the attention mechanism. All words in the sentence are assessed in parallel using self-attention. This difference in architecture makes transformer models very efficient in computing long distance dependencies (Tan et al., 2022). Since LSTMs are much older, they have been the go-to choice for much NLP research. Transformer models are less investigated yet. Attention mechanisms have been incorporated in LSTM networks to utilise benefits from both model types (Zhou and Wu, 2018). This paper aims to analyse the differences in model performance on structure recognition an POS-tagging of these two architectures.

Both LSTM and transformer models are inherently unstructured. They do not explicitly capture structure. Natural language on the other hand, is organised in a hierarchical structure, not merely sequential (Everaert et al., 2015). Despite this presumed problem, Recurrent Neural Networks (RNNs) have been shown to increase language modeling performance (Kombrink et al., 2011) and can learn simple artificial context-free grammars (Gers and Schmidhuber, 2001). Moreover, they have been found able to reproduce the nesting and indentation structure in programming languages (Karpathy et al., 2015).

Researchers have already investigated the LSTM's capability to capture syntax structure by examining syntax-sensitive dependencies (Linzen et al., 2016). This research showed that LSTMs achieve well in strongly supervised and restricted tasks such as number agreement and grammaticality judgements. However, performance decreased drastically in language modeling settings without supervision. The researchers suggest other architectural structures might be better suited to grasp the grammatical structure of language. RNNs where capable of detecting long distance dependencies with regards to number agreement, lagging not far behind human performance (Gulordava et al., 2018). Another research wherein an LSTM was asked to distinguish grammatically correct sentences from their incorrect counterparts, showed that the LSTM still performed significantly worse than a human control group (Marvin and Linzen, 2018). Moreover, the question remains whether these models actually understand the correct linguistic aspects that underlie these phenomena, because further research has indicated that there are distinct differences in the patterns in agreement errors between RNNs and humans (Linzen and Leonard, 2018). This result suggests that these models do not learn the linguistic structure of natural language in the same way as humans.

As described above it is evident that neural networks provide many opportunities to increase the perfomance on NLP tasks. However, this does come at a cost of transparancy. Large language models are so complex and require little supervision and parameters, such that they appear a black box. Multiple arguments have been brought forward to stress the importance of interpretable models as they are further applied in critical areas like healthcare and the criminal justice system (Kim, 2015). The precise meaning of interpretability in a model is not clearly defined resulting in differing claims about model transparency (Lipton, 2017). Attention mechanisms (Bahdanau et al., 2016) as used in the transformer model have been argued to provide transparency, as they provide weights to tokens in a text (Li et al., 2017) - under the assumption that these weights are a direct indicator of a models behaviour. This claim, however, was undermined in further research which concluded that attention weights should not be treated as meaningful explanations for model outputs (Jain and Wallace, 2019). This paper follows the approach of probing to attain insights in the neural models (Conneau et al., 2018). In particular structural probing introduced by Hewitt and Manning (2019).

3 Methods

3.1 Models

The LSTM model used in this paper was made available by Gulordava et al. (2018). The transformer model used in this paper is the distilled version of GPT2 (Sanh et al., 2020), retrieved from

the Huggingface transformer library.[1]

3.2 Data

For this paper the models are trained on a treebank corpus. This corpus is parsed and stored in a manner that facilitates tree construction. Moreover, the data contains POS-tags, which will be used to train en test the models. The treebank used is the English EWT database from the Universal Dependencies project [2]. For the tree construction we make use of the *conllu* library.

Important to notice is that transformer models can break up words by using Byte-Pair Encodings (BPE). The resulting chunks are then passed through the model and assigned with a representation. However, as we are interested in word-level representations, these word parts have to be combined again. We do this following the suggestion of Hewitt and Manning (2019), and compute the word representation by taking the average of its sub-word representations. Also, with these subwords it is important to make a distinction between word chunks after a space or in the middle of a word, as these are different tokens.

3.3 Structural probes

For the structural probing the set-up of Hewitt and Manning (2019) is used. With this approach the probe learns a linear transformation of a word representation space such that the transformed space embeds parse trees across all sentences. If this probe performs correctly, we have found part of the representation space of the model that is used to encode this structural syntax feature (re-create the closest parse-tree).

3.4 POS-tags

POS-tagging is a crucial task in NLP preprocessing that involves a model predicting the POS tag of a token based on a learned corpus (Martinez, 2012). To assess the models actual comprehension of natural language POS tagging they were tested on a control task (Hewitt and Liang, 2019). This control task assigns each word in the vocabulary to a new, meaningless, POS-tag according to a probability distribution based on the POS tag occurrence in the corpus. The POS-tagging task is then trained and tested by the model once again on these newly assigned control tags.

[1] https://github.com/huggingface/transformers
[2] https://universaldependencies.org/treebanks

3.5 Evaluation

For the POS tagging task the models performance is measured by the accuracy (Eq 1) and Matthews Correlation Coefficient (Eq 2) (Matthews, 1975). Accuracy returns the fraction of correct outputs over all outputs. Considering that the POS tag distribution of our dataset is not uniform, a more robust metric that is insensitive to class imbalances like the Matthews correlation coefficient (MCC) can be used. This metric ranges from -1 for a poor model to 1 for a good prediction model.

$$Accuracy = \frac{tp}{tp + fp} \quad (1)$$

$$MCC = \frac{tn * tp + fn * fp}{\sqrt{(tp + tn)(tp + fn)(fp + tn)(fp + fn)}} \quad (2)$$

The models are evaluated with these metrics on their normal POS-tagging and the control task. Selectivity is then a measure for the difference in performance on these two tasks in terms of accuracy (Eq 3) or MCC. It essentialy quantifies how much of the performance on the POS tagging task comes from an actual understanding of the linguistic structure or merely from pattern recognition.

$$Selectivity = Acc - Acc_{control} \quad (3)$$

One of the metrics used for the structural probe is, the UUAS metric which looks at correctly placed egdes in the graph. However, it is sensitive to small mistakes made at the root of the predicting parse tree, therefore making it harder to predict longer sentences. For distance in the parse tree, we compute the Spearman correlation (DSpr) between true and predicted parse tree distance for each word in the each sentence.

To get more insight into the models susceptibility to long-distance dependencies, the structure score performance is set against sentence length for all sentences in the corpus. This information is plotted to show the trends of transformer and LSTM models along different sentence lengths.

4 Experiments and Results

4.1 POS probe

For both model on both metrics the control task performed worse as can be seen by the selectivity in Table 1. The LSTM model outperformed GPT2 in the regular POS-tagging. Selectivity on the other hand is higher for GPT2 than for the LSTM, suggesting that GPT rely more on word identities than those on LSTM.

	Normal		Selectivity	
Model	Acc.	MCC	Acc.	MCC
GPT2	0.772	0.748	0.396	0.463
LSTM	**0.889**	**0.878**	0.317	0.359

Table 1: Accuracy and Matthews correlation coefficient (MCC) for GPT2 and LSTM for normal POS-tagging and selectivity (minus control task).

4.2 Structural probe

	Distance	
Model	UUAS	DSpr.
GPT2	0.529	0.636
LSTM	0.460	0.589

Table 2: Results of structural probes on the testset. For the distance probes, we show the Undirected Unlabeled Attachment Score (UUAS) and the average Spearman correlation of true to predicted distances, DSpr.

GPT2 scores better than LSTM on both structure metric as can be seen in Table 2.

4.3 Varying sentence length

The average UUAS value is plotted against sentence length in Figure 1. From these plots we observe a big skew with the model performance decreasing as the number of words in a sentence increases. This decrease appears to be steeper for LSTM than for GPT2, in accordance with the overall lower structure score for LSTM (see Table 2).

5 Discussion & Conclusion

The GPT2 transformer model shows a higher selectivity for the POS-tagging task with the control task than the LSTM model. Following the literature by Hewitt and Liang (2019) this means that the transformer model is relatively good in understanding the syntactic structure of the given text. However, as the overall accuracy of the LSTM model is higher it is unclear which model is better in absolute understanding of the probing task. Moreover, the overall POS-tagging accuracy is significantly lower than might be expected according to other research, indicating that the training of our model may not have worked perfectly. It might also be considered to remove very short sentences as they are arguably not real sentences. This might also lead to different results in the structure evaluation.

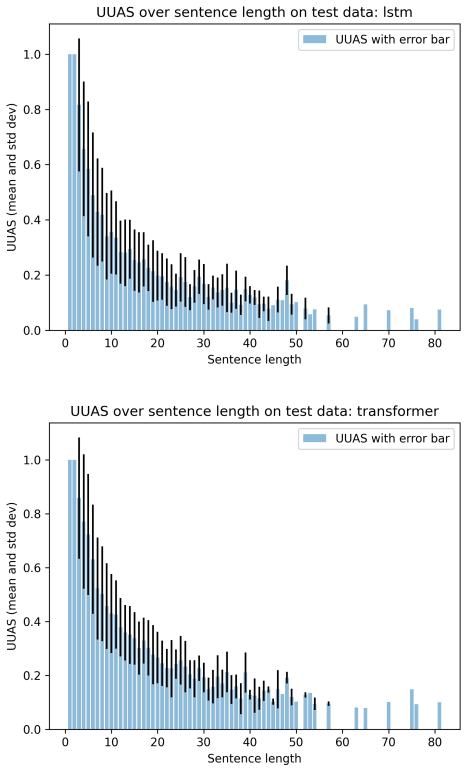


Figure 1: Average UUAS value per sentence length for LSTM (top) and GPT2 (bottom).

The structure UUAS value against sentence length showed a strong decrease in performance for increasing sentence lengths. This is in line with the distribution of sentence length within the dataset, which follows the same trend. Since the model is trained on this imbalance we expect it to perform worse on data that it has seen less of. Longer sentences might also perform worse because there are simply more possibilities for tree organization and thus there is more room for error. The UUAS value is unforgiving, one wrong choice in the root results in a very low score, other metrics like distance metric (from matrix) are more forgiving and as such may provide more insight for longer sentences.

All findings of this paper come from two models; one RNN (LSTM) and one transformer model (GPT2). To further support or reject the claims made in this research more models might be tested with multiple seed runs each. Different model sizes could also be taken into consideration to assess the generality of our findings. Future research may look into long-distance sentences dependencies within LLM, as this is a key aspect of understanding large texts and this paper has focused solely on structure within sentences.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv:1409.0473 [cs, stat].

Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166. Conference Name: IEEE Transactions on Neural Networks.

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.

Martin B. H. Everaert, Marinus A. C. Huybregts, Noam Chomsky, Robert C. Berwick, and Johan J. Bolhuis. 2015. Structures, Not Strings: Linguistics as Part of the Cognitive Sciences. *Trends in Cognitive Sciences*, 19(12):729–743.

F. A. Gers and E. Schmidhuber. 2001. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE transactions on neural networks*, 12(6):1333–1340.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. ArXiv:1902.10186 [cs].

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and Understanding Recurrent Networks. ArXiv:1506.02078 [cs].

Been Kim. 2015. *Interactive and interpretable machine learning models for human machine collaboration*. Thesis, Massachusetts Institute of Technology. Accepted: 2015-09-17T19:04:25Z.

Stefan Kombrink, Tomas Mikolov, Martin Karafiát, and Lukas Burget. 2011. Recurrent Neural Network Based Language Modeling in Meeting Recognition. pages 2877–2880.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Understanding Neural Networks through Representation Erasure. ArXiv:1612.08220 [cs].

Tal Linzen. 2018. What can linguistics and deep learning contribute to each other? ArXiv:1809.04179 [cs].

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.

Tal Linzen and Brian Leonard. 2018. Distinct patterns of syntactic agreement errors in recurrent networks and humans. ArXiv:1807.06882 [cs].

Zachary C. Lipton. 2017. The Mythos of Model Interpretability. ArXiv:1606.03490 [cs, stat].

Zachary C. Lipton, John Berkowitz, and Charles Elkan. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. ArXiv:1506.00019 [cs].

Angel R. Martinez. 2012. Part-of-speech tagging. *WIREs Computational Statistics*, 4(1):107–113. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wics.195.

Rebecca Marvin and Tal Linzen. 2018. Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

B. W. Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

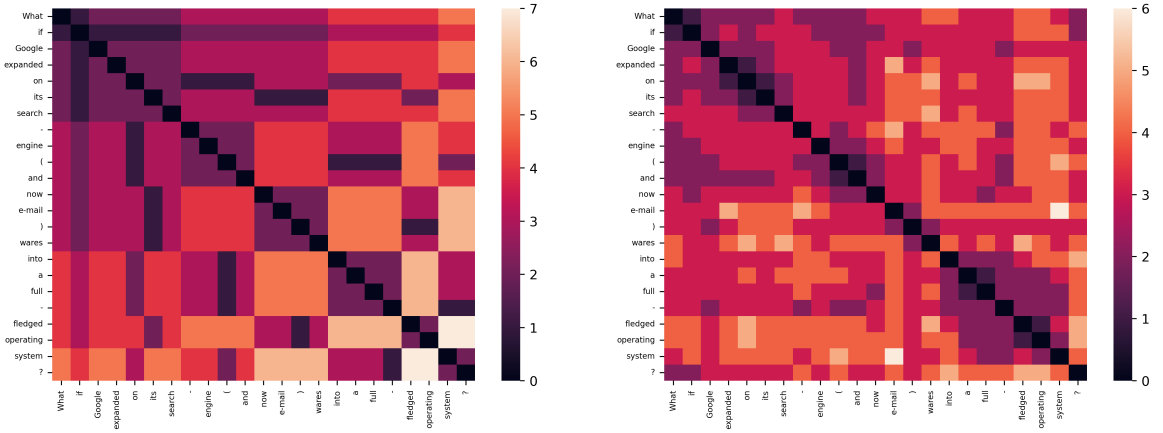
Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv:1910.01108 [cs].

Kian Long Tan, Chin Poo Lee, Kalaiarasi Sonai Muthu Anbananthen, and Kian Ming Lim. 2022. RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network. *IEEE Access*, 10:21517–21525. Conference Name: IEEE Access.

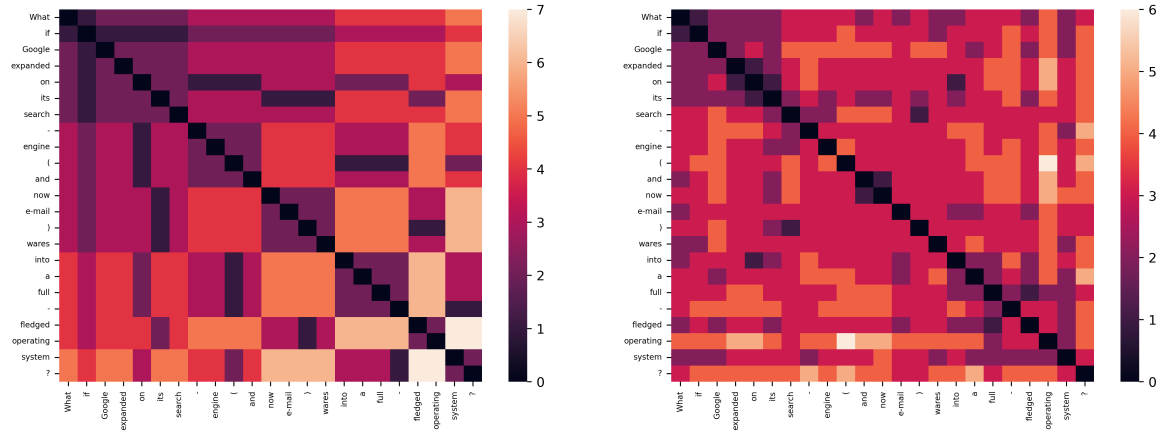
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. ArXiv:1706.03762 [cs].

Qimin Zhou and Hao Wu. 2018. NLP at IEST 2018: BiLSTM-Attention and LSTM-Attention via Soft Voting in Emotion Classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 189–194, Brussels, Belgium. Association for Computational Linguistics.

A Appendix



(a) LSTM



(b) Transformer

Figure 2: Gold (left) and predicted (right) distance matrices. Predicted matrices are rounded.

B Appendix

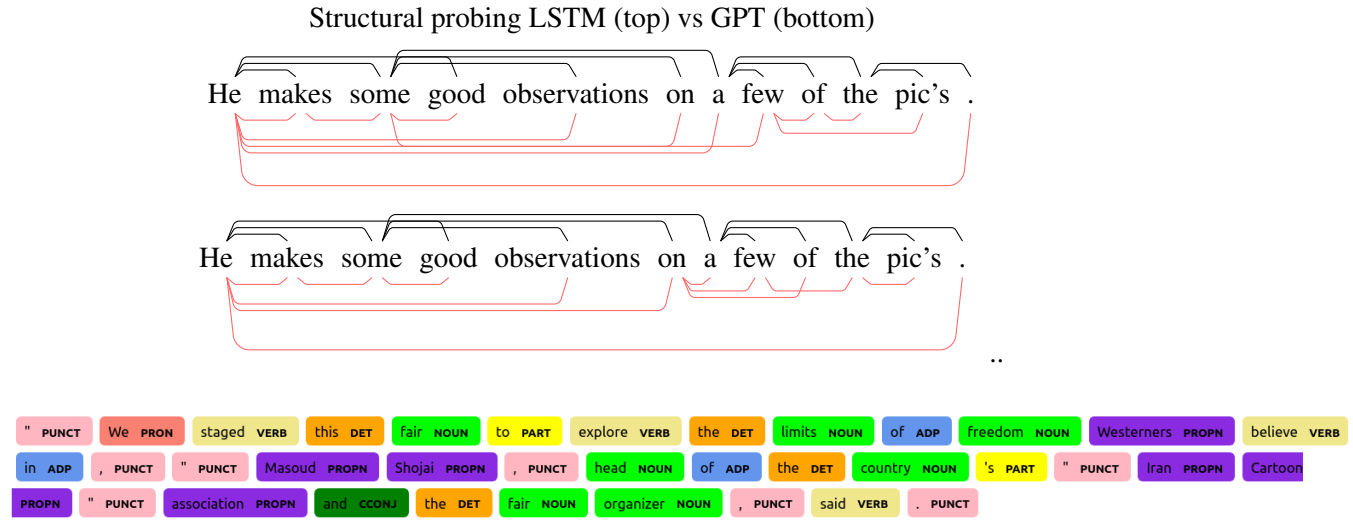


Figure 3: POS tag predicted by the probe using LSTM embeddings

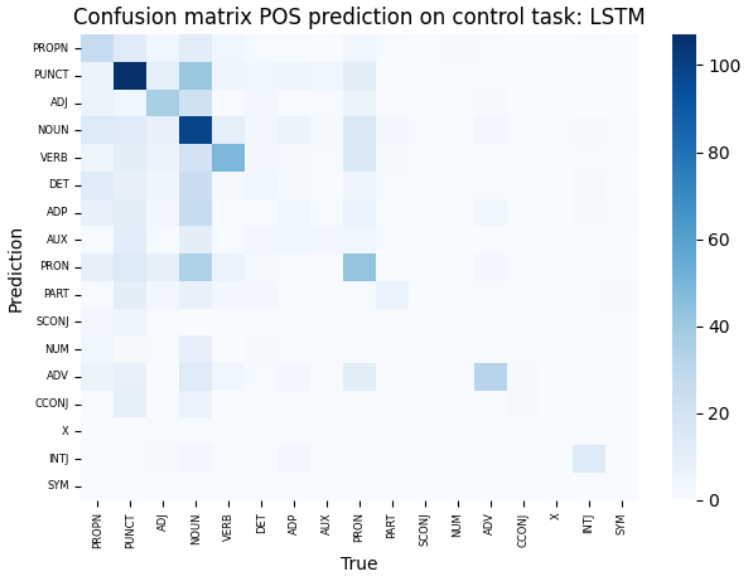


Figure 4: LSTM POS probe confusion matrix