Analysing the Importance of LLMs Embeddings' Components on Probing Linguistic Tasks

Pavel Bartenev [1 2] **Bair Mikhailov** [1 2] **Kseniia Petrushina** [1 2] **Julia Sergeeva** [1] **Daniil Shlenskii** [1]

Abstract

Large language models (LLMs) have helped researchers to achieve tremendous results in the field of NLP. However, work is still being done on their interpretability, part of which is contextualized embeddings from LLMs. Previous works demonstrated that some dimensions in LLMs' embeddings are important to the representational quality of these embeddings for task specific knowledge. In this study, we analyze components' importance of LLMs by probing on simple tasks. Our results (may) suggest that several embeddings' dimensions are directly responsible for definite linguistic properties.

Github repo: EmbeddingComponents
Presentation file: your link here to presentation file in github

1. Introduction

Various studies (Timkey & van Schijndel, 2021; **?**) have shown that representations produced by Large Language Models (LLMs) are usually dominated by a few outlier dimensions. These dimensions are characterized by large variance and magnitude in comparison with other dimensions, and it can be shown (Rudman et al., 2023) that they actually contain a lot of information about the encoded text. Particularly, some downstream tasks may be solved only by using the single outlier dimension of an embedding without resorting to any other information about the text.

Since the quality of embeddings produced by LLMs is crucial for solving any text-related task, the methods of evaluating this quality are found to be extremely important. The widely-used technique of evaluating embeddings on downstream tasks possesses noticeable downsides (Jabri et al., 2016; Lai & Hockenmaier, 2014). Among them is the possibility of hidden biases in complex tasks and the inability to clearly identify the information the model is relying upon. To address the issue, authors (Conneau et al., 2018) propose a framework, which consists of a set of simple probing tasks. These tasks target specific linguistic properties of the text, making it clear whether embeddings contain the information or not.

In this work, we provide a deep analysis of the outlier dimensions of the embeddings of LLMs and their significance for the linguistic properties of text representations. We use the framework of probing tasks (Conneau et al., 2018) in order to thoroughly validate which information the outlier dimensions hold and how it can be leveraged for embedding compression.

2. Related works

2.1. Embeddings Dimsensions

Research on BERT attention heads' purpose (Kovaleva et al., 2019) revealed that the model is overparametrized. Surprisingly, across all considered NLP tasks and datasets, disabling some attention heads leads to an increase in performance. This result implies that it can be possible to decrease number of parameters in BERT without losing quality significantly.

Thus, the research was aimed at finding important components of the model for making a prediction. In this work (Kovaleva et al., 2021) authors define outlier dimensions via output 'LayerNorm' layer of each transformer block: they compute mean and standard deviation of all scaling and bias parameters, and if some component lies too far from mean (e.g. out of 3σ bounds) for a large amount of layers (e.g. more than a half), it is pronounced as an outlier. Such components were investigated in terms of influence on MLM and some downstream tasks. It was shown that they have a significant impact on all the tasks under consideration: the performance of the model deteriorates if these outlier dimensions are not taken into account.

Outlier dimensions can be defined slightly differently (Rud-

[1] Skolkovo Institute of Science and Technology, Moscow, Russia [2] Moscow Institute of Physics and Technology, Moscow, Russia. Correspondence to: Kseniia Petrushina <petrushina.ke@phystech.edu>.

Final Projects of the Machine Learning 2023 Course, Skoltech, Moscow, Russian Federation, 2023.

man et al., 2023), that is, as dimensions in LLM representations whose variance is at least 5x larger than the average variance in the global vector space. With this definition the authors discovered several properties:

- Outlier dimensions from pre-training remain in the fine-tuned models.
- Simple threshold classifier on the dimension with the highest variance is enough to solve downstream tasks with minimal performance decline for several LLMs.
- There is a significant correlation between variance of the dimension and the performance of the simple threshold classifier in some downstream tasks.

All of that allowed authors to conclude that outlier dimensions contain task-specific knowledge for some LLMs.

2.2. Probing

Probing is one of the popular analysis methods, often used for investigating the encoded knowledge in language models (Conneau et al., 2018), (Tenney et al., 2019). This is typically carried out by training a set of diagnostic classifiers that predict a specific linguistic property based on the representations obtained from different layers.

Numerous studies use classifier's performance score to elucidate the type of knowledge encoded in various layers. To further investigate the reasons behind the layer-wise behavior, the role played by token representations different methods to interpret classifier's predictions can be used.

Some studies explored the role of token representations in the final performance (Mohebbi et al., 2021). They compute the attribution of each input token to the output labels which is called saliency score of an input token to classifier's decision. In addition to layerwise representations, subspaces that encode specific linguistic knowledge, such as syntax, have been a popular area of study. By designing a structural probe, (Hewitt & Manning, 2019) showed that there exists a linear subspace that approximately encodes all syntactic tree distances. Also one of the approaches to probe pretrained language models is fill-in-the-gap probing. (Pandit & Hou, 2021) applied fill-in-the-gap to probe bridging by formulating bridging anaphora resolution as a of-Cloze test.

Because it is difficult to interpret the embeddings of sentences on complex downstream tasks, simpler tasks were developed. To tackle evaluating sentence embeddings' quality 10 probing tasks were proposed (Conneau et al., 2018), that allowed to assess linguistic features directly. These tasks, evaluated through a Multi-Layer Perceptron (MLP) with sigmoid nonlinearity, range from predicting sentence length to semantic structures. Embeddings were obtained using encoders like BiLSTM and Gated ConvNets trained on diverse linguistic tasks. The study finds encoder architecture significantly impacts embeddings' structure with results on probing tasks being correlated with metrics on other downstream tasks. This approach offers a nuanced method to understand sentence embeddings' effectiveness and limitations.

We took the probing tasks proposed by (Conneau et al., 2018) and studied feature importances of diagnostic classifiers to investigate whether outlier dimensions play an important role in prediction or not.

2.3. Feature Importance

Feature importance is a popular approach to explain why ML model works the way they do. There are several different techniques how feature importances are being measured, most notably global and local. A modular global feature importance attempts to describe the importance of the feature for the entire model, while a local feature importance describes the importance of that feature for a specific input. (Saarela & Jauhiainen, 2021) paper compared different feature importance measures using logistic regression with L1 regularization (modular global and model-specific), random forest (modular global and model-specific), and after that for specific input cases they used LIME (local and model-agnostic).

It was shown that the most important features differ depending on the technique. Therefore, a combination of several explanation techniques could provide more reliable and trustworthy results. Also, in particular, local explanations should be used in the most critical cases such as false negatives.

On of the most notable ways of local interpretation is LIME (Ribeiro et al., 2016), a technique that explains the predictions of any classifier in an interpretable manner. Moreover, the authors also proposed a SP-LIME method, that selects a set of representative instances with explanations to address the "trusting the model" problem, via submodular optimization.

Another sophisticated approach to extracting the importance of features involved the game-theoretic concept of the Shapley vector (Lundberg & Lee, 2017). Aggregating ideas from Shapley values and other methods on feature importances evaluation authors propose their own approach unifying previuous ones. The results showed that this method allows to get feature importances that are more consistent with human intuition.

3. Project plan

- Train classifiers for probing tasks on embedding sentences from different encoders (e.g. BERT, RoBERTa, GPT-2).

- Extract the feature importance of the received classifiers using various methods (logistic regression coefficients/tree-based methods/Shapley value).
- Find outlier dimensions for the above-described encoder models.
- Conduct a comparative analysis of outlier dimensions and the most important features. Find the parts of embeddings that contain information about probing tasks.
- If possible, expand the study to more complex downstream tasks.

3.1. Methodology

3.1.1. Embeddings

For obtaining the embeddings of the sentences we used the encoder of Transformer architecture. The choice of the architecture is motivated by its efficiency in encoding a wide range of linguistic features and contexts.

The model can be viewed as a function $h : T \longrightarrow E$, where T is a $N \times L$ matrix of input embeddings of tokenized text with length of embeddings equal to L. E – output $N \times L$ matrix of embeddings. The embedding of the whole sentence can be obtained as the average of output embeddings:

$$e_{sent} = \frac{1}{n} \sum_{e_{out} \in E} e_{out}.$$

3.2. Experiments and Results

3.2.1. Logistic regression

We used logistic regression for solving probing tasks with obtained text embeddings.

3.2.2. SHapley Additive exPlanations of MLP

3.2.3. Gradient boosting

References

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

Hewitt, J. and Manning, C. D. 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)*, pp. 4129–4138, 2019.

Jabri, A., Joulin, A., and van der Maaten, L. Revisiting visual question answering baselines. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *Computer Vision – ECCV 2016*, pp. 727–739, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. Revealing the dark secrets of BERT. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4365–4374, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL https://aclanthology.org/D19-1445.

Kovaleva, O., Kulshreshtha, S., Rogers, A., and Rumshisky, A. BERT busters: Outlier dimensions that disrupt transformers. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3392–3405, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.300. URL https://aclanthology.org/2021.findings-acl.300.

Lai, A. and Hockenmaier, J. Illinois-LH: A denotational and distributional approach to semantics. In Nakov, P. and Zesch, T. (eds.), *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 329–334, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2055. URL https://aclanthology.org/S14-2055.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Mohebbi, H., Modarressi, A., and Pilehvar, M. T. Exploring the role of bert token representations to explain sentence probing results. *arXiv preprint arXiv:2104.01477*, 2021.

Pandit, O. and Hou, Y. Probing for bridging inference in transformer language models. *arXiv preprint arXiv:2104.09400*, 2021.

Ribeiro, M., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. In DeNero, J., Finlayson, M., and Reddy, S. (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-3020. URL https://aclanthology.org/N16-3020.

Rudman, W., Chen, C., and Eickhoff, C. Outlier dimensions encode task specific knowledge. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14596–14605, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.901. URL https://aclanthology.org/2023.emnlp-main.901.

Saarela, M. and Jauhiainen, S. Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, 3, 02 2021. doi: 10.1007/s42452-021-04148-9.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.

Timkey, W. and van Schijndel, M. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.372. URL https://aclanthology.org/2021.emnlp-main.372.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Pavel Bartenev (20% of work currently)

- Reviewing literate on the topic (1 paper)
- Preparing the Introduction of this report

Bair Mikhailov (20% of work currently)

- Reviewing literate on the topic (2 papers)
- Preparing the Abstract of this report
- Preparing the Appendix of this report

Kseniia Petrushina (20% of work currently)

- Coding the main algorithm
- Preparing the GitHub Repo
- Thinking about project plan

Julia Sergeeva (20% of work currently)

- Reviewing literate on the topic (2 papers)

Daniil Shlenskii (20% of work currently)

- Reviewing literate on the topic (2 papers)

B. Reproducibility checklist

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   □ No.
   □ Not applicable.

   General comment: If the answer is **yes**, students must explicitly clarify to which extent (e.g. which percentage of your code did you write on your own?) and which code was used.

   Students' comment: None

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   □ Yes.
   □ No.
   □ Not applicable.

   Students' comment:

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.

   □ Yes.
   □ No.
   □ Not applicable.

   Students' comment:

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment: None

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   □ Yes.
   □ No.
   ☑ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   □ Yes.
   □ No.
   □ Not applicable.

   Students' comment: None

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   □ Yes.
   □ No.
   □ Not applicable.

   Students' comment: None

9. The exact number of evaluation runs is included.

   □ Yes.
   □ No.
   □ Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.

    □ Yes.
    □ No.
    □ Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    □ Yes.
    □ No.
    □ Not applicable.

    Students' comment: None

12. Clearly defined error bars are included in the report.

    □ Yes.
    □ No.
    ☑ Not applicable.

    Students' comment: None

13. A description of the computing infrastructure used is included in the report.

    □ Yes.
    □ No.
    □ Not applicable.

    Students' comment: None