Investigating semantic subspaces of Transformer sentence embeddings through linear structural probing

Dmitry Nikolaev **Sebastian Padó**
Institute for Natural Language Processing, University of Stuttgart
dnikolaev@fastmail.com pado@ims.uni-stuttgart.de

Abstract

The question of what kinds of linguistic information are encoded in different layers of Transformer-based language models is of considerable interest for the NLP community. Existing work, however, has overwhelmingly focused on word-level representations and encoder-only language models with the masked-token training objective. In this paper, we present experiments with *semantic structural probing*, a method for studying sentence-level representations via finding a subspace of the embedding space that provides suitable task-specific pairwise distances between data-points. We apply our method to language models from different families (encoder-only, decoder-only, encoder-decoder) and of different sizes in the context of two tasks, semantic textual similarity and natural-language inference. We find that model families differ substantially in their performance and layer dynamics, but that the results are largely model-size invariant.

1 Introduction

It is more or less generally assumed that the success of bidirectional masked language models (MLMs), such as BERT (Devlin et al., 2019), on downstream tasks is largely due to the fact that in pre-training they learn to compute rich and well-structured representations of their inputs. More precisely, it is often argued that the task of masked language modelling encourages models to successively aggregate lexical/collocational, syntactic, and semantic information from the input text as the activations progress through encoder layers (Tenney et al., 2019). The extent to which BERT-like models follow the stages of the classical NLP pipeline (Niu et al., 2022) or accumulate contextual information (Kunz and Kuhlmann, 2022) has been questioned. However, the association of middle layers of MLMs with syntax and higher levels with semantic information is not widely disputed as a general principle and is taken for granted in many papers on model analysis and downstream applications (Chi et al., 2020; Li et al., 2021; Sharma et al., 2022).

Despite the high volume of literature on Transformer representations, these studies are mostly constrained in two ways: (i) they deal almost exclusively with word/token-level and not sentence-level embeddings, and (ii) the lion's share of attention is given to encoder-only MLMs, notably variants of BERT (cf. Reif et al., 2019; Hewitt and Manning, 2019; Vulić et al., 2020; Conia and Navigli, 2022). As a result, the representations computed by text-to-text models, such as T5, and causal language models, such as members of the GPT family, remain understudied. This can be largely attributed to the fact that the standard way of analyzing pretrained language models, namely *probing*, proceeds by applying linear classifiers to token representations at different layers (Belinkov, 2022). This approach is not as easily applicable to MLM-derived sentence representations or to representations computed by other types of models.[1] Conversely, while it is possible to provide nuanced analyses of causal Transformer-based models (Geva et al., 2021, 2022), such analyses are not easily transferable to MLMs (Nikolaev and Padó, 2023).

In this study, we propose a unified methodology for studying layer-specific sentence-level representations extracted from masked, text-to-text, and causal language models. We analyze these representations via *structural semantic probing*, largely inspired by Chi et al. (2020). Instead of directly predicting features of interest from representations, structural probing projects them onto lower-dimensional subspaces where distances are interpretable in terms of task properties, or where

[1] Cf., however, Liu et al. (2019) and works targeting representations computed by LSTM-based LMs: Giulianelli et al. (2018); Aina et al. (2019); Sorodoc et al. (2020); Sukumaran et al. (2022), and others. In this work, we focus on the processing of natural language, cf. Bhattamishra et al. (2020) and Traylor et al. (2021) on the ability of LMs to tackle formal languages.

Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 142–154
December 7, 2023. ©2023 Association for Computational Linguistics

different classes of data points are directly linearly separable. By varying the dimensionality of the projection space, we can gauge the amount of information contained in the embeddings.

While Chi et al. (2020) identify well-structured syntactic subspaces, i.e. those encoding the topology and labels of Universal Dependency trees, we target sentence-level semantic subspaces and carry out experiments on two semantic tasks, viz. sentence similarity and natural language inference (NLI). Our contributions are as follows:

1. We analyse the efficiency of solving different semantics-level downstream tasks using only suitably projected sentence embeddings derived from vanilla pre-trained encoder-only, encoder-decoder, and decoder-only models.

2. We conduct an extensive analysis of the informativeness of embeddings derived from different model layers using varying dimensionalities of projection subspaces. Many of the models we study have never been analysed in this way, and we find that their behaviour is influenced in interesting ways by both architecture and training regime.

3. We conduct our experiments at widely differing model scales: from BERT base, T5 mini, and OPT 125m to T5 XXL, Llama 13B, and OPT 66b. Our main finding is that the way information is structured across layers is largely *scale invariant*, with models sharing the same architecture and training regime demonstrating similar activation patterns.

4. We show that three major NLI datasets – SNLI, MNLI, and ANLI – lead to very different results when tackled with projected vanilla embeddings. While SNLI and MNLI, surprisingly, can be almost 'solved' with most vanilla models, ANLI, in contrast, is nearly completely opaque, and only embeddings from the biggest models are useful there.

The structure of the paper is as follows: § 2 introduces structural probing and its application; § 3 lays out our experimental setup; § 4 presents and discusses our findings, and § 5 concludes.

2 Semantic structural probing

In all our experiments, we assume that we are given a set of sentences $s_i \in \mathcal{S}$ and a corresponding set of sentence representations $r_{s_i,m,l} \in \mathcal{R}$, where each element is indexed with a sentence, a model from which it was derived, and the model layer. (Model and layer subscripts will be omitted when not needed.) Depending on the task, we also have labels of different types either for sentence pairs ($l_{i,j}$) or individual sentences (l_i). We target the following tasks:

1. Semantic textual similarity (STS): a pair of sentences is labelled with a number from 0 to 5, where 0 corresponds to the smallest degree of semantic similarity and 5 to the maximal degree. We map these labels to the range [0, 1] of semantic differences.

2. Textual entailment (TE): an ordered pair of sentences is labelled according to whether the second sentence is entailed by the first one or contradicts it. To simplify the analysis, we do not address neutral sentence pairs.

To study the semantic organisation of sentence representations, we aim to find a projection matrix M to a lower-dimensional space, such that we can directly 'read off' the answer from the application of the matrix to elements of $\mathcal{R}$.

For the **STS** task, we choose an M that minimises the differences between the gold-label similarities and the Euclidean distance between embeddings (averaged over the mini-batch):

$$\mathcal{L}_{\text{STS}} = (||Mr_{s_i} - Mr_{s_j}||_2 - l_{i,j})^2 \qquad (1)$$

This corresponds to learning an approximation to the Mahalanobis matrix M^TM, that is, to learning a distance metric in the embedding space. (This interpretation carries over to our other experiments.) This distance metric is optimised to correlate well with the manually provided similarity judgements. Correspondingly, we evaluate the performance of the probing approach by computing the Spearman correlation between $||Mr_{s_i} - Mr_{s_j}||_2$ and $l_{i,j}$.

The number of columns of M is equal to the dimensionality of the embedding space, but we can control the number of rows and thus vary the dimensionality of the projection subspace. In all experiments, we use the sequence of the powers of two from 2^1 to 2^9, augmented with the embedding dimension of the model (e.g., 768 for BERT base).

We apply a similar approach for a subset of data from the **TE** task: for sentence triplets where we have both an entailment e_i and a contradiction c_i

for a given premise p_i, we define $q_i = ||Mr_{p_i} - Mr_{e_j}||_2$ and $r_i = ||Mr_{p_i} - Mr_{c_j}||_2$ and minimise

$$\mathcal{L}_{\text{TE-triplet}} = [q_i - r_i]_+ \tag{2}$$

where $[\cdot]$ stands for $\max(0, \cdot)$. In this manner, we encourage premises to be closer to their entailments than to contradictions.

By replacing Euclidean distances with cosine similarities, we can further tackle any premise–hypothesis pair (p_i, h_i) by minimising $\mathcal{L}_{\text{TE-pair}} =$

$$\begin{cases} (1 - \text{Cos}(Mp_i, Mh_i))^2 & \text{if } l_i = \text{entail.,} \\ (-1 - \text{Cos}(Mp_i, Mh_i))^2 & \text{if } l_i = \text{contr.} \end{cases} \tag{3}$$

where Cos is cosine similarity. In this manner, we induce entailments to show positive cosine similarities to their premises, and contradictions to show negative similarities to their premises.[2]

We evaluate the TE models using accuracy. In the triplet setting, we count as hits all cases where $r_i - q_i < 0$. In the sentence-pair setting, we follow the intuition above and consider answers to be correct if $\text{Cos}(Mp_i, Mh_i) > 0$ for entailments and $\text{Cos}(Mp_i, Mh_i) \leq 0$ for contradictions.

It must be stressed that by 'evaluation' we mean a proxy measure of the informativeness of vanilla embeddings and not a measure of how well the models can solve the original task. The labels of the tasks themselves constitute an 'abuse of notation' as in all cases we are dealing with reformulations of the original tasks, which in the case of NLI involve a considerable simplification. Thus, numbers should not be compared to results on the original tasks. Nevertheless, we believe that our proxy tasks can provide interesting insights into the models.

Representation extraction The extraction of sentence representations depends on the architecture of the model. When working with encoder-only MLMs, such as BERT, we follow the standard practice of averaging all token representations in a given layer. When working with T5-type models, which have both an encoder and a decoder, we hypothesise that the heavy lifting in representation learning is being done by the encoder and apply the same approach to it (cf. Ni et al., 2022). For causal LM models, such as GPT-2, Llama, and OPT, we extract the representation of the last token of the input sentence.

Recall that since our goal is to probe general models, we always work with vanilla pre-trained versions with no fine-tuning. Our structural probing approach is, however, also applicable to fine-tuned models.

[2] Even though Cosine, as a symmetrical measure, is not an ideal match for asymmetrical entailment, it works well in practice (Reimers and Gurevych, 2019).

3 Models, data, and experimental setup

3.1 Models

We experiment with the following models:

- **MLMs**: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2020), and ELECTRA (Clark et al., 2020).
- **Text-to-text**: the original T5 series of models (Raffel et al., 2020) and the T5-efficient model series (Tay et al., 2022).
- **Causal LMs**: GPT-2 (Radford et al., 2019), Llama[3] and OPT (Zhang et al.).[4]

We aim at providing maximum coverage by model type and size for all tasks, but due to very high computational costs of running larger models on large datasets (T5 XXL and causal LMs with 7b+ parameters), even in inference mode, gaps remain.

3.2 Datasets

For the STS task, we use the STS benchmark (Cer et al., 2017) distributed with the `sentence-transformers` Python library.[5]

For the TE task, we use SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and ANLI (Nie et al., 2020), all distributed by HuggingFace.

See the Appendix for the sizes of data splits.

3.3 Experimental setup

All experiments are implemented using PyTorch and the `transformers` library (Wolf et al., 2020).[6] Projection matrices are implemented as single PyTorch linear layers without bias and are fit to data using AdamW (Loshchilov and Hutter, 2019) and the learning rate of 10^{-5}. A separate matrix is fitted for each combination of the model, layer, and subspace dimensionality.

[3] https://huggingface.co/openlm-research

[4] All model checkpoints were downloaded from HuggingFace.

[5] https://sbert.net/datasets/stsbenchmark.tsv.gz

[6] Scripts for conducting the analyses can be found at https://github.com/macleginn/semantic-subspaces-code

For the STS task, where the dataset is comparatively small, the optimisation is allowed to run for the maximum of 300 epochs with early stopping after 10 epochs without improvement on the development set. For the TE task, where training takes much longer, each optimisation is run for 5 epochs with the best checkpoint selected by performance on the development set. All the results are reported for the test set. In the context of the STS task, we conduct 10 runs of each experiment to assess the robustness of results to random initialisation of projection matrices.

4 Results

In this section, we first describe our presentation of the results of the experiments (§ 4.1) and then go over individual tasks and model types (§ 4.2).

4.1 Presentation of results

The result for each experiment run is a matrix with rows corresponding to projection-subspace dimensionalities and columns corresponding to layers. When it was feasible to run the experiments several times, we obtain a matrix of averages and a matrix of standard deviations. As we show below, these matrices exhibit interesting patterns of how semantic information is distributed in the models.

However, it is unwieldy to operate with a large number of such matrices, and for summary comparison of model architectures and sizes, we collapse them into by-layer and by-dimensionality vectors by applying the maximum function to the columns or the rows of the matrices respectively. In order to compare models of different sizes, which have different numbers of layers, we further map layer numbers to the interval [0, 1], encoding *relative layer position*, such that 0 corresponds to the embedding layer and 1 to the final layer, respectively. We visualise these results as line graphs.

4.2 Results by task

4.2.1 Semantic textual similarity

Results across model architectures We first assess to what extent sentence representations computed by encoder-only, text-to-text, and causal language models contain subspaces where distances between representations mirror their semantic distances according to human annotations. Here, and in the other experiments, our results provide a lower bound on the amount of structure and informativeness of the embeddings extracted from vanilla pretrained models. Assuming, however, that our probe provides a reasonable proxy for the informativeness of the embeddings, we can also ask which layer provides the richest embeddings and what is the minimal necessary dimensionality of the projection subspace to achieve good results.

	2	4	6	8	10	12	14	16	18	20	22	24
BERT large cased												
2	0.36	0.32	0.29	0.3	0.34	0.31	0.28	0.26	0.28	0.36	0.28	0.25
4	0.49	0.4	0.39	0.41	0.48	0.47	0.39	0.42	0.52	0.51	0.42	0.35
8	0.57	0.57	0.57	0.54	0.59	0.61	0.6	0.6	0.63	0.61	0.62	0.63
16	0.63	0.63	0.63	0.64	0.66	0.66	0.65	0.65	0.67	0.67	0.67	0.68
32	0.66	0.65	0.65	0.67	0.68	0.69	0.68	0.68	0.69	**0.69**	**0.7**	**0.7**
64	0.67	0.67	0.66	0.68	0.7	0.7	0.69	0.69	0.69	**0.69**	**0.71**	**0.71**
128	0.57	0.67	0.67	0.68	0.68	0.7	0.7	0.55	0.51	0.49	0.54	0.57
256	0.57	0.56	0.55	0.55	0.56	0.56	0.53	0.5	0.5	0.48	0.5	0.56
512	0.57	0.55	0.54	0.54	0.56	0.55	0.53	0.5	0.49	0.48	0.5	0.56
1024	0.57	0.55	0.54	0.54	0.56	0.55	0.52	0.5	0.49	0.47	0.49	0.56
1024′	0.57	0.55	0.54	0.53	0.57	0.55	0.52	0.5	0.51	0.49	0.5	0.58
RoBERTa large												
2	0.31	0.27	0.35	0.35	0.31	0.28	0.37	0.31	0.29	0.26	0.27	0.19
4	0.41	0.41	0.52	0.44	0.37	0.37	0.41	0.4	0.38	0.38	0.37	0.25
8	0.56	0.58	0.61	0.63	0.62	0.63	0.64	0.61	0.48	0.54	0.48	0.3
16	0.61	0.64	0.66	0.68	0.7	0.72	0.72	0.72	0.73	0.73	0.72	0.36
32	0.64	0.68	0.7	0.71	0.73	0.74	0.74	0.75	0.76	0.76	0.75	0.42
64	0.66	0.69	0.71	0.72	0.75	0.76	0.76	0.76	**0.77**	**0.77**	**0.77**	0.72
128	0.58	0.55	0.57	0.56	0.75	0.75	0.58	0.74	**0.77**	**0.77**	**0.77**	0.73
256	0.54	0.54	0.56	0.55	0.57	0.56	0.55	0.55	0.56	0.57	0.58	0.74
512	0.54	0.54	0.56	0.55	0.55	0.55	0.54	0.54	0.55	0.56	0.57	0.74
1024	0.54	0.54	0.56	0.54	0.55	0.55	0.54	0.54	0.54	0.55	0.56	0.52
1024′	0.53	0.53	0.57	0.57	0.57	0.57	0.58	0.58	0.57	0.58	0.58	0.47
ELECTRA large												
2	0.29	0.31	0.33	0.34	0.3	0.28	0.35	0.36	0.33	0.24	0.23	0.19
4	0.37	0.41	0.46	0.47	0.4	0.38	0.4	0.44	0.45	0.29	0.23	0.2
8	0.42	0.54	0.62	0.55	0.63	0.47	0.62	0.63	0.64	0.46	0.32	0.23
16	0.59	0.68	0.69	0.7	0.69	0.69	0.69	0.69	0.68	0.67	0.6	0.36
32	0.68	0.71	0.71	0.72	0.72	0.71	0.72	0.71	0.7	0.69	0.63	0.48
64	0.69	0.72	0.72	**0.73**	**0.73**	0.72	0.72	0.71	0.7	0.69	0.64	0.49
128	0.7	**0.73**	**0.73**	**0.74**	**0.73**	0.72	0.71	0.69	0.7	0.69	0.64	0.49
256	0.71	0.63	0.62	**0.73**	0.58	0.53	0.52	0.52	0.5	0.46	0.63	0.5
512	0.59	0.61	0.61	0.59	0.55	0.52	0.51	0.51	0.49	0.45	0.38	0.49
1024	0.59	0.6	0.6	0.58	0.55	0.52	0.51	0.51	0.48	0.44	0.38	0.31
1024′	0.57	0.59	0.6	0.57	0.54	0.53	0.54	0.54	0.51	0.44	0.35	0.23
T5 large												
2	0.27	0.28	0.28	0.25	0.25	0.25	0.24	0.23	0.21	0.21	0.16	0.2
4	0.37	0.36	0.36	0.34	0.36	0.33	0.33	0.29	0.28	0.27	0.22	0.25
8	0.45	0.45	0.43	0.42	0.42	0.41	0.41	0.36	0.32	0.32	0.3	0.31
16	0.49	0.5	0.5	0.48	0.48	0.46	0.45	0.4	0.36	0.36	0.33	0.35
32	0.52	0.53	0.53	0.52	0.51	0.49	0.49	0.42	0.39	0.37	0.35	0.39
64	0.54	0.56	0.56	0.54	0.53	0.53	0.51	0.44	0.4	0.39	0.36	0.45
128	0.55	0.57	0.57	0.56	0.55	0.53	0.53	0.45	0.4	0.4	0.36	**0.74**
256	0.56	0.57	0.57	0.56	0.56	0.54	0.54	0.45	0.41	0.4	0.37	**0.74**
512	0.56	0.58	0.58	0.57	0.56	0.54	0.54	0.46	0.41	0.4	0.37	0.73
1024	0.56	0.58	0.58	0.57	0.56	0.55	0.54	0.46	0.41	0.4	0.37	0.72
1024′	0.54	0.54	0.51	0.44	0.41	0.38	0.31	0.22	0.2	0.14	0.14	0.49

Table 1: Spearman correlations of sentence-similarity scores derived via projection from averaged-token representations by model, layer (columns), and subspace dimensionality (rows) with the STS benchmark scores. 1024′ stands for using vanilla representations without projection. The results are averaged over ten runs.

Table 1 shows the full results for MLMs. (For space considerations, odd-numbered layers were omitted: they continue the same pattern.) It can be seen that the task can be solved rather well using only projected vanilla embeddings and that, while RoBERTa shows better performance than BERT ($r = 0.77$ vs. 0.71), best results are achieved using the same setup: extracting representations from the layers close to the last one and projecting them to 64 or 128 dimensions. ELECTRA, whose performance is in between the classic MLMs ($r = 0.74$) can also be made to perform well by using 128-

GPT-2 large

	3	6	9	12	15	18	21	24	27	30	33	36
2	0.23	0.25	0.21	0.24	0.3	0.31	0.24	0.25	0.19	0.18	0.22	0.38
4	0.39	0.37	0.36	0.36	0.37	0.33	0.29	0.26	0.23	0.23	0.27	0.49
8	0.45	0.4	0.38	0.41	0.4	0.29	0.29	0.26	0.27	0.27	0.3	0.55
16	0.46	0.4	0.38	0.41	0.38	0.27	0.29	0.28	0.28	0.28	0.32	**0.57**
32	0.44	0.38	0.37	0.38	0.26	0.29	0.3	0.28	0.29	0.29	0.33	**0.56**
64	0.42	0.39	0.36	0.23	0.26	0.3	0.3	0.28	0.29	0.3	0.33	0.33
128	0.43	0.35	0.2	0.23	0.26	0.3	0.31	0.29	0.29	0.3	0.34	0.32
256	0.3	0.21	0.19	0.23	0.27	0.3	0.31	0.29	0.3	0.3	0.34	0.32
512	0.3	0.21	0.19	0.23	0.27	0.3	0.31	0.29	0.3	0.3	0.34	0.32
1280	0.29	0.21	0.2	0.23	0.27	0.3	0.31	0.29	0.3	0.3	0.34	0.32
1280′	0.28	0.21	0.19	0.22	0.27	0.3	0.31	0.29	0.31	0.32	0.35	0.31

Llama 7B

	1	2	5	8	11	14	17	20	23	26	29	32
2	0.11	0.09	0.21	0.17	0.24	0.37	0.36	0.35	0.3	0.24	0.23	0.43
4	0.11	0.12	0.34	0.24	0.5	0.49	0.45	0.44	0.33	0.27	0.26	0.53
8	0.11	0.24	0.42	0.44	0.56	0.55	0.52	0.4	0.28	0.3	0.3	**0.57**
16	0.11	0.17	0.47	0.56	**0.58**	0.56	0.5	0.3	0.29	0.32	0.33	0.54
32	0.12	0.3	0.51	0.56	**0.57**	0.53	0.3	0.3	0.3	0.32	0.34	0.35
64	0.12	0.39	0.51	0.54	0.54	0.31	0.3	0.3	0.3	0.32	0.34	0.35
128	0.12	0.41	0.51	0.52	0.49	0.3	0.29	0.3	0.3	0.33	0.34	0.34
256	0.12	0.42	0.5	0.49	0.31	0.29	0.29	0.3	0.3	0.33	0.34	0.34
512	0.13	0.43	0.49	0.44	0.3	0.29	0.29	0.3	0.3	0.33	0.34	0.33
4096	0.18	0.44	0.4	0.19	0.29	0.29	0.29	0.29	0.3	0.33	0.34	0.33
4096′	0.2	0.15	0.16	0.19	0.34	0.35	0.35	0.36	0.36	0.41	0.43	0.36

OPT 30B

	4	8	12	16	20	24	28	32	36	40	44	48
2	0.08	0.13	0.13	0.12	0.16	0.38	0.38	0.33	0.3	0.23	0.21	0.41
4	0.14	0.12	0.15	0.15	0.17	0.44	0.5	0.46	0.3	0.29	0.27	0.53
8	0.15	0.15	0.16	0.17	0.2	0.52	0.51	0.44	0.33	0.32	0.33	**0.56**
16	0.18	0.16	0.17	0.17	0.19	0.51	0.5	0.34	0.34	0.33	0.35	0.53
32	0.19	0.19	0.19	0.17	0.19	0.5	0.36	0.33	0.35	0.34	0.37	0.35
64	0.24	0.21	0.21	0.19	0.21	0.45	0.36	0.35	0.35	0.36	0.37	0.35
128	0.52	0.25	0.23	0.21	0.4	0.38	0.35	0.34	0.35	0.35	0.37	0.34
256	0.5	0.47	0.42	0.38	0.34	0.32	0.35	0.34	0.35	0.35	0.38	0.34
512	0.48	0.45	0.35	0.33	0.31	0.27	0.34	0.34	0.35	0.35	0.38	0.34
7168	0.3	0.29	0.28	0.26	0.23	0.26	0.34	0.33	0.35	0.35	0.38	0.33
7168′	0.19	0.19	0.2	0.19	0.2	0.28	0.41	0.42	0.44	0.46	0.49	0.39

Table 2: Spearman correlations of sentence-similarity scores derived via projection from averaged-token representations by model, layer (columns), and subspace dimensionality (rows) with the STS benchmark scores. ′ stands for using vanilla representations without projection. The results are averaged over ten runs.

dimensional subspace, but its best performance is achieved much earlier in the model, on layers 8–10, and then slowly degrades. This demonstrates that the specialisation of higher levels on semantic features, characteristic of BERT-like models (Li et al., 2021), does not apply straightforwardly to ELECTRA, which raises the question of what kind of hierarchy of linguistic features ELECTRA encodes. In line with the results by Chi et al. (2020), the use of very high-dimensional subspaces, with or without projection, leads to bad performance.

The encoder from T5 large demonstrates yet another pattern: the performance is low for almost all parameter combinations, but then suddenly jumps to 0.74 at the last layer. (The results on layer 23, not shown in the table, are very similar to those from layer 22.) Additionally, T5 seems to encode semantics in a higher-dimensional subspace, with projecting on 128 dimensions being the minimum and 1024 still working well.

The results from causal models, shown in Table 2, demonstrate a different consistent pattern. The best performance is worse, r = 0.56–0.58, and it is usually achieved in the last layer, similarly to T5, but the optimal dimensionality of the projections is much lower (between 8 and 32), despite the models' higher embedding dimensionality. Also, there is a clear cyclic development in performance across layers. E.g., with a dimensionality of 16, GPT-2 large first goes high (0.46), then low ($\approx$ 0.28), then high again (0.57). Llama 7B shows 1.5 cycles and OPT 30B two full cycles (cf. also Figure 2).

The relatively lower informativeness of causal models' representations compared to those provided by MLMs seems to support the argument that they are less suited for representation learning (Clark et al., 2020; Reimers, 2022). However, the fact that they are most informative in the last layer goes against the previous interpretation that the last layer of GPT-2 undergoes a representation collapse (Ethayarajh, 2019) and rather supports the argument that the extreme anisotropy of the last layer of GPT-2 is an artefact of inadequate similarity modelling (Timkey and van Schijndel, 2021).[7]

Results across model sizes We now study the connection between the semantic content of models' representations, as measured by our structural probe, and their size. Classic MLMs, such as BERT and RoBERTa, are only available in a few sizes, not counting various smaller distilled versions, such as TinyBERT (Jiao et al., 2020). Later models (both text-to-text and causal) were published in a larger size range. Aggregated results are presented visually in Figure 1 (cf. Section 4.1). The left pane shows the performance of encoder-only MLMs across two model sizes each. We see that while bigger models perform better, the distribution of the semantic information across layers is very similar across model sizes. This finding is further strengthened by the analysis of the performance of the T5-efficient, Llama, and OPT models shown in the right-hand pane of Figure 1. All T5-efficient models attain the best performance in the very last layer and show some loss of informativity in middle layers.

The three Llama models, shown in the left pane of Figure 2, follow the pattern from Table 2: the performance of the lower layers is almost zero, while middle layers attain maximum performance, which then decreases and goes up again at the end. The behaviour of the OPT models is even more

[7]Figures 5 and 6 in the Appendix provide a visualisation of the distribution of the performance by the normalised layer position and projection-space dimensionality across models.

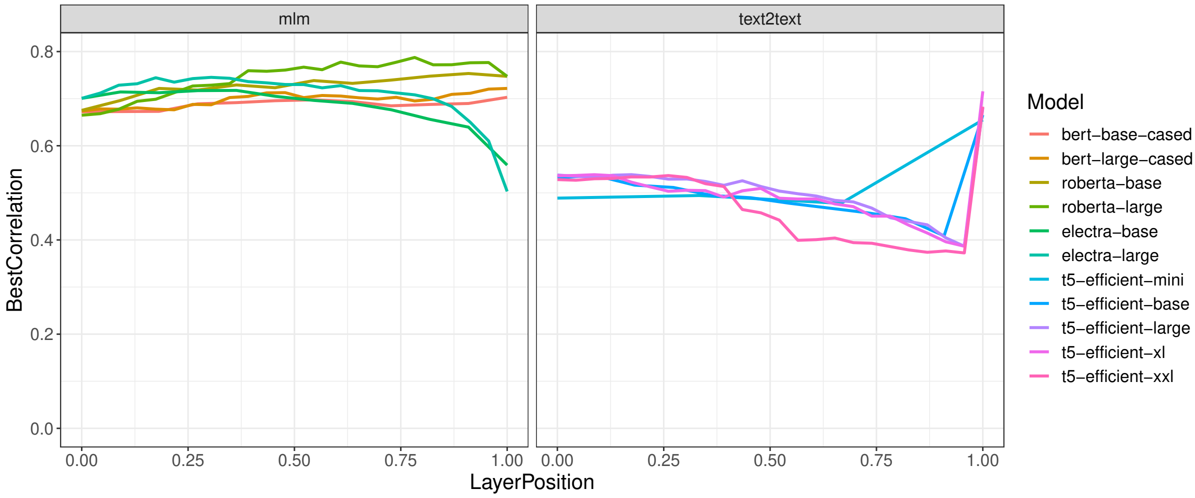

Figure 1: Performance of encoder-only and text-to-text MLMs on the STS probing task by layer and model size.

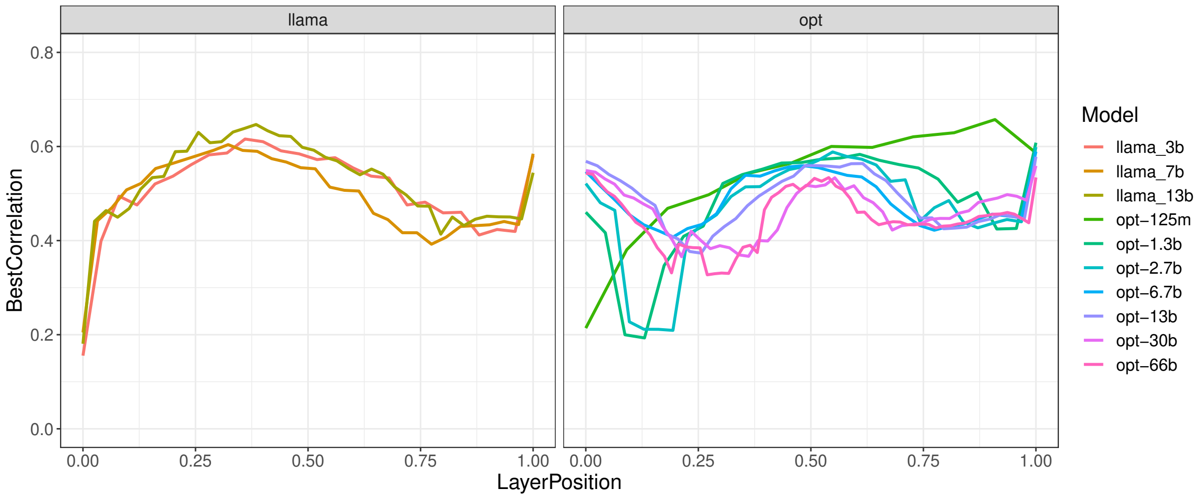

Figure 2: Performance of Llama and OPT on the STS probing task by layer and model size.

complicated: nearly all of them demonstrate the 'double dip' pattern, where as the layer number increases the performance first goes down, then up, then down again, and finally reaches the peak in the last layer. This oscillating pattern in the performance of causal LMs does not fully align with the the conclusions reached by Geva et al. (2021), who claim that there exists a progression of lexical, syntactic, and semantic features as information flows through decoder-only models language models.

OPT, interestingly, is also the only model class where we see a substantial effect of model size: the smallest model, OPT 125m, shows a steady increase in performance with a slight drop at the very end. It outperforms all larger models and nearly reaches the results of T5. This seems to suggest that extremely small causal LMs have non-trivial representation-learning capacities.

4.2.2 Natural Language Inference

In this section, we check if the distribution of semantic information across model layers we identified in the context of the STS task can also be observed in the context of NLI. We further check if the patterns are dependent on the dataset and on the exact operationalization: we contrast triplet-focused probing, which is structurally close to our STS operationalization, with cosine-similarity-based probing, which operates on the level of sentence pairs and permits us to cover more data.

Figure 3 shows the performance of the MLMs, text-to-text, and causal models of different sizes on ANLI, MNLI, and SNLI; best results by model class are summarised in Table 3. What comes to the fore in this analysis are stark differences between the three datasets, visible across all architectures.

ANLI presents the worst results across all model

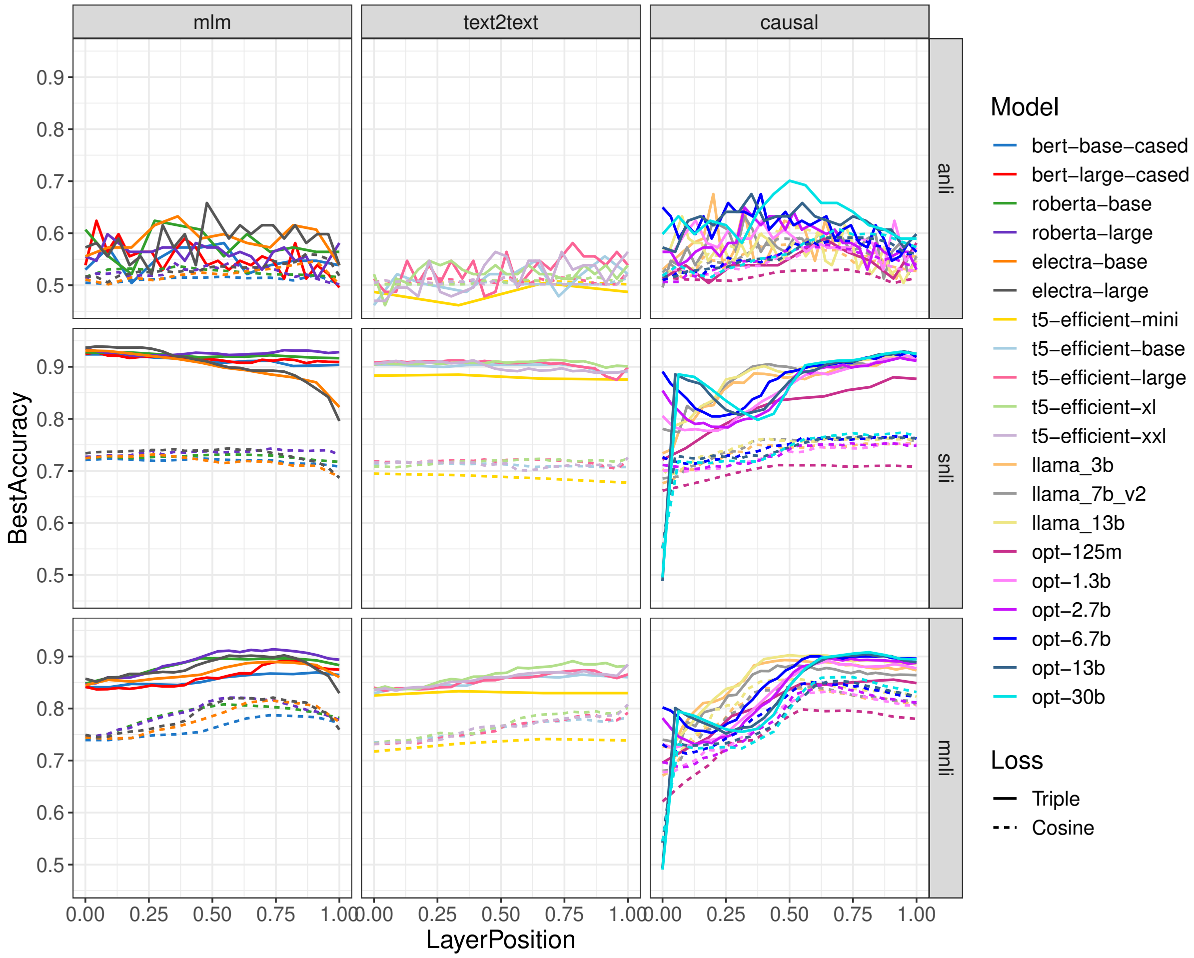

Figure 3: Performance of models on NLI across datasets, model types and sizes, and model layers.

Task	Triplet loss Model	Triplet loss Acc	Cosine loss Model	Cosine loss Acc
ANLI	OPT 30b	0.701	Llama 13b	0.613
	Llama 3b	0.675	OPT 30b	0.609
	ELECTRA l	0.658	ELECTRA l	0.561
SNLI	ELECTRA l	0.939	OPT 30b	0.773
	RoBERTa l	0.935	Llama 7b	0.764
	OPT 6.7b	0.929	RoBERTa l	0.743
MNLI	RoBERTa l	0.914	OPT 30b	0.86
	OPT 30b	0.908	Llama 13b	0.856
	ELECTRA l	0.902	ELECTRA l	0.821

Table 3: Best-performing model types (intervening models of the same type but different size were skipped) by task and setting. *Acc* stands for accuracy; *l*, for large.

types, albeit with interesting differences. While the T5-efficient models never do better than random guessing and the MLMs guess randomly in the cosine-similarity setting and sporadically achieve accuracies of ≈ 0.6 in the triplet setting, the models from the Llama and OPT families consistently achieve accuracies above 0.6 in *both* settings, squarely beating the encoder-equipped models.

The results on **SNLI** show the greatest differences between operationalizations. In the triplet setting, all encoder-based models achieve accuracies of ≈ 0.93 in their lower layers, and the results then remain stable or degrade (ELECTRA). Causal models attain similar results in the upper layers, and T5-efficient models demonstrate slightly lower results regardless of the layer. In the cosine-similarity setting, however, the task becomes much harder, with no model showing accuracy above 0.77, and causal models again showing best performance.

The differences between layers are most pronounced in **MNLI**. In the triplet setting, MLMs show the best performance in middle layers, while text-to-text and causal models achieve slightly worse results in upper layers. In the cosine-similarity setting, however, middle and upper layers of causal models again demonstrate the best performance, approaching 0.86.

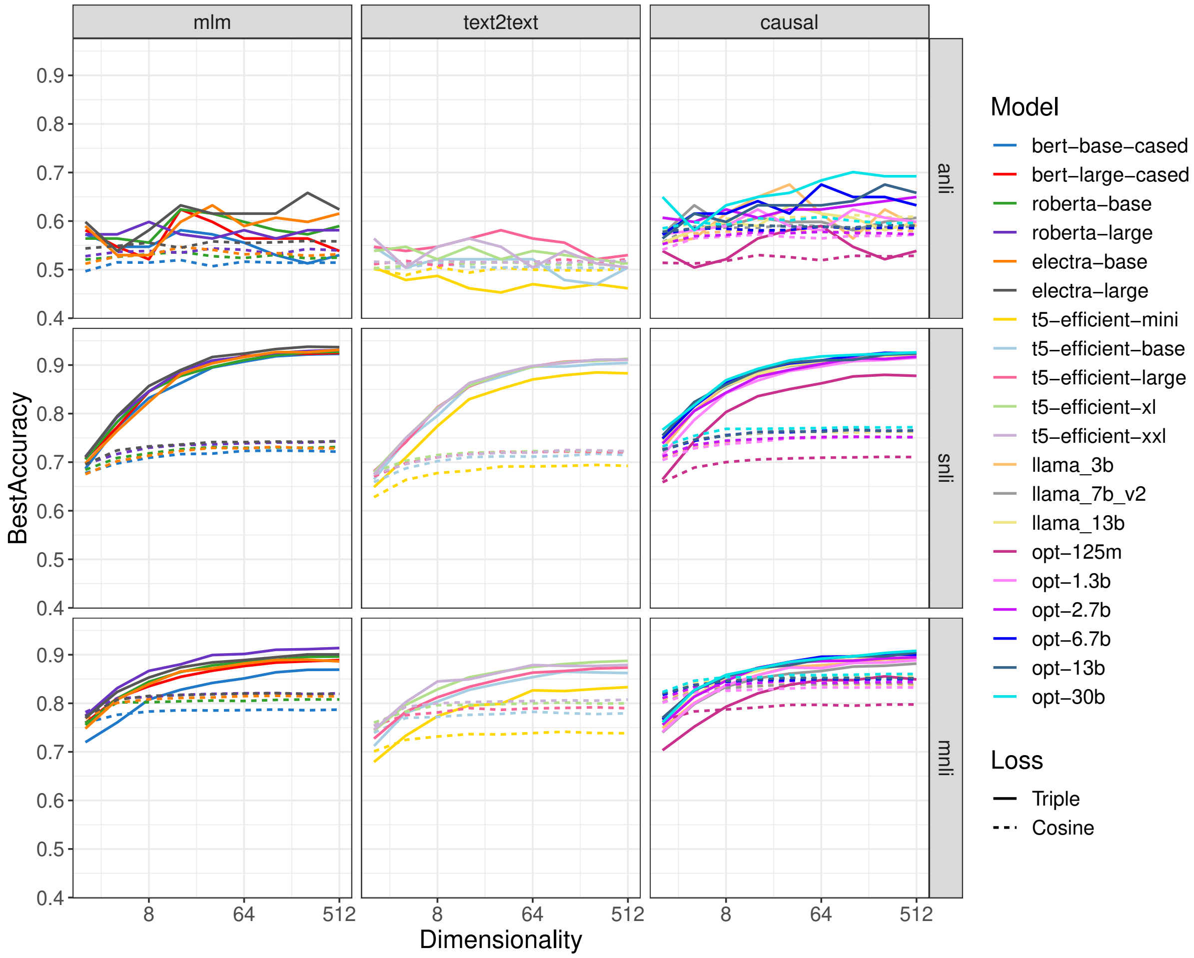

Figure 4: Performance of models on NLI across datasets, model types and sizes, and projection-space dimensions.

On the whole, causal models demonstrate surprisingly good results, outperforming T5-efficient, competing with MLMs in the triplet setting, and consistently outperforming them in the more challenging cosine-similarity setting.

Difference between layers The observations on the distribution of information across model layers made based on the STS task are largely repeated: Llama models tend to achieve peak performance in middle layers, while larger OPT models have a dip in performance between early and late layers, with the best performance attained near the end. The largest OPT models are also distinguished by an almost uninformative first layer.

Effect of subspace dimensionality Fig. 4 shows that on MNLI and SNLI all models types achieve peak performance with a dimensionality of at least 128 in the triplet-loss setting. This is in contrast to the STS task, where only T5 profited from a dimensionality above 64. However, no model can profit from more than 16 dimensions for the cosine-similarity setting, which highlights the influence of the finer details of probing methodology on the experimental results. The results on ANLI are generally inconclusive, as performance is low and unstable throughout; only OPT-30b seems to systematically gain from dimensionalities above 128.

Effect of model size Up to a certain point, the size of the model is of a much smaller importance than the architecture and training regime, and even in the finer details of their performance, differently-sized Llama models resemble each other more than the OPT models that are close to them in parameter count. When it comes to the best performance on the probing task, however, the most successful model is nearly invariably also the biggest in its class, with the cosine setting being the most size demanding.

5 Conclusion

Despite a surge of interest in prompting techniques targeting large decoder-only language models (Liu et al., 2023), there are still settings where vector representations of sentences remain a competitive alternative, e.g. semantic search and information retrieval (Thakur et al., 2021; Zhuang et al., 2023). Therefore, it seems worthwhile to investigate sentence representations from pre-trained models so as to not only better understand models themselves but also guide practical applications.

The results of our study suggest two general observations. First, no architecture is best suited for representation learning, and the informativeness of vanilla sentence representations can only be measured with regard to a particular task. Thus, while the pre-trained RoBERTa provides the best representations for semantic textual similarity, beating much larger Llama 13b or OPT 30b and 66b, when it comes to NLI, causal models can provide more informative embeddings even at smaller model sizes, in line with the findings of Muennighoff (2022) regarding the informativeness of causal model embeddings for semantic search.

Secondly, different models arrive at very different patterns of information processing across layers. Most surprisingly, ELECTRA, despite its similarities to BERT, demonstrates a degradation in performance on all surveyed tasks in its upper layers, which begs the question of what kind of linguistic hierarchy this model encodes. Similarly, Llama and OPT models, despite sharing the same architecture, also show markedly differing patterns of information restructuring.

In this study, we targeted two rather general semantics-oriented tasks. However, the proposed methodology can be applied to other problems – straightforwardly to regression tasks, such as political scaling (Glavaš et al., 2017) or emotion-intensity estimation (Zad et al., 2021), but also to classification tasks, as long as they support a reasonable similarity-based reformulation. We leave the exploration of these areas to future work.

Limitations

The results of this study depend on a long series of design choices as to the particular ways of extracting sentence embeddings, reformulating the downstream tasks, choosing the loss function, etc. We believe that the choices we made are justifiable and help to provide a strong lower bound on the informativeness of sentence representations, but the results we obtained are still dependent on them and different operationalization may lead to somewhat different conclusions.

A more general issue with this type of analysis is the fact that the notion of *semantics* as encoded by LMs is not well defined, and while STS and NLI are both reasonable approximations, there are differences in the way the surveyed models encode information relevant for these tasks, which, among other things, points to the importance of lexical effects. Disentangling these aspects is an important area for future work on model interpretability.

Finally, the validation and test splits of the ANLI dataset in the triplet setting are small, which leads to noticeable instability of the performance of all models, except for OPT 30b.

References

Laura Aina, Kristina Gulordava, and Gemma Boleda. 2019. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7096–7116, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Simone Conia and Roberto Navigli. 2022. Probing for predicate argument structures in pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4622–4632, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. Unsupervised cross-lingual scaling of political texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 688–693, Valencia, Spain. 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.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jenny Kunz and Marco Kuhlmann. 2022. Where does linguistic information emerge in neural language models? Measuring gains and contributions across layers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4664–4676, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. 2021. How is BERT surprised? Layerwise detection of linguistic anomalies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4215–4228, Online. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations, New Orleans, 6-9 May 2019*.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022.

Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Dmitry Nikolaev and Sebastian Padó. 2023. The universe of utterances according to BERT. In *Proceedings of the 15th International Conference on Computational Semantics (IWCS)*.

Jingcheng Niu, Wenjie Lu, and Gerald Penn. 2022. Does BERT rediscover a classical NLP pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. *Advances in Neural Information Processing Systems*, 32.

Nils Reimers. 2022. OpenAI GPT-3 text embeddings – Really a new state-of-the-art in dense text embeddings? Blog post.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of EMNLP/IJCNLP*, pages 3980–3990. Association for Computational Linguistics.

Rishab Sharma, Fuxiang Chen, Fatemeh Fard, and David Lo. 2022. An exploratory study on code attention in BERT. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, pages 437–448.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. 2020. Probing for referential information in language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4177–4189, Online. Association for Computational Linguistics.

Priyanka Sukumaran, Conor Houghton, and Nina Kazanina. 2022. Do LSTMs see gender? Probing the ability of LSTMs to learn abstract syntactic rules.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Traylor, Roman Feiman, and Ellie Pavlick. 2021. AND does not mean OR: Using formal languages to study language models' representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 158–167, Online. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Samira Zad, Maryam Heidari, James H Jr Jones, and Ozlem Uzuner. 2021. Emotion detection of textual data: An interdisciplinary survey. In *2021 IEEE World AI IoT Congress (AIIoT)*. IEEE.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning T5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2308–2313.

A Dataset-split sizes

A.1 STS task

Train: 5749; dev: 1500; test: 1379.

A.2 NLI task

A.2.1 Euclidean triplet loss

- **ANLI** Train: 217940; dev: 116; test: 117.
- **MNLI** Train: 261775; dev: 6692; test: 6703.
- **SNLI** Train: 345241; dev: 3256; test: 3247.

A.2.2 Cosine-similarity loss

- **ANLI** Train: 94076; dev: 2132; test: 2132.
- **MNLI** Train: 261775; dev: 6692; test: 6703.
- **SNLI** Train: 367384; dev: 6765; test: 6781.

B Details of model performance

Performance of the models on the STS task by layer and by projection-space dimensionality is shown in Fig. 5 and 6 respectively.

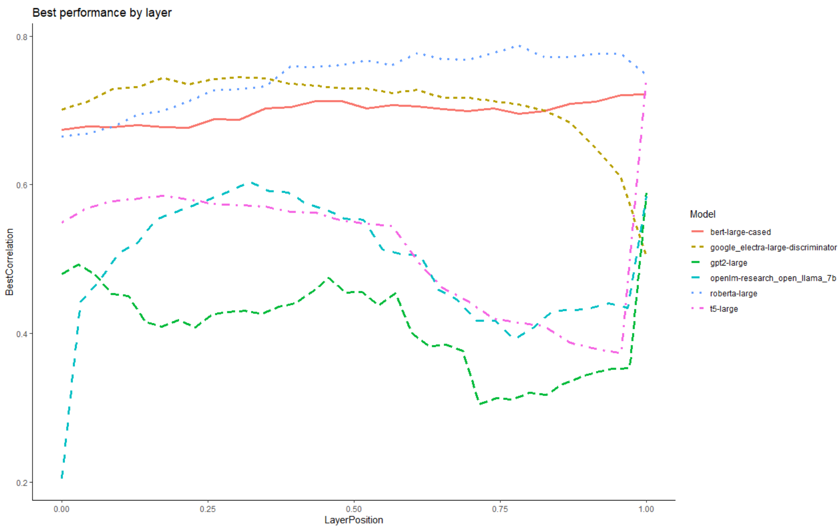

Figure 5: Best performance on the STS task by layer.

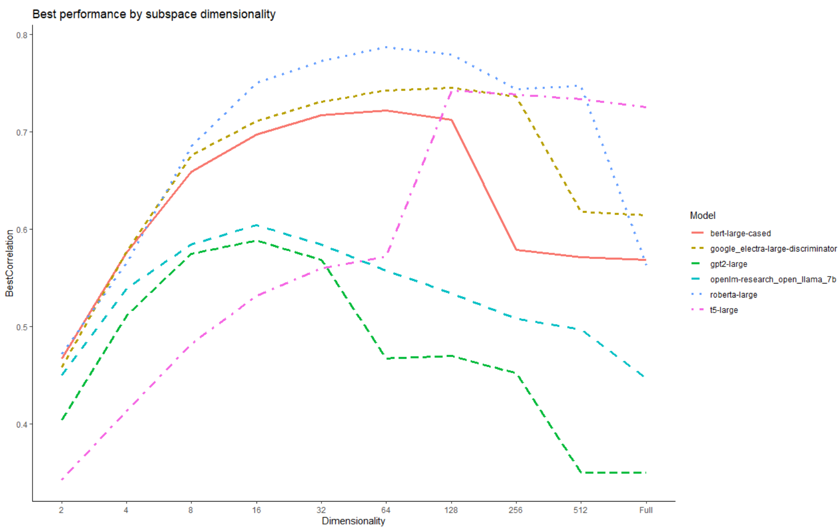

Figure 6: Best performance on the STS task by projection-space dimensionality.