BEYOND TOKENS IN LANGUAGE MODELS: INTERPRETING ACTIVATIONS THROUGH TEXT GENRE CHUNKS

Nicholas Pochinkov
Independent
work@nicky.pro

Éloïse Benito-Rodriguez
Independent
eloise_benito@laposte.net

Einar Urdshals
Chalmers Technical University
einarurdshals@gmail.com

Jasmina Nasufi
Lund University
jasminanasufi9@gmail.com

ABSTRACT

Understanding Large Language Models (LLMs) is key to ensure their safe and beneficial deployment. This task is complicated by the difficulty of interpretability of LLM structures, and the inability to have all their outputs human-evaluated. In this paper, we present the first step towards a predictive framework, where the genre of a text used to prompt an LLM, is predicted based on its activations. Using `Mistral-7B` and two datasets, we show that genre can be extracted with F1-scores of up to 98% and 71% using `scikit-learn` classifiers. Across both datasets, results consistently outperform the control task, providing a proof of concept that text genres can be inferred from LLMs with shallow learning models.

1 INTRODUCTION

As language models continue to improve in performance and are applied in a growing number of sectors, it is becoming more important to understand how these models work and monitor their outputs. However, the current paradigm of transformer interpretability focuses almost exclusively on predicting single tokens outputs, with some limited research looking at a specific component of text such as truthfulness or emotion.

We set the ground for looking at language model outputs on the scale of clusters of tokens as interpretable natural language separations. By looking at larger blocks of text, we aim to make it easier to interpret the highest-level of predictions that a language model is making, and to make it easier to monitor large scale language models outputs.

We achieve this by building a small dataset of diverse language outputs, and make a machine- and human-curated natural language labelling of different chunks of these texts. In addition to this dataset, we also use an established dataset. We then show that the text chunks in these two datasets are interpretable and easy to classify using simple probes, and that these outputs may be relatively natural units of study.

2 RELATED WORK

Most mechanistic interpretability research focuses on understanding the effects neuron activations have on single token activations (Geva et al., 2021; Chan et al., 2022; Conmy et al., 2023; nostalgebraist, 2020; Belrose et al., 2023; Olsson et al., 2022). While this is an important and natural bottom-up lens through which to understand models, it stands that a more top-down approach may be underexplored.

Recent work on investigating language model activations and Representation Engineering (Zou et al., 2023; Burns et al., 2022; Turner et al., 2023; Gurnee & Tegmark, 2023; Meng et al., 2022), suggests that it is possible to extract human-understandable concepts with their activations. These results support the hypothesis that a top-down approach to language model interpretability like the method described in this paper is tractable.

In addition, research on modularity and activation sparsity in language models (Liu et al., 2023; Pochinkov & Schoots, 2023; Zhang et al., 2022; 2023b; Pfeiffer et al., 2023), as well as the achievement of high pruning ratios for task-specific models, (Xu et al., 2022) suggest there may be separable ways of looking at different components of text.

Research on image models, including interpretability research (Voss et al., 2021; Olah et al., 2017; Mordvintsev et al., 2018), and research into the machine unlearning of specific classes and concepts (Foster et al., 2023; Bourtoule et al., 2021; Nguyen et al., 2022), highlights that it is possible to understand how models activate not only on the specific pixel values, but also on larger scale concepts. Our work tries to lay the groundwork for finding these larger scale concepts in text.

There exists a large variety of research into using language models for text embedding (Muennighoff et al., 2022) and classification (Minaee et al., 2021; Howard & Ruder, 2018) on various datasets and tasks, including news categorisation (Zhang et al., 2015), sentiment analysis (Maas et al., 2011), question answering (Rajpurkar et al., 2016), semantic relatedness (Marelli et al., 2014), and retrieval augmentation (Thakur et al., 2021; Lin et al., 2022). However, these are focused on using language models to perform specific tasks, rather than understanding how a language model performs tasks.

Relatedly, there is research on topic analysis of single texts (De Paoli, 2023) and large corpuses of text (Gauthier & Wallace, 2022), as well as research into qualitative deductive coding, (Zhang et al., 2023a; Gebreegziabher et al., 2023; Rietz & Maedche, 2021). The research is often narrow in scope, such as finding aggregate themes of outputs from specific online communities (Tai et al., 2023; Chew et al., 2023), classifying types of questions (Xiao et al., 2023), or extracting and reformatting knowledge (Shi et al., 2023). Our research, instead of topics, aims to probe into the working of language models to understand on a high-level, how different multi-token scale components are represented in the activations, and thus investigate text categories (Drew, 2023) and block components.

3 METHOD

The goal of this work is to investigate the extent to which different classifiers manage to predict the category of a text chunk, given the activations and residual stream that this text generates in a pre-trained model. To this end, we create a dataset and train various classifiers as prediction models. We describe the process and our methods in the following subsections.

3.1 DATASETS

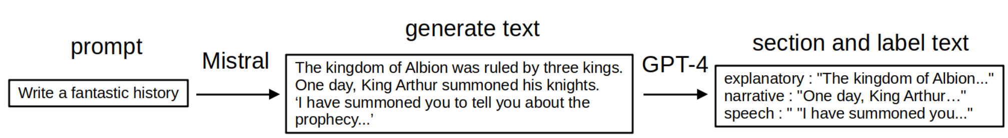

Figure 1: Generation of the labelled dataset.

As shown in Figure 1, we build our dataset by creating a list of 669 prompts, which are constructed such that they can elicit varied texts from a list of categories and topics. We use them with `Mistral-7B-Instruct-v2.0` (Albert Q. Jiang, 2023) to generate texts. These generated texts are then passed on to `GPT-4` (OpenAI et al., 2023), which is asked to segment the text into chunks according to categories and subsequently label each chunk with the selected category. We obtain 3914 distinct text chunks. The categories we consider are: instructional, explanatory, speech, narrative, and code. The labelling is reviewed by humans and mislabelled text sections are removed. The prompts and examples of the labelled text are shown in the appendix.

The category distribution of this synthetic dataset is as follows: 'instructional': 1159, 'explanatory': 699, 'speech': 548, 'narrative': 542, 'code': 290.

The primary motivation for using an open-weights model like `Mistral-7B-Instruct-v2.0` to generate the texts, is that it allows us to access its activations. This lays the groundwork for future work where we predict the sequence of labels based on the activations produced by the prompt.

We wanted a second dataset that was official so that our results would not be based solely on the dataset we created. The official dataset that we chose is `Corpus of Online Registers of English` (CORE) (Veronika Laippala & Pyysalo, 2022), which we modified to have fewer categories and a single label per text.

The category distribution of the CORE dataset is as follows: 'News Report': 6992, 'Informational': 5917, 'Opinion': 4594, 'Sports Report': 1975, 'Personal Blog': 1849, 'Persuasion': 950, 'Discussion': 832, 'Instructional': 214.

3.2 Training procedure

We work with the Large language model `Mistral-7B` (Albert Q. Jiang, 2023) with 7-billion parameters. The model is accessed via the Hugging Face Transformers library (Taylor et al., 2022) and run using PyTorch (Paszke et al., 2019). We prompt the LLM using text chunks from our two datasets, and we study the residual stream and activations at the output of each attention and MLP layer, as described in (Radford et al., 2019). Specifically, a text chunk produces activations a_i^j in layer j of the LLM for chunk c_i.

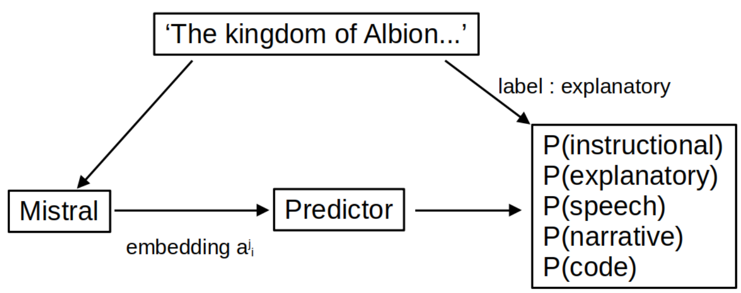

Figure 2: This figure illustrates the training procedure of prediction models on the task of predicting the category of a text section. The activation a_i^j of chunk c_i in the j-th layer is extracted from `Mistral-7B`.

The means of activations a_i^j serve as input for the probe models we train to predict the correct category for each labelled text chunk. We employ a range of commonly used shallow learning classifiers as our probe models from the scikit-learn library Pedregosa et al. (2011). We call them using the Lazy Predict library Shankar Rao Pandala (2022), which enables us to test several scikit-learn algorithms on the same dataset. We keep the vanilla hyperparameters, apart from `max_iter` which was fixed at 100000, otherwise some models would not be callable. We scale our data by removing the mean and scaling to unit variance with `StandardScaler()`.

Probing activations is notoriously subject to all kinds of spurious reasons for high classification accuracy (Ravichander et al., 2021). It demands heavy use of controls in order to validate the study. We drew inspiration from (Hewitt & Liang, 2019) to provide baselines: for each probe, we built a probe trained and tested on `Mistral-7B` activations and residual stream, but where all the parameters are random. We use the random model probe results as a control task. If the probes on the original `Mistral-7B` show better performance than the control task, this will mean that our probes do not rely on spurious correlation, but on true representations in activations.

We divide our dataset into 80% for the train set and 20% for the test set.

To evaluate our predictors, we use the Macro F1 score metric (Margherita Grandini, 2020) to give the same weight to all classes, even in an unbalanced dataset, including both recall and precision.

3.3 Dimensional Reduction

To study how the model might represent different aspects of text, we use a text embedding model, `Qwen3-Embedding-0.6B` Zhang et al. (2025) to encode the split chunks of text. For dimensionality reduction and analysis, we use PHATE Moon et al. (2017), and show the labeled category. We randomly select 200 samples from each category so as not to overload the display and make it readable.

4 Results

4.1 Dimensional Reduction

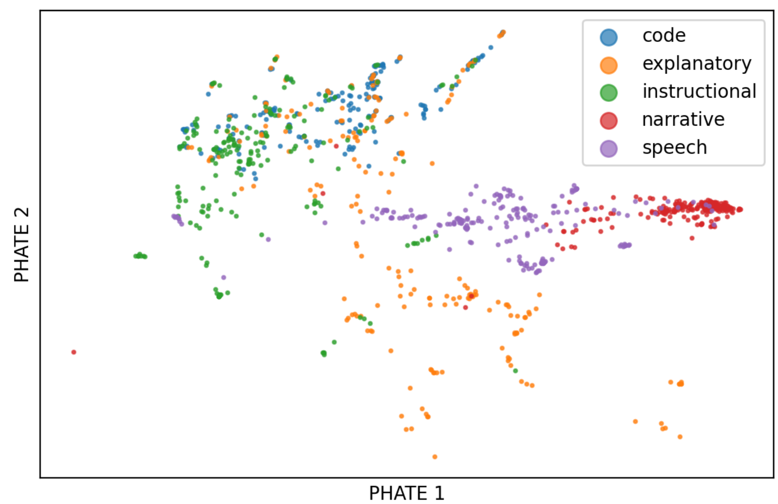

Figure 3: The PHATE dimensionality reduction for the synthetic dataset. We observe that there is some correspondence between the clusters and the labeled categories.

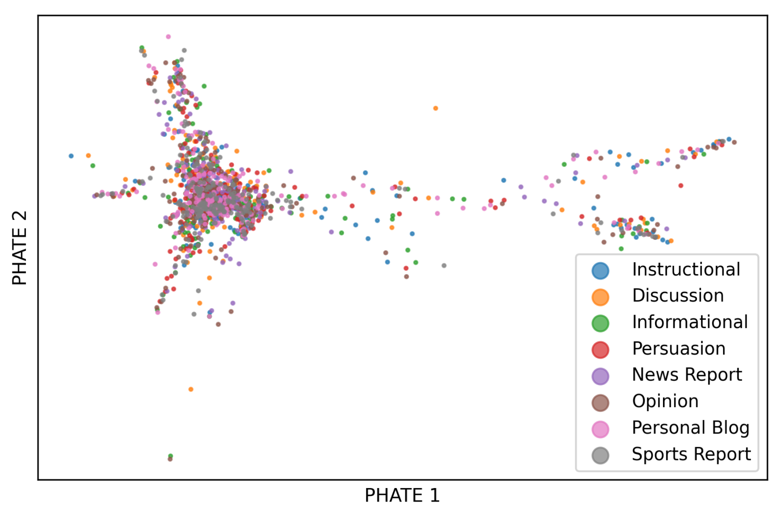

Figure 4: The PHATE dimensionality reduction for the CORE dataset. We see that there is a lot of overlap between clusters and the labelled categories.

Looking at Figure 3, the category does not fully explain the clusters observed in dimensionality reduction, but we see the relatively high correspondence between text categories and various clusters in the embedding space as a promising signal that the encoding of these themes is being captured within the models.

Looking at Figure 4, we see a great overlap with clusters that are difficult to distinguish. This could be explained because there are more classes in the corpus dataset, but also because the categories are much more similar in this dataset.

4.2 Labelling Accuracy

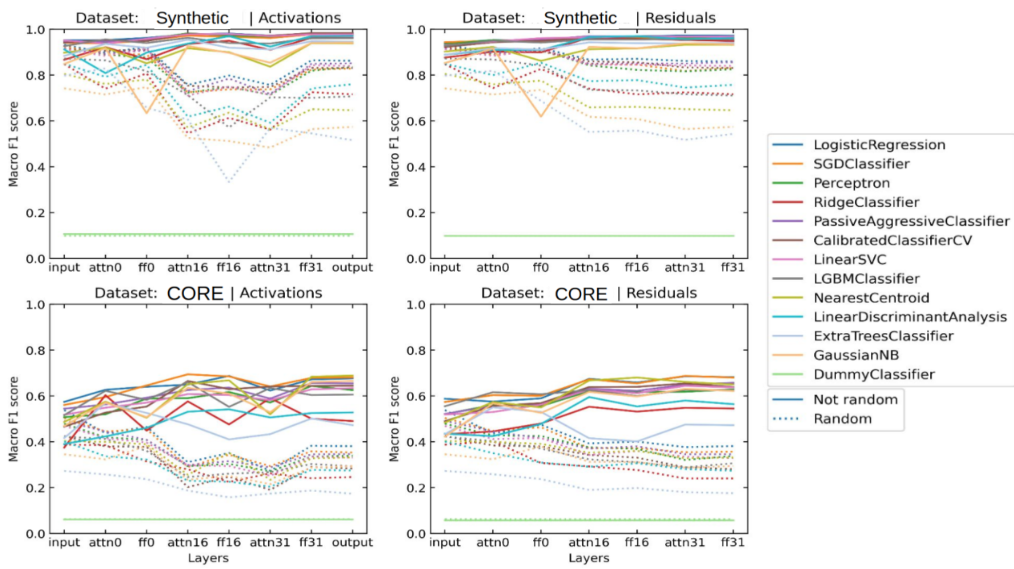

Figure 5: F1-score performance as a function of the layer fraction for prediction models on the task of predicting the category of a text section. The activation a_i^j of chunk c_i at the j-th layer has been extracted from `Mistral-7B`.

In Figure 5 we see the macro F1 score performance for various probe models predicting the category.

The x-axis indicates the layer from which the activations have been extracted, and the colors indicate the predictor that has been employed.

The plain lines correspond to the probing with the test task, whereas the dotted lines correspond to the control task.

The left graphic shows the performance of probe models based on the activations, the right graphic shows the performance based on the residual stream. The top graphic is based on the synthetic dataset, the bottom graphic is based on the CORE dataset.

We see that the deeper we go in the model, the higher the performances.

We see that when the models are trained with the synthetic dataset, the results are largely better than with the CORE dataset, reaching an F1 score of 0.98 whereas the best models built on the CORE dataset reach 0.71. This can be explained by the number of classes which is more important in the CORE dataset. But as we saw in the dimensional reduction results, the label clusters with PHATE dimensional reduction are reasonable with the synthetic dataset, whereas they are almost indistinguishable with the CORE dataset. We suspect this could mean that text categories are similar in the CORE dataset. Our probe results could be a confirmation of this hypothesis. But contrary to the dimensional reduction, the probes distinguish the labels to a certain extent (better than the random task).

5 Discussion and Conclusions

Overall, we find the results promising, and they preliminarily support the hypothesis that models represent high-level multi-token text structures in relatively easy to identify ways. Starting out with 2 datasets, we found that we can achieve a high F1 score for shallow learning probe models, better than the control task. Additionally, our embedding analysis shows interesting results for how models represent concepts.

We saw that the performance of our probe method is superior to the control task, revealing the presence of the representation that we are searching for in the attention patterns. The deeper we go in the Transformer, the more accurate the representations are. The performance is highly dependent on the dataset; we expect that the more different the classes are from each other, the higher the performance will be.

5.1 Limitations

We study the activations of `Mistral-7B`, but we could also probe the activations of other models to see if the results are consistent. Additionally, we use only two datasets, leaving room to investigate additional datasets to determine what exactly drives the performance.

5.2 Future Work and Broader Impacts

This is a starting point for looking at language model outputs on a larger scale than the token scale.

So far, our results indicate that even with a few relatively overlapping text categories, we are able to probe the model relatively accurately. Future work could investigate scaling up the number of text categories, possibly to include text topic of discussion or other properties of text, and to find better ways of incorporating the overlap of multiple text categories.

Future work could try to use these findings and probes to find longer timescale predictions on which categories are likely to emerge in the future of the residual stream, improving our ability to verify the trustworthiness of models stating their intended future actions.

References

Arthur Mensch-Chris Bamford Devendra Singh Chaplot Diego de las Casas Florian Bressand Gianna Lengyel Guillaume Lample Lucile Saulnier Lélio Renard Lavaud Marie-Anne Lachaux Pierre Stock Teven Le Scao Thibaut Lavril Thomas Wang Timothée Lacroix William El Sayed Albert Q. Jiang, Alexandre Sablayrolles. Mistral 7b. 2023.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pp. 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019. URL `https://doi.org/10.1109/SP40001.2021.00019`.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing: a method for rigorously testing interpretability hypotheses. AI Alignment Forum, 2022. URL `https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN`.

Robert Chew, John Bollenbacher, Michael Wenger, Jessica Speer, and Annice Kim. Llm-assisted content analysis: Using large language models to support deductive coding. *arXiv preprint arXiv:2306.14924*, 2023.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *CoRR*, abs/2304.14997, 2023. doi: 10.48550/arXiv.2304.14997. URL https://doi.org/10.48550/arXiv.2304.14997.

Stefano De Paoli. Performing an inductive thematic analysis of semi-structured interviews with a large language model: An exploration and provocation on the limits of the approach. *Social Science Computer Review*, pp. 08944393231220483, 2023.

C. Drew. 18 text types (with examples) – writing styles explained. Helpful Professor, September 2023. URL https://helpfulprofessor.com/text-types/.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. *CoRR*, abs/2308.07707, 2023. doi: 10.48550/ARXIV.2308.07707. URL https://doi.org/10.48550/arXiv.2308.07707.

Robert P Gauthier and James R Wallace. The computational thematic analysis toolkit. *Proceedings of the ACM on Human-Computer Interaction*, 6(GROUP):1–15, 2022.

Simret Araya Gebreegziabher, Zheng Zhang, Xiaohang Tang, Yihao Meng, Elena L Glassman, and Toby Jia-Jun Li. Patat: Human-ai collaborative qualitative coding with explainable interactive rule synthesis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–19, 2023.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (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. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Annual Meeting of the Association for Computational Linguistics*, 2018. URL https://api.semanticscholar.org/CorpusID:40100965.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature, 2022.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Ré, and Beidi Chen. Deja vu: Contextual sparsity for efficient llms at inference time. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22137–22176. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23am.html.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 1–8, 2014.

Giorgio Visani Margherita Grandini, Enrico Bagli. Metrics for multi-class classification: An overview. *arXiv preprint arXiv:2008.05756*, 2020.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021.

Kevin R Moon, David van Dijk, Zheng Wang, William Chen, Matthew J Hirn, Ronald R Coifman, Natalia B Ivanova, Guy Wolf, and Smita Krishnaswamy. Phate: a dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data. *BioRxiv*, 120378, 2017.

Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022. doi: 10.48550/ARXIV.2210.07316. URL `https://arxiv.org/abs/2210.07316`.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *CoRR*, abs/2209.02299, 2022. doi: 10.48550/arXiv.2209.02299. URL `https://doi.org/10.48550/arXiv.2209.02299`.

nostalgebraist. interpreting gpt: the logit lens. *LessWrong*, 2020. URL `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Josh OpenAI, Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulic, and Edoardo Maria Ponti. Modular deep learning. *CoRR*, abs/2302.11529, 2023. doi: 10.48550/arXiv.2302.11529. URL `https://doi.org/10.48550/arXiv.2302.11529`.

Nicky Pochinkov and Nandi Schoots. Dissecting large language models. In *Socially Responsible Language Modelling Research*, 2023.

Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. Better language models and their implications. *OpenAI Blog https://openai.com/blog/better-language-models*, 1(2), 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295/.

Tim Rietz and Alexander Maedche. Cody: An ai-based system to semi-automate coding for qualitative research. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2021.

Breno Batista da Silva Shankar Rao Pandala. Lazy predict, 2022. URL https://lazypredict.readthedocs.io/en/latest/.

Yucheng Shi, Hehuan Ma, Wenliang Zhong, Gengchen Mai, Xiang Li, Tianming Liu, and Junzhou Huang. Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs. *arXiv preprint arXiv:2305.03513*, 2023.

Robert H Tai, Lillian R Bentley, Xin Xia, Jason M Sitt, Sarah C Fankhauser, Ana M Chicas-Mosier, and Barnas G Monteith. An examination of the use of large language models to aid analysis of textual data. *bioRxiv*, pp. 2023–07, 2023.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *CoRR*, abs/2211.09085, 2022. doi: 10.48550/arXiv.2211.09085. URL https://doi.org/10.48550/arXiv.2211.09085.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. doi: 10.48550/arXiv.2308.10248. URL https://doi.org/10.48550/arXiv.2308.10248.

Miika Oinonen Aki-Juhani Kyröläinen Anna Salmela Douglas Biber Jesse Egbert Veronika Laippala, Samuel Rönnqvist and Sampo Pyysalo. Register identification from the unrestricted open web using the corpus of online registers of english. *Language Resources and Evaluation*, 2022. URL https://github.com/TurkuNLP/CORE-corpus.

Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch specialization. *Distill*, 2021. doi: 10.23915/distill.00024.008. https://distill.pub/2020/circuits/branch-specialization.

Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, pp. 75–78, 2023.

Runxin Xu, Fuli Luo, Chengyu Wang, Baobao Chang, Jun Huang, Songfang Huang, and Fei Huang. From dense to sparse: Contrastive pruning for better pre-trained language model compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11547–11555, 2022.

He Zhang, Chuhao Wu, Jingyi Xie, ChanMin Kim, and John M Carroll. Qualigpt: Gpt as an easy-to-use tool for qualitative coding. *arXiv preprint arXiv:2310.07061*, 2023a.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*, 2015. URL https://api.semanticscholar.org/CorpusID:368182.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 877–890. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.71. URL https://doi.org/10.18653/v1/2022.findings-acl.71.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Chaojun Xiao, Xu Han, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Emergent modularity in pre-trained transformers, 2023b. URL https://openreview.net/forum?id=XHuQacT6sa6.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

The code and data used in our experiments are available at the GitHub repository: `https://github.com/Aza-Spearal/Trajectories-Probing`. We did not include the CORE dataset due to its size.

A Dataset Generation

Our categories are defined in the following way:

narrative: a spoken or written account of connected events; a story.
explanatory: a statement or account that makes something clear.
instructional: detailed information about how something should be done or operated; a direction or order.
speech: spoken words.

We generate our dataset by following these steps:

1. Manually create a list with prompts that can elicit varied texts from an LLM. The goal is that each element of the list can prompt an LLM to generate text pertaining to particular genres. Additionally, the prompts should be formulated in an open-ended way. The categories we consider are: { instructional, speech, narrative, explanatory, code }. We write 68 prompts that way.

2. To have a more diversified dataset, we create synthetic prompts with `Mistral-7B-Instruct-v2.0` (Albert Q. Jiang, 2023) by using the following prompt:
   "Please generate a prompt list inspired by the list below."
   list_to_string(prompts)

   We remove the prompts that are gibberish or off-topic.

3. We prompt `Mistral` with our list of prompts to generate the texts with a limit of 500 tokens by output.

4. Ask GPT-4 to section and label each text from step 3 according to the following genre categories. We use the following prompt to achieve this:
   "Please return a json list that sections the text below and labels it according to one of these categories: instructional, narrative, explanatory, speech, code, other. Please escape characters such as "\n". Here is how you should format the output: [\n {"text": ..., "category": ...}, \n {"text": ..., "category": ...}, \n ... \n]"
   (text_sample)

   The labelling is reviewed by humans and mislabelled text sections are removed.

You can find some examples of how the sectioning and labelling looks like in section A.1.

The genres distribution of this dataset is as follows: 'instructional': 1159, 'explanatory': 699, 'speech': 548, 'narrative': 542, 'code': 290.

A.1 Examples

A.1.1 Example 1: Narrative

Prompt: Create a tale about a person who creates a machine that can predict the future.

- **"text"**: "Title: The Oracular Apparatus", **"category"**: other

- **"text"**: "Once upon a time, in the heart of the bustling city of Zephyra, resided an unassuming man named Archimedes. He was a renowned inventor, scientist, and philosopher, known for his relentless curiosity and insatiable thirst for knowledge. However, Archimedes had a secret ambition, a dream that would change the course of history - the invention of a machine that could predict the future.", **"category"**: narrative

- **"text"**: "Archimedes toiled away in his laboratory, day and night, fueled by his determination and driven by his relentless pursuit. He contemplated the concept of time, the tangible yet elusive force that governed all existence. He pondered over the intricacies of the universe and the delicate balance of cause and effect. His mind was a whirlwind of thoughts, and he knew that if he could unlock the secrets of time, he could create a machine that could see into the future.", **"category"**: narrative

- **"text"**: "Driven by his obsession, Archimedes began constructing a device that he came to call the Oracular Apparatus. The machine was a marvel, a majestic contraption that resembled a vast, mechanical cosmos. It was adorned with gears and cogs, pendulums and pulleys, all intricately connected to form a harmonious, self-contained entity. The apparatus was designed to observe the patterns and connections in the universe and to decipher the inherent formulae that governed the passage of time.", **"category"**: explanatory

- **"text"**: "As Archimedes fine-tuned his machine, he discovered that it was capable of depicting the future in the form of visions, images that revealed the events that were yet to unfold. Mesmerized by his creation, Archimedes spent hours gazing into the Oracular Apparatus, fascinated by the glimpses of the future that unfurled before his eyes.", **"category"**: narrative

- **"text"**: "News of Archimedes' Oracular Apparatus spread like wildfire. People from all corners of Zephyra flocked to the inventor's laboratory, seeking answers to their most pressing questions and desperate for a peek into the future. The machine's accuracy was unparalleled, and its prophecies became the talk of the town.", **"category"**: narrative

- **"text"**: "However, as time went by,", **"category"**: other

A.2 Example 2: Coding

Prompt: "Implement a class in C++ representing a geometric shape, with methods to calculate its area and perimeter.

- **"text"**: "Here's an example of a simple Shape base class and a Derived class Circle that follows your requirements:", **"category"**: other

- **"text"**:

```
// C++ program with Shape and Circle classes
#include <iostream>
#include <cmath>

// Base Shape class
class Shape {
public:
    virtual double area() const = 0;
    virtual double perimeter() const = 0;
};

// Circle class derived from Shape
class Circle : public Shape {
private:
    double radius;

public:
    Circle(double r) : radius(r) {}

    double area() const override {
        return M_PI * pow(radius, 2);
    }

    double perimeter() const override {
        return 2 * M_PI * radius;
     }
};

// Test function
int main() {
    Circle circle(5);
    std::cout << "Area: " << circle.area() << std::endl;
    std::cout << "Perimeter: " << circle.perimeter() << std::endl;

    return 0;
}

```

  , **"category"**: code
- **"text"**: "In this example, we have a base ‘Shape‘ class with pure virtual functions ‘area()‘ and ‘perimeter()‘. The ‘Circle‘ class is derived from ‘Shape‘, and it provides the implementation for both methods. In the provided ‘main‘ function, we create a ‘Circle‘ object and calculate its area and perimeter.", **"category"**: explanatory